

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**«Δημιουργία αυτόματης διαδικασίας
σχεδιασμού πύργων ανεμογεννητριών
μέσω προγραμματιστικής διεπαφής
τριδιάστατου στερεού μοντελοποιητή CAD».**



Κώτσιας Δημοσθένης Α.Μ:42860
Μανουσέλης Δημήτριος Α.Μ.: 43005

Επιβλέπων Καθηγητής:
Δρ. Στεργίου Κωνσταντίνος

Ιούλιος 2019

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός πυλώνων χερσαίων (onshore) ανεμογεννητριών μέσω της προγραμματιστικής διεπαφής τρισδιάστατου παραμετρικού μοντελοποιητή Visual Basic for Application και του 3D προγράμματος Inventor. Μέσω του προγράμματος δίνεται η δυνατότητα στο χρήστη να επιλέξει τα κομμάτια από τα οποία θα αποτελείται ο πύργος με τις διαστάσεις τους καθώς και το είδος τους (κυλινδρικό ή κωνικό). Ο κώδικας που δημιουργήθηκε μπορεί να χρησιμοποιηθεί από εταιρίες ανανεώσιμων πηγών ενέργειας που δραστηριοποιούνται στον τομέα κατασκευής ανεμογεννητριών. Μέσω της αυτοματοποίησης που παρέχει στο σχεδιασμό, δύναται να μειωθεί το κόστος της μελέτης των πύργων των ανεμογεννητριών.

Εξετάζοντας το αιολικό δυναμικό στις χερσαίες περιοχές της Ευρώπης, παρατηρούμε αμέσως τις τεράστιες δυνατότητες ανάπτυξης ανεμογεννητριών. Παρατηρούμε επίσης την πλεονεκτική θέση της χώρας μας που εμφανίζεται να διαθέτει στο Αιγαίο πέλαγος μια από τις πλουσιότερες, σε αιολικό δυναμικό, περιοχές της Ευρώπης. Για να έχουμε όμως μεγαλύτερη απόδοση στην ισχύ μιας ανεμογεννήτριας πρέπει να αυξήσουμε το ύψος της έτσι ώστε να εκμεταλλευτούμε τις υψηλότερες ταχύτητες ανέμου σε μεγαλύτερα ύψη. Αυτό με την σειρά του θα επιφέρει πιο μεγάλα φορτία και επομένως μεγαλύτερες καταπονήσεις στην βάση του πυλώνα της ανεμογεννήτριας. Ως συνέπεια τούτου θα πρέπει να αυξήσουμε την αντοχή του πυλώνα είτε αυξάνοντας το πάχος του υλικού της διατομής είτε αυξάνοντας τη διάμετρο της διατομής μέχρι ενός ορίου λόγω του περιορισμένου πλάτους του οδικού δικτύου που αποτρέπει την μεταφορά τμημάτων του πυλώνα της ανεμογεννήτριας με μεγάλες διατομές. Τα παραπάνω συνεπάγονται επίσης αύξηση του κόστους γι αυτό επίσης είναι πολύ σημαντικό και το υλικό κατασκευής τους.

Στο πρώτο κεφάλαιο γίνεται μια αναφορά στην ανάγκη της σύγχρονης κοινωνίας να στραφεί στις διάφορες μορφές των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ). Στη συνέχεια δίνεται μια έμφαση στην αιολική ενέργεια με τα προτερήματα και τα μειονεκτήματά της καθώς και στον ρόλο της στην οικονομία και στο περιβάλλον.

Στο δεύτερο κεφάλαιο επιχειρείται μια ιστορική αναδρομή στην εξέλιξη των ανεμογεννητριών καθώς και στο ρόλο τους στην ζωή του ανθρώπου.

Στο τρίτο κεφάλαιο αναπτύσσονται τα είδη των ανεμογεννητριών χωριζόμενα κυρίως σε δύο μεγάλες κατηγορίες (οριζοντίου και καθέτου άξονα) με τα πλεονεκτήματα και τα μειονεκτήματα κάθε ενός σχεδιασμού. Επίσης αναφέρονται και επεξηγούνται τα διάφορα εσωτερικά και εξωτερικά τμήματα από τα οποία αποτελούνται.

Στο τέταρτο κεφάλαιο αναφέρονται τα είδη των πύργων των ανεμογεννητριών σε συνδυασμό με τις διάφορες επιλογές των υλικών κατασκευής τους καθώς και το είδος των φορτίων και οι καταπονήσεις που δέχονται. Όλα τα παραπάνω είναι απαραίτητα έτσι ώστε να εξασφαλιστεί η σωστή στήριξη του πύργου και ως εκ τούτου η καλή λειτουργία των ανεμογεννητριών.

Στο πέμπτο και τελευταίο κεφάλαιο περιγράφεται βήμα βήμα η δημιουργία του κώδικα μέσω του προγράμματος Inventor σε γλώσσα προγραμματισμού Visual Basic for Application (VBA) για την κατασκευή των διαφόρων τμημάτων του πύργου μιας ανεμογεννήτριας. Περιγράφονται αναλυτικά όλα τα απαραίτητα στάδια με σχέδια και εικόνες που κατατοπίζουν το χρήστη. Στο τέλος για την καλύτερη κατανόηση του προγράμματος περιγράφονται και δύο παραδείγματα εφαρμογής.

Abstract

Wind Energy is rapidly growing the last years in order to reach the target of the European Union to produce a significant amount of its energy (20%) from renewable sources by 2020. Due to the fact that the cost of extracting fossil fuels is continuously growing, new technologies are trying to lower the cost of harvesting energy from the wind. The objective of this thesis is the development of a computer program for constructing wind turbine towers. The program is based on the commercial software Visual Basic for Application built in the design program Inventor. It offers the user to choose between two geometrical shapes (cylindrical and conical) as parts of the wind turbine tower as well as the dimensions of the parts. It is user friendly, fast and reliable making it a useful tool for engineers.

Πίνακας Περιεχομένων

Περίληψη	1
Abstract	3
Πίνακας Εικόνων	6
Αιολική Ενέργεια	9
Εννοιολογική σημασία του Ανέμου	9
Σχήμα 1.1. Παγκόσμια συνολική δυναμικότητα αιολικής ενέργειας για το έτος 2017	10
Ανάγκη για ανανεώσιμες πηγές ενέργειας	11
Είδη Ανανεώσιμων πηγών ενέργειας	12
Πλεονεκτήματα Αιολικής Ενέργειας	13
Μειονεκτήματα Αιολικής Ενέργειας	14
Ο Ρόλος της Αιολικής Ενέργειας στην Οικονομία	15
Οικονομικά οφέλη της αιολικής ενέργειας:	16
ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΑΝΕΜΟΓΕΝΝΗΤΡΙΩΝ	17
Οι Αρχές της Αιολικής Ενέργειας (5000 π.Χ. - 1300 μ.Χ.)	17
Ανεμόμυλοι στον δυτικό κόσμο (1300 - 1875 Α.Δ.)	19
Ο ρόλος των μικρότερων συστημάτων	22
Πρώτη χρήση του ανέμου για την παραγωγή ηλεκτρικής ενέργειας μεγάλης κλίμακας	23
Πρωτοπόροι μικρών συστημάτων	25
"Μαζική" ισχύς από τον άνεμο	26
Ευρωπαϊκή Ανάπτυξη	27
Οι Ανεμογεννήτριες σήμερα	29
Είδη Ανεμογεννητριών	31
Ανεμογεννήτριες Κάθετου Άξονα (VAWTs)	31
Ο ανεμοκινητήρας Savonius	31
Ο ανεμοκινητήρας Darrieus	32
Ανεμογεννήτριες Οριζοντίου Άξονα (HAWTs)	35
Πύργος της ανεμογεννήτριας (Tower)	39
Κατηγορίες πύργων ανεμογεννητριών	41
Σωληνοειδής χαλύβδινος πύργος (tubular steel tower)	41
Υβριδικός πύργος σκυροδέματος/χάλυβα (concrete/steel hybrid tower)	42
Σύνθετος πυλώνας από ινοπλισμένα πολυμερή υλικά	43

Δικτυωτός πύργος (Lattice Tower).....	43
Απαιτήσεις σχεδιασμού πυλώννα ανεμογεννήτριας	46
Φορτία Πύργου	47
Δικτυωτή Α/Γ της General Electric	49
Το μέλλον στο σχεδιασμό των ανεμογεννητριών (Χωρίς Πτερύγια).....	51
Γλώσσα Προγραμματισμού VISUAL BASIC for APPLICATIONS.....	53
Δημιουργία Κώδικα.....	54
Εισαγωγή.....	54
Αρχικές Ενέργειες.....	54
Κατασκευή Κυλινδρικού Κομματιού	55
Κατασκευή Κωνικού Κομματιού	63
Κατασκευή Κωνικής Φλάντζας.....	67
Κατασκευή Κομματιού Βάσης (με εξωτερική φλάντζα)	71
Παραδείγματα εφαρμογής 1	74
Παράδειγμα εφαρμογής 2	79
Βιβλιογραφία	86

Πίνακας Εικόνων

Σχήμα 1.1. Παγκόσμια συνολική δυναμικότητα αιολικής ενέργειας για το έτος 2017	10
Σχήμα 1.2. Σε αυτό το περσικό σχεδιασμό “rapemone”, ο ρότορας μπορεί μόνο να συγκεντρώσει το ήμισυ του ανέμου που χτυπά την περιοχή συλλογής.	18
Σχήμα 1.3. Ανεμόμυλοι με πανιά για την άντληση νερού στο νησί της Κρήτης	
Σχήμα 1.4. Ένας αμερικανικός rapemone του 19ου αιώνα, εμπνευσμένος από τον περσικό, που πιθανώς χρησιμοποιούταν ως στεγνωτήριο ρούχων.	19
Σχήμα 1.5. Ένας πρώτος μύλος οριζόντιου άξονα με πτερύγια στην ακτή της Μεσογείου.	20
Σχήμα 1.6. Ένας ολλανδικός ανεμόμυλος (1994) που διαθέτει τμήματα αεροδυναμικής αιχμής (πάνω δεξιά). Ο μηχανισμός που χρησιμοποιείται για την περιστροφή του ρότορα στον άνεμο και τα παράθυρα του πρώτου ορόφου όπου ζούσαν οι άνθρωποι είναι εύκολα αντιληπτά.	22
Σχήμα 1.7. Ένας ανεμόμυλος άντλησης νερού με χαλύβδινο νερό στην Αμερικάνικη Midwest (τέλη του 1800).	23
Σχήμα 1.8. Η βρωμιά μετάλλων στο Cleveland, Ohio, 1888. Η πρώτη χρήση ενός μεγάλου ανεμόμυλου για την παραγωγή ηλεκτρικής ενέργειας. Σημειώστε τον άνθρωπο που κόβει το γκαζόν στα δεξιά.	24
Σχήμα 1.9. M.L. Jacobs ρυθμίζοντας τον μηχανισμό αλλαγής κλίσης που ενεργοποιείται με ελατήριο σε ένα JacobsWind-electric το 1977.	26
Σχήμα 1.10. Η ανεμογεννήτρια 1,25-μεγαβάτ του PalmerPutnam ήταν ένα από τα θαύματα της τεχνολογίας στα τέλη της δεκαετίας του 1930, αλλά το άλμα σε κλίμακα ήταν υπερβολικά μεγάλο για τα διαθέσιμα υλικά.	27
Σχήμα 1.11. Ναι, αυτό είναι ένα αεροσκάφος που συγκρατεί τα τρία πτερύγια του "GedserMollen." Τα υαλοβάμβακα αργότερα εξάλειψαν αυτή την απαίτηση σχεδιασμού.	28
Σχήμα 1.12. Οι ανεμογεννήτριες του Hutter, όπως και άλλες γερμανικές συσκευές των μέσων του 20ού αιώνα, προωθήθηκαν για την εποχή τους.	29
Σχημα 1.13. Βασικά μέρη διάταξης για την εκμετάλλευση της αιολικής ενέργειας καθώς επίσης και διάγραμμα ροής της ενέργειας.	30
Σχήμα 1.14. Αυτοστηριζόμενος Darrieus.	31
Σχήμα 1.15. πάνω αριστερά: Ανεμοκινητήρας Darrieus, πάνω δεξιά: Ανεμοκινητήρας Savonius κάτω αριστερά : Hybrid Darrieus-Savonius και κάτω δεξιά: Ανεμοκινητήρας τύπου H.	33
Σχήμα 1.16. Τύποι Ανεμοκινητήρων κατακόρυφου άξονα.	34
Σχήμα 1.17. Τεχνικά μέρη μιας ανεμογεννήτριας οριζοντίου άξονα.	37
Σχήμα 1.18. Τύποι ανεμοκινητήρων οριζόντιου άξονα.	38

Σχήμα 1.19. Πορεία αύξησης του ύψους του πύργου και της διαμέτρου του ρότορα στις Α/Γ από τη δεκαετία του 80.	40
Σχήμα 1.20. Τα τρία επικρατέστερα είδη πύργων Α/Γ.	40
Σχήμα 1.21. Η μεγαλύτερη Α/Γ στον κόσμο με σωληνωτό πύργο κοντά στις ακτές του Liverpool στην Μεγάλη Βρετανία. (Burbo Bank extension windfarm in Merseyside).	42
Σχήμα 1.22. Σχεδιασμός κυλινδρικού κελύφους με υβριδική – σύμμικτη διατομή ινοπλισμένων πολυμερών (fiber-glass) – σκυροδέματος.	43
Σχήμα 1.23. Πλήθος ποδών δικτυωτών πύργων.	45
Σχήμα 1.24. ο μεγαλύτερος δικτυωτός πύργος βρίσκεται στο Laasow (Brandenburg) στη Γερμανία με ύψος 160 m.	45
Σχήμα 1.25. Παραλαβή φόρτισης σε δικτυωτούς πύργους.	49
Σχήμα 1.26. Δικτυωτός πύργος της General Electric.	50
Σχήμα 1.27. Εσωτερικό δικτυωτού πύργου General Electric (Είναι ορατή και η επένδυση.	50
Σχήμα 1.28. Κέντρο ελέγχου στη βάση της Α/Γ.	51
Σχήμα 1.29. Α/Γ τύπου Vortex.	53
Σχήμα 1.30. Επιλογή του Workplane.....	56
Σχήμα 1.31. Δημιουργία εξωτερικού κύκλου.	57
Σχήμα 1.32. Δημιουργία εσωτερικού κύκλου.	57
Σχήμα 1.33. Επιλογή επιφάνειας μεταξύ των κύκλων.	58
Σχήμα 1.34. Επιλογή της εντολής Extrude.	59
Σχήμα 1.35. Δημιουργία κατάλληλου Workplane πάνω στο κομμάτι κατασκευής.	60
Σχήμα 1.36. Δημιουργία εξωτερικού και εσωτερικού κύκλου για δημιουργία φλάντζας.	61
Σχήμα 1.37. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στη φλάντζα.	62
Σχήμα 1.38. Τελικό κομμάτι Βάσης (με εξωτερική κάτω φλάντζα).	63
Σχήμα 1.39. Δημιουργία εξωτερικού κύκλου.	65
Σχήμα 1.40. Δημιουργία εσωτερικού κύκλου.	66
Σχήμα 1.41. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στο κωνικό κομμάτι.	67
Σχήμα 1.42. Δημιουργία κατάλληλου Workplane πάνω στο κομμάτι κατασκευής.	68
Σχήμα 1.43. Δημιουργία εξωτερικού και εσωτερικού κύκλου για δημιουργία φλάντζας.	69

Σχήμα 1.44. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στη φλάντζα.	70
Σχήμα 1.45. Ολοκληρωμένη κατασκευή κομματιού βάσης.	73
Σχήμα 1.46. Μέσω του 1ου User Form ο χρήστης επιλέγει τον αριθμό των κομματιών που θα αποτελείται ο πύργος.	74
Σχήμα 1.47. Μέσω του 2ου User Form ο χρήστης επιλέγει την μορφή των κομματιών που θα αποτελείται ο πύργος (κυλινδρικά η κωνικά κομμάτια).	75
Σχήμα 1.48. Κομμάτι Κορυφής.	76
Σχήμα 1.49. Κωνικό κομμάτι.	76
Σχήμα 1.50. Κομμάτι Βάσης.	77
Σχήμα 1.51. Ολοκληρωμένη μορφή πύργου.	77
Σχήμα 1.52. Ολοκληρωμένη μορφή πύργου σε μηχανολογικό σχέδιο.	78
Σχήμα 1.53. Μέσω του 1ου User Form ο χρήστης επιλέγει τον αριθμό των κομματιών που θα αποτελείται ο πύργος.	79
Σχήμα 1.54. Μέσω του 2ου User Form ο χρήστης επιλέγει την μορφή των κομματιών που θα αποτελείται ο πύργος (κυλινδρικά η κωνικά κομμάτια).	80
Σχήμα 1.55. Κομμάτι Κορυφής.	81
Σχήμα 1.56. Κωνικό κομμάτι.	82
Σχήμα 1.57. 1ο Κυλινδρικό κομμάτι.	82
Σχήμα 1.58. 2ο Κυλινδρικό κομμάτι.	83
Σχήμα 1.59. Κομμάτι Βάσης.	83
Σχήμα 1.60. Ολοκληρωμένη μορφή πύργου.	84
Σχήμα 1.61. Ολοκληρωμένη μορφή πύργου σε μηχανολογικό σχέδιο.	85

Αιολική Ενέργεια

Εννοιολογική σημασία του Ανέμου

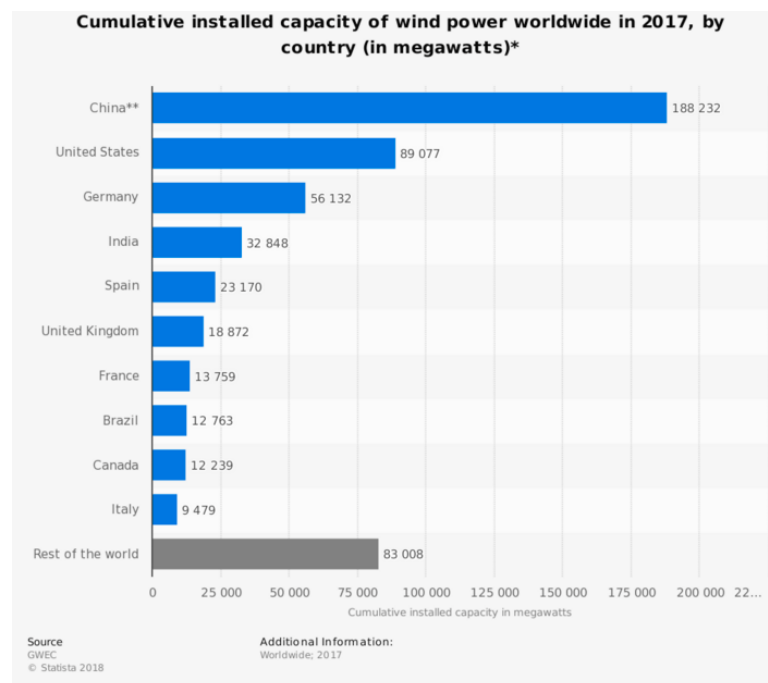
Άνεμος χαρακτηρίζεται κάθε μετακίνηση του ατμοσφαιρικού αέρα πάνω στη επιφάνεια της γης ανεξαρτήτως κατεύθυνσης. Η οριζόντια συνιστώσα στην κίνηση χαρακτηρίζεται ως άνεμος ενώ η κατακόρυφη συνιστώσα ως ρεύμα. Ανοδικό θα χαρακτηρίζεται ένα ρεύμα εάν η κίνηση είναι από κάτω προς τα πάνω ενώ καθοδικό, το αντίθετο. Αυτή η κίνηση του ατμοσφαιρικού αέρα προκαλείται από τη διαφορά θερμοκρασίας του αέρα που με τη σειρά της δημιουργεί υπό ορισμένες προϋποθέσεις, διαφορές στη βαρομετρική πίεση μεταξύ παρακείμενων τόπων. Επομένως, η αιολική ενέργεια είναι έμμεση μορφή ηλιακής ενέργειας. Περίπου 1-2 % της ηλιακής ακτινοβολίας που προσπίπτει στην επιφάνεια της γης μετατρέπεται σε άνεμο. Καθώς ο αέρας θερμαίνεται στις τροπικές ζώνες αρχικά ανυψώνεται στην ατμόσφαιρα και κατόπιν οδεύει προς τους πόλους. Η κίνηση των αέριων μαζών επηρεάζεται σημαντικά από την περιστροφή της γης, η επίδραση της οποίας είναι μεγαλύτερη στον ισημερινό και μηδενική στους πόλους, από την αναλογία της επιφάνειας της ξηράς προς την αντίστοιχη της θάλασσας, από τα μορφολογικά χαρακτηριστικά της ξηράς (βουνά, πεδιάδες) καθώς και από τις εποχές του χρόνου. Ο συνδυασμός, λοιπόν, της προσλαμβανόμενης από την ατμόσφαιρα και τη γη ηλιακής ακτινοβολίας με την ανομοιομορφία του γήινου ανάγλυφου και την περιστροφή της γης γύρω από το άξονά της, είναι η γενεσιουργός αιτία του ανέμου. Οι δυνάμεις που ρυθμίζουν τη κίνηση του είναι: η δύναμη βαροβαθμίδας, η δύναμη Coriolis, η τριβή και η φυγόκεντρη δύναμη

Αιολική ενέργεια είναι η ενέργεια που παράγεται από την εκμετάλλευση του πνέοντος ανέμου. Η ενέργεια αυτή χαρακτηρίζεται "ήπια μορφή ενέργειας" και περιλαμβάνεται στις "καθαρές" πηγές, όπως συνηθίζονται να λέγονται οι πηγές ενέργειας που δεν εκπέμπουν ή δεν προκαλούν ρύπους. Η αρχαιότερη μορφή εκμετάλλευσης της αιολικής ενέργειας ήταν τα ιστία (πανιά) των πρώτων ιστιοφόρων πλοίων και πολύ αργότερα οι ανεμόμυλοι στην ξηρά. Ονομάζεται αιολική γιατί στην ελληνική μυθολογία ο Αίολος ήταν

ο θεός του ανέμου.

Η αιολική ενέργεια μπορεί να αξιοποιηθεί χρησιμοποιώντας κατάλληλους μηχανισμούς και διατάξεις, τις ανεμογεννήτριες. Η κινητική ενέργεια του ανέμου μετατρέπεται αρχικά σε μηχανική και ακολούθως σε ηλεκτρική, μέσω των ανεμογεννητριών. Με την εγκατάσταση περισσότερων ανεμογεννητριών στην ίδια θέση, σχηματίζονται αιολικά πάρκα. Ένα μεγάλο αιολικό πάρκο μπορεί να αποτελείται από αρκετές εκατοντάδες μεμονωμένες ανεμογεννήτριες οι οποίες συνδέονται με το ηλεκτρικό δίκτυο μεταφοράς ηλεκτρικής ενέργειας. Υπεράκτια αιολικά πάρκα μπορούν να αξιοποιήσουν πιο συχνούς και ισχυρούς ανέμους από τις χερσαίες εγκαταστάσεις και έχουν λιγότερο ορατή επίπτωση στο τοπίο, αλλά το κόστος κατασκευής είναι σημαντικά υψηλότερο.

Η αιολική ενέργεια αποτελεί σήμερα μια ελκυστική λύση στο πρόβλημα της ηλεκτροπαραγωγής. Το «καύσιμο» είναι άφθονο, αποκεντρωμένο και δωρεάν. Δεν εκλύονται αέρια θερμοκηπίου και άλλοι ρύποι, και οι επιπτώσεις στο περιβάλλον είναι μικρές σε σύγκριση με τα εργοστάσια ηλεκτροπαραγωγής από συμβατικά καύσιμα. Επίσης, τα οικονομικά οφέλη μιας περιοχής από την ανάπτυξη της αιολικής βιομηχανίας είναι αξιοσημείωτα.



Σχήμα 1.1. Παγκόσμια συνολική δυναμικότητα αιολικής ενέργειας για το έτος 2017

Ανάγκη για ανανεώσιμες πηγές ενέργειας

Η αναζήτηση μεθόδων εκμετάλλευσης των ανανεώσιμων πηγών ενέργειας δημιουργήθηκε ως απάντηση στην καταστροφή του περιβάλλοντος και στην εξάντληση των μη ανανεώσιμων πηγών ενέργειας, δύο φαινόμενα, συνέπεια της Βιομηχανικής Επανάστασης. Από τον 18^ο αιώνα παρατηρείται η γέννηση της βιομηχανίας και η μεγάλης κλίμακας εκμετάλλευση φυσικών πόρων χωρίς να γίνονται άμεσα αντιληπτές οι καταστροφικές συνέπειες για το περιβάλλον.

Πιο συγκεκριμένα, στην Αγγλία του 19^{ου} αιώνα η τεράστια αστικοποίηση οδήγησε σε επιδημίες, φτώχεια και εξαθλίωση, γεγονότα που έφεραν για πρώτη φορά στην επιφάνεια τις περιβαλλοντικές επιπτώσεις της ανεξέλεγκτης εκβιομηχάνισης (Homburg,1998). Η συνθήκη αυτή, οδήγησε για πρώτη φορά τη δημιουργία επιτροπής προκειμένου να διαπιστωθεί το μέγεθος της βλάβης που προκαλείτο από τα αέρια που δημιουργούνται σε ορισμένες παραγωγικές διαδικασίες. Η μελέτη έδειξε ότι η μόλυνση που είχε επέλθει ήταν σε μεγάλο βαθμό καταστροφική για τη χλωρίδα υποβιβάζοντας τις επιπτώσεις του φαινομένου στην πανίδα.

Σύγχρονες έρευνες έχουν σημειώσει ότι η συγκέντρωση του διοξειδίου του άνθρακα στην ατμόσφαιρα αυξήθηκε από περίπου 280 μέρη ανά εκατοστό (ppm) στον προβιομηχανικό κόσμο σε 380 μέρη ανά εκατοστό (ppm) στο σύγχρονο κόσμο (Hill & Kolb,2007). Οι συνέπειες της συγκέντρωσης διοξειδίου του άνθρακα στην ατμόσφαιρα μπορούν να αποβούν μοιραίες για την πορεία του πλανήτη. Δύο από τις κυριότερες επιπτώσεις περιλαμβάνουν την αύξηση της θερμοκρασίας σε παγκόσμιο επίπεδο και την μείωση του pH των ωκεανών. Με τη σειρά της η αύξηση της θερμοκρασίας του πλανήτη προκαλεί λιώσιμο των πάγων στους πόλους, αυξάνοντας έτσι τη στάθμη των ωκεανών απειλώντας προς εξαφάνιση ολόκληρες περιοχές. Επιπλέον, το ίδιο φαινόμενο αποτελεί παράγοντα κινδύνου για τους έμβιους οργανισμούς καθώς είναι προσαρμοσμένοι να επιβιώνουν σε ορισμένες κλιματικές συνθήκες. Όσον αφορά τη μείωση του pH των ωκεανών, παρατηρείται καταστροφή του θαλάσσιου πλαγκτόν με αποτέλεσμα το θάνατο λόγω αστίας μεγάλων πληθυσμών θαλάσσιων οργανισμών.

Οι ξαφνικές αυτές αλλαγές στο οικοσύστημα οδήγησαν σε προβληματισμό μεγάλο μέρος της επιστημονικής κοινότητας και των οργανισμών περιβαλλοντικής προστασίας. Έτσι, ξεκίνησε

γύρω στη δεκαετία του 1960 η αναζήτηση νέων τρόπων εκμετάλλευσης των φυσικών πόρων με το λιγότερο δυνατό κόστος για τον πλανήτη.

Τα τελευταία 30 χρόνια η κρίση του πετρελαίου, η επικείμενη εξάντληση των ορυκτών καυσίμων, η αυξανόμενη μόλυνση του περιβάλλοντος, το φαινόμενο του θερμοκηπίου, και η βούληση των κρατών για εθνική και ανεξάρτητη ενεργειακή πολιτική με την αξιοποίηση των εγχώριων πηγών ενέργειας οδηγούν στη χρήση των ανανεώσιμων πηγών ενέργειας που δίνουν σε μεγάλο βαθμό λύση στα παραπάνω προβλήματα. Πηγές ενέργειας όπως η αιολική, η ηλιακή κ.α. έχουν ένα χαρακτηριστικό που τις δίνει την ονομασία ανανεώσιμες, ότι δηλαδή από τη φύση τους ανανεώνονται διαρκώς και προσφέρονται στον άνθρωπο δωρεάν για εκμετάλλευση.

Όπως θα αναφερθεί λεπτομερώς και στη συνέχεια, η κατασκευή ανεμογεννητριών εμπίπτει σε αυτό το πλαίσιο αναγκών.

Είδη Ανανεώσιμων πηγών ενέργειας

- I. Αιολική ενέργεια: Η κινητική ενέργεια που παράγεται από τη δύναμη του ανέμου και μετατρέπεται σε μηχανική ή/και ηλεκτρική. Χρησιμοποιήθηκε παλιότερα σε διάφορες εφαρμογές όπως η άντληση νερού και η άλεση σε δημητριακών σε ανεμόμυλους.
- II. Ηλιακή ενέργεια: Πρόκειται για το σύνολο των διαφόρων μορφών ενέργειας που προέρχονται από τον Ήλιο. Το φως και η θερμότητα που ακτινοβολούνται, απορροφούνται από τη Γη και μετατρέπονται σε άλλες μορφές ενέργειας. Η τεχνολογία σήμερα αξιοποιεί την ηλιακή ενέργεια μέσω τριών συστημάτων, τα θερμικά ηλιακά, τα παθητικά ηλιακά και τα φωτοβολταϊκά συστήματα, ενώ η χρήση της για την παραγωγή ηλεκτρισμού έχει αρχίσει να κερδίζει έδαφος, με την βοήθεια της πολιτικής προώθησης των Ανανεώσιμων Πηγών Ενέργειας από το ελληνικό κράτος και την Ευρωπαϊκή Ένωση.
- III. Υδραυλική ενέργεια: Τα μικρά υδροηλεκτρικά έργα (ισχύος μέχρι 10 MW) που αξιοποιούν τις υδατοπτώσεις, με στόχο την παραγωγή ηλεκτρικής ενέργειας ή τη μετατροπή τους σε μηχανική. Θεωρούνται από τις πιο διαδεδομένες ΑΠΕ.
- IV. Βιομάζα: Χρησιμοποιεί τους υδατάνθρακες των φυτικών οργανισμών (κυρίως αποβλήτων της βιομηχανίας ξύλου, τροφίμων και ζωοτροφών και της βιομηχανίας

ζάχαρης) με σκοπό την αποδέσμευση της ενέργειας που δεσμεύτηκε απ' το φυτό με τη φωτοσύνθεση. Ακόμα μπορούν να χρησιμοποιηθούν αστικά απόβλητα και απορρίμματα. Μπορεί να δώσει βιοαιθανόλη και βιοαέριο, που είναι καύσιμα πιο φιλικά προς το περιβάλλον από τα παραδοσιακά. Είναι μια πηγή ενέργειας με πολλές δυνατότητες και εφαρμογές που θα χρησιμοποιηθεί πλατιά στο μέλλον.

- V. Γεωθερμική ενέργεια: Προέρχεται από τη θερμότητα που παράγεται απ' τη ραδιενεργό αποσύνθεση των πετρωμάτων της γης και εμπεριέχεται σε φυσικούς ατμούς, σε επιφανειακά ή υπόγεια θερμά νερά και σε θερμά ξηρά πετρώματα. Είναι εκμεταλλεύσιμη εκεί όπου η θερμότητα αυτή ανεβαίνει με φυσικό τρόπο στην επιφάνεια, π.χ. στους θερμοπίδακες ή στις πηγές ζεστού νερού. Μπορεί να χρησιμοποιηθεί είτε απευθείας για θερμικές εφαρμογές είτε για την παραγωγή ηλεκτρισμού.

- VI. Ενέργεια από τη θάλασσα:

Ενέργεια από παλίρροιες: Εκμεταλλεύεται τη βαρύτητα του Ήλιου και της Σελήνης, που προκαλεί ανύψωση της στάθμης του νερού. Το νερό αποθηκεύεται καθώς ανεβαίνει και όταν ξανακατεβαίνει διέρχεται μέσα από στροβίλους, παράγοντας ηλεκτρισμό. Ενέργεια από κύματα: Εκμεταλλεύεται την κινητική ενέργεια των κυμάτων της θάλασσας.

Ενέργεια από τους ωκεανούς: Εκμεταλλεύεται τη διαφορά θερμοκρασίας ανάμεσα στα στρώματα του ωκεανού, κάνοντας χρήση θερμικών κύκλων. Βρίσκεται στο στάδιο της έρευνας.

Πλεονεκτήματα Αιολικής Ενέργειας

- ✓ Ο άνεμος παρέχεται δωρεάν και είναι ανανεώσιμος επομένως είναι στο χέρι των ανθρώπων να τον χρησιμοποιήσουν και να τον εκμεταλλευτούν προς όφελός τους.
- ✓ Είναι φιλική προς το περιβάλλον με αμελητέες επιδράσεις στην πανίδα καθώς δεν καίγονται ορυκτά καύσιμα για την παραγωγή ηλεκτρισμού και επομένως δεν επιβαρύνει το τοπικό περιβάλλον με επικίνδυνους αέριους ρύπους όπως μονοξείδιο του άνθρακα, διοξείδιο του θείου καθώς και καρκινογόνα μικροσωματίδια όπως γίνεται με τους συμβατικούς σταθμούς παραγωγής ηλεκτρικής ενέργειας.

- ✓ Οι ανεμογεννήτριες καταλαμβάνουν λιγότερο χώρο από ένα μέσο σταθμό παραγωγής ηλεκτρικής ενέργειας. Καταλαμβάνουν μόνο μερικά τετραγωνικά μέτρα για τη βάση τους, και αυτό επιτρέπει στο έδαφος γύρω από τη μονάδα να χρησιμοποιηθεί για πολλούς σκοπούς, όπως η ανάπτυξη της γεωργίας.
- ✓ Νεότερες τεχνολογίες κάνουν την εξαγωγή της αιολικής ενέργειας πολύ πιο αποτελεσματική και κατά επέκταση οικονομική. Το αρχικό κόστος κατασκευής και λειτουργίας θα συνεχίσει να μειώνεται, συμπαρασύροντας την τιμή της κιλοβατώρας (KWh) σε αντίθεση με την τιμή των ορυκτών καυσίμων που θα συνεχίσει να ανεβαίνει λόγω εξάντλησης των αποθεμάτων και τις δυσκολίες εξόρυξης των καινούργιων αποθεμάτων σε μεγάλα βάθη.
- ✓ Ενισχύει την ενεργειακή ανεξαρτησία κάθε χώρας.
- ✓ Δημιουργεί θέσεις απασχόλησης στην περιφέρεια
- ✓ Απομακρυσμένες περιοχές, όπως ορεινά χωριά, που δεν είναι συνδεδεμένα με το δίκτυο ηλεκτρικής ενέργειας μπορούν να χρησιμοποιήσουν ανεμογεννήτριες για να παράγουν το δικό τους ηλεκτρικό ρεύμα. Άρα είναι ανεξάρτητη από κεντρικά δίκτυα διανομής.
- ✓ Οι ανεμογεννήτριες είναι διαθέσιμες σε μια ποικιλία μεγεθών, που σημαίνει ένα ευρύ φάσμα ανθρώπων και επιχειρήσεων μπορούν να τις χρησιμοποιήσουν.

Μειονεκτήματα Αιολικής Ενέργειας

- Η δύναμη του ανέμου δεν είναι σταθερή αλλά κυμαινόμενη ανάλογα με την ένταση του ανέμου. Αυτό σημαίνει ότι δεν παράγουν την ίδια ποσότητα ηλεκτρικής ενέργειας όλη την περίοδο λειτουργίας τους. Σε πολλές περιοχές, η ένταση ανέμων είναι πολύ χαμηλή για να υποστηρίξει μια ανεμογεννήτρια ή ένα αιολικό πάρκο
- Οι ανεμογεννήτριες παράγουν γενικά λιγότερη ηλεκτρική ενέργεια κατά μέσο όρο από ένα σταθμό παραγωγής ενέργειας που χρησιμοποιεί ορυκτά καύσιμα, οπότε απαιτείται να κατασκευαστούν πολλαπλές ανεμογεννήτριες, ώστε να έχουν το ίδιο ενεργειακό αντίκτυπο.
- Η μετάδοση της ηλεκτρικής ενέργειας από αιολικά πάρκα εξ αποστάσεως μπορεί να

είναι ένα σημαντικό εμπόδιο για τις επιχειρήσεις κοινής ωφέλειας, δεδομένου ότι πολλές φορές οι ανεμογεννήτριες δεν βρίσκονται γύρω από τα αστικά κέντρα.

➤ Διαμαρτυρίες ή/και αναφορές αντιμετωπίζει συνήθως κάθε προτεινόμενη ανάπτυξη αιολικών πάρκων. Οι άνθρωποι αισθάνονται οπτική όχληση δεδομένου ότι οι ανεμογεννήτριες είναι ορατές από μεγάλη απόσταση και θεωρούν ότι πρέπει να γίνεται προσπάθεια ενσωμάτωσης τους με τέτοιο τρόπο ώστε να μην διαταράσσεται το φυσικό κάλος του τοπίου.

➤ Ενώ το ποσοστό των πτηνών που εκλείπουν ετησίως από πρόσκρουση τους σε ανεμογεννήτριες είναι μόλις 0.5%, παρόλα αυτά πρέπει να λαμβάνεται υπόψη η χωροθέτηση τους σε τυχόν προστατευόμενες περιοχές και να εξετάζεται η τοποθέτηση συστήματος υπερήχων. Περιβαλλοντολόγοι έχουν παραπονεθεί ότι οι μεγάλες ανεμογεννήτριες είναι επικίνδυνες όταν τοποθετούνται πάνω στις μεταναστευτικές διαδρομές των πουλιών.

➤ Η αποθήκευση της περίσσειας ενέργειας που παράγεται από τις ανεμογεννήτριες, με τη μορφή των μπαταριών, υδρογόνου ή άλλων μορφών εξακολουθεί να χρειάζεται έρευνα και ανάπτυξη για να καταστούν εμπορικά βιώσιμες.

➤ Η ηχορύπανση από τις εμπορικές ανεμογεννήτριες είναι μερικές φορές παρόμοια με ένα μικρό κινητήριο τζετ και αποτελεί πρόβλημα σε όσους κατοικούν μέσα σε λίγες εκατοντάδες μέτρα από μια ανεμογεννήτρια. Ο εκπεμπόμενος θόρυβος που προέρχεται από τα περιστρεφόμενα μηχανικά τμήματα και από την περιστροφή των πτερυγίων εκτιμάται σε περίπου 44db σε απόσταση 200m για ταχύτητα ανέμου 8 m/s. Ωστόσο νέες τεχνολογίες έχουν μειώσει αισθητά τα επίπεδα της ηχορύπανσης.

Ο Ρόλος της Αιολικής Ενέργειας στην Οικονομία

Η αξιοποίηση της αιολικής ενέργειας είναι μια από τις πιο θετικές εξελίξεις που δημιουργήθηκε στη σύγχρονη εποχή ως αποτέλεσμα της ανόδου των τιμών του πετρελαίου, τη κλιματική αλλαγή, την υποβάθμιση του περιβάλλοντος, την εξάντληση των αποθεμάτων ορυκτών καυσίμων και την εξάρτηση από εξωτερικές πηγές ενέργειας. Πράγματι, η Ευρωπαϊκή Ένωση έχει θέσει ως δεσμευτικό στόχο του 20% του ενεργειακού της εφοδιασμού να προέρχεται από αιολική ενέργεια και άλλες ανανεώσιμες πηγές μέχρι το 2020. Για την επίτευξη αυτού του στόχου του 20% της ενέργειας, περισσότερο από το ένα τρίτο της

ευρωπαϊκής ζήτησης για ηλεκτρική ενέργεια θα πρέπει να προέρχεται από ανανεώσιμες πηγές ενέργειας, με την αιολική ενέργεια να αναμένεται να συνεισφέρει από 14 μέχρι 18%.

Οικονομικά οφέλη της αιολικής ενέργειας:

- ✓ Οικονομική ανάπτυξη και δημιουργία θέσεων εργασίας:

Το 2010, οι επενδύσεις σε νέα αιολικά πάρκα στην Ευρώπη ανήλθαν σε € 13 δισ. Ευρώ. Το 2010, πάνω από το 40% όλων των νέων εγκαταστάσεων παραγωγής ηλεκτρικής ενέργειας στην ΕΕ ήταν από ανανεώσιμες πηγές ενέργειας. Η αιολική ενέργεια μόνο αντιπροσώπευε το 17%. Περίπου 190.000 άνθρωποι στην ΕΕ ήταν σε σχετική με την αιολική ενέργεια απασχόληση το 2010. Η βιομηχανία αιολικής ενέργειας θα μπορούσε να δημιουργήσει έως και 271.000 νέες θέσεις εργασίας στην ΕΕ μέχρι το 2020.

- ✓ Καθαρότερο περιβάλλον:

Τα 84 GW της αιολικής ενέργειας που έχουν εγκατασταθεί μέχρι τα τέλη του 2010 στην ΕΕ, απέφυγαν την εκπομπή 119 εκατομμυρίων τόνων CO₂. Επιπλέον, η αναμενόμενη επένδυση των € 192 δισεκατομμυρίων Ευρώ στην αιολική ενέργεια την περίοδο 2011 - 2020 θα αποφύγει € 85 δισεκατομμύρια Ευρώ σε αξία κόστους εκπομπής CO₂ κατά τη διάρκεια της ίδιας περιόδου.

- ✓ Ενεργειακή ανεξαρτησία:

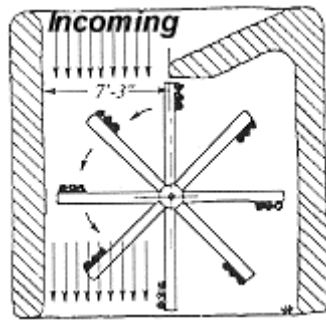
Η Ευρώπη εισάγει σήμερα περισσότερο από το ήμισυ της ενέργειας της, ποσοστό που αναμένεται να φτάσει το 70% μέσα στα επόμενα 20 με 30 χρόνια. Η εγκατεστημένη ισχύς αιολικής ενέργειας μέχρι το τέλος του 2010 ήταν, σε ένα σύνηθες έτος ανέμου, ίση με το 5,3% της ηλεκτρικής ενέργειας της Ευρωπαϊκής Ένωσης. Από το 2011 - 2020 η αιολική ενέργεια θα αποφύγει € 138 δισεκατομμύρια Ευρώ σε κόστος ορυκτών καυσίμων.

ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΑΝΕΜΟΓΕΝΝΗΤΡΙΩΝ

Οι Αρχές της Αιολικής Ενέργειας (5000 π.Χ. - 1300 μ.Χ.)

Από την πρώτη καταγεγραμμένη ιστορία, οι άνθρωποι έχουν εκμεταλλευτεί την ενέργεια του ανέμου. Πλοία αξιοποιούσαν την αιολική ενέργεια για την κίνησή τους κατά μήκος του ποταμού Νείλου ήδη από το 5000 π.Χ. ενώ μέχρι και το 200 π.Χ. ανεμόμυλοι χρησιμοποιούνταν στην Κίνα για την άντληση νερού. Η ιστορία της αιολικής ενέργειας παρουσιάζει μια γενική εξέλιξη από τη χρήση απλών ελαφρών συσκευών που οδηγούνται από τις δυνάμεις του αέρα, σε συσκευές με πιο βαριά υλικά και τελικά στην αυξημένη χρήση ελαφρών αεροδυναμικών διατάξεων και κατασκευών στην σύγχρονη εποχή. Πρέπει να σημειωθεί ότι η αεροδυναμική ανύψωση (η δύναμη που κάνει τα αεροπλάνα να πετάνε) δεν είναι μια σύγχρονη ιδέα που ήταν άγνωστη στους αρχαίους. Η παλαιότερη γνωστή χρήση της αιολικής ενέργειας, φυσικά, είναι το ιστιοφόρο, και αυτή η τεχνολογία είχε σημαντικό αντίκτυπο στην μεταγενέστερη ανάπτυξη ανεμόμυλων με πανιά. Οι αρχαίοι ναυτικοί αντιλαμβάνονταν την ανύψωση και τη χρησιμοποιούσαν καθημερινά, παρόλο που δεν είχαν ακόμα τις απαραίτητες γνώσεις φυσικής για να εξηγήσουν πώς και γιατί λειτουργούσε.

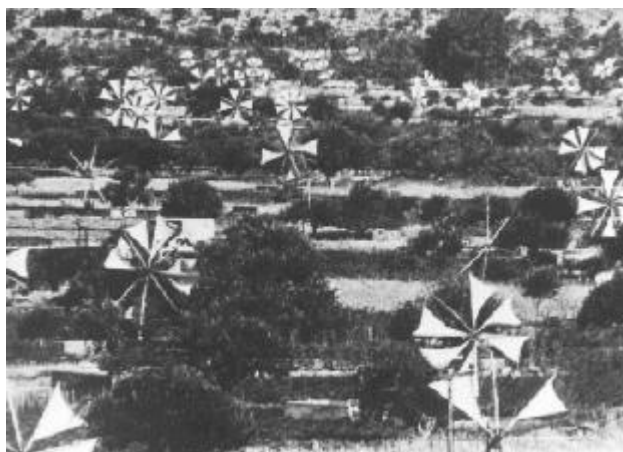
Οι πρώτοι ανεμόμυλοι αναπτύχθηκαν για να αυτοματοποιήσουν καθήκοντα όπως η άλεση του σιταριού και η άντληση νερού. Ο αρχαιότερος γνωστός σχεδιασμός είναι το σύστημα κατακόρυφου άξονα που αναπτύχθηκε στην Περσία περίπου το 500-900 μ.Χ. Η πρώτη χρήση ήταν φαινομενικώς η άντληση νερού αλλά η ακριβής μέθοδος μεταφοράς του νερού δεν είναι γνωστή διότι δεν υπάρχουν ζωγραφιές ή σχέδια παρά μόνο προφορικές αναφορές. Το πρώτο γνωστό τεκμηριωμένο σχέδιο είναι ένας Περσικός ανεμόμυλος, ο οποίος είχε κατακόρυφα πανιά και αποτελούταν από από καλάμια ή ξύλα δεμένα μεταξύ τους που ήταν συνδεδεμένα με τον κεντρικό κατακόρυφο άξονα με οριζόντιους ορθοστάτες (Σχήμα 1). Μια αμερικανική προσέγγιση του 19ου αιώνα αυτής της συσκευής εμφανίζεται στην Εικόνα 2.



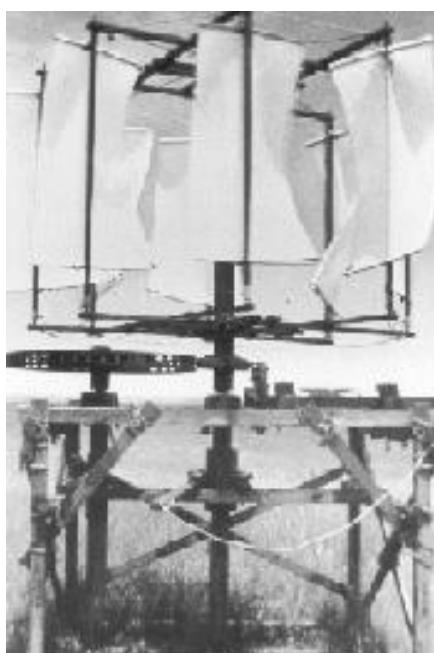
Σχήμα 1.2. Σε αυτό το περσικό σχεδιασμό “panemone”, ο ρότορας μπορεί μόνο να συγκεντρώσει το ήμισυ του ανέμου που χτυπά την περιοχή συλλογής.

Η άλεση των σιτηρών ήταν η πρώτη τεκμηριωμένη εφαρμογή αιολικής ενέργειας και ήταν πολύ απλή. Η πέτρα άλεσης τοποθετήθηκε στον ίδιο κατακόρυφο άξονα. Ο μηχανισμός των μύλων ήταν συνήθως κλεισμένος σε ένα κτίριο, το οποίο περιλάμβανε έναν τοίχο ή ασπίδα για να εμποδίσει τον εισερχόμενο άνεμο να χτυπάει επάνω στον ρότορα και να τον επιβραδύνει. Οι ανεμόμυλοι κάθετων αξόνων χρησιμοποιήθηκαν επίσης στην Κίνα, η οποία συχνά θεωρείται ως η γενέτειρά τους. Ενώ η πεποίθηση ότι ο ανεμόμυλος εφευρέθηκε στην Κίνα περισσότερο από 2000 χρόνια πριν, είναι ευρέως διαδεδομένο και μπορεί να είναι ακριβές, αλλά η παλαιότερη πραγματική τεκμηρίωση ενός κινέζικου ανεμόμυλου ήταν το 1219 μ.Χ. από τον Κινέζο γεωγνώστη YehluChhu-Tshai. Εδώ επίσης, οι πρωτογενείς εφαρμογές προφανώς ήταν η λείανση των κόκκων και η άντληση νερού.

Μια από τις πιο γραφικές και επιτυχημένες εφαρμογές αιολικής ενέργειας (και μία που εξακολουθεί να υπάρχει) είναι η εκτεταμένη χρήση των μηχανών άντλησης νερού στο νησί της Κρήτης. Εδώ, κυριολεκτικά εκατοντάδες ανεμόμυλοι με πανιά αντλούν νερό για καλλιέργειες και ζώα.



Σχήμα 1.3. Ανεμόμυλοι με πανιά για την άντληση νερού στο νησί της Κρήτης



Σχήμα 1.4. Ένας αμερικανικός ranemone του 19ου αιώνα, εμπνευσμένος από τον περσικό, που πιθανώς χρησιμοποιούταν ως στεγνωτήριο ρούχων.

Ανεμόμυλοι στον δυτικό κόσμο (1300 - 1875 Α.Δ.)

Οι πρώτοι ανεμόμυλοι που εμφανίστηκαν στη δυτική Ευρώπη ήταν της διαμόρφωσης οριζόντιου άξονα. Ο λόγος για την ξαφνική εξέλιξη από την περσική σχεδίαση κάθετου άξονα είναι άγνωστος, αλλά το γεγονός ότι οι ευρωπαϊκοί υδρόμυλοι είχαν επίσης διαμόρφωση οριζόντιου άξονα, και προφανώς χρησίμευαν ως τεχνολογικό μοντέλο για τους πρώτους ανεμόμυλους, μπορούν να μας δώσουν ένα μέρος της απάντησης. Ένας άλλος λόγος μπορεί

να είναι η υψηλότερη απόδοση λόγω δομικής κατασκευής των οριζόντιων μηχανών σε σχέση με τις κατακόρυφες μηχανές, οι οποίες χάνουν έως και το 50% της επιφάνειας συλλογής τους λόγω των απαιτήσεων θωράκισης του ρότορα. Οι πρώτες απεικονίσεις (1270 μ.Χ.) δείχνουν ένα μύλο τεσσάρων πτερυγίων τοποθετημένα σε κεντρικό άξονα, το οποίο ήταν ήδη αρκετά προηγμένο τεχνολογικά σε σχέση με τους Περσικούς ανεμόμυλους. Αυτοί οι ανεμόμυλοι χρησιμοποιούσαν ξύλινα γρανάζια (κωνικά) για να μετατρέψουν την οριζόντια κίνηση σε κάθετη ώστε να κινήσουν την πέτρα για την άλεση. Ο μηχανισμός αυτός έπειτα χρησιμοποιήθηκε σε ανεμόμυλους και υδρόμυλους που εφευρέθηκε από τον Vitruvius. Ήδη από το 1390, οι Ολλανδοί θέλησαν να βελτιώσουν τον σχεδιασμό του πύργου που είχε εμφανιστεί κάπως νωρίτερα κατά μήκος της Μεσογείου (Εικόνα 4).



Σχήμα 1.5. Ένας πρώτος μύλος οριζώντιου άξονα με πτερύγια στην ακτή της Μεσογείου.

Ήδη από το 1390, οι Ολλανδοί αποφάσισαν να βελτιώσουν τον σχεδιασμό του πύργου, που είχε εμφανιστεί κάπως νωρίτερα στη Μεσόγειο. (Εικόνα 4). Οι Ολλανδοί εγκατέστησαν κατ'ουσίαν τον σιδερένιο ανεμόμυλο στην κορυφή ενός πολυώροφου πύργου, με ξεχωριστά δάπεδα αφιερωμένα στην άλεση των σιτηρών, την αφαίρεση του φλοιού των σιταριών, την αποθήκευση των σιτηρών και στο κάτω μέρος των διαμερισμάτων έμεναν οι άνθρωποι με τις οικογένειές τους. Τόσο ο μεταλλικός ανεμόμυλος όσο και ο μεταγενέστερος σχεδιασμός του έπρεπε να προσανατολίζονται χειροκίνητα στον άνεμο, σπρώχνοντας ένα μεγάλο μοχλό στο

πίσω μέρος του μύλου. Η βελτιστοποίηση της ενέργειας και της ισχύος του ανεμόμυλου και η προστασία του από τις ζημιές στα πτερύγια κατά τη διάρκεια των καταιγίδων ήταν μεταξύ των πρωταρχικών εργασιών του σιδηρουργού του ανεμόμυλου. Πρωταρχική βελτίωση των ευρωπαϊκών ανεμόμυλων ήταν η χρήση κατάλληλα σχεδιασμένων πανιών που δημιούργησαν αεροδυναμική ανύψωση. (Εικόνα 5). Αυτό το χαρακτηριστικό γνώρισμα βελτίωσε την απόδοση του ρότορα σε σύγκριση με τους περσικούς ανεμόμυλους, επιτρέποντας έτσι την αύξηση της ταχύτητας του ρότορα, η οποία επέτρεψε επίσης καλύτερη άλεση και άντληση νερού. Η διαδικασία τελειοποίησης των πάνινων πτερυγίων του ανεμόμυλου, αυξάνοντας την αποτελεσματικότητα, χρειάστηκε 500 χρόνια. Μέχρι τη στιγμή που ολοκληρώθηκε η διαδικασία, τα πανιά ανεμόμυλων είχαν όλα τα κύρια χαρακτηριστικά γνωρίσματα που αναγνωρίζονται από τους μοντέρνους σχεδιαστές ως κρίσιμα για την απόδοση των σύγχρονων λεπίδων ανεμογεννητριών, συμπεριλαμβανομένης της 1)κάμψης κατά μήκος της αιχμής, 2)τοποθέτηση της σπάτουλας 3) κέντρο βάρους στην ίδια θέση 1/4 χορδών και 4)μη γραμμική συστροφή της λεπίδας από τη ρίζα στην άκρη (Drees, 1977). Ορισμένα μοντέλα παρουσίαζαν επίσης αεροδυναμικά φρένα, spoilers και πτερύγια. Το μηχάνημα που φαίνεται στο σχήμα 4 (το οποίο λειτουργούσε με δύο από τους φίλους του αντλώντας νερό περίπου ένα μέτρο από μια λίμνη άρδευσης σε μια άλλη στην Ολλανδία το 1994) διαθέτει τμήματα αεροδυναμικής αιχμής. Αυτοί οι μύλοι ήταν ο «ηλεκτροκινητήρας» της προ-βιομηχανικής Ευρώπης. Οι εφαρμογές ήταν ποικίλες, κυμαινόμενες από την κοινή άντληση νερού, την άρδευση ή την άντληση αποστράγγισης χρησιμοποιώντας έναν τροχό σέσουλας (μονή ή διαδοχική), λείανση με κόκκους (και πάλι, χρησιμοποιώντας μονές ή πολλαπλές πέτρες), πριόνισμα της ξυλείας και επεξεργασία άλλων εμπορευμάτων όπως τα μπαχαρικά, το κακάο, τα χρώματα και τις βαφές, και ο καπνός. Συνεχίζοντας τον 19ο αιώνα, η χρήση των μεγάλων πύργων μειώθηκε με την αυξημένη χρήση ατμομηχανών. Η επόμενη εξέλιξη της ανάπτυξης της αιολικής ενέργειας συνέβη πολλά χιλιόμετρα δυτικά.



Σχήμα 1.6. Ένας ολλανδικός ανεμόμυλος (1994) που διαθέτει τμήματα αεροδυναμικής αιχμής (πάνω δεξιά). Ο μηχανισμός που χρησιμοποιείται για την περιστροφή του ρότορα στον άνεμο και τα παράθυρα του πρώτου ορόφου όπου ζούσαν οι άνθρωποι είναι εύκολα αντιληπτά.

Ο ρόλος των μικρότερων συστημάτων

Για εκατοντάδες χρόνια, η πιο σημαντική εφαρμογή των ανεμόμυλων στο επίπεδο διαβίωσης ήταν η μηχανική άντληση νερού χρησιμοποιώντας σχετικά μικρά συστήματα με διαμέτρους ρότορα ενός έως μερικών μέτρων. Αυτά τα συστήματα τελειοποιήθηκαν στις Ηνωμένες Πολιτείες κατά τη διάρκεια του 19ου αιώνα, ξεκινώντας με τον ανεμόμυλο Halladay το 1854 και συνεχίζοντας με τα σχέδια Aermotor και Dempster, τα οποία χρησιμοποιούνται ακόμα σήμερα. Οι πρώτοι μύλοι είχαν τέσσερα ξύλινα πτερύγια σαν κουπί. Ακολούθησαν μύλοι με λεπτές ξύλινες σχάρες που καρφώθηκαν σε ξύλινες ζάντες. Τα περισσότερα από αυτά τα ελαιοτριβεία είχαν ουρές για να τους προσανατολίσουν στον άνεμο, αλλά μερικοί ήταν μύλοι που έτρεχαν με καιρικές συνθήκες που λειτουργούσαν κάτω από τον πύργο. Ο έλεγχος ταχύτητας ορισμένων μοντέλων παρέχεται με αρθρωτά τμήματα λεπίδων, έτσι ώστε να αναδιπλώνουν σαν ομπρέλα σε μεγάλους ανέμους, πράγμα που μείωσε την περιοχή σύλληψης του δρομέα για να μειώσει την ώθηση. Η πιο σημαντική βελτίωση του αμερικανικού ανεμόμυλου τύπου ανεμιστήρα ήταν η ανάπτυξη χαλύβδινων λεπίδων το 1870 (Σχήμα 5).

Τα χαλύβδινα πτερύγια θα μπορούσαν να γίνουν ελαφρύτερα και να εργαστούν σε πιο αποτελεσματικά σχήματα. Δούλεψαν πολύ καλά, στην πραγματικότητα, ότι η υψηλή ταχύτητά τους απαιτούσε μια μειωτική (αργή) ταχύτητα για να γυρίσουν τις τυποποιημένες αντλίες με την απαιτούμενη ταχύτητα. Μεταξύ του 1850 και του 1970, πάνω από έξι εκατομμύρια κυρίως μικρές (1 ίππων ή λιγότερο) μηχανικές ανεμογεννήτριες που

εγκαταστάθηκαν, εγκαταστάθηκαν μόνο στις Η.Π.Α. Η κύρια χρήση ήταν η άντληση νερού και οι κυριότερες εφαρμογές ήταν οι πότισμα των αποθεμάτων και οι ανάγκες των νοικοκυριών στο σπίτι. Πολύ μεγάλοι ανεμόμυλοι, με ρότορες έως και 18 μέτρα σε διάμετρο, χρησιμοποιήθηκαν για την άντληση νερού για τις σιδηροδρομικές αμαξοστοιχίες ατμού που παρείχαν την κύρια πηγή εμπορικής μεταφοράς σε περιοχές όπου δεν υπήρχαν πλωτές ποτάμιες. Στα τέλη του 19ου αιώνα, ο επιτυχημένος "αμερικανικός" σχεδιασμός ανεμόμυλων πολλαπλών λεπίδων χρησιμοποιήθηκε στον πρώτο μεγάλο ανεμόμυλο για να παράγει ηλεκτρική ενέργεια.

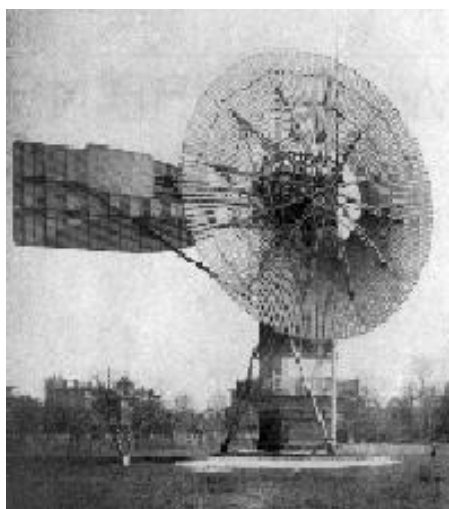


Σχήμα 1.7. Ένας ανεμόμυλος άντλησης νερού με χαλύβδινο νερό στην Αμερικάνικη Midwest (τέλη του 1800)

Πρώτη χρήση του ανέμου για την παραγωγή ηλεκτρικής ενέργειας μεγάλης κλίμακας

Η πιο προφανής επίδραση στην αιολική ενέργεια του 20ου αιώνα ήταν η αυξανόμενη χρήση ηλεκτρισμού. Αλλά αυτό ξεκίνησε με μια ματιά στο παρελθόν. Η πρώτη χρήση ενός μεγάλου ανεμόμυλου για την παραγωγή ηλεκτρισμού ήταν ένα σύστημα που χτίστηκε στο Κλίβελαντ του Οχάιο το 1888 από τον Charles F. Brush. Το μηχάνημα Βούρτσας (που εμφανίζεται στα δεξιά) ήταν ένας postmill με έναν πολλαπλών πτερυγίων ρότορα "πύργου-φράχτη" 17 μέτρων σε διάμετρο, που διαθέτει μια μεγάλη ουρά αρθρωτή για να γυρίσει τον ρότορα έξω από τον άνεμο. Ήταν ο πρώτος ανεμόμυλος για να ενσωματώσει ένα κιβώτιο ταχυτήτων (με

αναλογία 50: 1) για να μετατρέψει μια γεννήτρια συνεχούς ρεύματος στην απαιτούμενη λειτουργική ταχύτητα (στην περίπτωση αυτή 500 σ.α.λ.). Παρά τη σχετική επιτυχία της λειτουργίας της για 20 χρόνια, ο ανεμόμυλος Brush έδειξε τους περιορισμούς του δρομολογητή χαμηλής ταχύτητας και υψηλής αντοχής για εφαρμογές παραγωγής ηλεκτρικής ενέργειας. Τα 12 κιλοβάτ που παρήχθησαν από τον ρότορα 17 μέτρων του παρακμάζουν δίπλα στα 70-100 κιλοβάτ που παράγονται από έναν σύγχρονο, ανελκυστήρα τύπου, συγκρίσιμου μεγέθους. Το 1891, ο Dane Poul LaCour ανέπτυξε την πρώτη μηχανή ανεμογεννήτριας για να ενσωματώσει τις αρχές αεροδυναμικής σχεδίασης (στροφεία με χαμηλή αντοχή, τεσσάρων λεπίδων που ενσωματώνουν πρωτόγονα σχήματα αεροτομής) που χρησιμοποιούνται στα καλύτερα ευρωπαϊκά πύργους πύργων. Η υψηλότερη ταχύτητα της κίνησης του ρότορα LaCour έκαναν τα μύδια αυτά αρκετά πρακτικά για την παραγωγή ηλεκτρικής ενέργειας. Μέχρι το τέλος του Α΄ Παγκοσμίου Πολέμου, η χρήση ηλεκτρικών μηχανών παραγωγής 25 κιλοβάτ είχε εξαπλωθεί σε ολόκληρη τη Δανία, αλλά φθηνότερα και μεγαλύτερα εργοστάσια παραγωγής ατμού με ορυκτά καύσιμα έβαλαν σύντομα τους επιχειρηματίες αυτών των ελαιοτριβείων εκτός λειτουργίας. Μέχρι το 1920, οι δύο κυρίαρχες διαμορφώσεις ρότορα (τύπου ανεμιστήρα και ιστία) είχαν δοκιμαστεί και βρέθηκαν ανεπαρκείς για τη δημιουργία αξιόλογων ποσοτήτων ηλεκτρικής ενέργειας. Η περαιτέρω ανάπτυξη των ηλεκτρικών συστημάτων ανεμογεννήτριας στις Ηνωμένες Πολιτείες εμπνεύστηκε από το σχεδιασμό ελίκων αεροπλάνων και (αργότερα) χαμηλών ταχυτήτων μονοπλάνων με ταχύτητα ανέμου.



Σχήμα 1.8. Η βρωμιά μετάλλων στο Cleveland, Ohio, 1888. Η πρώτη χρήση ενός μεγάλου ανεμόμυλου για την παραγωγή ηλεκτρικής ενέργειας. Σημειώστε τον άνθρωπο που κόβει το γκαζόν στα δεξιά.

Πρωτοπόροι μικρών συστημάτων

Οι πρώτες μικρές ανεμογεννήτριες με ηλεκτρική ισχύ χρησιμοποίησαν απλώς τροποποιημένους έλικες για την οδήγηση γεννητριών συνεχούς ρεύματος. Μέχρι τα μέσα της δεκαετίας του 1920, 1 έως 3 κίλοβατώρες ανεμογεννήτριες που αναπτύχθηκαν από εταιρείες όπως το Parris-Dunn και το Jacobs Wind-electric βρίσκουν ευρεία χρήση στις αγροτικές περιοχές της Μεσογείου Great Plains. (Μια μονάδα Jacobs 3 κίλοβατώρες εμφανίζεται δεξιά και ρυθμίζεται από ένα Jacobs ML σε Jacobs στο Rocky Flats το 1977.)

Τα συστήματα αυτά εγκαθίστανται αρχικά για να παρέχουν φωτισμό για τα αγροκτήματα και να φορτίζουν τις μπαταρίες που χρησιμοποιούνται για την τροφοδοσία κρυστάλλων σκηνικά. Αλλά η χρήση τους επεκτάθηκε σε μια ολόκληρη σειρά συσκευών με κινητήρα με άμεσο ρεύμα, όπως ψυγεία, καταψύκτες, πλυντήρια ρούχων και ηλεκτρικά εργαλεία. Όμως, όσο περισσότερες συσκευές τροφοδοτούνται από τις πρώτες γεννήτριες ανέμου, τόσο περισσότερο η διαλείπουσα λειτουργία τους έγινε πρόβλημα. Η εικονική κατάρρευση των συστημάτων αυτών επιταχύνθηκε στα τέλη της δεκαετίας του 1930 και της δεκαετίας του 1940 από δύο παράγοντες: τη ζήτηση των αγροκτημάτων για όλο και μεγαλύτερες ποσότητες ενέργειας κατόπιν ζήτησης και τη Μεγάλη Ύφεση, που ώθησε την ομοσπονδιακή κυβέρνηση των ΗΠΑ να διεγείρει τις υποβαθμισμένες αγροτικές οικονομίες επεκτείνοντας το ηλεκτρικό δίκτυο σε όλες τις περιοχές. Έχουν γίνει πολλά για αυτή την εξέλιξη και πόσο φρικτό ήταν να παρέμβει η κυβέρνηση. (Σε αυτό το σημείο, στα περισσότερα ντοκιμαντέρ για την αιολική ενέργεια, υπάρχει μια φευγαλέα φωνή μιας αρμονικής και μια βολή ενός σκουριασμένου ανεμοστρόβιλου). Αλλά αμφιβάλλω ότι οι αγρότες που βοήθησαν τα νέα ηλεκτρικά δίκτυα θα μοιραστούν αυτό το συναίσθημα. Και η αυξανόμενη ζήτηση για ηλεκτρική ενέργεια που δημιουργήθηκε από την ανεμογεννήτρια, σε συνδυασμό με την αδυναμία της τεχνολογίας να προσαρμοστεί, βοήθησε να καταστεί η κατάσταση αναπόφευκτη. Η πρώιμη επιτυχία των ανεμογεννητριών Midwest έθεσε στην πραγματικότητα τη σκηνή για τη δυνατότητα μεγαλύτερης ανάπτυξης της αιολικής ενέργειας στο μέλλον. Ενώ η αγορά νέων μικρών ανεμογεννητριών οποιουδήποτε τύπου είχε διαβρωθεί σε μεγάλο βαθμό στις Ηνωμένες

Πολιτείες μέχρι το 1950, η χρήση μηχανικού και ηλεκτρικού συστήματος συνεχίστηκε σε ολόκληρη την Ευρώπη και σε θυελλώδη και ξηρά κλίματα όπως αυτά που βρίσκονται σε μέρη της Αφρικής και της Αυστραλίας.



Σχήμα 1.9. M.L. Jacobs ρυθμίζοντας τον μηχανισμό αλλαγής κλίσης που ενεργοποιείται με ελατήριο σε ένα JacobsWind-electric το 1977.

"Μαζική" ισχύς από τον άνεμο

Η ανάπτυξη συστημάτων μετατροπής αιολικής ενέργειας σε χύδην-ηλεκτρικού ρεύματος χρησιμοποιήθηκε για πρώτη φορά στη Ρωσία το 1931 με την ανεμογεννήτρια 100kW Balaclava. Αυτό το μηχάνημα λειτουργεί για περίπου δύο χρόνια στην ακτή της Κασπίας Θάλασσας, δημιουργώντας 200.000 kWh ηλεκτρικής ενέργειας. Τα επακόλουθα πειραματικά αιολικά πάρκα στις Ηνωμένες Πολιτείες, τη Δανία, τη Γαλλία, τη Γερμανία και τη Μεγάλη Βρετανία κατά την περίοδο 1935-1970 έδειξαν ότι θα λειτουργούσαν μεγάλης κλίμακας ανεμογεννήτριες, αλλά δεν κατέληξαν σε πρακτική μεγάλη ηλεκτρική ανεμογεννήτρια. Η μεγαλύτερη ήταν η μηχανή Smith-Putnam μήκους 1,25 μεγαβάτ (Σχήμα 8), που εγκαταστάθηκε στο Βερμόντ το 1941. Αυτός ο σχεδιασμός οριζόντιας άξονας χαρακτηριζόταν από έναν ρότορα διαμέτρου 175 ποδιών προσανατολισμένο προς τα κάτω στον άνεμο του πύργου. Ο ρότορας ανοξείδωτου χάλυβα 16 τόνων χρησιμοποίησε έλεγχο πλήρους πλήκτρου πλήκτρων για να διατηρήσει τη λειτουργία του σε 28 σ.α.λ. Το 1945, μετά από μερικές μόνο εκατοντάδες ώρες διακοπτόμενης λειτουργίας, μία από τις λεπίδες έσπασε κοντά στον κόμβο, προφανώς ως αποτέλεσμα κόπωσης μετάλλου. Αυτό δεν προκαλεί έκπληξη λαμβάνοντας υπόψη τα τεράστια φορτία που πρέπει να έχουν δημιουργηθεί σε μια δομή που είχε πολλά κοινά με ένα γιγαντιαίο σετ περιστρεφόμενου σκυροδέματος.



Σχήμα 1.10. Η ανεμογεννήτρια 1,25-μεγαβάτ του PalmerPutnam ήταν ένα από τα θαύματα της τεχνολογίας στα τέλη της δεκαετίας του 1930, αλλά το άλμα σε κλίμακα ήταν υπερβολικά μεγάλο για τα διαθέσιμα υλικά.

Ευρωπαϊκή Ανάπτυξη

Οι ευρωπαϊκές εξελίξεις συνεχίστηκαν μετά τον Δεύτερο Παγκόσμιο Πόλεμο, όταν η προσωρινή έλλειψη ορυκτών καυσίμων οδήγησε σε υψηλότερο ενεργειακό κόστος. Όπως και στις Ηνωμένες Πολιτείες, η πρωταρχική εφαρμογή για τα συστήματα αυτά ήταν η διασύνδεση με το ηλεκτρικό δίκτυο. Στη Δανία, η ανεμογεννήτρια 200 kW GedserMill λειτούργησε με επιτυχία μέχρι τις αρχές της δεκαετίας του 1960, όταν η πτώση των τιμών των ορυκτών καυσίμων έκανε πάλι την αιολική ενέργεια μη ανταγωνιστική με μονάδες παραγωγής ηλεκτρικής ενέργειας με ατμό. Αυτό το μηχάνημα χαρακτηρίστηκε από έναν τρίχρονο ανεμογεννήτη ρότορα με σταθερά πτερύγια ανύψωσης που χρησιμοποίησαν μηχανική ανεμόμυλου ενισχυμένη με δομή υποστήριξης αεροσκάφους. Το σχέδιο ήταν πολύ λιγότερο μηχανικά πολύπλοκο από το σχέδιο Smith-Putnam. Στην πραγματικότητα, δεν ήταν τόσο μακριά από τον ανεμόμυλο της εποχής του 1920 στην Roullacour (γεγονός που λειτούργησε προς όφελός του).



Σχήμα 1.11. Ναι, αυτό είναι ένα αεροσκάφος που συγκρατεί τα τρία πτερύγια του "GedserMollen." Τα υαλοβάμβακα αργότερα εξάλειψαν αυτή την απαίτηση σχεδιασμού.

Στη Γερμανία, ο καθηγητής UlrichHutter ανέπτυξε μια σειρά προωθημένων οριζόντιων σχεδίων μεσαίου μεγέθους, τα οποία χρησιμοποίησαν μοντέρνα λεπίδες από πολυεστερικό και πλαστικό με διαμήκη κλίση για να παρέχουν ελαφρύ βάρος και υψηλή απόδοση. Αυτή η προσέγγιση σχεδιασμού επιδιώκει να μειώσει τις φέρουσες και κατασκευαστικές βλάβες με την «αποδέσμευση» των αεροδυναμικών φορτίων αντί να «αντέχει» τους όπως και η δανική προσέγγιση. Ένα από τα πιο καινοτόμα χαρακτηριστικά σχεδίασης φορτίων ήταν η χρήση ενός ρουλεμάν στο πλήμνη του ρότορα που έδινε τη δυνατότητα στον στροφέα να "τείνει" σε απόκριση των ριπών ανέμου και της κατακόρυφης διάτμησης του ανέμου. Τα προηγμένα σχέδια του Hutter πέτυχαν πάνω από 4000 ώρες λειτουργίας πριν τελειώσουν τα πειράματα το 1968. Η μεταπολεμική δραστηριότητα στη Δανία και τη Γερμανία υπαγόρευσε σε μεγάλο βαθμό τις δύο μεγάλες προσεγγίσεις σχεδιασμού οριζόντιου άξονα που θα προκύψουν όταν επιστρέψει η προσοχή στην ανάπτυξη ανεμογεννητριών στις αρχές της δεκαετίας του 1970. Οι Δανοί επεξεργάστηκαν το απλό, σταθερό βήμα, το σχεδιασμό GedserMill, χρησιμοποιώντας προηγμένα υλικά, βελτιωμένο αεροδυναμικό σχεδιασμό και αεροδυναμικά στοιχεία ελέγχου για να μειώσουν μερικές από τις αδυναμίες του. Οι τεχνολογικές καινοτομίες των γερμανικών μηχανών ελαφρού βάρους, υψηλότερης απόδοσης, όπως η μεντεσέδες στο κέντρο του ρότορα, χρησιμοποιήθηκαν αργότερα από

τους σχεδιαστές των ΗΠΑ. Η ανάπτυξη σύγχρονων δρομέων κάθετου άξονα ξεκίνησε στη Γαλλία από τον G.J.M. O Darrieus στη δεκαετία του 1920. Από τους πολλούς σχεδιασμένους ρότορες Darrieus, ο σημαντικότερος είναι ένας ρότορας που περιλαμβάνει λεπτούς, καμπυλωμένους, λεπίδες διατομής αεροτομής προσαρτημένους στην κορυφή και στον πυθμένα ενός περιστρεφόμενου κατακόρυφου σωλήνα. Σημαντικές εργασίες ανάπτυξης αυτής της έννοιας δεν ξεκίνησαν μέχρι την επανεξέταση της ιδέας στα τέλη της δεκαετίας του 1960 από δύο канаδούς ερευνητές. Οι προσπάθειες των ΗΠΑ με την ιδέα Darrieus στα Sandia National Laboratories ξεκίνησαν μετά το εμπάργκο πετρελαίου του 1973, με την είσοδο του Αμερικανικού Ομοσπονδιακού Προγράμματος Αιολικής Ενέργειας στον κύκλο ανάπτυξης της αιολικής ενέργειας.

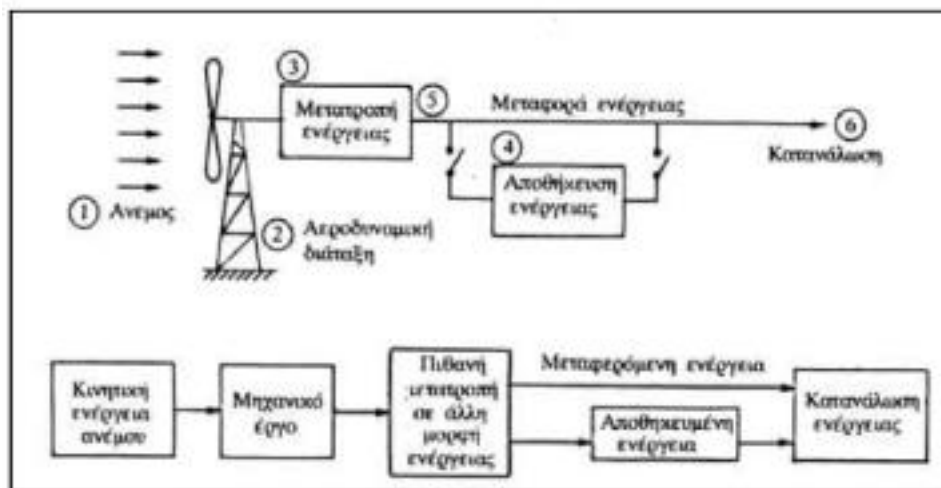


Σχήμα 1.12. Οι ανεμογεννήτριες του Hutter, όπως και άλλες γερμανικές συσκευές των μέσων του 20ού αιώνα, προωθήθηκαν για την εποχή τους.

Οι Ανεμογεννήτριες σήμερα

Οι ανεμογεννήτριες είναι μηχανές που μετατρέπουν την κινητική ενέργεια του ανέμου σε ηλεκτρική ενέργεια. Κατά τη λειτουργία τους, ο αέρας περιστρέφει τα πτερύγια τα οποία είναι συνδεδεμένα με έναν περιστρεφόμενο άξονα. Αυτός ο άξονας περνάει σε ένα κιβώτιο μετάδοσης κίνησης το οποίο συνδέεται με έναν δεύτερο άξονα μέσω του οποίου κινείται μια γεννήτρια για την παραγωγή ηλεκτρικού ρεύματος. Οι σημερινές ανεμογεννήτριες μπορούν

να παράγουν από μερικές δεκάδες ή εκατοντάδες W μέχρι μερικά MW.



Σχήμα 1.13 Βασικά μέρη διάταξης για την εκμετάλλευση της αιολικής ενέργειας καθώς επίσης και διάγραμμα ροής της ενέργειας.

Οι σύγχρονες ανεμογεννήτριες είναι πολύ πιο μεγάλες και πιο πολύπλοκες, τόσο στον εξωτερικό όσο και στον εσωτερικό σχεδιασμό τους. Μπορούν να ταξινομηθούν σύμφωνα με τον προσανατολισμό των αξόνων τους σε σχέση με τη ροή του ανέμου σε οριζοντίου άξονα (HAWTs) και καθέτου άξονα (VAWTs):

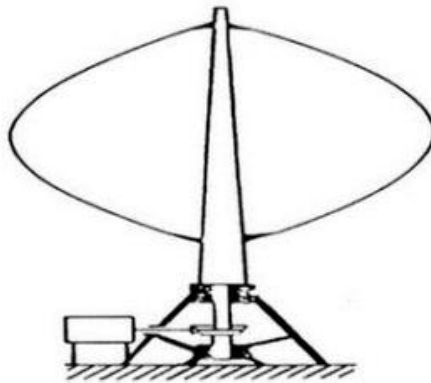
- i. Οριζοντίου άξονα, στους οποίους ο άξονας περιστροφής του δρομέα είναι παράλληλος προς την κατεύθυνση του ανέμου.
- ii. Οριζοντίου άξονα (Cross Wind), στους οποίους ο άξονας περιστροφής είναι παράλληλος προς την επιφάνεια της γης αλλά κάθετος στην κατεύθυνση ροής του ανέμου.
- iii. Καθέτου άξονα στους οποίους ο άξονας περιστροφής είναι κάθετος στην επιφάνεια της γης και κάθετος στη ροή του ανέμου.

Το μεγαλύτερο ποσοστό ανεμογεννητριών που χρησιμοποιούνται σήμερα είναι οριζοντίου άξονα.

Είδη Ανεμογεννητριών

Ανεμογεννήτριες Κάθετου Άξονα (VAWTs)

Στις ανεμογεννήτριες κάθετου άξονα η άτρακτος είναι τοποθετημένη σε κατακόρυφο άξονα και αποτελεί συνήθως τον πύργο στήριξης της αιολικής μηχανής. Συνηθέστερα ο πύργος αυτός προσδένεται και με συρματόσκοινα, ενώ τώρα τελευταία κατασκευάζονται πύργοι αυτοστηριζόμενοι. Όλος ο εξοπλισμός βρίσκεται στο επίπεδο του εδάφους για εύκολη εγκατάσταση και συντήρηση αλλά αυτό σημαίνει μεγαλύτερο αποτύπωμα για τον ανεμοκινητήρα, το οποίο είναι μεγάλο μειονέκτημα κυρίως στις αγροτικές περιοχές.



Σχήμα 1.14: Αυτοστηριζόμενος Darrieus

Οι ανεμοκινητήρες αυτού του τύπου είναι κατασκευαστικά απλούστεροι του ανεμοκινητήρα οριζοντίου άξονα γιατί:

- Δεν απαιτούν περύγιο ή σύστημα αυτοματισμού για τον προσανατολισμό του δρομέα στη διεύθυνση πνοής του ανέμου.
- Το σύστημα μετατροπής της μηχανικής ενέργειας του δρομέα σε άλλη μορφή ενέργειας βρίσκεται στο έδαφος, στη βάση του ανεμοκινητήρα.
- Τα έξοδα αυτοματισμού, συντήρησης ή επισκευών είναι σαφώς μικρότερα σε σύγκριση με τον ανεμοκινητήρα οριζοντίου άξονα.

Ο ανεμοκινητήρας Savonius

Οι ανεμοκινητήρες αυτού του τύπου πρωτοπαρουσιάστηκαν το 1931 από τον Savonius και τα βασικά τους χαρακτηριστικά είναι ο χαμηλός συντελεστής ισχύος ($C_{pmax}=0.19$), ο μικρός λόγος ταχύτητας ακροπτερυγίου $\lambda=(\omega R/V)=1$, για βέλτιστο

συντελεστή ισχύος (δηλαδή μικρή ακραία περιφερειακή ταχύτητα), το περιορισμένο μέγεθος και η εξαιρετική απλότητα και οικονομικότητα της κατασκευής. Πάρα τα μειονεκτήματα, το τελευταίο πλεονέκτημα, σε συνδυασμό με το γεγονός ότι δεν χρειάζεται σύστημα προσανατολισμού προς τον άνεμο, είχε δώσει ώθηση σε μια σειρά από έρευνες πάνω στο δρομέα Savonius για την εύρεση του καλύτερου συνδυασμού των διαφόρων παραμέτρων. Τέτοιες παράμετροι είναι ο λόγος ύψους προς διάμετρο, το άνοιγμα χ μεταξύ των πτερυγίων, ο αριθμός των πτερυγίων (2, 3 ή περισσότερα), το σχήμα των πτερυγίων και άλλα. Ο δρομέας Savonius λόγω του ότι είναι κατακόρυφου άξονα και παρουσιάζει συμμετρία προσανατολίζεται αυτόματα κατά τη διεύθυνση του ανέμου και δεν έχει συνεπώς ανάγκη των ακριβών διατάξεων προσανατολισμού του οριζοντίου άξονα. Έχει το πλεονέκτημα να εκκινεί μόνος του και μάλιστα παρουσιάζει μεγάλη ροπή εκκίνησης και λειτουργίας. Επίσης παρουσιάζει μεγάλο βάρος ανά μονάδα ισχύος και η επιφάνεια του είναι τριάντα φορές μεγαλύτερη από τους άλλους ανεμοκινητήρες για την ίδια ισχύ. Οι αεροδυναμικές μελέτες για τον δρομέα αυτό είναι πάρα πολύ λίγες γεγονός που οφείλεται στο περιορισμένο βιομηχανικό ενδιαφέρον και την ιδιαίτερη δυσκολία μελέτης ροής γύρω από τον δρομέα. Το πεδίο ταχυτήτων είναι χρονικά μεταβαλλόμενο με περιοδικότητα διπλάσια της γωνιακής ταχύτητας του δρομέα. Το πεδίο χαρακτηρίζεται από έντονες αποκολλήσεις της ροής και από εκπομπή δινών από την ακμή εκφυγής των πτερυγίων γεγονός που κάνει τις γνωστές τεχνικές μετρήσεις ταχυτήτων δύσκολο αν όχι αδύνατο να χρησιμοποιηθούν. Για τους παραπάνω λόγους δεν υπάρχουν ακόμα και σήμερα πλήρεις θεωρητικές μελέτες σχεδίασης του δρομέα.

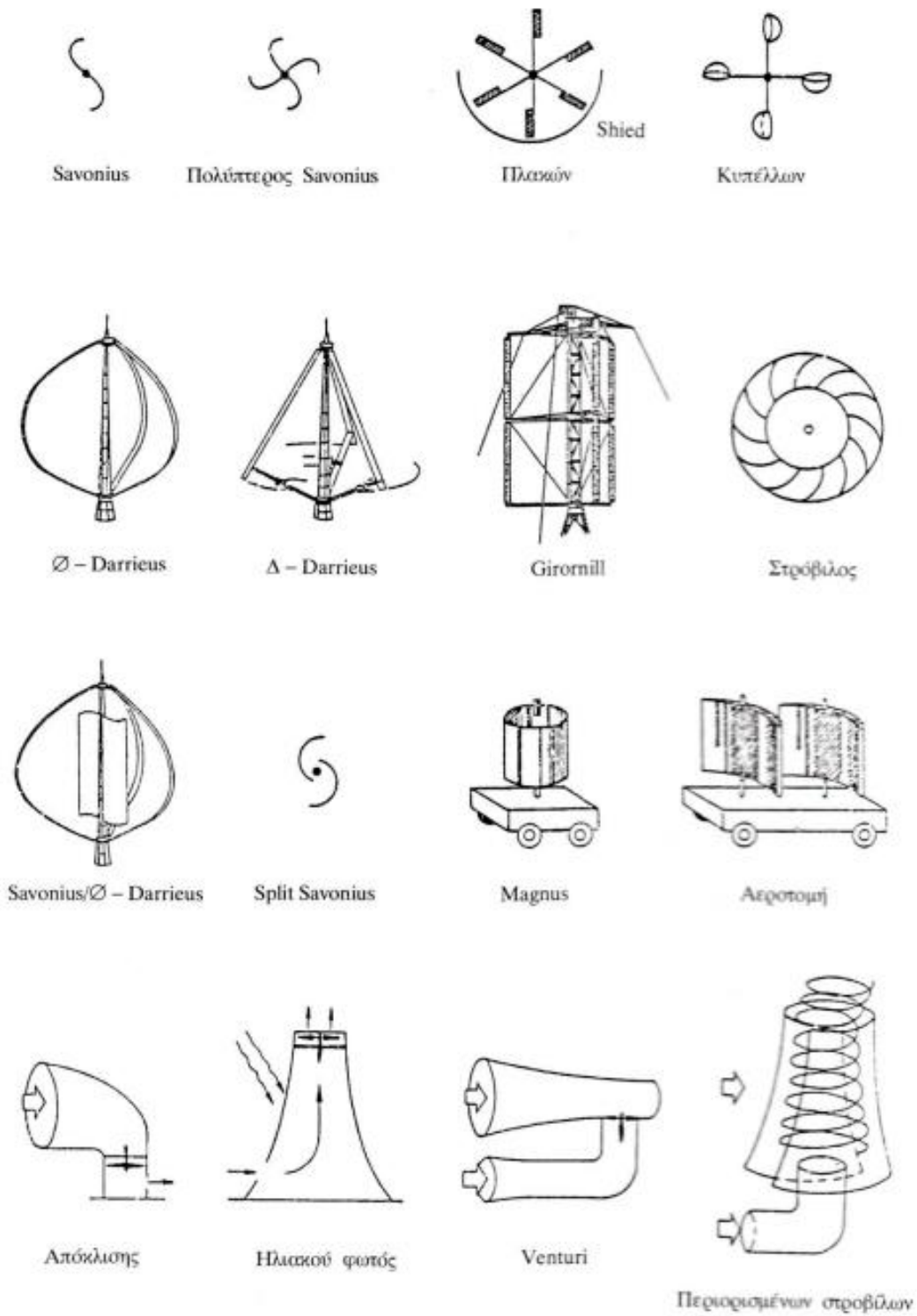
Ο ανεμοκινητήρας Darrieus

Οι ανεμογεννήτριες με κάθετο άξονα είναι αρκετά σπάνιες και για την ακρίβεια η μόνη που είναι επί του παρόντος στην εμπορική παραγωγή είναι ο ανεμοκινητήρας του Darrieus. Επινοήθηκε από τον Γάλλο μηχανικό G.J.M.c γύρω στα 1920 και έτυχε εκτεταμένης ανάπτυξης και εφαρμογής στον Καναδά, κυρίως στη δεκαετία του 1970 οπότε και έγινε ευρύτερα γνωστός. Ο ανεμοκινητήρας Darrieus είναι μια μηχανή που χαρακτηρίζεται από καμπυλωτά πτερύγια (Egg beater). Έχει σχετικά χαμηλή αρχική ροπή εκκίνησης και ως εκ τούτου έχει το μειονέκτημα να μη ξεκινάει μόνος του όταν φυσάει ο άνεμος. Συνδυασμός όμως του ανεμοκινητήρα Darrieus και ενός μικρού Savonius επιλύει το τεχνολογικό αυτό

πρόβλημα. Μια άλλη παραλλαγή είναι η λεγόμενη ανεμογεννήτρια μορφής Η. Σε αυτόν τον τύπο, τα πτερύγια είναι ευθύγραμμα και συνδέονται με τον κατακόρυφο άξονα περιστροφής με μικρούς στυλίσκους. Παρά τις πειραματικές προσπάθειες βελτιστοποίησης της λειτουργίας αυτού του τύπου, μέχρι και σήμερα δεν έχει επιτευχθεί ο χαρακτηρισμός τους ως οικονομικά βιώσιμων



Σχήμα 1.15. πάνω αριστερά: Ανεμοκινητήρας Darrieus, πάνω δεξιά: Ανεμοκινητήρας Savonius κάτω αριστερά : Hybrid Darrieus-Savonius και κάτω δεξιά: Ανεμοκινητήρας τύπου Η



Σχήμα 1.16. Τύποι Ανεμοκινητήρων κατακόρυφου άξονα

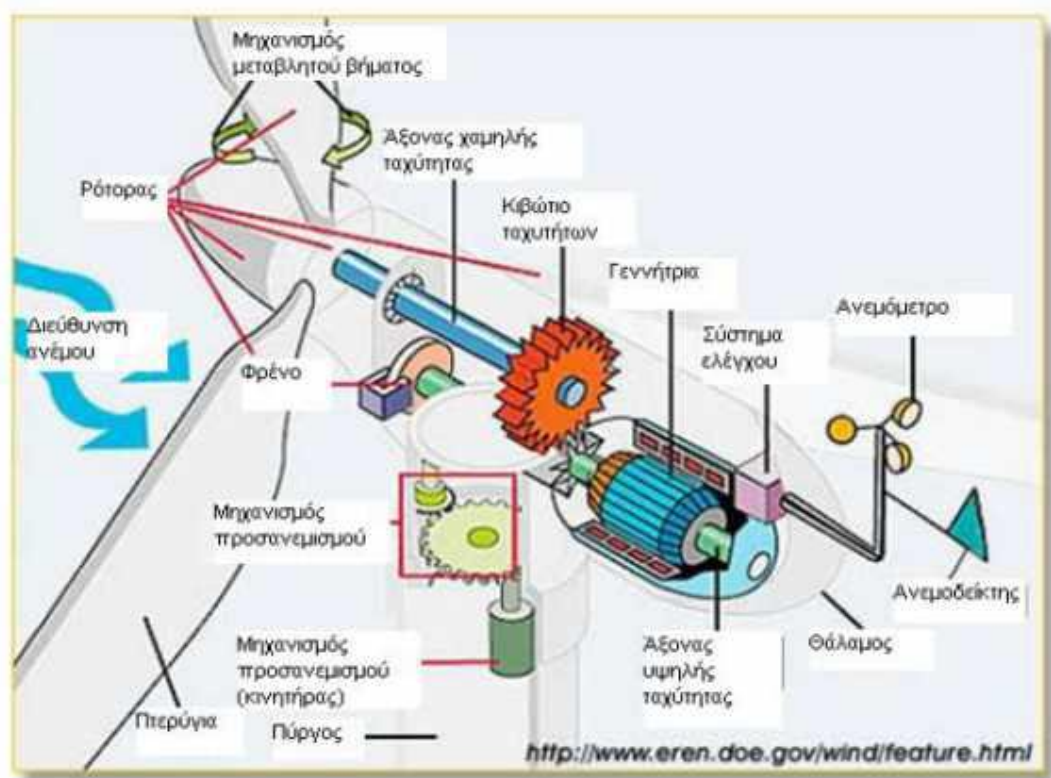
Ανεμογεννήτριες Οριζοντίου Άξονα (HAWTs)

Όπως υπονοείται από το όνομα, ο άξονας των HAWT είναι τοποθετημένος οριζόντια, παράλληλα με το έδαφος. Για την μεγιστοποίηση δέσμευσης της κινητικής ενέργειας του ανέμου απαιτείται όπως το επίπεδο του δρομέα του ανεμοκινητήρα να είναι κάθετο στην κατεύθυνση του ανέμου. Για το σκοπό αυτό στους μεν μικρής ισχύος ανεμοκινητήρες υπάρχει συνήθως πτερύγιο που ευθυγραμμίζει τον άξονα του δρομέα στον άνεμο, στους δε μεγάλους ανεμοκινητήρες εφαρμόζονται συστήματα αυτοματισμού της ρύθμισης της σωστής θέσης του δρομέα ως προς τον άνεμο μέσω σερβομηχανισμού. Τα κυριότερα μέρη που απαρτίζουν μία ανεμογεννήτρια οριζοντίου άξονα (HAWT) είναι:

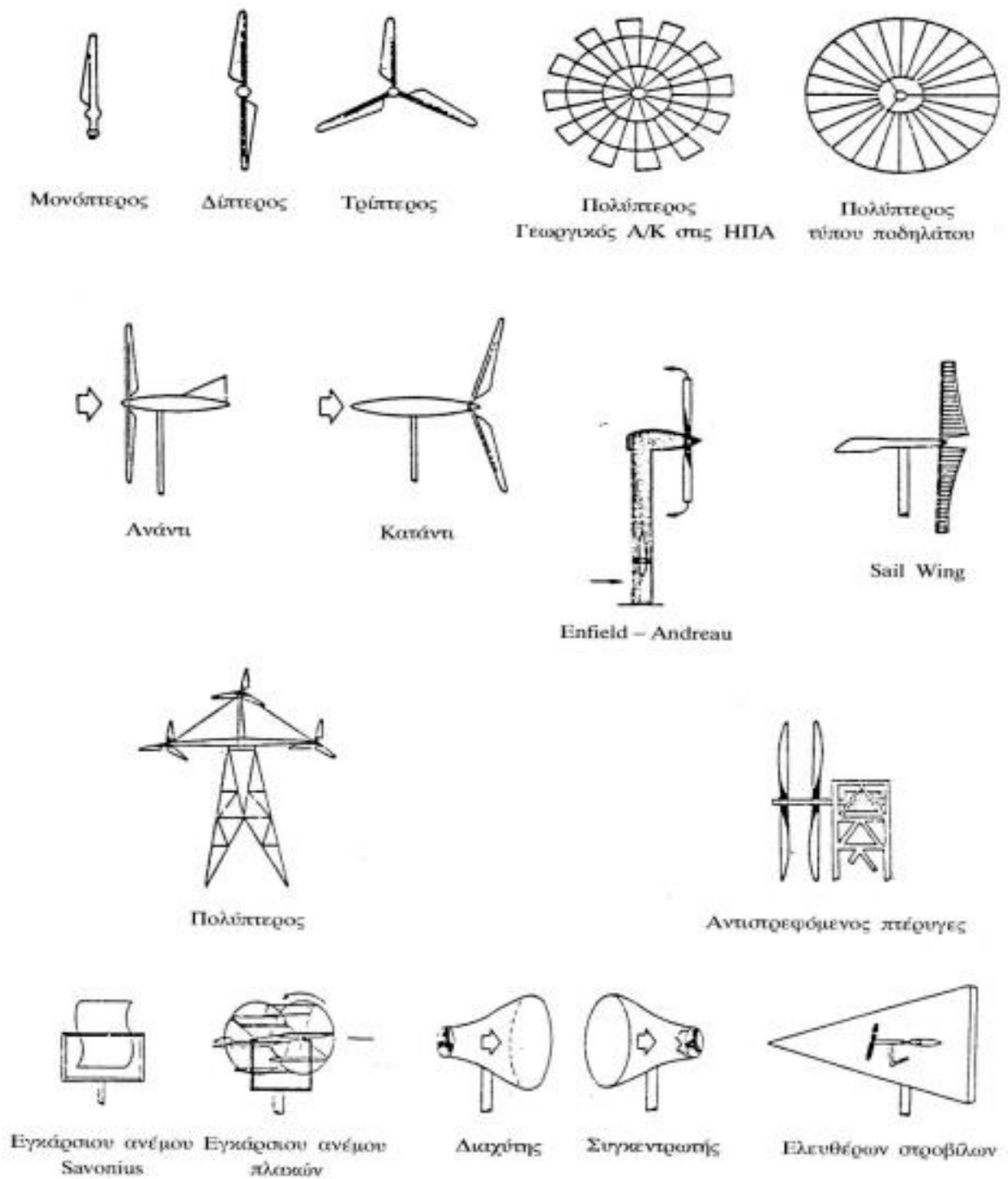
- Ο **πύργος (Tower)** που αποτελεί τη βάση στήριξης της ανεμογεννήτριας. Είναι κατασκευασμένος από χάλυβα και είναι συνήθως σωληνωτός ή τύπου δικτυώματος. Κριτήριο για την επιλογή του πύργου είναι εκτός από το κόστος του, η ευκολία μεταφοράς του στον τόπο της εγκατάστασης και η ευκολία ανέγερσης του
- Ο **ρότορας (Rotor)** που αποτελεί ουσιαστικά τη διάταξη που περιλαμβάνει το διακλαδωτή (Hub) και τα πτερύγια (Blades).
- Τη **πλήμνη (Hub)** που είναι μια μεταλλική κατασκευή πάνω στην οποία στερεώνονται τα πτερύγια. Στις ανεμογεννήτριες μεταβλητού βήματος, στο εσωτερικό του διακλαδωτή περιέχεται ειδική διάταξη για τη μεταβολή του βήματος των πτερυγίων.
- Τα **πτερύγια (Blades)** που είναι αεροδυναμικά σχεδιασμένα έτσι ώστε να εκμεταλλεύονται με βέλτιστο τρόπο την αιολική ενέργεια και μέσω της περιστροφικής τους κίνησης να τη μετατρέπουν σε κινητική. Στους μικρούς ανεμοκινητήρες τα πτερύγια είναι κατασκευασμένα από πολυουρεθάνη, υαλόνημα και ξύλο. Στους μεσαίου μεγέθους δρομείς χρησιμοποιούνται υαλονήματα με εναλλαγή της κατεύθυνσης των υαλονημάτων σε πολλαπλές στρώσεις ενώ στους μεγάλου μεγέθους ανεμοκινητήρες χρησιμοποιείται η τεχνολογία των ελίκων των αεροσκαφών (ανθρακόνημα με μηχανές αυτόματου τυλίγματος – Hamilton Standard). Κάθε πτερύγιο έχει ένα σύστημα προστασίας από κεραυνούς που αποτελείται από υποδοχείς κεραυνού στην άκρη του και έναν χάλκινο συρμάτινο αγωγό στο εσωτερικό του πτερυγίου.

- **Ο θάλαμος (Nacelle):** Βρίσκεται στη κορυφή του πύργου, κατασκευάζεται συνήθως από fiber glass και περιέχει τα παρακάτω βασικά τεχνικά μέρη της ανεμογεννήτριας:
- Τη **πλάκα εδράσεως (machine foundation):** Το μπροστινό μέρος της πλάκας εδράσεως της ατράκτου είναι κατασκευασμένο από χυτό χάλυβα και είναι η βάση για το σύστημα κίνησης το οποίο μεταδίδει τις δυνάμεις και τη ροπή από το δρομέα στον πύργο μέσω του συστήματος περιστροφής.
- Το **κιβώτιο ταχυτήτων (gearbox):** Είναι ο κύριος μηχανισμός που μεταδίδει τη ροπή από τον ρότορα στη γεννήτρια και είναι βιδωμένο στην πλάκα εδράσεως. Αυξάνει την ταχύτητα περιστροφής σε 1000-1800 rpm.
- Τον **άξονα χαμηλής ταχύτητας (Low-speed shaft)** που κινείται από το ρότορα με περίπου 30-60 rpm και στη συνέχεια μεταφέρει την ισχύ μέσω του κιβωτίου ταχυτήτων στον άξονα υψηλής ταχύτητας.
- Το **σύστημα προσανεμισμού (yaw system):** Το σύστημα προσανεμισμού επιτρέπει να περιστρέφει ο θάλαμος με τον ρότορα προς την κατεύθυνση που φυσάει ο άνεμος και είναι ένα απλό σύστημα με ρουλεμάν με ενσωματωμένη τριβή. Το σύστημα μεταδίδει τις δυνάμεις από τον θάλαμο και ρότορα στον πύργο.
- Τον **ελεγκτή (controller)** ο οποίος εκκινεί τη μηχανή για ταχύτητες ανέμου μεγαλύτερες από 4 m/s και την κλείνει όταν φτάνουν τα 25 m/s περίπου
- Το **σύστημα πέδησης (brake system):** Η ανεμογεννήτρια φρενάρει με πλήρη μεταβολή του βήματος των πτερυγίων. Ο ρόλος του συστήματος πέδησης είναι να σταματά την περιστροφή του άξονα σε περίπτωση υπερφόρτωσης ισχύος ή βλάβης του συστήματος. Επιπλέον μέσω υδραυλικού συστήματος ασκείται πίεση σε ένα δισκόφρενο που βρίσκεται στον κύριο άξονα υψηλής ταχύτητας. Αυτό το δισκόφρενο χρησιμεύει και ως φρένο στάθμευσης.
- Τη **γεννήτρια (generator):** Η γεννήτρια παραγωγής ηλεκτρικού ρεύματος είναι ασύγχρονη τετραπολική και υδρόψυκτη. Η μεταβλητή ταχύτητα επιτρέπει την μεταβολή της ταχύτητας του ρότορα εντός μιας ευρείας περιοχής ταχυτήτων. Αυτό, μειώνει τις διακυμάνσεις στην παροχή ρεύματος στο σύστημα ηλεκτρικού δικτύου καθώς και ελαχιστοποιεί τα φορτία σε ζωτικά μέρη του στροβίλου. Επιπλέον, το σύστημα μεταβλητής ταχύτητας βελτιστοποιεί την παραγωγή ενέργειας, ειδικότερα σε χαμηλές ταχύτητες ανέμου. Η έξοδος της γεννήτριας πρέπει να είναι συμβατή με το δίκτυο διανομής ηλεκτρικής ενέργειας.

- Το σύστημα ψύξης και κλιματισμού (cooling system): Εάν η θερμοκρασία του αέρα στο εσωτερικό της ατράκτου υπερβεί ένα ορισμένο επίπεδο, βαλβίδες τύπου θυρίδας θα ανοίγουν προς τα έξω. Ένας ανεμιστήρας φέρνει εξωτερικό αέρα για την ψύξη του εσωτερικού της ατράκτου.



Σχήμα 1.17. Τεχνικά μέρη μιας ανεμογεννήτριας οριζοντίου άξονα



Σχήμα 1.18. Τύποι ανεμοκινητήρων οριζόντιου άξονα

Πύργος της ανεμογεννήτριας (Tower)

Ο πύργος είναι ένα σημαντικό δομικό στοιχείο της ανεμογεννήτριας γιατί εκτός του ότι μεταφέρει φορτία από την κορυφή στα θεμέλια και χρησιμοποιείται για τη στήριξη του ρότορα, το ύψος του καθορίζει και το μέγεθος των πτερυγίων και άρα και την παραγόμενη ενέργεια. Επομένως όσο ψηλότερος είναι ο πύργος, τόσο μεγαλύτερο είναι και το ποσό της παραγόμενης ενέργειας (π.χ. με αύξηση κατά 20 μέτρα από το ένα πρότυπο ύψος 80 μέτρων μπορεί να επιτευχθεί μία επιπλέον παραγωγή ενέργειας γύρω στο 5%). Αυτό συμβαίνει διότι ψηλότερα υπάρχει μεγαλύτερο διαθέσιμο αιολικό δυναμικό και μειώνεται η επιρροή των εμποδίων του εδάφους, όπως δέντρων και κτηρίων, που δημιουργούν αναταράξεις στην ροή του ανέμου.

Ο πύργος γενικά πρέπει να εξασφαλίζει έναν ασφαλή, αποτελεσματικό και οικονομικό σχεδιασμό. Το κόστος του μπορεί να ανέλθει έως και το 20% του σύνολο της κατασκευής. Για λόγους συντήρησης των στοιχείων του θα πρέπει επίσης να παρέχει εύκολη πρόσβαση σ' αυτόν. Με την αύξηση του ύψους του πύργου (άνω των 100 m) εμφανίζονται προβλήματα όπως η μεταφορά των μελών, η ανέγερση του πύργου, το μεγάλο βάρος καθώς και οι μεγάλες διαστάσεις στη βάση που πρέπει να επιλυθούν. Η διαδικασία στησίματος της μηχανής σε μεγάλες μονάδες εξαρτάται και από την ευκολία οδικής πρόσβασης στη θέση, την ύπαρξη ικανοποιητικού ανυψωτικού μέσου τόσο σε ανυψωτική ικανότητα όσο και σε ύψος ανύψωσης.

Τα υλικά που χρησιμοποιούνται για την κατασκευή των πύργων ανεμογεννητριών είναι ο χάλυβας, το οπλισμένο και το προεντεταμένο σκυρόδεμα ή συνδυασμούς αυτών. Με βάση αυτά τα υλικά, υπάρχουν διάφοροι διαθέσιμοι τύποι που μπορούν να ικανοποιήσουν τις απαραίτητες τεχνικές προδιαγραφές. Συνθετικά υλικά κατασκευής όπως το fiber glass βρίσκονται ακόμα σε ερευνητικό επίπεδο.

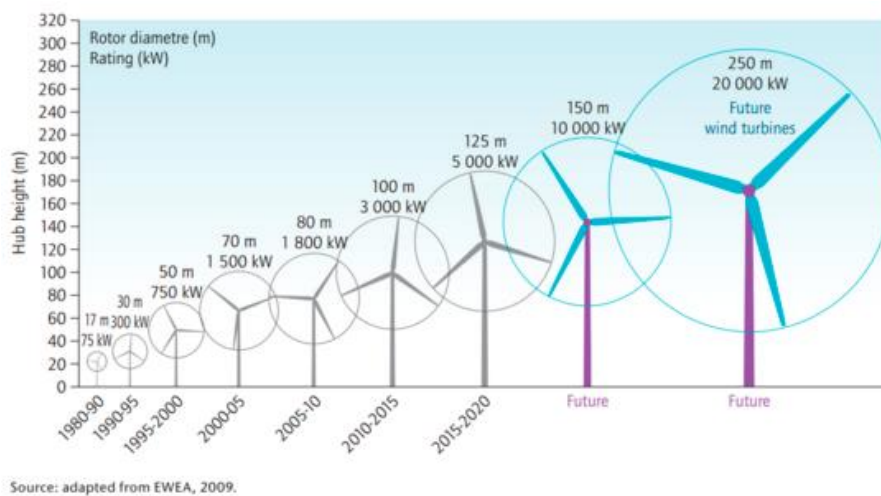
Τα τελευταία χρόνια παρατηρείται η τάση για κατασκευή ολοένα και ψηλότερων πύργων. Στο μέλλον αναμένονται ακόμα ψηλότεροι πύργοι, επίσης, γιατί στην ώριμη πλέον αγορά αιολικής ενέργειας οι πιο παραγωγικές θέσεις έχουν ήδη χρησιμοποιηθεί. Το μέσο ύψος των πύργων που έχουν εγκατασταθεί στην Ευρώπη σήμερα είναι περίπου 80 μέτρα.

Τέλος, κρίσιμη παράμετρος θεωρείται το αισθητικό αποτέλεσμα της κατασκευής του πύργου που επηρέασε και καθόρισε σε μεγάλο βαθμό την επιλογή της τυπολογίας των

πύργων ανεμογεννητριών, τουλάχιστον έως και σήμερα.

Όσον αφορά τον σχεδιασμό των πύργων στη σύγχρονη εποχή έχουν επικρατήσει τρεις τύποι διατάξεων, οι οποίοι έχουν κατασκευαστεί σε μεγάλη κλίμακα ανά το κόσμο: η μονοπολική διάταξη σωλήνα (tubular tower), ο δικτυωτός πύργος (lattice tower) με τριποδική διάταξη σωλήνων (three-legged tower) και η μονοπολική διάταξη με αντιστήριξη καλωδίων (guy-wired pole tower).

Συμπερασματικά, ο προσδιορισμός του τύπου και του ύψους του πύργου κρίνεται απαιτητικός και ιδιαίτερα καθοριστικός για την οικονομική, κατασκευαστική και λειτουργική απόδοση της ανεμογεννήτριας.



Σχήμα 1.19. Πορεία αύξησης του ύψους του πύργου και της διαμέτρου του ρότορα στις Α/Γ από τη δεκαετία του 80



Σχήμα 1.20. Τα τρία επικρατέστερα είδη πύργων Α/Γ

Κατηγορίες πύργων ανεμογεννητριών

Οι πύργοι ανεμογεννητριών χαρακτηρίζονται ως ευαίσθητες κατασκευές υπό τη δράση του ανέμου. Κατά κύριο λόγο, συναντώνται χαλύβδινοι σωληνωτοί ή δικτυωτού τύπου είτε σκυροδέματος. Από τους παραπάνω, περιγράφονται οι δύο περιπτώσεις των χαλύβδινων ως επικρατέστερες

Σωληνοειδής χαλύβδινος πύργος (tubular steel tower).

Σήμερα οι σωληνοειδής πύργοι αποτελούν τον επικρατέστερο τύπο, κυρίως λόγω της ταχείας συναρμολόγησης και ανέγερσης τους στο εργοτάξιο. Βασικά κατασκευάζονται από ελάσματα φύλλων χάλυβα κομμένα, επεξεργασμένα και συγκολλημένα μεταξύ τους. Μέχρι το ύψος των 20 m, μπορούν να κατασκευαστούν ως ένα ενιαίο τμήμα και να μεταφερθούν στο εργοτάξιο ενώ άλλοι με ύψος άνω των 100 m, αποτελούνται από πολλά επιμέρους τμήματα των 20-30 m, συνδεδεμένα μεταξύ του με διπλές φλαντζωτές συνδέσεις προεντεταμένων κοχλιών, προκειμένου να μπορούν να αποφευχθούν οι επί τόπου συγκολλήσεις. Κατασκευάζονται ακολουθώντας κωνικό σχήμα με τη διάμετρο της διατομής να είναι μεγαλύτερη πλησιέστερα στη βάση (κυμαίνεται από 4.5m) σε περίπου 2m στη κορυφή, προκειμένου να επιτευχθεί η επιθυμητή αντοχή καθώς και για να γίνει και εξοικονόμηση υλικού. Το στατικό σύστημα είναι απλό στη σύλληψη, καθώς η κατασκευή αντιμετωπίζεται ως πρόβολος. Το βασικό τους πλεονέκτημα αφορά την εξοικονόμηση υλικού καθώς αυτός ο τύπος απαιτεί μόλις τη μισή ποσότητα υλικού σε σχέση με έναν σωληνωτό πύργο ίδιας στιβαρότητας. Ταυτόχρονα, το βασικό τους μειονέκτημα έγκειται στο γεγονός ότι αισθητικά δεν αποτελούν ελκυστική λύση. Βέβαια υποστηρίζεται και η άποψη ότι το αισθητικό τους αποτέλεσμα είναι αρνητικό μόνο από κοντινή απόσταση ενώ από μακριά κρίνεται ότι εντάσσονται ομαλότερα στο γύρω περιβάλλον. Ταυτόχρονα δεν προκαλούν αισθητική όχληση λόγω αντανάκλασεων όπως οι σωληνωτοί πύργοι. Οι νέοι πύργοι που υπερβαίνουν τα 100 μέτρα έχουν διάμετρο βάσης πάνω από 5 μέτρα το οποίο είναι πρόβλημα σε πολλές χώρες, καθώς το μέγιστο μεταφερόμενο οδικώς μέγεθος είναι μικρότερο από 4,9 μέτρα. Οι μεγαλύτερες Α/Γ στο κόσμο κατασκευάστηκαν από την Δανεική εταιρία Dong Energy και βρίσκονται κοντά στην ακτή του Liverpool με ύψος 195 m και ισχύ 8 MW η κάθε μια σε ένα

σύνολο 32 (Σχήμα 1.)



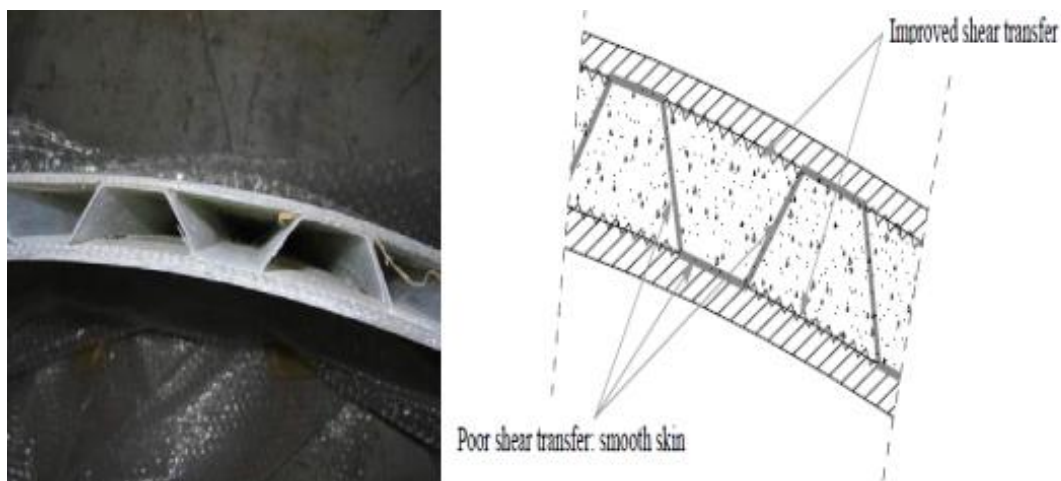
Σχήμα 1.21. Η μεγαλύτερη Α/Γ στον κόσμο με σωληνωτό πύργο κοντά στις ακτές του Liverpool στην Μεγάλη Βρετανία. (Burbo Bank extension windfarm in Merseyside)

Υβριδικός πύργος σκυροδέματος/χάλυβα (concrete/steel hybrid tower).

Η ιδέα πίσω από τη δημιουργία ενός υβριδικού πύργου σκυροδέματος και χάλυβα είναι να γίνει χρήση σκυροδέματος για το ευρύ κάτω μέρος και χρήση χάλυβα για το άνω μέρος του πύργου. Στο άνω μέρος σχεδιάζεται ένας συμβατικός χαλύβδινος σωληνοειδής πύργος χωρίς να υπάρχει κίνδυνος σύγκρουσης με τους περιορισμούς μεταφοράς, καθώς θα έχει μικρότερο ύψος και συμπερασματικά μικρότερη διάμετρο βάσης. Επίσης καθιστά ευκολότερο το σχεδιασμό του μέρους του σκυροδέματος ώστε να πάρει τις ιδιοσυχνότητες που θέλουμε. Βέβαια υπάρχει ένα επιπλέον κόστος που πρέπει να συμπεριληφθεί που αφορά την σύνδεση του μέρους του σκυροδέματος με το μέρος του χαλύβδινου πύργου. Μία άλλη εφαρμογή των υβριδικών πύργων σκυροδέματος/χάλυβα είναι σε περιοχές όπου το κόστος του χάλυβα είναι πολύ ψηλό (όπως για παράδειγμα στην Βραζιλία). Επιπλέον η ευκολότερη μεταφορά λόγω μικρότερων διαστάσεων των τμημάτων του πύργου μειώνει το συνολικό κόστος. Από την άλλη μεριά το μεγαλύτερο πρόβλημα είναι το αυξημένο βάρος. Υβριδικοί πύργοι χρησιμοποιούνται ευρέως από την πολυεθνική εταιρία κατασκευής ανεμογεννητριών Enercon.

Σύνθετος πυλώνας από ινοπλισμένα πολυμερή υλικά.

Νέες κατασκευαστικές λύσεις προσφέρει και η τεχνολογία των σύνθετων ινοπλισμένων πολυμερών υλικών καθώς παρατείνει την διάρκεια ζωής, μειώνει το κόστος συντήρησης της ανεμογεννήτριας, αυξάνει τα δυναμικά χαρακτηριστικά απόσβεσης και αυξάνει την αντοχή σε κόπωση. Επίσης προσφέρει μια εναλλακτική λύση για περιπτώσεις όπου η συνθήκες μεταφοράς και εγκατάστασης είναι δύσκολες. Σημαντική επίσης είναι η συμπεριφορά του υλικού αυτού σε υγρές συνθήκες, όπως στην περίπτωση θαλάσσιων ανεμογεννητριών



Σχήμα 1.22. Σχεδιασμός κυλινδρικού κελύφους με υβριδική – σύμμικτη διατομή ινοπλισμένων πολυμερών (fiber-glass) – σκυροδέματος

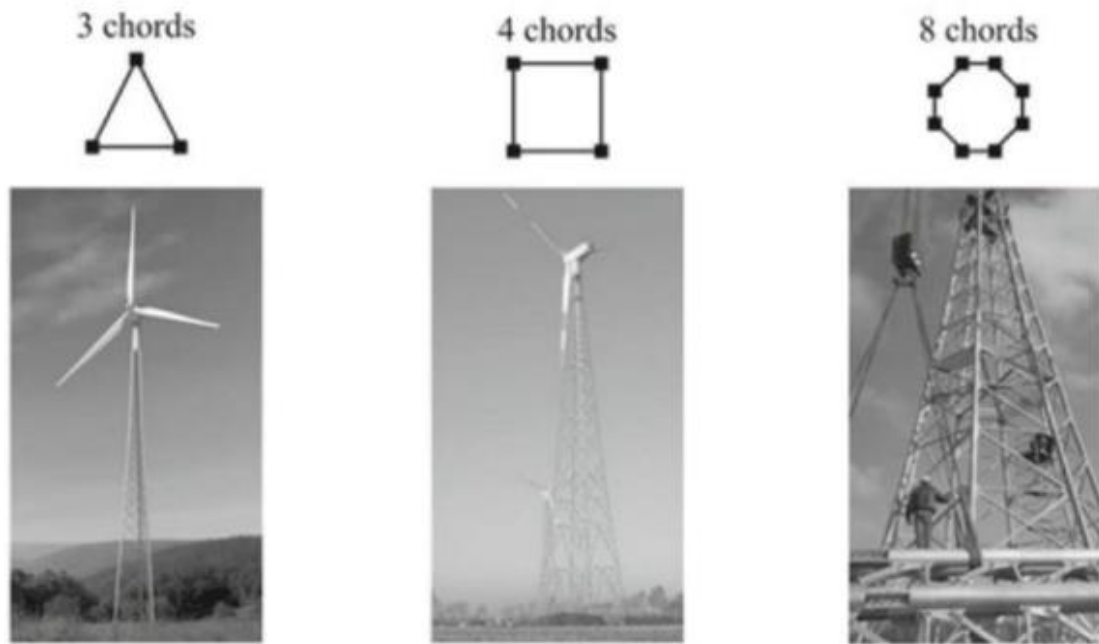
Δικτυωτός πύργος (Lattice Tower)

Κατά τις αρχές της εμπορικής χρήσης της αιολικής ενέργειας, αυτός ο τύπος πύργου αποτέλεσε την επικρατέστερη επιλογή. Χρησιμοποιήθηκαν πολύ κατά το παρελθόν, όπου οι ανεμογεννήτριες ήταν πιο μικρές σε μέγεθος (λιγότερο από ένα MW). Στην πορεία των χρόνων όμως επήλθε η αντικατάστασή τους σχεδόν πλήρως από τους σωληνωτούς πύργους. Πρόσφατα το ενδιαφέρον στην κατεύθυνση των δικτυωτών πύργων έχει επανέλθει στο προσκήνιο λόγω της ευκολίας που προσφέρουν σε περιπτώσεις που το επιθυμητό ύψος ξεπερνά τα 100 m.

Το μεγαλύτερο τους μειονέκτημα είναι η οπτική και αισθητική εικόνα που δημιουργούν στο φυσικό τοπίο που εγκαθίστανται. Βέβαια υποστηρίζεται και η άποψη ότι το αισθητικό τους αποτέλεσμα είναι αρνητικό μόνο από κοντινή απόσταση ενώ από μακριά κρίνεται ότι εντάσσονται ομαλότερα στο γύρω περιβάλλον. Ταυτόχρονα δεν προκαλούν αισθητική όχληση λόγω αντανάκλασεων όπως οι σωληνωτοί πύργοι Έχουν επίσης μεγαλύτερο κατασκευαστικό κόστος και κόστος συντήρησης.

Είναι όμως ελαφρύτερες από οποιαδήποτε άλλη διάταξη σχεδιασμού, χρησιμοποιώντας 50% λιγότερο υλικό σε σχέση με τις αντίστοιχες με σωληνοειδή πύργο της ίδιας στιβαρότητας. Δημιουργούν λόγω σχεδιασμού λιγότερες αναταράξεις αέρα καθώς είναι διαπερατές. Η φυσική σημασία αυτής της εξοικονόμησης υλικού οφείλεται στα μεγάλα πλάτη των κάτω τμημάτων. Η ανάγκη του υλικού να παραλάβει ένταση είναι αντιστρόφως ανάλογη προς το πλάτος. Με μια σωληνοειδή λεπτότοιχη κατασκευή υπάρχει κίνδυνος λυγισμού κάτι το οποίο περιορίζει τη μέγιστη διάμετρο. Ένα σχέδιο δικτυώματος δεν λυγίζει σαν ένα κέλυφος. Ο κίνδυνος λυγισμού των επιμέρους μελών περιορίζεται με εισαγωγή πολλών δοκών που δίνουν στο δικτυωτό πύργο την χαρακτηριστική εμφάνιση του. Οι δικτυωτοί πύργοι μπορούν να κατασκευαστούν με διαφορετικό αριθμό ποδών. Συγκεκριμένα, υπάρχει η δυνατότητα κατασκευής τους με τρεις, τέσσερις και οκτώ πόδες με συχνότερη επιλογή αυτή των τεσσάρων ποδών. Έτσι τα φορτία μεταφέρονται από την κορυφή της ανεμογεννήτριας μέχρι τη θεμελίωση και τελικά στο έδαφος.

Ο ψηλότερος δικτυωτός πύργος βρίσκεται στο Laasow (Brandenburg) στη Γερμανία, με την νασέλα της ανεμογεννήτριας να βρίσκεται σε ύψος 160 μέτρων. Ο συγκεκριμένος πύργος έχει πλάτος στην κορυφή του ίσο με 2,9 μέτρα, ενώ στην βάση του, το πλάτος του φτάνει τα 29 μέτρα. Η διάμετρος του ρότορα είναι 90 m με ισχύ 2.5 MW. Ο ψηλότερος δικτυωτός πύργος βρίσκεται στο Laasow (Brandenburg) στη Γερμανία, με την νασέλα της ανεμογεννήτριας να βρίσκεται σε ύψος 160 μέτρων. Ο συγκεκριμένος πύργος έχει πλάτος στην κορυφή του ίσο με 2,9 μέτρα, ενώ στην βάση του, το πλάτος του φτάνει τα 29 μέτρα. Η διάμετρος του ρότορα είναι 90 m με ισχύ 2.5 MW.



Σχήμα 1.23. Πλήθος ποδών δικτυωτών πύργων



Σχήμα 1.24. ο μεγαλύτερος δικτυωτός πύργος βρίσκεται στο Laasow (Brandenburg) στη Γερμανία με ύψος 160 m.

Απαιτήσεις σχεδιασμού πυλώνα ανεμογεννήτριας

Προκειμένου να εξασφαλιστεί ένας ικανοποιητικός σχεδιασμός του πυλώνα της ανεμογεννήτριας πρέπει να ικανοποιηθούν ορισμένες απαιτήσεις ως προς τις διαστάσεις, την πλαστική ροπή αντοχής, τον λυγισμό, της μετατόπισης της κορυφής του πυλώνα, τις φλάντζες συνδέσεως και των συγκολλήσεων. Επίσης όρια στην θεμελιώδη ιδιοσυχνότητα του πύργου πρέπει να εξεταστούν.

1. **Διαστάσεις:** Εξαιτίας περιορισμών στην μεταφορά, η εξωτερική διάμετρος του πύργου δεν μπορεί να ξεπεράσει τα 4,5 m. Επιπρόσθετα λόγω περιορισμών του χάλυβα που μπορεί να επεξεργαστεί σε κύλινδρο χρησιμοποιώντας βασικό εξοπλισμό το πάχος δεν μπορεί να ξεπεράσει τα 40 mm.

2. **Πλαστική οριακή ανάλυση:** Στην οριακή κατάσταση αστοχίας ελέγχονται οι μέγιστες ισοδύναμες τάσεις κατά von Mises σε κάθε σημείο του πυλώνα και συγκρίνονται με το όριο θραύσης του χάλυβα.

3. **Λυγισμός:** Ελέγχονται οι επιτρεπόμενες τάσεις στην οριακή κατάσταση λυγισμού σε κάθε σημείου του πυλώνα.

4. **Μετατόπιση της κορυφής του πυλώνα:** Η μέγιστη επιτρεπόμενη μετατόπιση της κορυφής του πυλώνα τίθεται 1% του ύψους του πυλώνα έτσι ώστε να αποφευχθεί υπερβολική κίνηση που θα παρεμπόδιζε την ομαλή λειτουργία της ανεμογεννήτριας.

5. **Θεμελιώδης ιδιοσυχνότητα:** Για να αποφύγουμε το φαινόμενο του συντονισμού, η θεμελιώδης ιδιοσυχνότητα της κατασκευής πρέπει να είναι σε ασφαλή απόσταση από τις συχνότητες διέγερσης στην φάση λειτουργίας του ρότορα της ανεμογεννήτριας. Συνηθισμένες τιμές για συχνότητες λειτουργίας είναι ανάμεσα σε 0,23 με 0,52 Hz για μικρές ανεμογεννήτριες, και 0,10 με 0,30 για μεγάλες ανεμογεννήτριες. Η θεμελιώδης ιδιοσυχνότητα του πύργου θα πρέπει να παραμείνει πάνω από την μεγαλύτερη συχνότητα λειτουργίας μιας συγκεκριμένης ανεμογεννήτριας πολλαπλασιασμένη με έναν συντελεστή, τυπικά ανάμεσα σε 1,1 και 2, έτσι ώστε να αποφευχθεί ο συντονισμός σε οποιοδήποτε σημείο κατά το διάστημα λειτουργίας.

6. **Κυκλικές Φλάντζες Συνδέσεως:** Η σύνδεση των τμημάτων του πυλώνα γίνεται μέσω κυκλικών φλάντζών μαζί με υψηλά προεντεταμένους κοχλίες (HV) και ελέγχεται η σύνδεση

αυτή στην οριακή κατάσταση αστοχίας. Επίσης οι προεντεταμένοι κοχλίες οφείλουν να ελεγχθούν και σε κόπωση.

7. Συγκολλήσεις: Ελέγχονται οι επιτρεπόμενες τάσεις στην οριακή κατάσταση αστοχίας σε κάθε σημείο κατά ύψος του πυλώνα όπου υπάρχει σύνδεση με συγκόλληση.

Φορτία Πύργου

Τα θεωρούμενα φορτία του πύργου αποτελούνται από το βάρος της νασέλας, του ρότορα, το ίδιο βάρος του πύργου και τα φορτία από τον άνεμο. Το ίδιο βάρος του ρότορα και της νασέλας εφαρμόζονται στην κορυφή του πύργου και δίνονται από την κατασκευάστρια εταιρία της ανεμογεννήτριας. Τα φορτία από τον άνεμος και το ίδιο βάρος του πύργου υπολογίζονται χρησιμοποιώντας κατάλληλους τύπους.

Το ίδιο βάρος της κατασκευής: Υπολογίζεται από κατάλληλο λογισμικό με χρήση πεπερασμένων στοιχείων, ως συνάρτηση της γεωμετρίας και της μονάδας μάζας του χάλυβα. Η συνεισφορά του βοηθητικού εξοπλισμού στο συνολικό βάρος του πύργου μπορεί να αμεληθεί.

Άνεμος: Η φόρτιση του ανέμου στον πύργο, υπολογίζετε για συγκεκριμένα δυναμικά χαρακτηριστικά και γεωμετρία της κατασκευής, επιβάλλεται κατά ύψος και περιφερειακά σε όλες τις επιφάνειες του αφού ολοκληρωθεί η σχετική εξίσωση κατανομής σύμφωνα με την ακριβή μέθοδο του Ευρωκώδικα [1-1-4]. Εναλλακτικά μπορεί να χρησιμοποιηθεί η απλοποιητική εξίσωση κατανομής που προτείνεται από τους Βανιτοπουλος et al. ως εξής: Η κατανομή της ανεμοπίεσης καθ' ύψος [z] του πύργου δίνετε ως συνάρτηση της διαμέτρου [D], από τις εξισώσεις (z, D σε [m], Fw η δύναμη από τον άνεμο σε [kN/m]):

Για $z \leq 2,00\text{m}$: $Fw = 0,51 * D$ ενώ για

Για $z > 2,00\text{m}$: $Fw = 0,013 * \ln(20z) * [(\ln(20z) + 7)] * D$

Όπου η εξωτερική διάμετρος D μεταβάλλεται καθ' ύψος σύμφωνα με τη σχέση:

$$D = -0,01775 * z + 4,30266$$

Η συνάρτηση μεταβολής είναι διαφορετική για κάθε συνολικό ύψος (H) του εκάστοτε πύργου ανεμογεννήτριας που εξετάζεται.

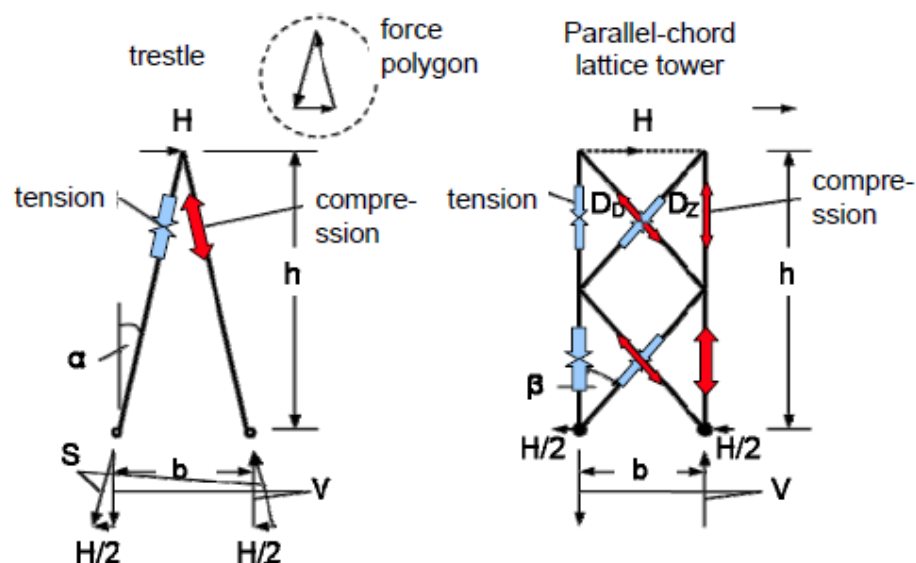
Στους δικτυωτούς πύργους, μπορούν να χρησιμοποιηθούν είτε στηρίξεις μορφής K είτε να διαμορφωθεί πύργος με κατακόρυφους πόδες.

Στην περίπτωση των στηρίξεων μορφής K, η παραλαβή της φόρτισης καθορίζεται από

την κλίση των πόδων. Τα διαγώνια στοιχεία στήριξης δεν είναι απαραίτητα σε αυτή την περίπτωση. Οι εσωτερικές δυνάμεις των χαλύβδινων στοιχείων βρίσκονται σε ισορροπία, οι τιμές των δυνάμεων θλίψης και εφελκυσμού είναι ίσες μεταξύ τους και τα κατακόρυφα φορτία οδηγούν σε ανατροπή της βάσης.

Από την άλλη, σε πύργους παράλληλων πόδων, τα οριζόντια φορτία παραλαμβάνονται ως ζεύγη θλίψης και εφελκυσμού στους πόδες και στα διαγώνια στοιχεία από την κορυφή έως τη θεμελίωση της κατασκευής.

Και για τα δύο συστήματα είναι σημαντική η διάκριση μεταξύ των στοιχείων που υποβάλλονται σε θλίψη και αυτών σε εφελκυσμό. Για τα μέλη υπό εφελκυσμό η αντοχή των κοχλιών στους κόμβους είναι κρίσιμη ενώ για τα στοιχεία υπό θλίψη καθοριστικός θεωρείται ο στρεπτοκαμπτικός λυγισμός. Σε κάθε περίπτωση κρίνονται σημαντικά τα φορτία κόπωσης. Η επιλογή των διατομών της κατασκευής γίνεται με βάση τις αυξανόμενες καμπτικές ροπές από την κορυφή προς τη θεμελίωση. Συνήθως, χρησιμοποιούνται διατομές μονών ή διπλών γωνιακών με ίσα ή άνισα πέλματα. Επίσης, η υπό κλίση ροή και οι ριπές του ανέμου στα πτερύγια, οδηγούν στην ανάπτυξη στρεπτικών ροπών καθ' ύψος του πύργου. Πρέπει να αναφερθεί ότι οι δικτυωτοί πύργοι διαθέτουν μικρότερη στρεπτική δυσκαμψία και ικανότητα παραλαβής φορτίων σε σχέση με τους σωληνωτούς. Συνολικά, στους δικτυωτούς πύργους, οι βασικές παράμετροι που εξετάζονται είναι η ανάλυση τάσεων, οι έλεγχοι λυγισμού των μελών, οι έλεγχοι των συνδέσεων και τελικώς να εξετασθεί η παραμέτρο της κόπωσης.



Σχήμα 1.25. Παραλαβή φόρτισης σε δικτυωτούς πύργους.

Δικτυωτή Α/Γ της General Electric

Η General Electric (GE) δραστηριοποιείται στον τομέα των ανανεώσιμων πηγών ενέργειας όπως η αιολική ενέργεια, η υδροηλεκτρική ενέργεια αλλά και η ηλιακή. Ειδικότερα στον τομέα της αιολικής ενέργειας προσφέρει λύσεις τόσο σε επίπεδο ηπειρωτικών περιοχών όσο και για ανεμογεννήτριες στη θάλασσα.

Τα τελευταία χρόνια παρουσιάστηκε από την εταιρεία ο χωρικός δικτυωτός πύργος (Space Frame Tower). Ο συγκεκριμένος πύργος διαμέτρου βάσης 10m, προσφέρει τη δυνατότητα τοποθέτησης του ρότορα σε ύψος 139m ενώ ταυτόχρονα παρουσιάζει την ικανότητα εύκολης μεταφοράς των χαλύβδινων μελών και συναρμολόγησης προς διαμόρφωση της κατασκευής σε οποιαδήποτε περιοχή ακόμα και δυσπρόσιτη. Η παραγωγή του ξεκίνησε περί το 2015.

Για την αντιμετώπιση του φαινομενικά ισχυρότερου μειονεκτήματος των δικτυωτών πύργων που δεν είναι άλλο από το αισθητικό αποτέλεσμα της κατασκευής, έχουν ληφθεί ειδικά και καινοτόμα μέτρα. Συγκεκριμένα, χρησιμοποιείται μια πολυεστερικού υφάσματος επένδυση που προσφέρει αισθητικό αποτέλεσμα όμοιο με αυτό του συμπαγούς πύργου ενώ ταυτόχρονα το στατικό σύστημα αξιοποιεί τα πλεονεκτήματά του δικτυώματος χωρίς καμία διαφοροποίηση. Ακόμα, η επένδυση παρέχει προστασία για το προσωπικό της συντήρησης του πύργου καθώς και των διαφόρων εγκαταστάσεων όπως το τμήμα ελέγχου, από τα καιρικά φαινόμενα. Ταυτόχρονα με τα παραπάνω, η επένδυση επιτρέπει στο φως να εισέρχεται στο εσωτερικό του χώρου, διευκολύνοντας έτσι την παρουσία του ειδικού προσωπικού συντήρησης του πύργου .

Σύμφωνα με τα δεδομένα της GE, ο πύργος αυτός μπορεί να συναρμολογηθεί και να ανεγερθεί σε χρονικό διάστημα αντίστοιχο της κατασκευής ενός σωληνωτού πύργου ανεμογεννήτριας αντίστοιχου ύψους. Έτσι τα πλεονεκτήματα του δικτυωτού πύργου όπως έχουν αναφερθεί και προηγουμένως, αξιοποιούνται πλήρως ενώ παράλληλα εξαλείφεται το μειονέκτημα του μεγάλου χρόνου συναρμολόγησης. Συγκεκριμένα, όσον αφορά την ποσότητα του χρησιμοποιούμενου υλικού εκτιμάται ότι ο δικτυωτός πύργος θα απαιτεί περίπου 20-30% λιγότερο χάλυβα σε σχέση με έναν σωληνωτό πύργο ύψους 100m.



Σχήμα 1.26. Δικτυωτός πύργος της General Electric



Σχήμα 1.27. Εσωτερικό δικτυωτού πύργου General Electric (Είναι ορατή και η επένδυση των όψεων)



Σχήμα 1.28. Κέντρο ελέγχου στη βάση της Α/Γ

Το μέλλον στο σχεδιασμό των ανεμογεννητριών (Χωρίς Πτερύγια)

Μια ισπανική εταιρεία που ονομάζεται Vortex Bladeless προτείνει ένα ριζοσπαστικό νέο τρόπο για να εκμεταλλεύεται την αιολική ενέργεια που θα ενισχύσει και πάλι αυτό που βλέπετε έξω από το παράθυρο ενός αυτοκινήτου σε κίνηση. Η ιδέα είναι τους είναι η ανεμογεννήτρια Vortex, μια ανεμογεννήτρια χωρίς πτερύγια που μοιάζει με ένα γιγαντιαίο έλασμα που σέρνεται στον ουρανό. Η Α/Γ Vortex έχει τους ίδιους στόχους με τις συμβατικές ανεμογεννήτριες δηλαδή να μετατρέψει την κινητική ενέργεια του αέρα σε ηλεκτρική με εντελώς όμως διαφορετικό τρόπο. Αντί της κυκλικής κίνησης μιας έλικας, η Α/Γ Vortex εκμεταλλεύεται αυτό που είναι γνωστό ως στροβιλισμός, ένα αεροδυναμικό αποτέλεσμα που παράγει ένα μοτίβο περιδίνησης. Ο στροβιλισμός έχει από καιρό θεωρηθεί εχθρός των μηχανικών, οι οποίοι προσπαθούν κατά το σχεδιασμό των κατασκευών να τον αποφύγουν αφού μπορεί να προκαλέσει μέχρι και την καταρρευση τους, όπως η γέφυρα Tacoma Narrows Bridge εξαιτίας των ταλαντώσεων που προκαλούν.

Όπου οι άλλοι σχεδιαστές και μηχανικοί βλέπουν τον κίνδυνο, οι ιδρυτές της Vortex Bladeless - David Suriol, David Yáñez και Raul Martín - βλέπουν την ευκαιρία. Έτσι αναρωτήθηκαν: "γιατί δεν προσπαθούμε να χρησιμοποιήσουμε αυτήν την ενέργεια, αντί να την αποφύγουμε", Η ερευνητική ομάδα ξεκίνησε το 2010 με σκοπό να βρεθεί ένας τρόπος να μετατραπεί αυτή η ενέργεια των ταλαντώσεων σε χρήσιμη και παραγωγική ενέργεια. Το σχήμα της Α/Γ Vortex αναπτύχθηκε υπολογιστικά για να εξασφαλίσει ότι οι δίνες (vortices) που δημιουργούνται από το άνεμο να 'χτυπούν' ταυτόχρονα όλο το μήκος του ιστού της Α/Γ. Στο σημερινό πρωτότυπο αυτής της Α/Γ, ο επιμήκης κώνος είναι κατασκευασμένος από ένα σύνθετο υλικό από ίνες και ανθρακόνημα, το οποίο επιτρέπει στην κατασκευή να δονείται όσο το δυνατόν περισσότερο (μια αύξηση της μάζας μειώνει τη φυσική συχνότητα). Στη βάση του κώνου υπάρχουν δύο μαγνητικοί δακτύλιοι που απωθούνται μεταξύ τους και οι οποίοι λειτουργούν ως ένα είδος μη ηλεκτρικού κινητήρα. Όταν ο κώνος ταλαντώνεται προς μια κατεύθυνση, οι απωθητικοί μαγνήτες το τραβούν προς την άλλη κατεύθυνση, σαν μια ελαφριά ώθηση για να αυξηθεί η κίνηση του ιστού ανεξάρτητα από την ταχύτητα του ανέμου. Αυτή η κινητική ενέργεια μετατρέπεται τότε σε ηλεκτρική μέσω ενός μετατροπέα που πολλαπλασιάζει τη συχνότητα της ταλάντωσης του ιστού για να βελτιώνει το βαθμό απόδοσης με την οποία το σύστημα συλλέγει την ενέργεια. Οι κατασκευαστές της Vortex Α/Γ

υποστηρίζουν το γεγονός ότι επειδή δεν περιλαμβάνονται γρανάζια, μπουλόνια ή μηχανικά κινούμενα μέρη, καθιστούν την κατασκευή της αλλά και την συντήρηση της φθηνότερη.

Οι ιδρυτές ισχυρίζονται ότι η μια Vortex Mini A/Γ η οποία έχει ύψος περίπου 41 πόδια, μπορεί να συλλάβει μέχρι και το 40% της ισχύος του ανέμου σε ιδανικές συνθήκες (δηλαδή όταν ο άνεμος φυσάει γύρω στα 26 μίλια ανά ώρα). Με βάση τις δοκιμές που έγιναν αποδείχθηκε ότι η Vortex Mini A/Γ συνέλαβε 30% λιγότερο της ισχύος του ανέμου έναντι των συμβατικών ανεμογεννητριών, αλλά αυτό το έλλειμμα αντισταθμίζεται από το γεγονός ότι μπορούν να τοποθετηθούν διπλάσιες σε αριθμό A/Γ τύπου Vortex στον ίδιο χώρο σε σχέση με τις τυπικές A/Γ με πτερύγια. Η εταιρία υποστηρίζει ότι υπάρχουν κάποια σαφή πλεονεκτήματα για το μοντέλο τους. Μερικά από αυτά είναι ότι είναι λιγότερο δαπανηρή η κατασκευή τους, εντελώς σιωπηλή και ασφαλέστερη για πτηνά, καθώς δεν έχουν πτερύγια. Επίσης αναφέρουν ότι το κόστος τους θα είναι περίπου 51% μειωμένο έναντι των παραδοσιακών A/Γ των οποίων το σημαντικότερο κόστος προέρχεται από το κόστος των πτερυγίων και το σύστημα στήριξης. Τέλος έχουν καλή φυσική εικόνα αφού μοιάζουν με σπαράγγια!



Σχήμα 1.29. A/Γ τύπου Vortex

Γλώσσα Προγραμματισμού VISUAL BASIC for APPLICATIONS

Η Visual Basic for Application (VBA) είναι μια εξειδικευμένη μορφή της Microsoft Visual Basic®. Δημιουργήθηκε με βάση την γνωστή Basic και χρησιμοποιεί πληθώρα ίδιων ή παρόμοιων χαρακτηριστικών και παρόμοιο σχεδιαστικό περιβάλλον. Σκοπός της δεν είναι η αντικατάσταση της Ms VB αλλά η δυνατότητα χρήσης της με σκοπό τον προγραμματισμό των λειτουργιών ενός λογισμικού ξενιστή. Η VBA δεν υπάρχει ως ξεχωριστό λογισμικό. Είναι όμως ενσωματωμένη σε άλλα προγράμματα όπως MS Office, StarOffice, AutoCad, MicroStation κτλ. Η χρήση της συνίσταται στην αυτοματοποίηση κάποιων διαδικασιών (ή στην προσθήκη κάποιων νέων) τις οποίες, είτε δεν πρόβλεψε ο κατασκευαστής, είτε πρόκειται για λειτουργίες που χρησιμεύουν σε μας, αλλά όχι και σε άλλους χρήστες. Είναι λοιπόν το εργαλείο εκείνο το οποίο θα μας βοηθήσει να προσαρμόσουμε το υπάρχον λογισμικό στις ανάγκες μας.

Ο προγραμματισμός της VBA διαφέρει από τον αντίστοιχο της Ms VB σε αρκετά σημεία. Για παράδειγμα τα εργαλεία που χρησιμοποιούνται στις περισσότερες περιπτώσεις έχουν παραπλήσια ονόματα αλλά όχι ίδια. Επίσης κάθε λογισμικό ξενιστής, πχ το Excel εισάγει τις δικές του εντολές και συναρτήσεις στη διάθεση του χρήστη. Με αυτό τον τρόπο δίνεται η δυνατότητα στο χρήστη να υπολογίσει μια συνάρτηση «τραβώντας» δεδομένα από ορισμένα κελιά του Excel.

Δημιουργία Κώδικα

Εισαγωγή

Το πρόγραμμα το οποίο δημιουργήθηκε αυτοματοποιεί τη διαδικασία σχεδίασης πύργου ανεμογεννητριών. Ο χρήστης μέσω του προγράμματος θα μπορεί μέσα σε λίγα λεπτά επιλέγοντας και προσθέτοντας στοιχεία για τον πύργο να έχει έναν ολοκληρωμένο σχεδιαστικά πύργο ανεμογεννήτριας. Ο χρήστης μπορεί να επιλέξει ο πύργος που θα κατασκευάσει να είναι από ένα (1) έως τρία (3) κομμάτια είτε αυτά είναι κυλινδρικά είτε κωνικά. Υπάρχουν και δύο κομμάτια της Βάσης και της Κορυφής τα οποία θα υπάρχουν σε κάθε πύργο και θα είναι κυλινδρικά. Αυτά τα κομμάτια ο χρήστης δε μπορεί να τα αφαιρέσει αλλά μπορεί να τους δώσει τις επιθυμητές διαστάσεις. Τέλος αφού ο χρήστης έχει επιλέξει από πόσα και τι μορφής κομμάτια θα είναι ο πύργος που επιθυμεί να κατασκευάσει, μπορεί να βάλει τις διαστάσεις σε κάθε κομμάτι και με το κουμπί «ΚΑΤΑΣΚΕΥΗ» να δοθεί η τελική μορφή του πύργου με ενωμένα όλα τα κομμάτια (assembly).

Αρχικές Ενέργειες

Βήμα 1^ο

Ξεκινάμε γράφοντας τον παρακάτω κώδικα για να δημιουργήσουμε ένα αρχείο στο οποίο να μπορούμε να δουλέψουμε και να αποθηκεύσουμε τον κώδικά μας.

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

Κατασκευή Κυλινδρικού Κομματιού

Βήμα 2°

Ορισμός μεταβλητών

Dim a As Double

Dim b As Double

Dim c As Double

Dim d As Double

Βήμα 3°

Μετατροπή κειμένου σε αριθμό και σύνδεση μεταβλητής με textbox

a = CDbI(TextBox1.Text)

b = CDbI(TextBox2.Text)

c = CDbI(TextBox3.Text)

Βήμα 4°

Μετατροπή μονάδων για να είναι σε mm ώστε ο χρήστης να χρησιμοποιήσει το πρόγραμμα σε mm (κατά ISO)

a = a / 10

b = b / 10

c = c / 10

Βήμα 5°

Μετατροπή από διάμετρο σε ακτίνα και για εύρεση εσωτερικής διαμέτρου

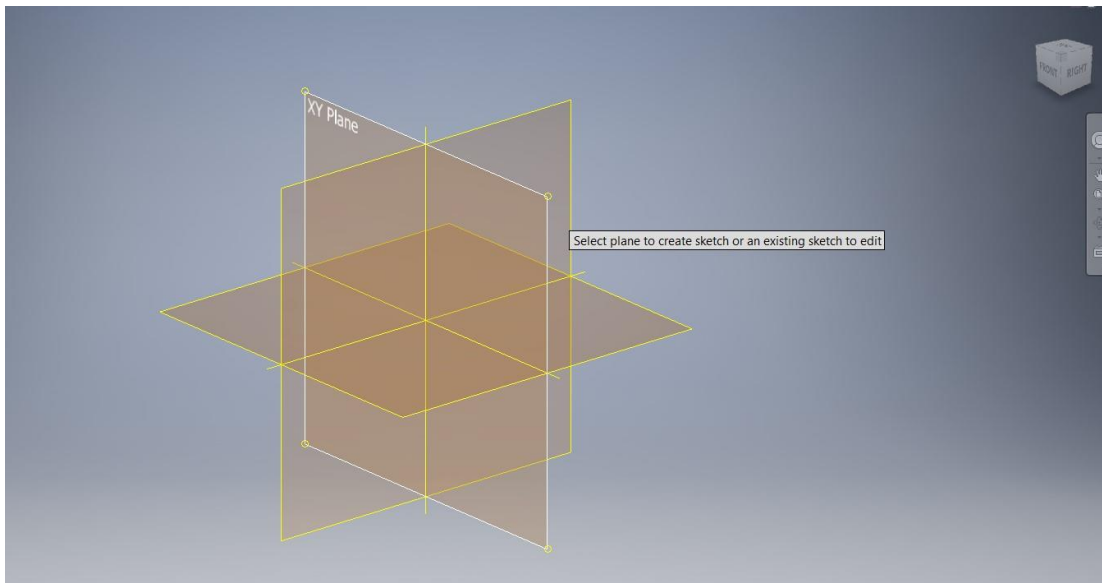
b = b / 2

d = b - c

Βήμα 6°

Επιλογή Workplane

Dim oSketch1 As PlanarSketch



Σχήμα 1.30. Επιλογή του Workplane.

Βήμα 7^ο

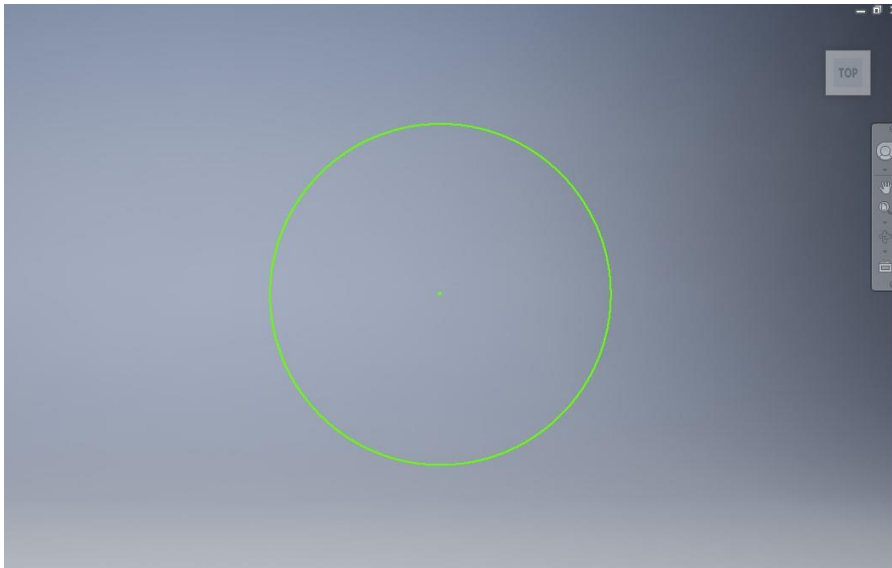
Ορισμός σημείου (0,0)

```
Set oSketch1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))
```

Βήμα 8^ο

Δημιουργία και ορισμός εξωτερικού κύκλου στο σημείο (0,0) με διάμετρο b

```
Dim oSkPnts1 As SketchPoints  
Set oSkPnts1 = oSketch1.SketchPoints  
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)  
Dim oCircs1 As SketchCircles  
Set oCircs1 = oSketch1.SketchCircles  
Dim oCirc1 As SketchCircle  
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```



Σχήμα 1.31. Δημιουργία εξωτερικού κύκλου.

Βήμα 9^ο

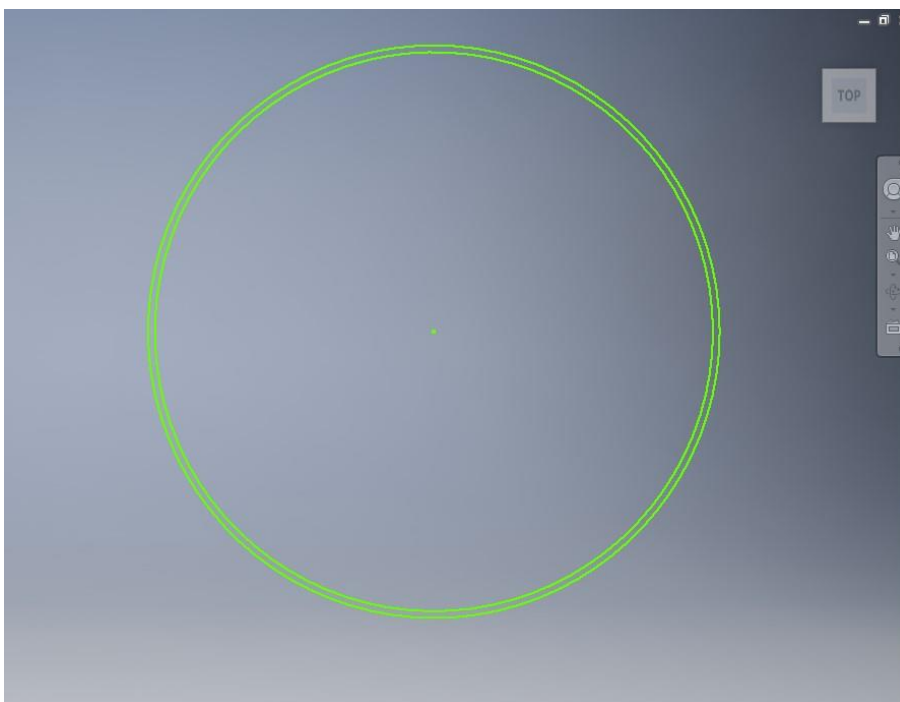
Δημιουργία και ορισμός εσωτερικού κύκλου στο σημείο (0,0) με διάμετρο d

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```



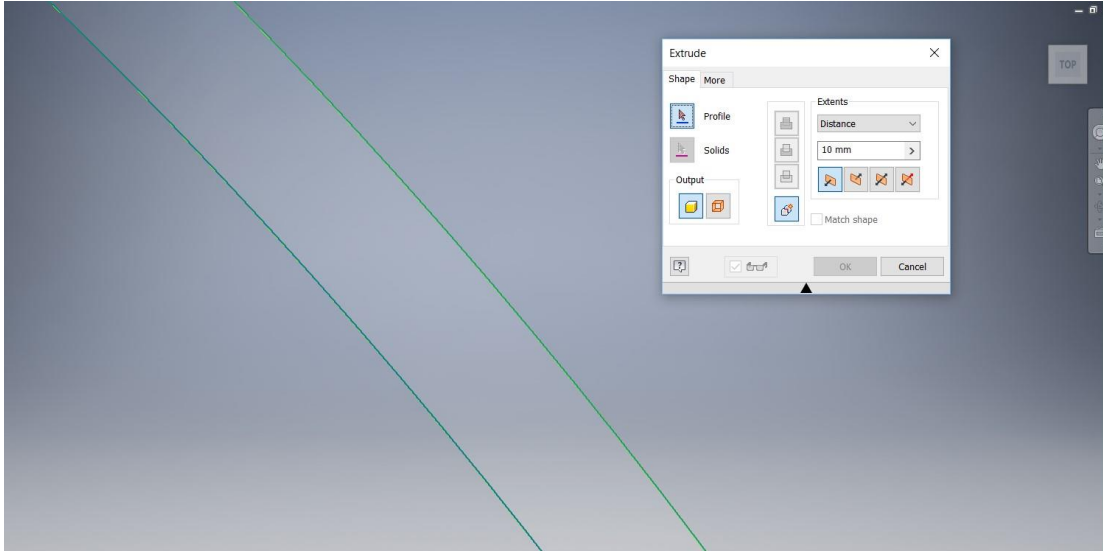
Σχήμα 1.32. Δημιουργία εσωτερικού κύκλου.

Βήμα 10°

Δημιουργία και επιλογή επιφάνειας μεταξύ των κύκλων

Dim oProfile1 As Profile

Set oProfile1 = oSketch1.Profiles.AddForSolid



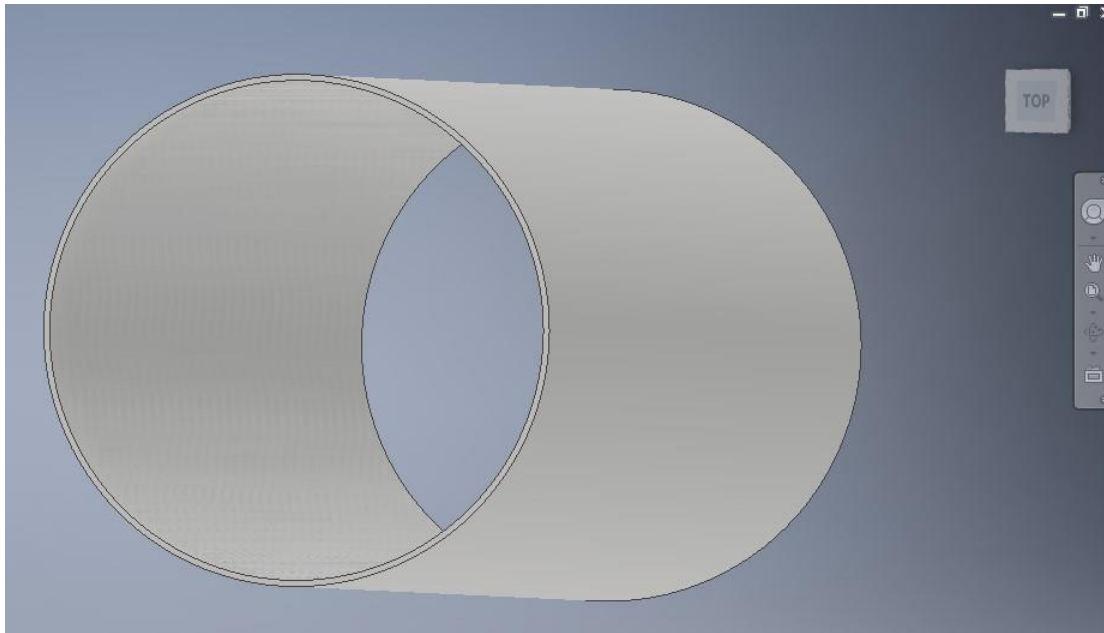
Σχήμα 1.33. Επιλογή επιφάνειας μεταξύ των κύκλων.

Βήμα 11°

Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στον κύλινδρο

Dim oExtrusion1 As ExtrudeFeature

Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile1, a, kPositiveExtentDirection, kJoinOperation)



Σχήμα 1.34. Επιλογή της εντολής Extrude.

Βήμα 12°

Δημιουργία πάνω Φλάντζας (ακολουθήθηκαν τα ίδια βήματα με την δημιουργία του κυλίνδρου)

Dim fd As Double

Dim fb As Double

fd = Cdbl(TextBox5.Text)

fb = Cdbl(TextBox4.Text)

fd = fd / 10

fb = fb / 10

fd = fd / 2

Βήμα 13°

Δημιουργία και επιλογή κατάλληλου Workplane

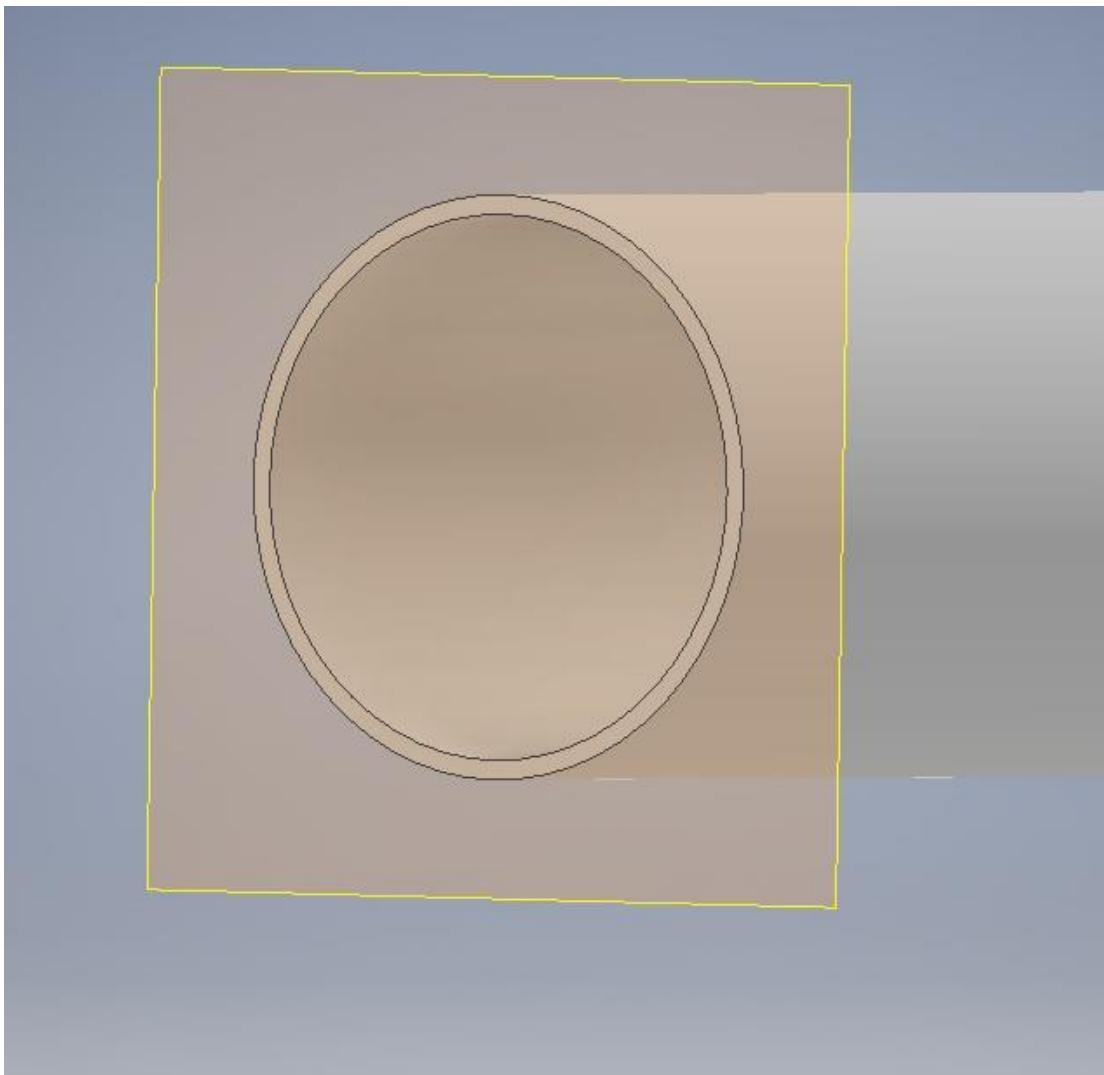
```
Dim oWPlane1 As WorkPlane
```

```
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
```

```
Set oSketchf2 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```



Σχήμα 1.35. Δημιουργία κατάλληλου Workplane πάνω στο κομμάτι κατασκευής.

Βήμα 14°

Ορισμός σημείου (0,0)

DimoSkPntsf2 AsSketchPoints

Set oSkPntsf2 = oSketchf2.SketchPoints

Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)

Βήμα 15°

Δημιουργία και ορισμός εξωτερικού κύκλου στο σημείο (0,0) με διάμετρο d

Dim oCircsf3 As SketchCircles

Set oCircsf3 = oSketchf2.SketchCircles

Dim oCircf3 As SketchCircle

Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)

Βήμα 16°

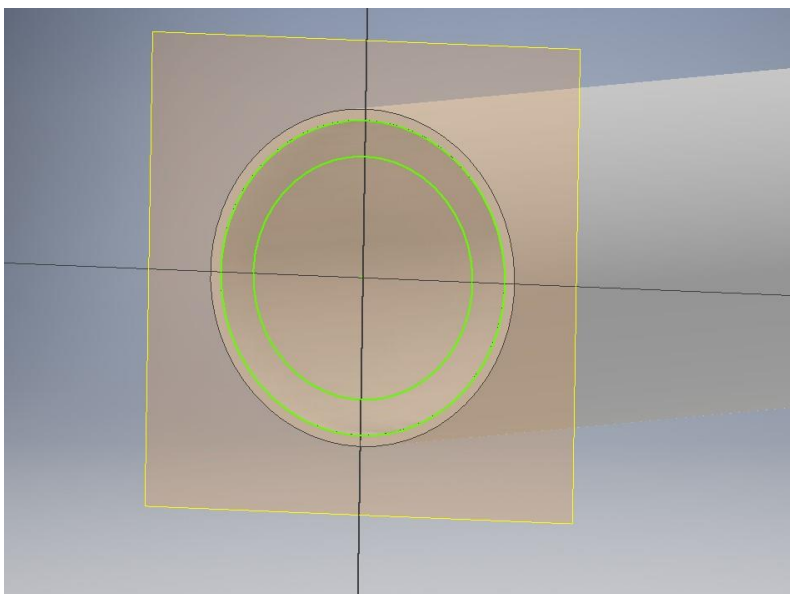
Δημιουργία και ορισμός εξωτερικού κύκλου στο σημείο (0,0) με διάμετρο fd

Dim oCircsf4 As SketchCircles

Set oCircsf4 = oSketchf2.SketchCircles

Dim oCircf4 As SketchCircle

Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)



Σχήμα 1.36. Δημιουργία εξωτερικού και εσωτερικού κύκλου για δημιουργία φλάντζας.

Βήμα 17°

Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στη φλάντζα και γίνονται κάποιες πράξεις για να βγαίνει το d σε mm

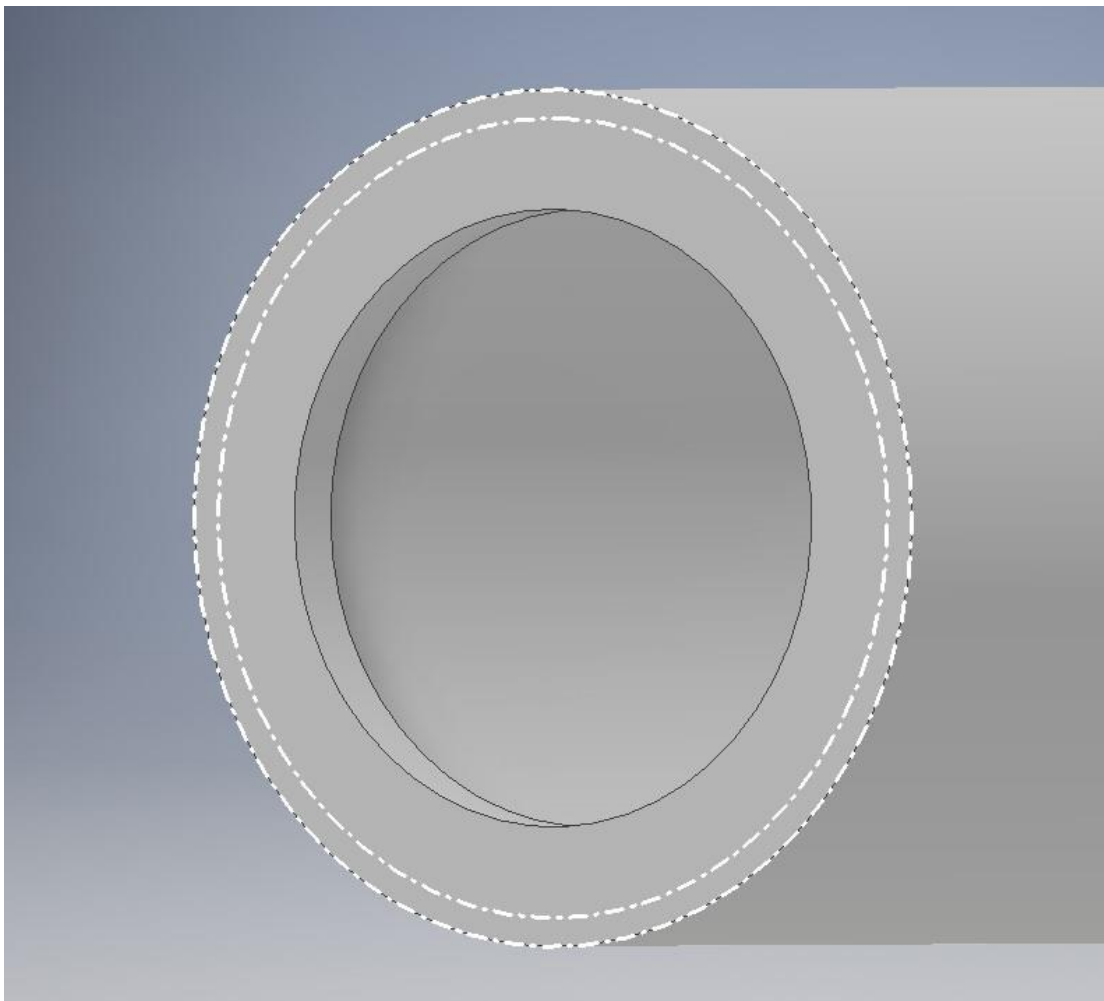
```
Dim oProfile2 As Profile
```

```
Set oProfile2 = oSketch2.Profiles.AddForSolid
```

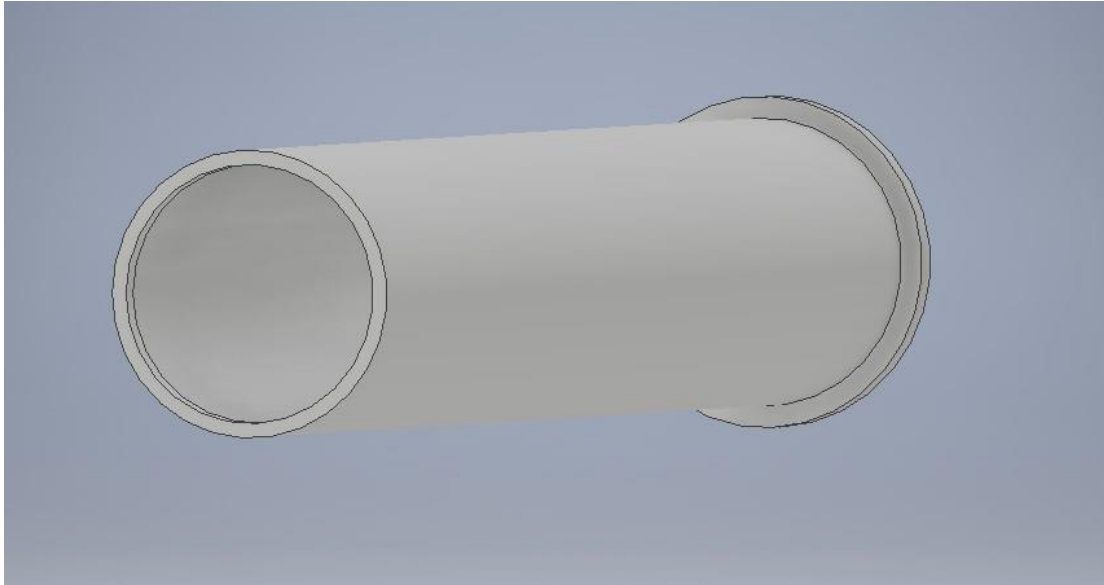
```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```



Σχήμα 1.37. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στη φλάντζα.



Σχήμα 1.38. Τελικό κομμάτι Βάσης (με εξωτερική κάτω φλάντζα).

Σημείωση: Η Κορυφή όπως και ένα κυλινδρικό κομμάτι δημιουργούνται με τον ίδιο τρόπο μόνο που έχουν στο κάτω μέρος τους εσωτερική φλάντζα.

Κατασκευή Κωνικού Κομματιού

Βήμα 18°

Ορισμός μεταβλητών

Dim j As Double

Dim l As Double

Dim m As Double

Dim n As Double

Dim o AsDouble

Βήμα 19°

Μετατροπή κειμένου σε αριθμό και σύνδεση μεταβλητής με textbox

j = CDbl(TextBox6.Text)

l = CDbl(TextBox9.Text)

m = CDbl(TextBox8.Text)

Βήμα 20°

Μετατροπή μονάδων για να είναι σε mm

$$j = j / 10$$

$$l = l / 10$$

$$m = m / 10$$

Βήμα 21°

Μετατροπή από διάμετρο σε ακτίνα και για εύρεση εσωτερικής διαμέτρου

$$l = l / 2$$

$$m_i = b - m$$

Βήμα 22°

Δημιουργία Κλίσης

$$n = (b - l) / \text{Sqr}((j)^2 + (b - l)^2)$$

Βήμα 23°

Επιλογή Workplane(ίδιο workplane με πάνω φλάντζα)

Dim oSketch2 As PlanarSketch

Set oSketch2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSketch3 As PlanarSketch

Set oSketch3 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Βήμα 24°

Δημιουργία δυο σημείων στις συντεταγμένες (0,0)

```
Dim oSkPnts2 As SketchPoints
```

```
Set oSkPnts2 = oSketch2.SketchPoints
```

```
Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oSkPnts3 As SketchPoints
```

```
Set oSkPnts3 = oSketch3.SketchPoints
```

```
Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)
```

Βήμα 25°

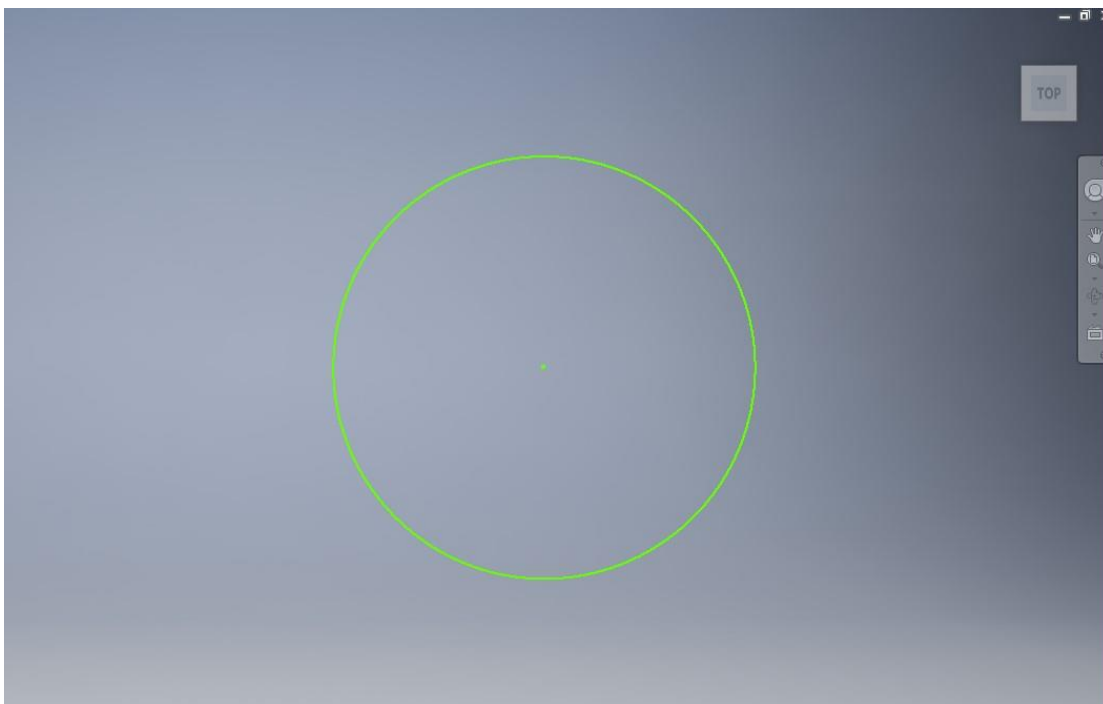
Δημιουργία και ορισμός εξωτερικού κύκλου στο σημείο (0,0) με διάμετρο b (όμοια με επάνω φλάντζα του κυλινδρικού κομματιού)

```
Dim oCircs3 As SketchCircles
```

```
Set oCircs3 = oSketch2.SketchCircles
```

```
Dim oCirc3 As SketchCircle
```

```
Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)
```



Σχήμα 1.39. Δημιουργία εξωτερικού κύκλου.

Βήμα 26°

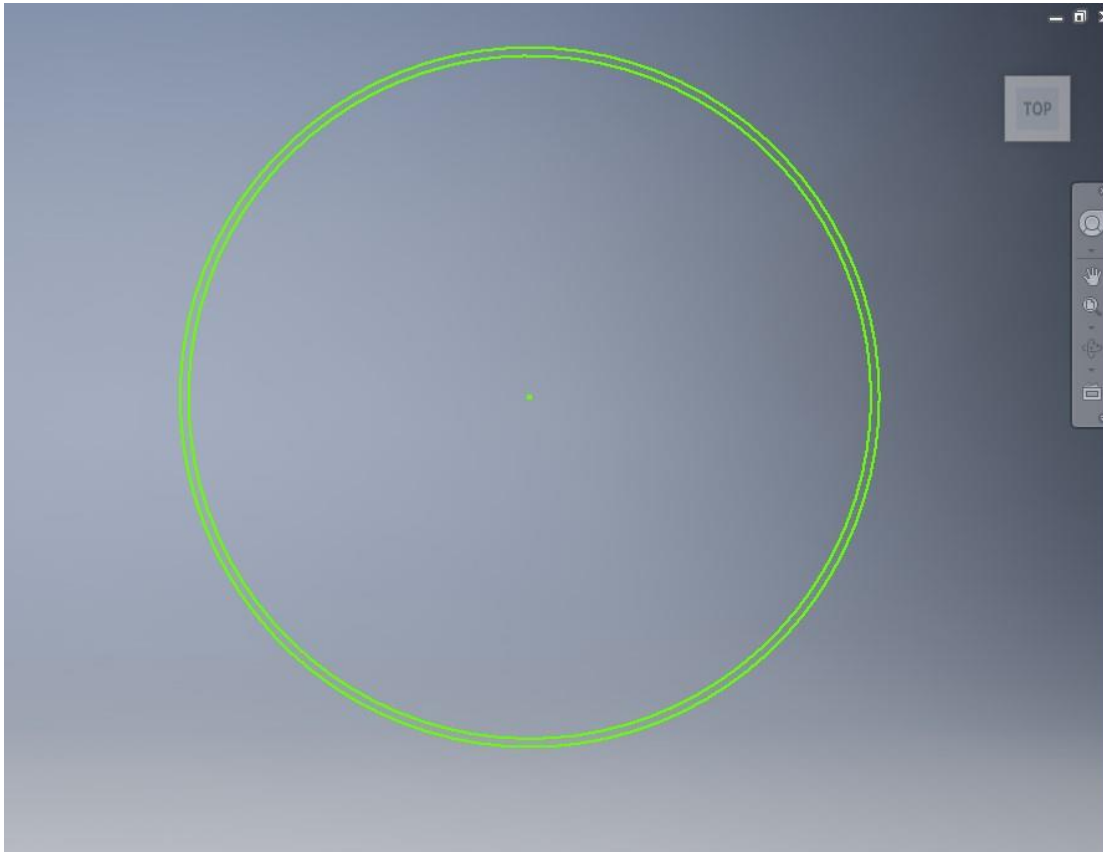
Δημιουργία και ορισμός εσωτερικού κύκλου στο σημείο (0,0) με διάμετρο mi

```
Dim oCircs4 As SketchCircles
```

```
Set oCircs4 = oSketch3.SketchCircles
```

```
Dim oCirc4 As SketchCircle
```

```
Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)
```



Σχήμα 1.40. Δημιουργία εσωτερικού κύκλου.

Βήμα 27°

Δημιουργούμε δύο προφίλ για να χρησιμοποιήσουμε τις εντολές Extrude και Cut στο αντίστοιχο προφίλ.

```
Dim oProfile2 As Profile
```

```
Set oProfile2 = oSketch2.Profiles.AddForSolid
```

```
Dim oProfile3 As Profile
```

```
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

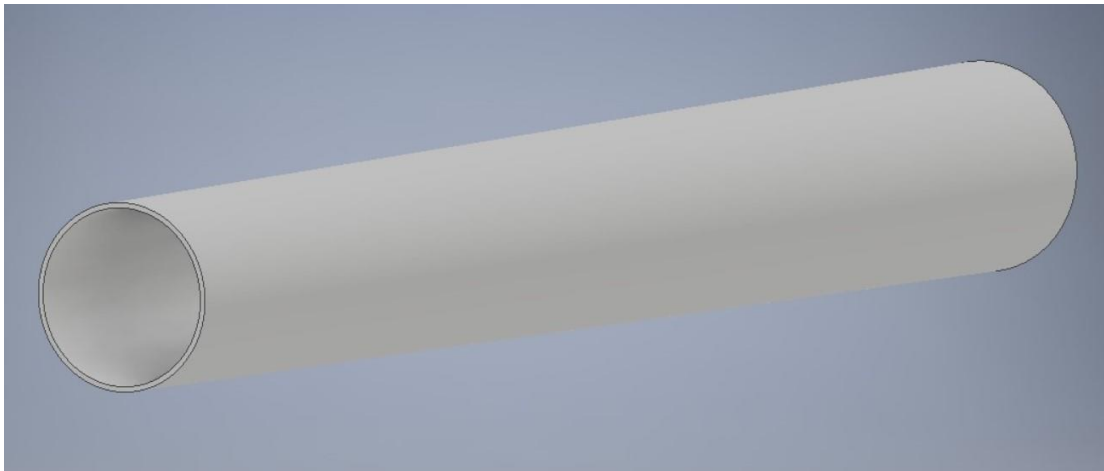
Βήμα 28°

Δημιουργία Extrude και Cut στα Profiles που δημιουργήσαμε.

```
Dim oExtrusion2 As ExtrudeFeature
```

```
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)
```

```
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)
```



Σχήμα 1.41. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στο κωνικό κομμάτι.

Κατασκευή Κωνικής Φλάντζας

Βήμα 29°

Ορισμός μεταβλητών

```
Dim fe As Double
```

```
Dim ffAs Double
```

Βήμα 30°

Μετατροπή κειμένου σε αριθμό και σύνδεση μεταβλητής με textbox

```
fe = Cdbl(TextBox10.Text)
```

```
ff = Cdbl(TextBox11.Text)
```

Βήμα 31°

Μετατροπή μονάδων για να είναι σε mm

$$fe = fe / 10$$

$$ff = ff / 10$$

Βήμα 32°

Μετατροπή από διάμετρο σε ακτίνα και για εύρεση εσωτερικής διαμέτρου

$$fe = fe / 2$$

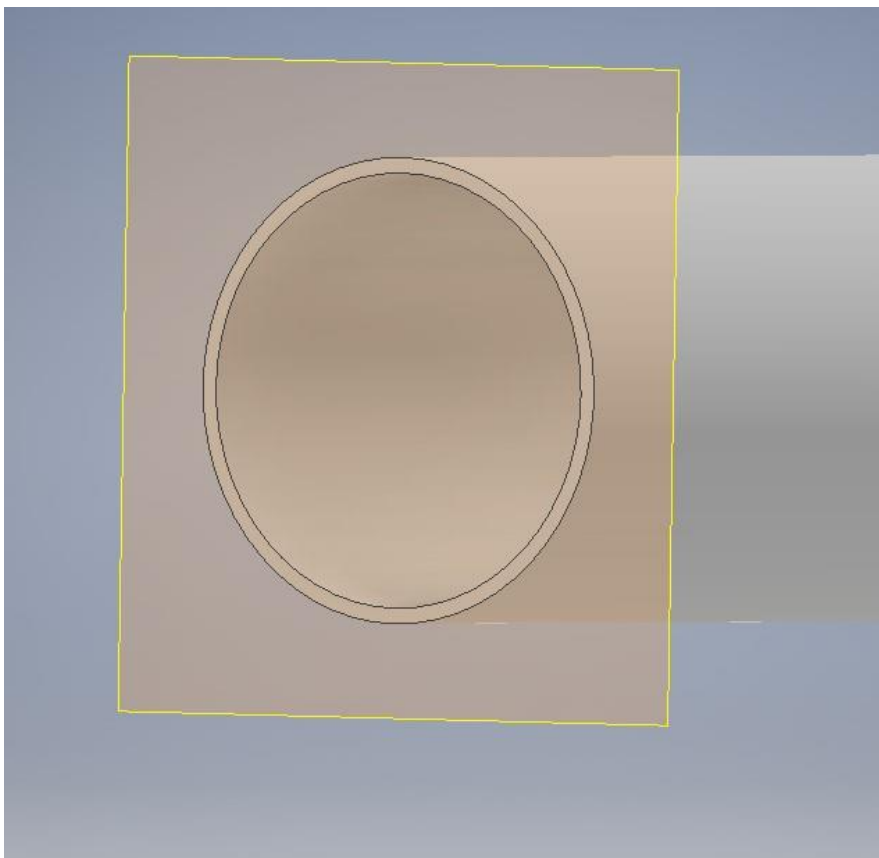
Βήμα 33°

Επιλογή Workplane(ίδιο με το προηγούμενο workplane)

Dim oSketchf3 As PlanarSketch

Set oSketchf3 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))



Σχήμα 1.42. Δημιουργία κατάλληλου Workplane πάνω στο κομμάτι κατασκευής.

Βήμα 34°

Δημιουργία ενός σημείου στις συντεταγμένες (0,0)

Dim oSkPntsf3 As SketchPoints

Set oSkPntsf3 = oSketchf3.SketchPoints

Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)

Βήμα 35°

Δημιουργία και ορισμός εξωτερικού κύκλου στο σημείο (0,0) με διάμετρο mi (όμοια με το προηγούμενο)

Dim oCircsf5 As SketchCircles

Set oCircsf5 = oSketchf3.SketchCircles

Dim oCircf5 As SketchCircle

Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)

Βήμα 36°

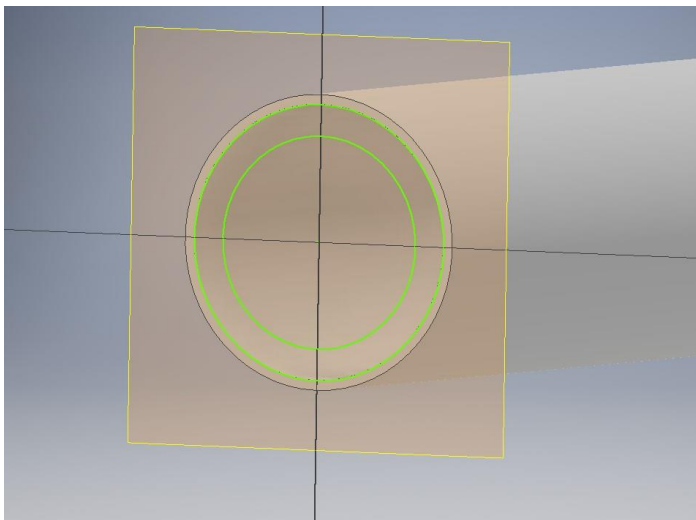
Δημιουργία και ορισμός εσωτερικού κύκλου στο σημείο (0,0) με διάμετρο fe

Dim oCircsf6 As SketchCircles

Set oCircsf6 = oSketchf3.SketchCircles

Dim oCircf6 As SketchCircle

Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)



Σχήμα 1.43. Δημιουργία εξωτερικού και εσωτερικού κύκλου για δημιουργία φλάντζας.

Βήμα 37°

Δημιουργούμε ένα προφίλ για να χρησιμοποιήσουμε την εντολή Extrude

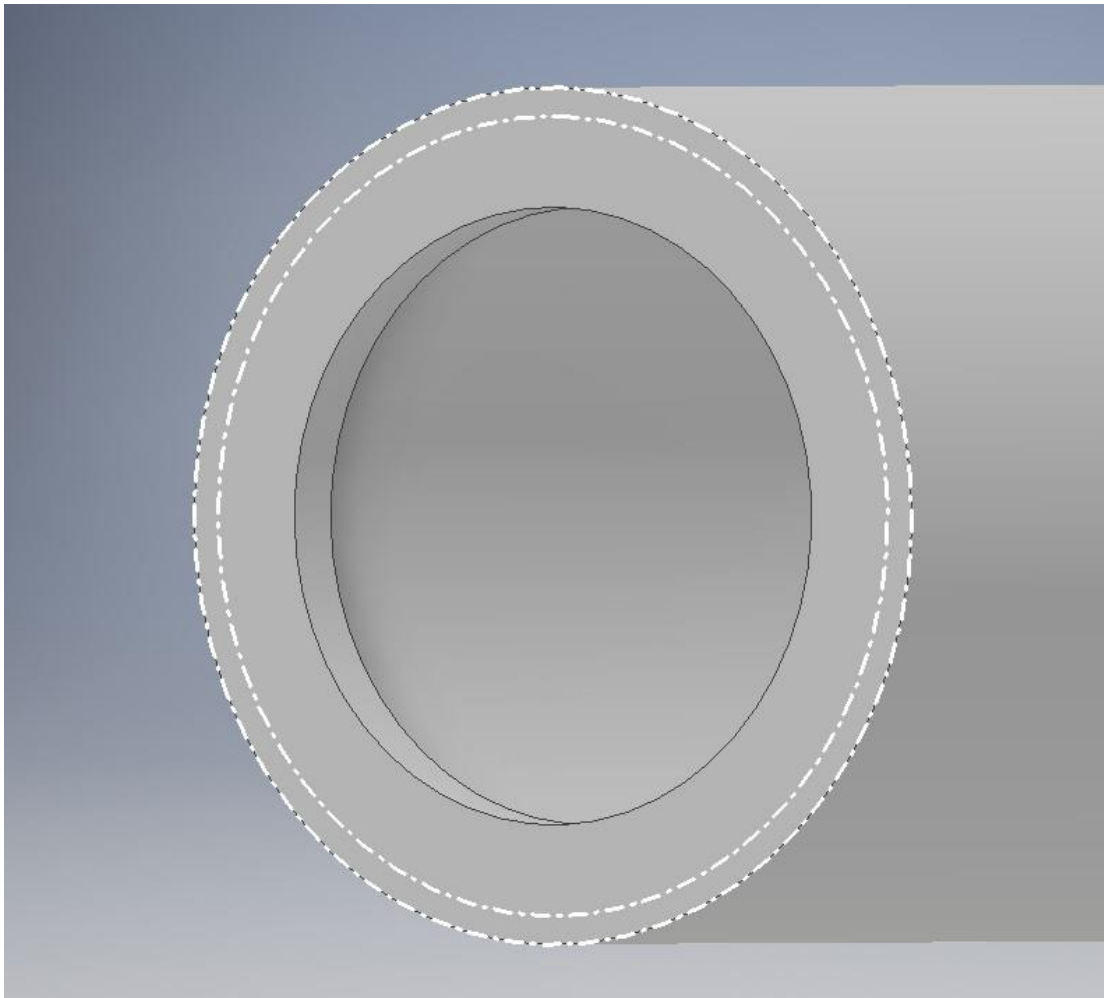
```
Dim oProfile3 As Profile
```

```
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

Βήμα 38°

Δημιουργία Extrude στο προφίλ που δημιουργήσαμε.

```
Set oextrude3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```



Σχήμα 1.44. Με την εντολή Extrude δίνεται ο απαραίτητος όγκος στη φλάντζα.

Η επάνω φλάντζα δημιουργείται με τον ίδιο τρόπο, εκτός από τη δημιουργία ενός ακόμα workplane στο σημείο όπου τελειώνει το κωνικό κομμάτι. (βλέπε παρακάτω).

Βήμα 39°

Δημιουργία μεταβλητής για το workplane

$fi = a + j$

Όπου μεταβλητή a και j είναι το μήκος του κυλινδρικού και κωνικού κομματιού αντίστοιχα.

Βήμα 40°

Δημιουργία workplane

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi)

Βήμα 41°

Επιλογή αντίστοιχου workplane

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Κατασκευή Κομματιού Βάσης (με εξωτερική φλάντζα)

Ακολουθούμε την ίδια διαδικασία όπως ενός κυλινδρικού κομματιού. Η φλάντζα στο επάνω μέρος παραμένει εσωτερική ενώ την φλάντζα στο κάτω μέρος του κομματιού την κάνουμε εξωτερική.

Dim fa As Double

Dim fc As Double

$fa = CDbl(\text{TextBox15.Text})$

$fc = CDbl(\text{TextBox17.Text})$

$fa = fa / 10$

$fc = fc / 10$

$fc = fc / 2$


```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf1 As SketchCircles
```

```
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
```

```
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
```

```
Dim oCircsf2 As SketchCircles
```

```
Set oCircsf2 = oSketchf1.SketchCircles
```

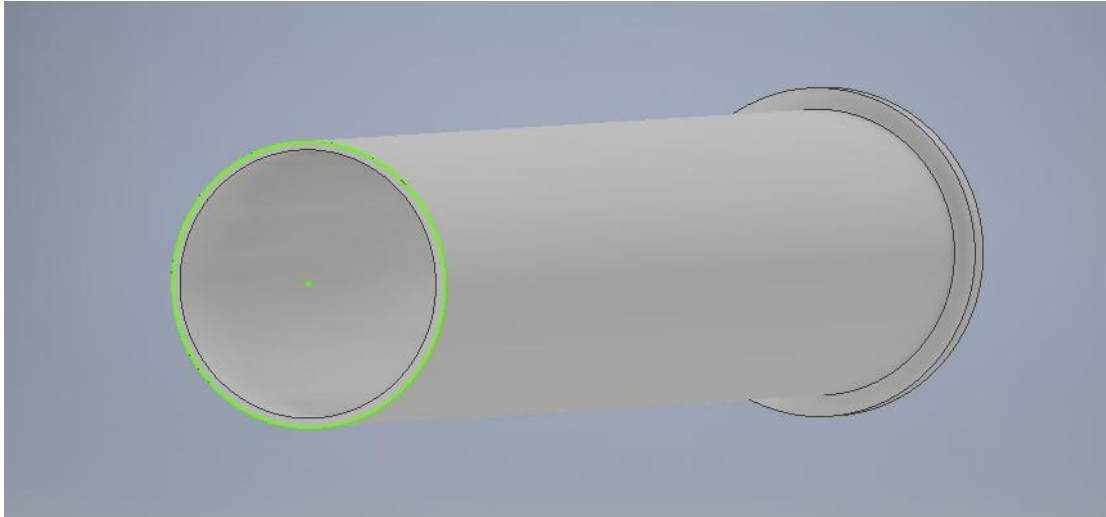
```
Dim oCircf2 As SketchCircle
```

```
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
```

```
Dim oProfilef1 As Profile
```

```
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```



Σχήμα 1.45. Ολοκληρωμένη κατασκευή κομματιού βάσης.

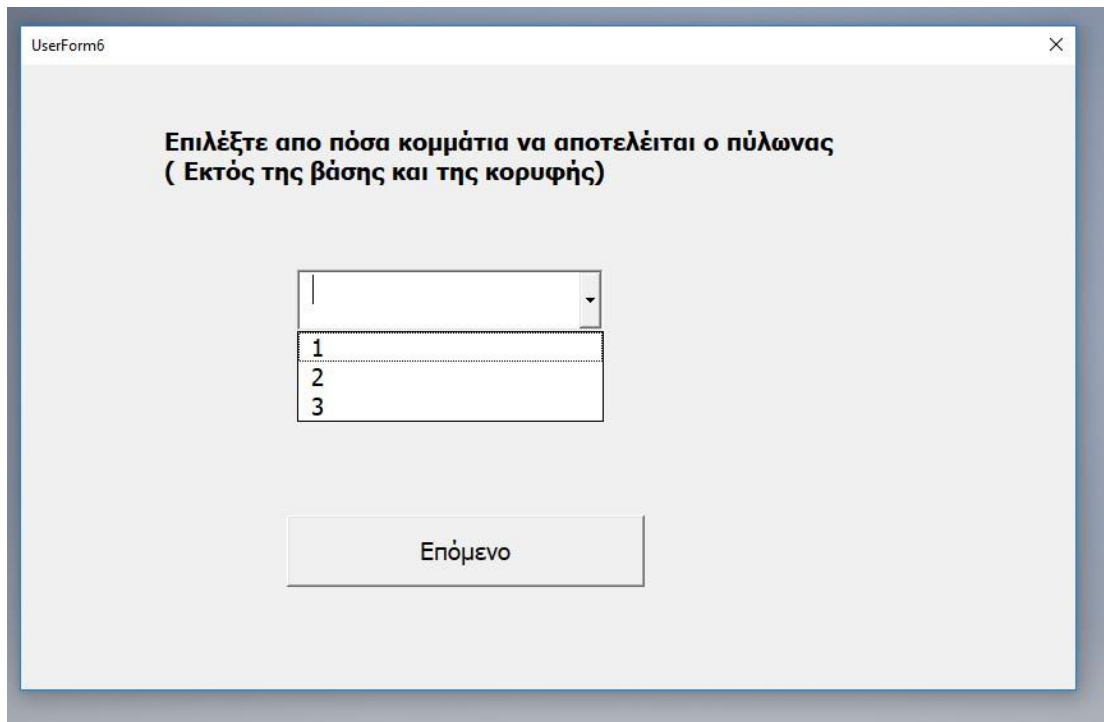
Παραδείγματα εφαρμογής 1

Το πρώτο παράδειγμα αποτελείται από ένα κωνικό κομμάτι (εκτός βάσης και κορυφής)

Σημείωση: Τα κομμάτια της βάσης και της κορυφής δημιουργούνται αυτόματα σύμφωνα με τα δεδομένα που θα βάλει ο χειριστής στο ενδιαμέσο κομμάτι.

Αρχικά επιλέγουμε από πόσα κομμάτια θέλουμε να αποτελείται ο πύλωνας μας.

Για αυτό το παράδειγμα επιλέξαμε να αποτελείται από ένα (1) κομμάτι.

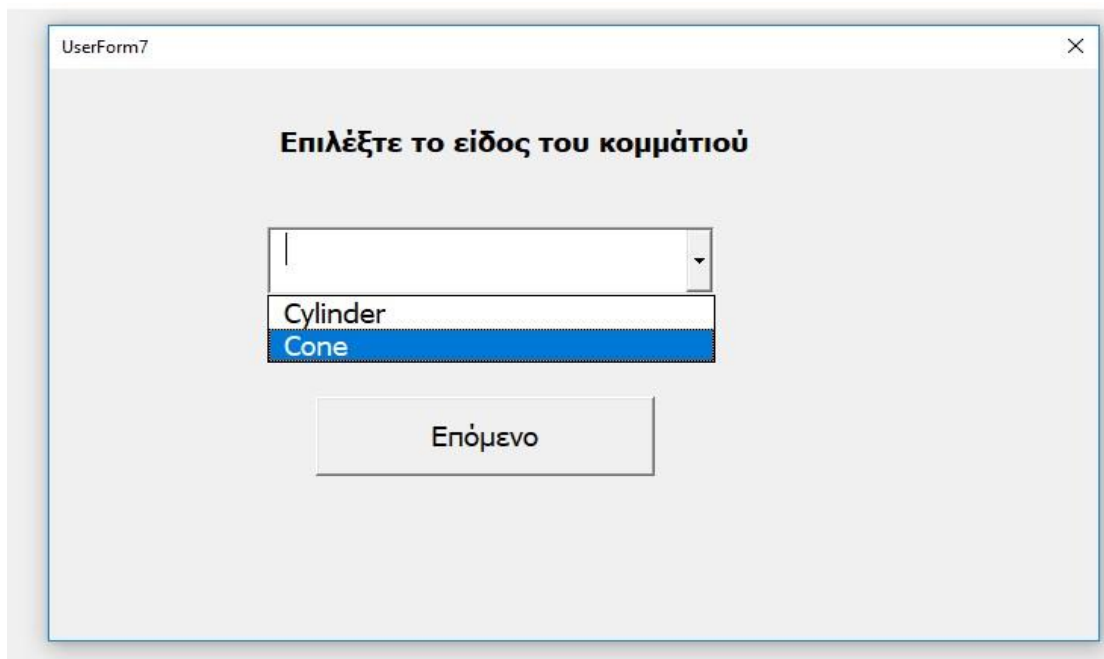


The screenshot shows a window titled "UserForm6" with a close button in the top right corner. The main content area contains the following text: "Επιλέξτε απο πόσα κομμάτια να αποτελείται ο πύλωνας (Εκτός της βάσης και της κορυφής)". Below this text is a dropdown menu with a white background and a downward-pointing arrow on the right side. The dropdown menu is open, showing three options: "1", "2", and "3". Below the dropdown menu is a rectangular button with the text "Επόμενο" (Next).

Σχήμα 1.46. Μέσω του 1^{ου} User Form ο χρήστης επιλέγει τον αριθμό των κομματιών που θα αποτελείται ο πύργος.

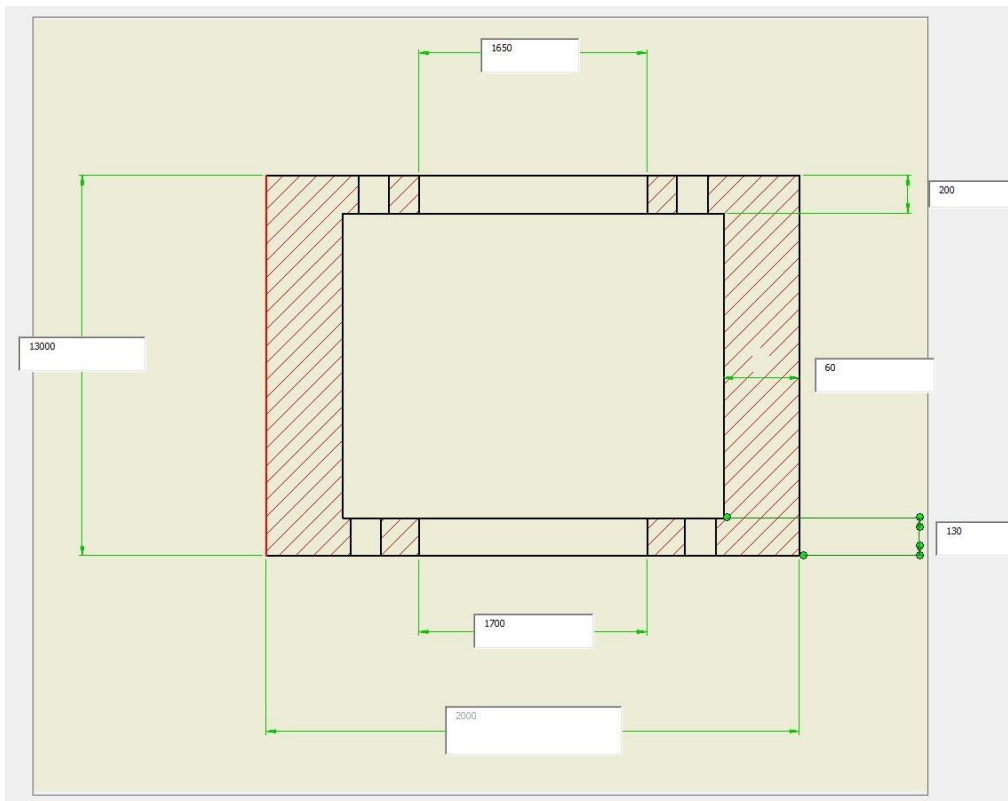
Σε δεύτερο βήμα επιλέγουμε αν το κομμάτι που θέλουμε να κατασκευάσουμε θα είναι κυλινδρικό ή κωνικό.

Εμείς επιλέξαμε για το πρώτο παράδειγμα κωνικό κομμάτι.

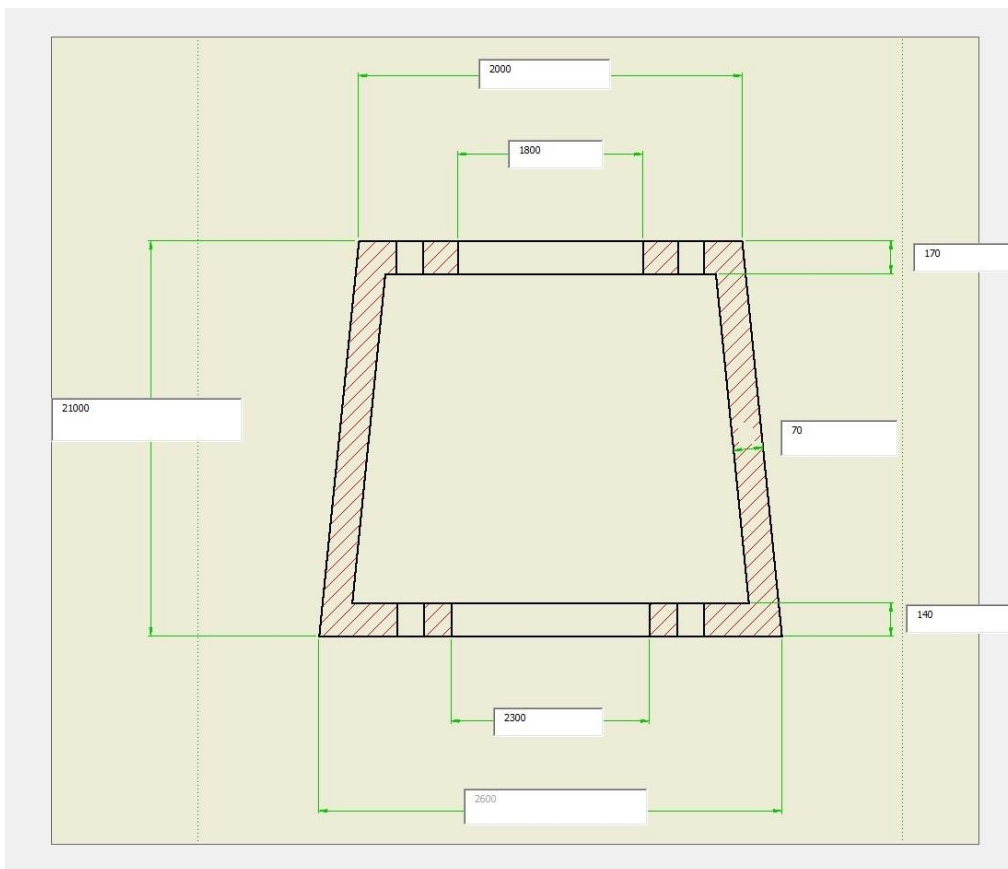


Σχήμα 1.47. Μέσω του 2^{ου} User Form ο χρήστης επιλέγει την μορφή των κομματιών που θα αποτελείται ο πύργος (κυλινδρικά η κωνικά κομμάτια).

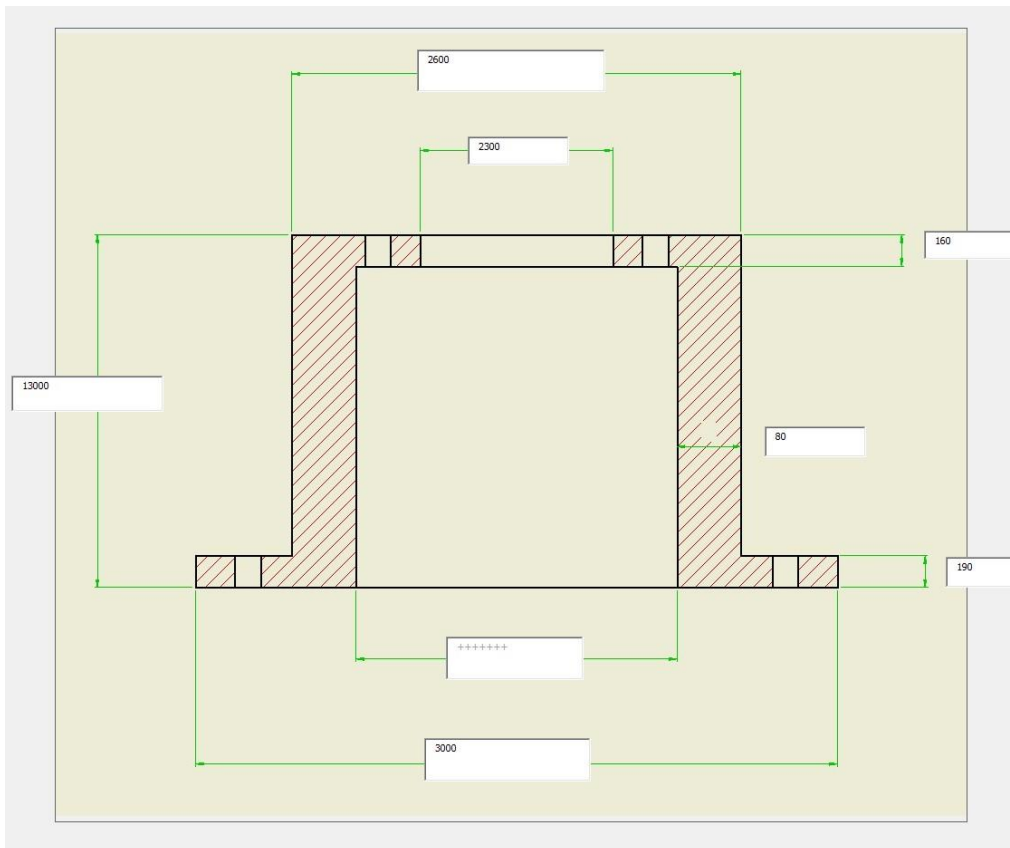
Στο επόμενο βήμα βάζουμε τις επιθυμητές διαστάσεις και πατάμε το κουμπί Κατασκευή για να κατασκευάσουμε τον πυλώνα ή το κουμπί Κλείσιμο αν κάναμε λάθος και θέλουμε να πάμε πίσω στην αρχική και να δημιουργήσουμε εκ νέου έναν διαφορετικό πυλώνα.



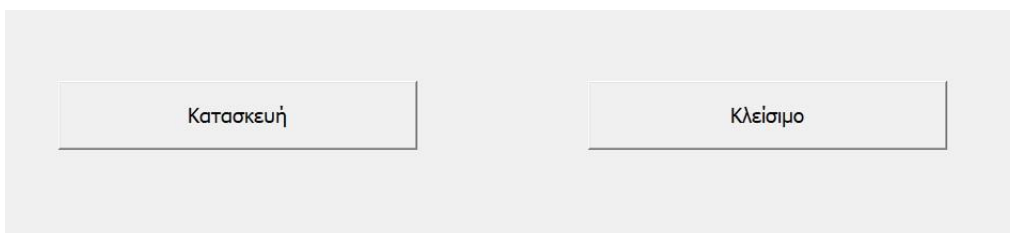
Σχήμα 1.48. Κομμάτι Κορυφής.



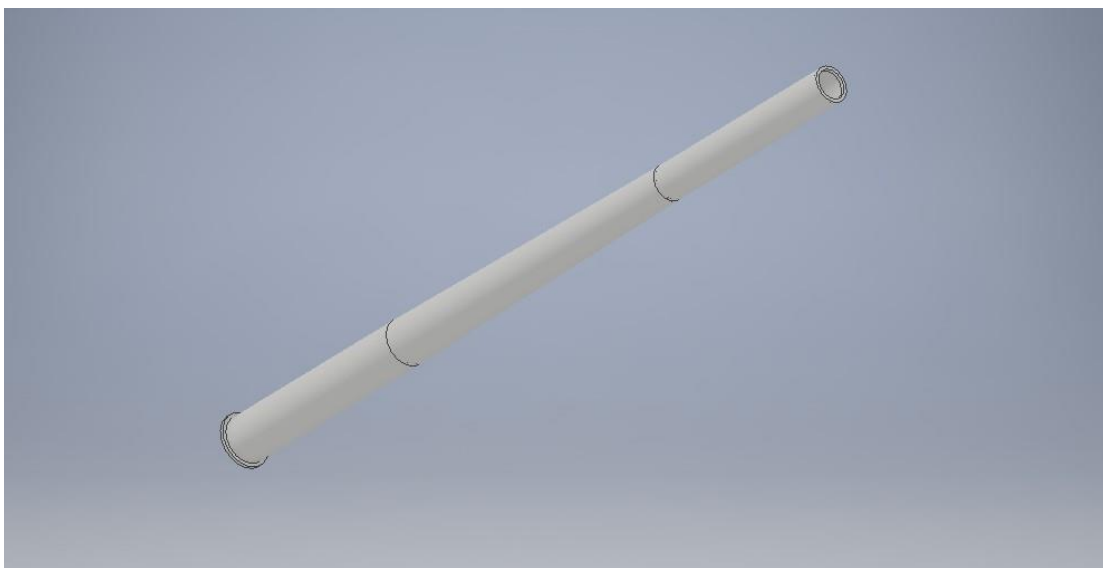
Σχήμα 1.49. Κωνικό κομμάτι.



Σχήμα 1.50. Κομμάτι Βάσης.

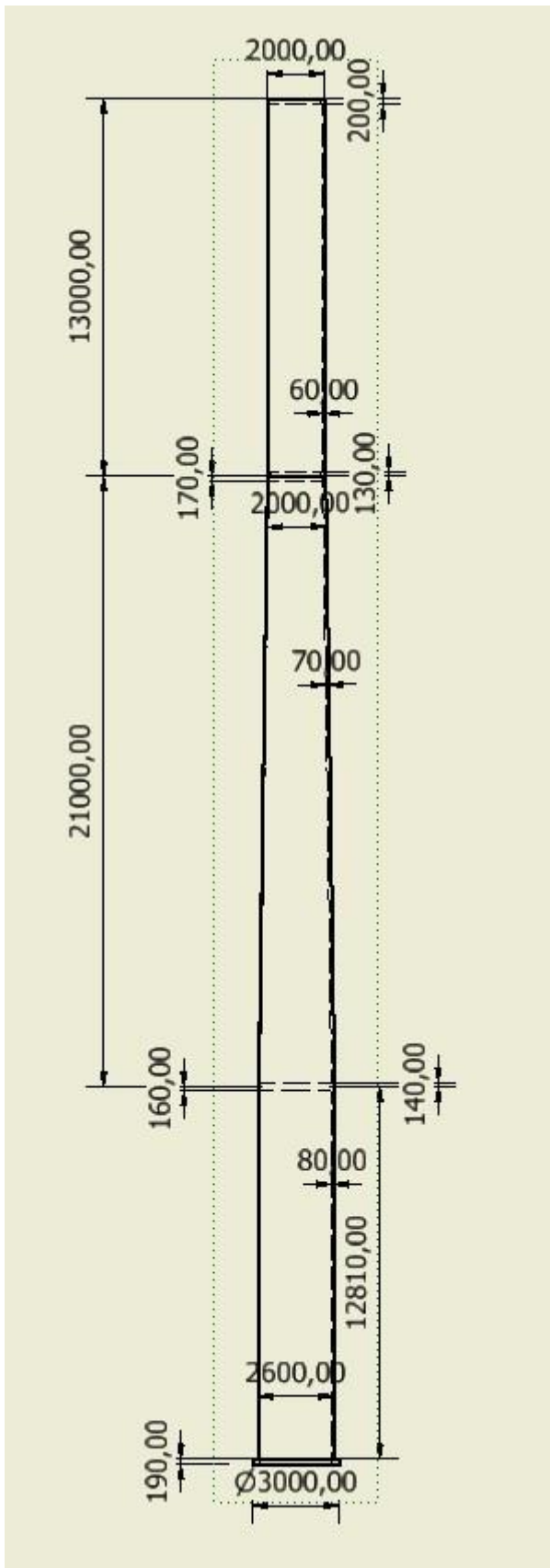


Ο πυλώνας που δημιουργήσαμε σε 3D φαίνεται στο παρακάτω σχήμα.



Σχήμα 1.51. Ολοκληρωμένη μορφή πύργου.

Σε μηχανολογικό σχέδιο (πρόοψη)



Σχήμα 1.52. Ολοκληρωμένη μορφή πύργου σε μηχανολογικό σχέδιο.

Παράδειγμα εφαρμογής 2

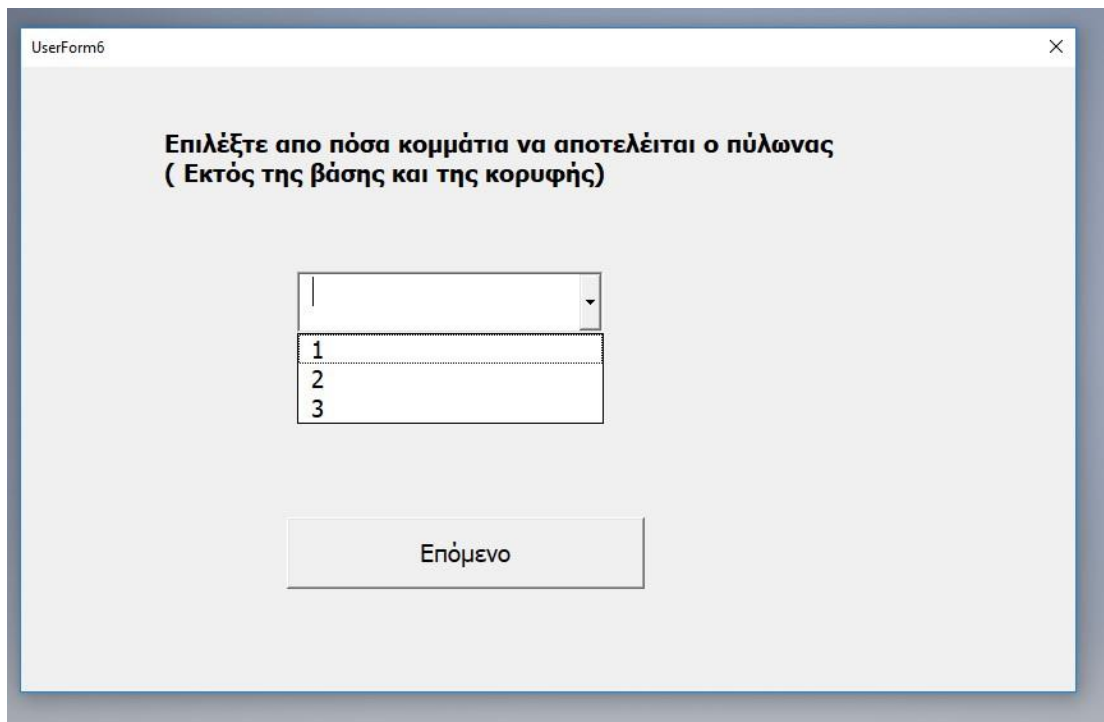
Το δεύτερο παράδειγμα αποτελείται από τρία κομμάτια (εκτός βάσης και κορυφής).

Πρώτα δύο κυλινδρικά κομμάτια και έπειτα ένα κωνικό πριν το κομμάτι της κορυφής διότι όπως προαναφέρθηκε δεν μπορούμε να βάλουμε το κωνικό πιο κάτω από τα κυλινδρικά κομμάτια.

Σημείωση: Τα κομμάτια της βάσης και της κορυφής δημιουργούνται αυτόματα σύμφωνα με τα δεδομένα που έχει βάλει ο χειριστής στο κομμάτια που έρχονται σε επαφή με τα κομμάτια της βάσης και της κορυφής αντίστοιχα. Δηλαδή το κομμάτι της βάσης θα πάρει δεδομένα από το πρώτο κυλινδρικό κομμάτι και το κομμάτι της κορυφής από το τρίτο κομμάτι (το κωνικό).

Αρχικά επιλέγουμε από πόσα κομμάτια θέλουμε να αποτελείται ο πύλωνας μας.

Για αυτό το παράδειγμα επιλέξαμε να αποτελείται από τρία (3) κομμάτια.



UserForm6

Επιλέξτε απο πόσα κομμάτια να αποτελείται ο πύλωνας
(Εκτός της βάσης και της κορυφής)

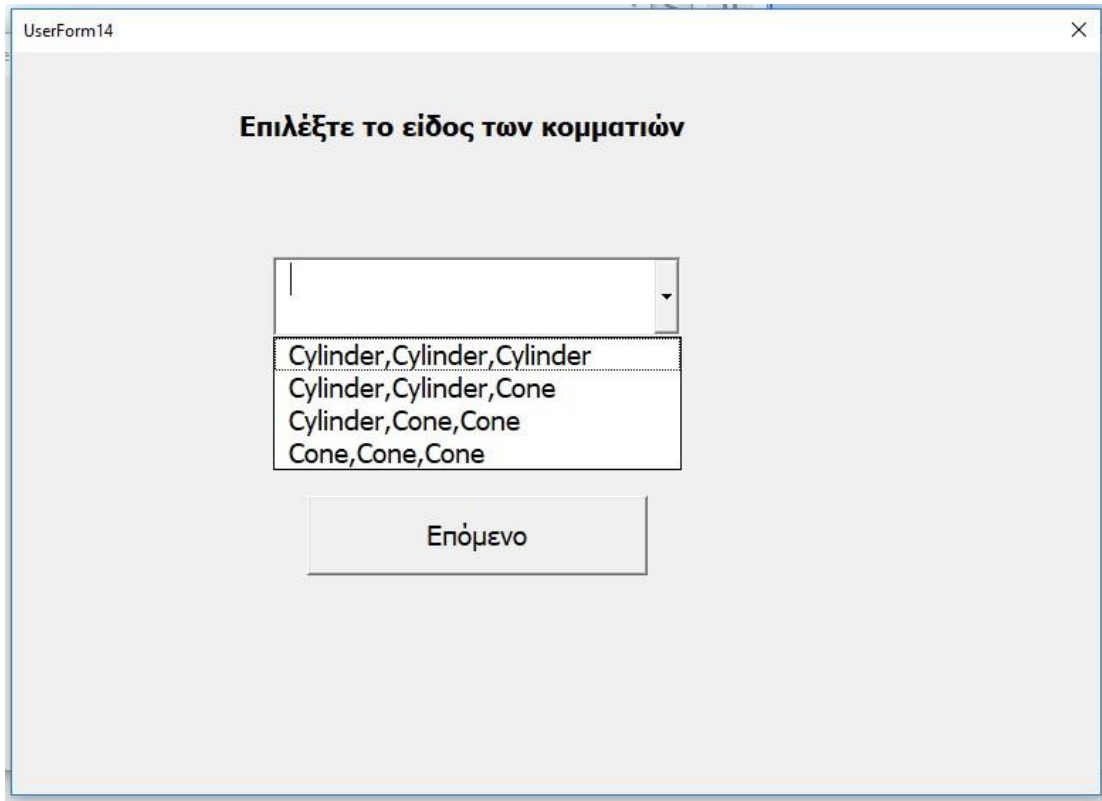
1
2
3

Επόμενο

Σχήμα 1.53. Μέσω του 1^{ου} User Form ο χρήστης επιλέγει τον αριθμό των κομματιών που θα αποτελείται ο πύργος.

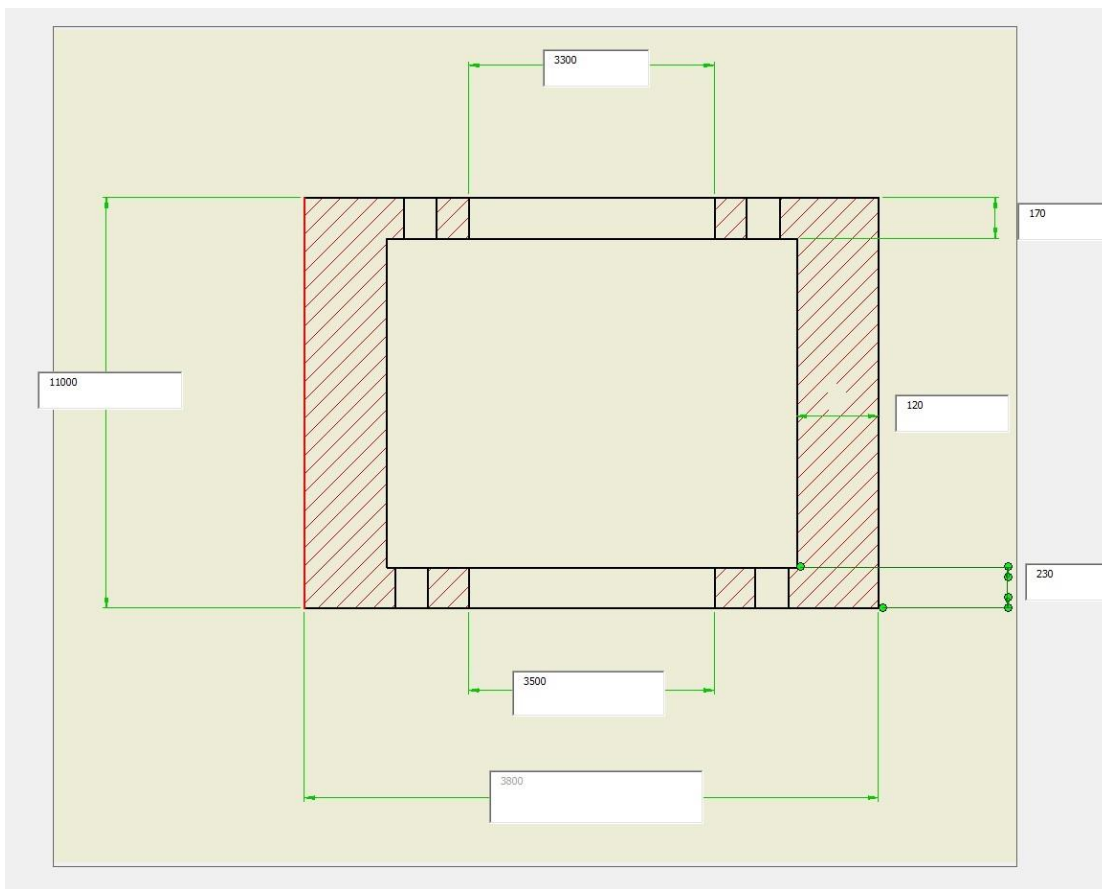
Σε δεύτερο βήμα επιλέγουμε τη μορφή των κομματιών από τις ήδη διαμορφωμένες επιλογές που μας δίνονται.

Εμείς επιλέξαμε την δεύτερη επιλογή , δηλαδή κύλινδρος, κύλινδρος , κώνος.

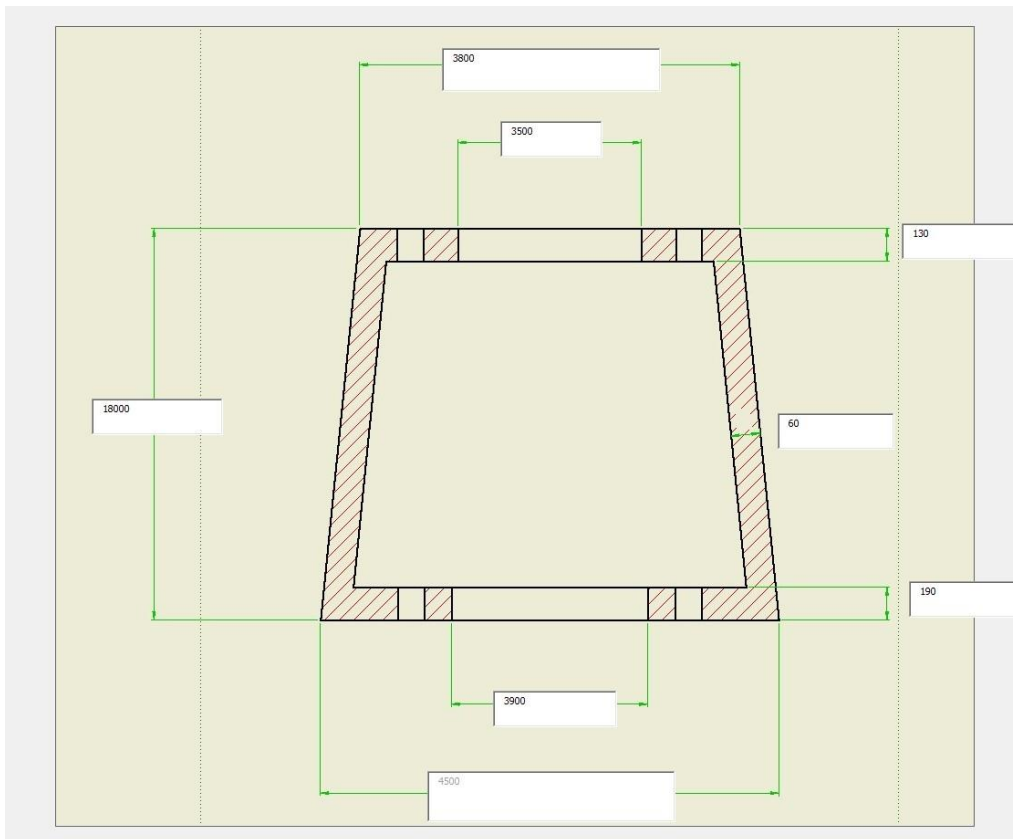


Σχήμα 1.54. Μέσω του 2^{ου} User Form ο χρήστης επιλέγει την μορφή των κομματιών που θα αποτελείται ο πύργος (κυλινδρικά ή κωνικά κομμάτια).

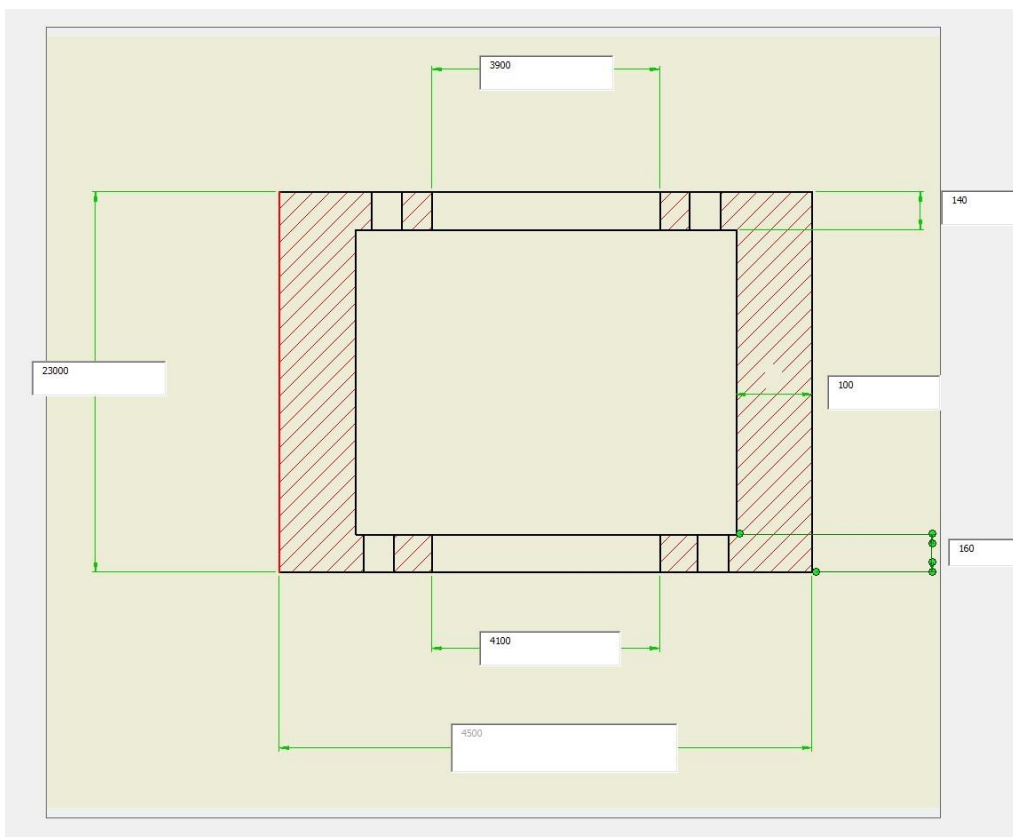
Στο επόμενο βήμα βάζουμε τις επιθυμητές διαστάσεις και πατάμε το κουμπί Κατασκευή για να κατασκευάσουμε τον πυλώνα ή το κουμπί Κλείσιμο αν κάναμε λάθος και θέλουμε να πάμε πίσω στην αρχική και να δημιουργήσουμε εκ νέου έναν διαφορετικό πυλώνα.



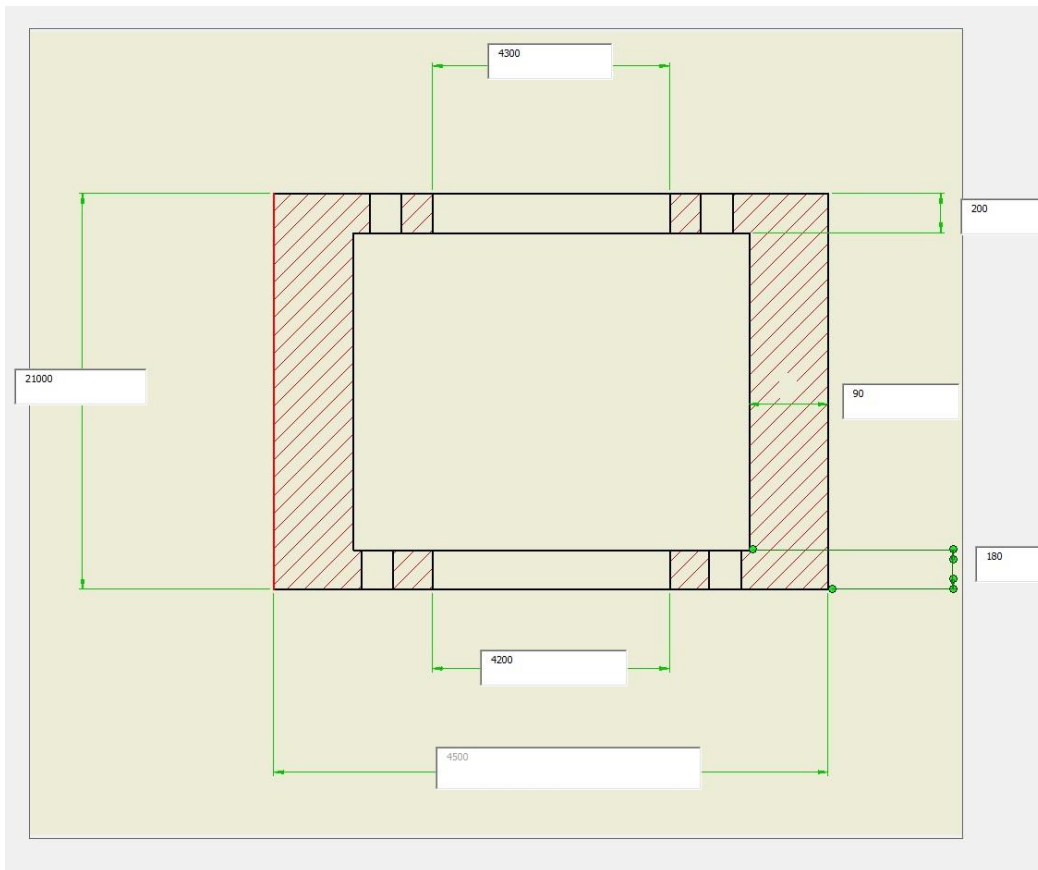
Σχήμα 1.55. Κομμάτι Κορυφής.



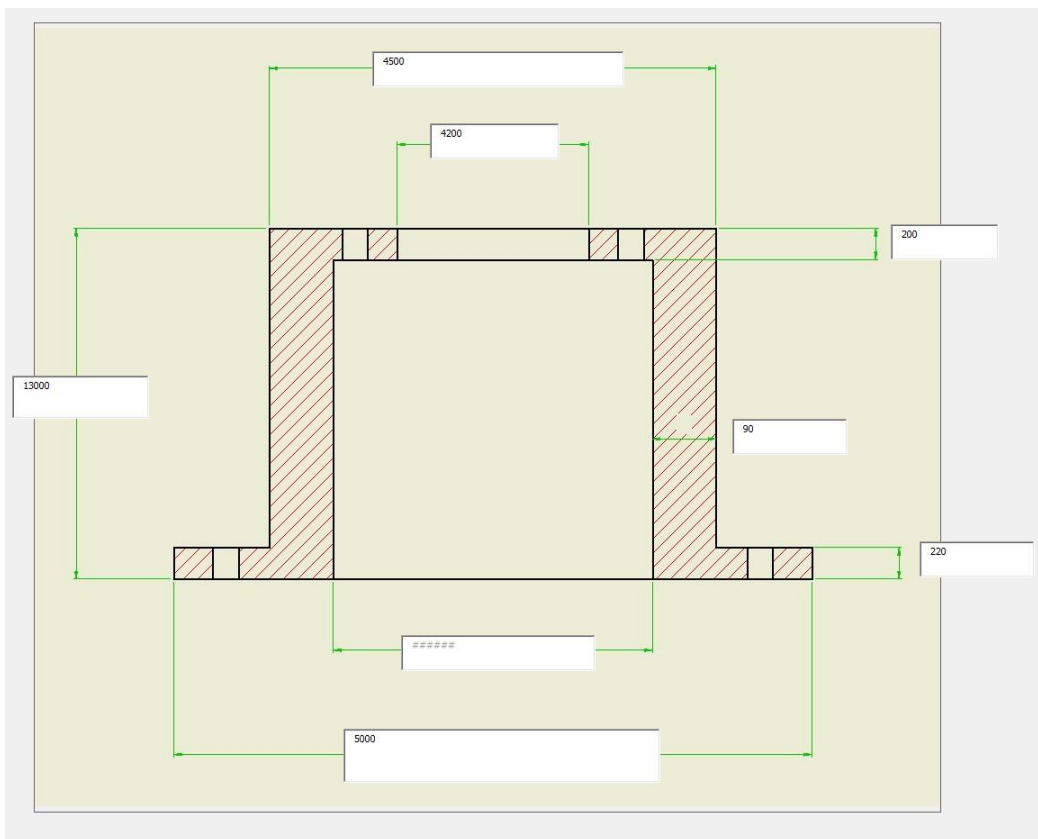
Σχήμα 1.56. Κωνικό κομμάτι.



Σχήμα 1.57. 1° Κυλινδρικό κομμάτι.

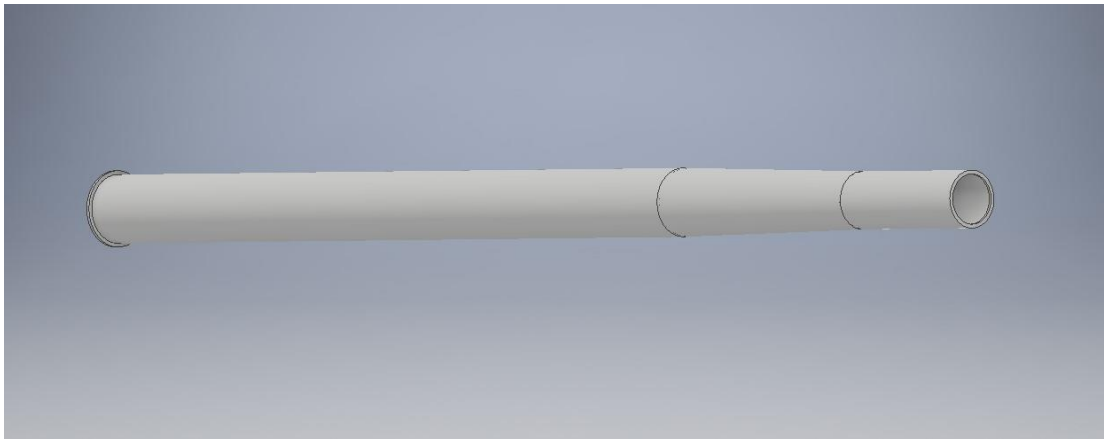


Σχήμα 1.58. 2^ο Κυλινδρικό κομμάτι.



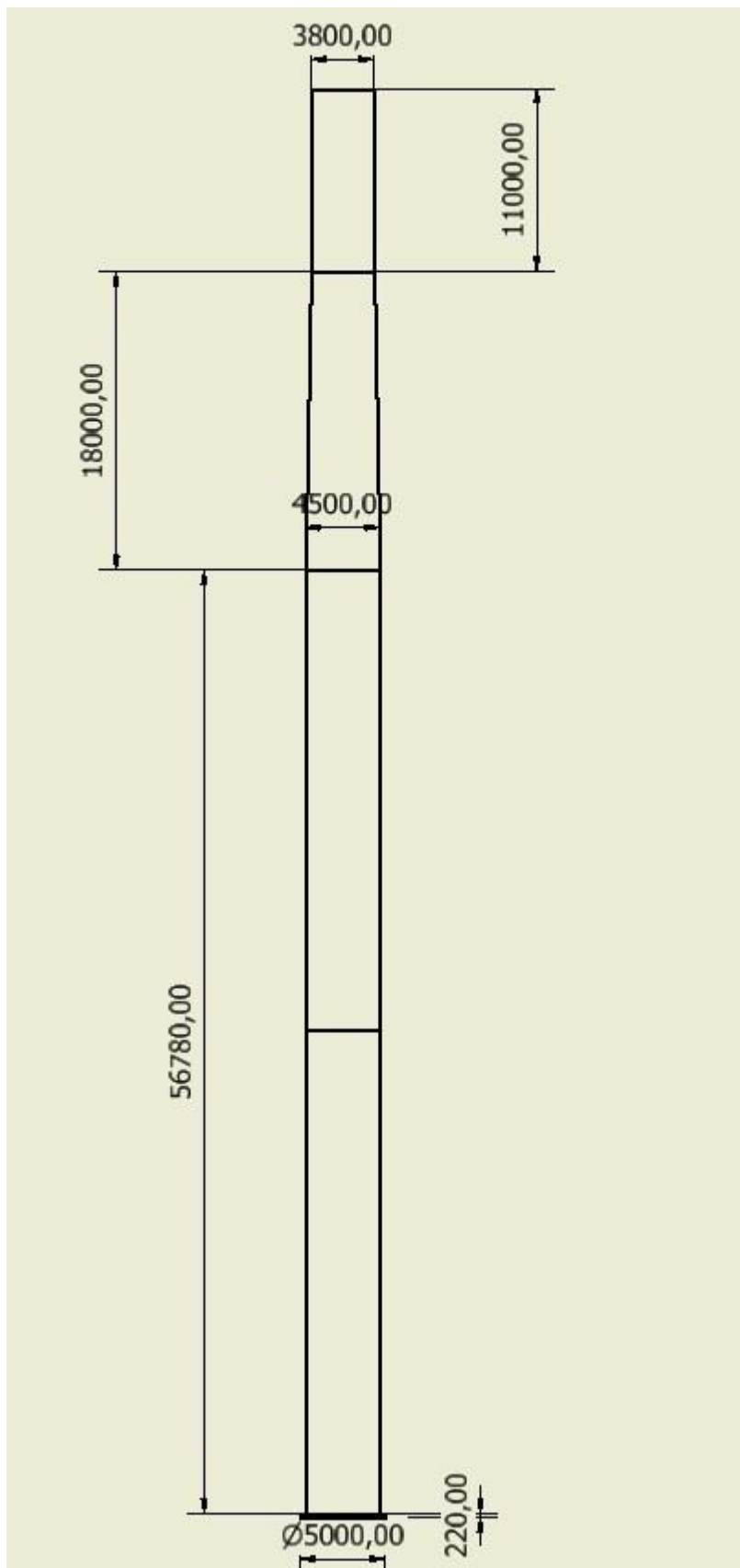
Σχήμα 1.59. Κομμάτι Βάσης.

Ο πυλώνας που δημιουργήσαμε σε 3D φαίνεται στο παρακάτω σχήμα.



Σχήμα 1.60. Ολοκληρωμένη μορφή πύργου.

Σε μηχανολογικό σχέδιο (πρόοψη)



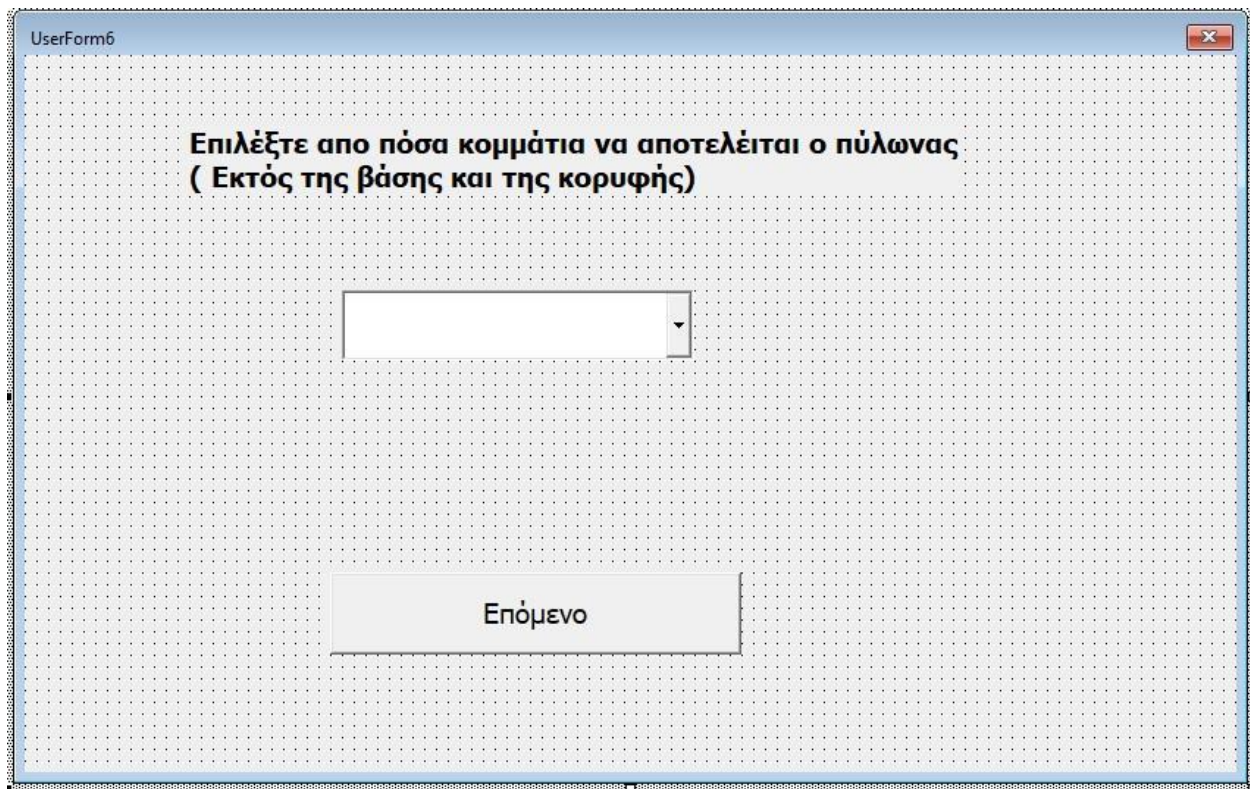
Σχήμα 1.61. Ολοκληρωμένη μορφή πύργου σε μηχανολογικό σχέδιο.

Βιβλιογραφία

1. <https://science.howstuffworks.com/environmental/green-science/wind-power2.htm>
2. <http://users.auth.gr/~kvek/ProgramVBv132.pdf>
3. Danish Wind Industry Association, “Wind Shear”, www.windpower.com
4. Energy Farming Wind GMBH, “Additional yield due to larger hub heights”, https://www.efiwind.de/fileadmin/user_upload/downloads/me_durch_nh_deu.pdf
5. General Electric Renewable Energy, “Space Frame Tower”, <https://www.ge.com/renewableenergy/wind-energy>
6. General Electric Renewable Energy, “The Renewable Energy Era”, https://www.gepowerconversion.com/sites/gepc/files/Whitepaper%20GE_Renewables_Digital_020216_4.pdf
7. <https://vortexbladeless.com/>
8. <https://www.theguardian.com/environment/2017/may/17/mersey-wind-turbines-liverpool-uk-wind-technology>
9. EN 1991-1-4 (2005), Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions, 2005, European Committee for Standardization (CEN): Brussels
10. Παύλος-Ραφαήλ Ντόβας, 2018, “Δικτυωτοί Πύργοι Ανεμογεννητριών, Βιβλιογραφική Έρευνα”, Διπλωματική εργασία, ΑΠΘ
11. Μπεργελές, Γ., “Ανεμοκινητήρες”, Εκδόσεις Συμεών, 2005
12. EN-1993-3-1(2006), Eurocode 3: Design of steel structures - Part 3-1: Towers, masts and chimneys, 2006, European Committee for Standardization (CEN): Brussels
13. Μπίρδας Μιχαήλ, “Βέλτιστος Δομικός Σχεδιασμός Ανεμογεννητριών.”, Διπλωματική Εργασία, ΕΜΠ
14. Baniotopoulos, Charalambos; Borri , Claudio; Stathopoulos, Theodore (2011) – “Environmental Wind Engineering and Design of Wind Energy Structures.”
15. Γεώργιος Θ. Λιβιτσάνος, 2015, “Βελτιστοποίηση σχεδιασμού πυλώνων Ανεμογεννητριών Σύμμικτης Διατομής Χάλυβα- Σκυροδέματος”, Πολυτεχνική Σχολή, Τμήμα Πολιτικών Μηχανικών, Πανεπιστήμιο Πατρών.
16. Καραγιάννη Αθηνά, 2016, “Εναέριες Ανεμογεννήτριες τύπου “Αετός””, Διπλωματική Εργασία, ΣΗΜΜΥ, ΕΜΠ

Παράρτημα

Userform 1



UserForm6

Επιλέξτε απο πόσα κομμάτια να αποτελείται ο πύλωνας
(Εκτός της βάσης και της κορυφής)

Επόμενο

Κώδικας του button

```
Private Sub CommandButton1_Click()  
If (ComboBox1.Text = "") Then  
i = MsgBox("Invalid selection", vbCritical, "Wrong ")  
ComboBox1.Text = ""  
Elseif (ComboBox1.Text = 1) Then  
UserForm7.Show  
Elseif (ComboBox1.Text = 2) Then  
UserForm11.Show  
Elseif (ComboBox1.Text = 3) Then  
UserForm14.Show  
Else  
i = MsgBox("Wrong !", vbCritical, "Wrong ")  
ComboBox1.Text = ""
```


End If

End Sub

Κώδικας του userform

```
Private Sub UserForm_Activate()
```

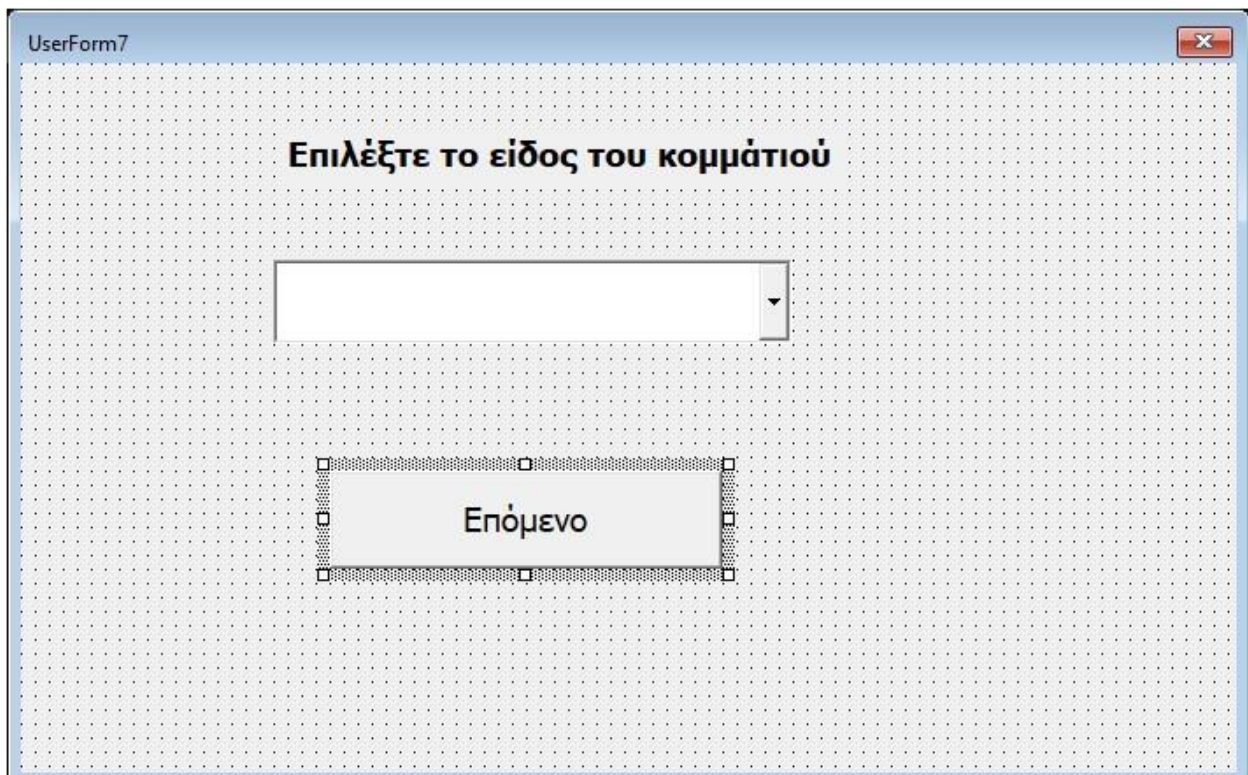
```
Me.ComboBox1.AddItem (1)
```

```
Me.ComboBox1.AddItem (2)
```

```
Me.ComboBox1.AddItem (3)
```

```
End Sub
```

Userform 2



Κώδικας του button

```
Private Sub CommandButton1_Click()
```

```
If (ComboBox1.Text = "") Then
```

```
a = MsgBox("Invalid selection", vbCritical, "Wrong ")
```

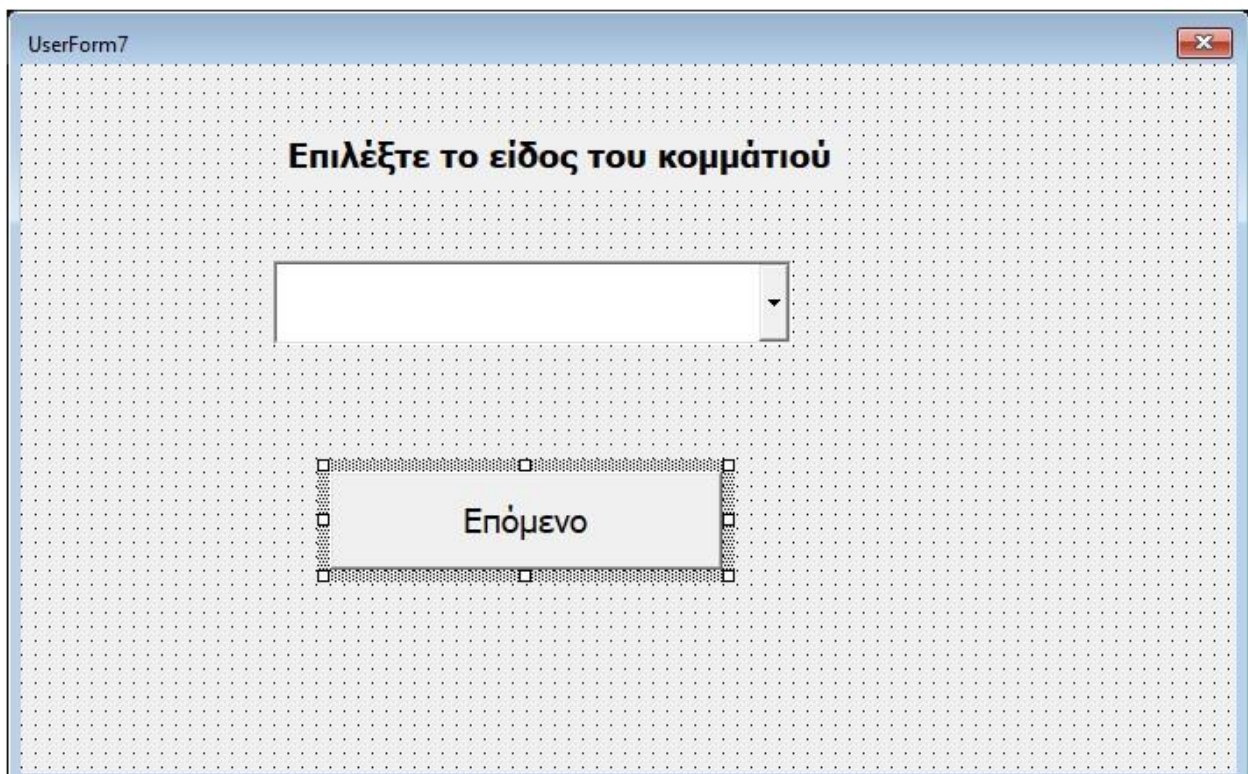
```
Elseif (ComboBox1.Text = "Cylinder") Then
```

```
UserForm9.Show  
Elseif (ComboBox1.Text = "Cone") Then  
UserForm8.Show  
Else  
a = MsgBox("Invalid selection", vbCritical, "Wrong ")  
ComboBox1.Text = ""  
End If  
End Sub
```

Κώδικας του userform 2

```
Private Sub UserForm_Activate()  
Me.ComboBox1.AddItem ("Cylinder")  
Me.ComboBox1.AddItem ("Cone")  
End Sub
```

Userform 3



Κώδικας του button

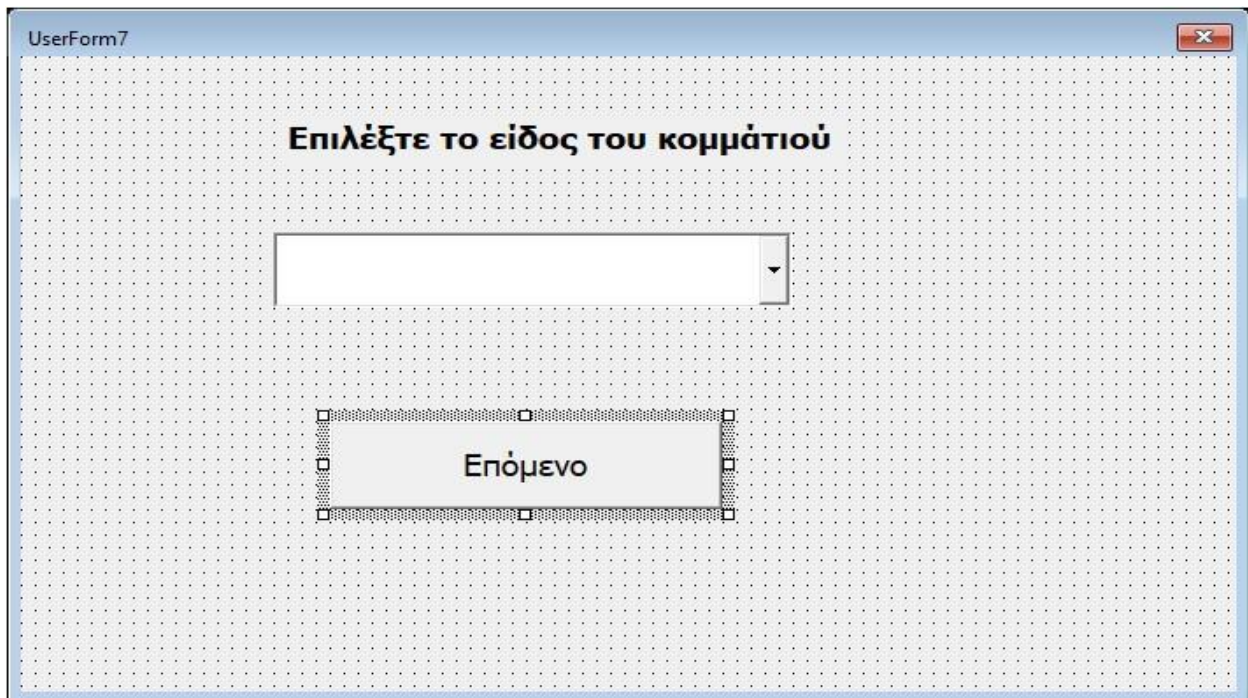
```
Private Sub CommandButton1_Click()
```

```
If (ComboBox1.Text = "") Then
a = MsgBox("Invalid selection", vbCritical, "Wrong ")
Elseif (ComboBox1.Text = "Cylinder,Cylinder") Then
UserForm10.Show
Elseif (ComboBox1.Text = "Cylinder,Cone") Then
UserForm12.Show
Elseif (ComboBox1.Text = "Cone,Cone") Then
UserForm13.Show
Else
a = MsgBox("Invalid selection", vbCritical, "Wrong ")
ComboBox1.Text = ""
End If
End Sub
```

Κώδικας του userform 3

```
Private Sub UserForm_Activate()
Me.ComboBox1.AddItem ("Cylinder,Cylinder")
Me.ComboBox1.AddItem ("Cylinder,Cone")
Me.ComboBox1.AddItem ("Cone,Cone")
End Sub
```

Userform 4



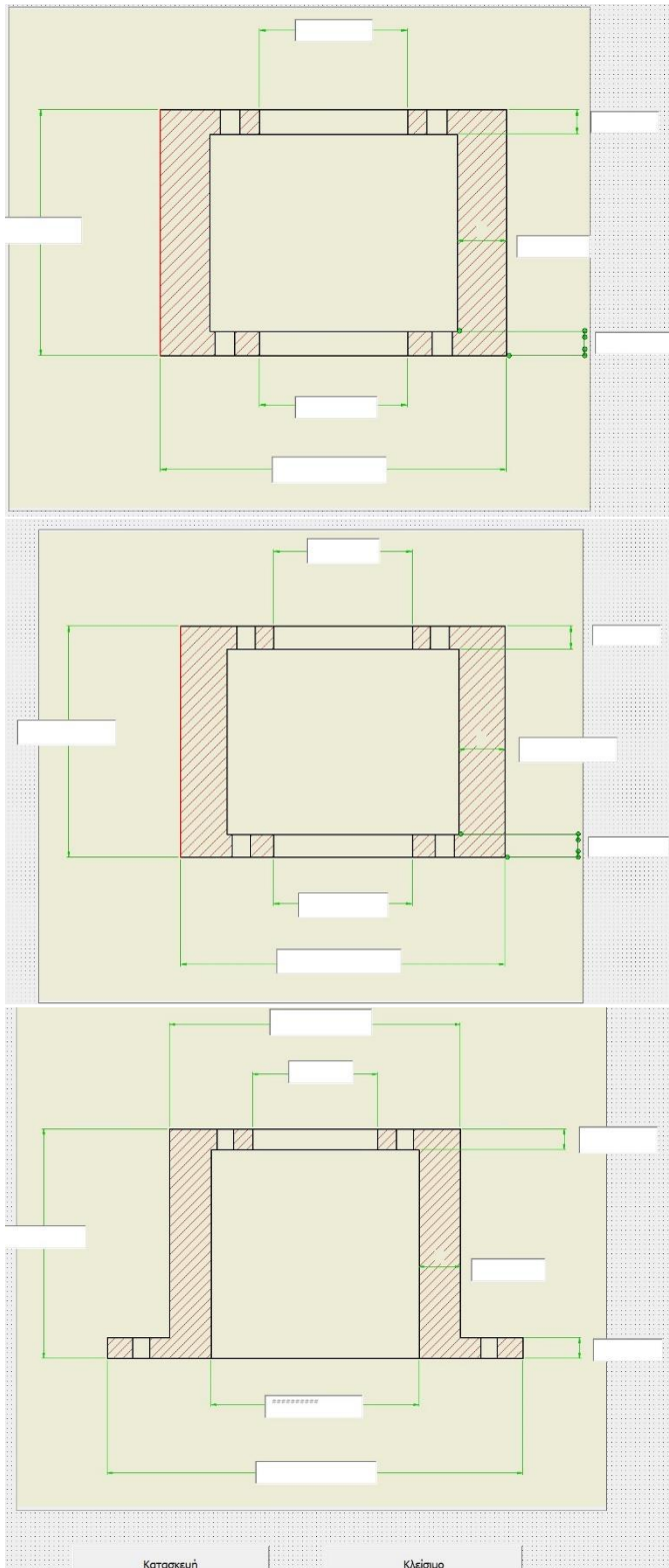
Κώδικας του button

```
Private Sub CommandButton1_Click()  
If (ComboBox1.Text = "") Then  
a = MsgBox("Invalid selection", vbCritical, "Wrong ")  
ElseIf (ComboBox1.Text = "Cylinder,Cylinder,Cylinder") Then  
UserForm15.Show  
ElseIf (ComboBox1.Text = "Cylinder,Cylinder,Cone") Then  
UserForm16.Show  
ElseIf (ComboBox1.Text = "Cylinder,Cone,Cone") Then  
UserForm17.Show  
ElseIf (ComboBox1.Text = "Cone,Cone,Cone") Then  
UserForm18.Show  
Else  
a = MsgBox("Invalid selection", vbCritical, "Wrong ")  
ComboBox1.Text = ""  
End If  
End Sub
```

Κώδικας του userform 4

```
Private Sub UserForm_Activate()  
Me.ComboBox1.AddItem ("Cylinder,Cylinder,Cylinder")  
Me.ComboBox1.AddItem ("Cylinder,Cylinder,Cone")  
Me.ComboBox1.AddItem ("Cylinder,Cone,Cone")  
Me.ComboBox1.AddItem ("Cone,Cone,Cone")  
End Sub
```

Userform 5



Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```

Dim oPartDoc As PartDocument
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,
kMetricSystemOfMeasure, kISO_DraftingStandard))

'Set a reference to the component definition
Dim oCompDef As PartComponentDefinition
Set oCompDef = oPartDoc.ComponentDefinition
Dim oTG As TransientGeometry
Set oTG = ThisApplication.TransientGeometry
'PART A CYLINDER
Dim a As Double
Dim b As Double
Dim c As Double
Dim d As Double
a = Cdbl(TextBox1.Text)
b = Cdbl(TextBox2.Text)
c = Cdbl(TextBox3.Text)
a = a / 10
b = b / 10
c = c / 10
b = b / 2
d = b - c
Dim oSketch1 As PlanarSketch
Set oSketch1 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It
em(2))
Dim oSkPnts1 As SketchPoints
Set oSkPnts1 = oSketch1.SketchPoints
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)

```

```
Dim oCircs1 As SketchCircles
```

```
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = Cdbl(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

```
fc = fc / 2
```

```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```



```
Dim oCircsf1 As SketchCircles
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
Dim oCircsf2 As SketchCircles
Set oCircsf2 = oSketchf1.SketchCircles
```

```
Dim oCircf2 As SketchCircle
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
Dim oProfilef1 As Profile
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```

'part 1 flantza panw

```
Dim fd As Double
Dim fb As Double
fd = Cdbl(TextBox5.Text)
fb = Cdbl(TextBox4.Text)
fd = fd / 10
fb = fb / 10
fd = fd / 2
```

```
Dim oWPlane1 As WorkPlane
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ
Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
Set oSketchf2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It
em(4))
```

```

Dim oSkPntsf2 As SketchPoints
Set oSkPntsf2 = oSketchf2.SketchPoints
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
Dim oCircsf3 As SketchCircles
Set oCircsf3 = oSketchf2.SketchCircles

Dim oCircf3 As SketchCircle
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
Dim oCircsf4 As SketchCircles
Set oCircsf4 = oSketchf2.SketchCircles

Dim oCircf4 As SketchCircle
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
Dim oProfilef2 As Profile
Set oProfilef2 = oSketchf2.Profiles.AddForSolid

Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)

d = ((d * 2) * 10)
TextBox60.Value = d

'part 2
Dim e As Double
Dim f As Double
Dim g As Double
e = Cdbl(TextBox64.Text)
f = Cdbl(TextBox65.Text)
e = e / 10
f = f / 10
g = b - f

```

```
Dim oSketch41 As PlanarSketch
```

```
Set oSketch41 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Ite  
m(4))
```

```
Dim oSkPnts41 As SketchPoints
```

```
Set oSkPnts41 = oSketch41.SketchPoints
```

```
Call oSkPnts41.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs51 As SketchCircles
```

```
Set oCircs51 = oSketch41.SketchCircles
```

```
Dim oCirc51 As SketchCircle
```

```
Set oCirc51 = oCircs51.AddByCenterRadius(oSkPnts41(1), b)
```

```
Dim oCircs61 As SketchCircles
```

```
Set oCircs61 = oSketch41.SketchCircles
```

```
Dim oCirc61 As SketchCircle
```

```
Set oCirc61 = oCircs61.AddByCenterRadius(oSkPnts41(1), g)
```

```
Dim oProfile41 As Profile
```

```
Set oProfile41 = oSketch41.Profiles.AddForSolid
```

```
Dim oExtrusion31 As ExtrudeFeature
```

```
Set oextrude31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile41, e, kPositiveExtentDirection, kJoinOperation)
```

```
'part 2 flantza katw
```

```
Dim fg As Double
```

```
Dim fh As Double
```

```
fg = Cdbl(TextBox63.Text)
```

```
fh = Cdbl(TextBox62.Text)
```

```
fg = fg / 10
```

fh = fh / 10

fh = fh / 2

Dim oSketchf51 As PlanarSketch

Set oSketchf51 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf51 As SketchPoints

Set oSkPntsf51 = oSketchf51.SketchPoints

Call oSkPntsf51.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf91 As SketchCircles

Set oCircsf91 = oSketchf51.SketchCircles

Dim oCircf91 As SketchCircle

Set oCircf91 = oCircsf91.AddByCenterRadius(oSkPntsf51(1), g)

Dim oCircsf101 As SketchCircles

Set oCircsf101 = oSketchf51.SketchCircles

Dim oCircf101 As SketchCircle

Set oCircf101 = oCircsf101.AddByCenterRadius(oSkPntsf51(1), fh)

Dim oProfilef51 As Profile

Set oProfilef51 = oSketchf51.Profiles.AddForSolid

Set oextruf51 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef51, fg, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza panw

Dim fi As Double

Dim fj As Double

Dim fki As Double

fi = Cdbl(TextBox66.Text)

fj = Cdbl(TextBox67.Text)

fi = fi / 10

fj = fj / 10

fki = e + a

fj = fj / 2

Dim oWPlane31 As WorkPlane

Set oWPlane31 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fki)

Dim oSketchf61 As PlanarSketch

Set oSketchf61 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf61 As SketchPoints

Set oSkPntsf61 = oSketchf61.SketchPoints

Call oSkPntsf61.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf111 As SketchCircles

Set oCircsf111 = oSketchf61.SketchCircles

Dim oCircf111 As SketchCircle

Set oCircf111 = oCircsf111.AddByCenterRadius(oSkPntsf61(1), g)

Dim oCircsf121 As SketchCircles

Set oCircsf121 = oSketchf61.SketchCircles

Dim oCircf121 As SketchCircle

Set oCircf121 = oCircsf121.AddByCenterRadius(oSkPntsf61(1), fj)

```
Dim oProfilef61 As Profile
```

```
Set oProfilef61 = oSketchf61.Profiles.AddForSolid
```

```
Set oextrudef61 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
    oProfilef61, fi, kNegativeExtentDirection, kJoinOperation)
```

```
'part 3
```

```
Dim p As Double
```

```
Dim q As Double
```

```
Dim r As Double
```

```
p = Cdbl(TextBox18.Text)
```

```
q = Cdbl(TextBox14.Text)
```

```
p = p / 10
```

```
q = q / 10
```

```
r = b - q
```

```
Dim oSketch4 As PlanarSketch
```

```
Set oSketch4 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
Dim oSkPnts4 As SketchPoints
```

```
Set oSkPnts4 = oSketch4.SketchPoints
```

```
Call oSkPnts4.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs5 As SketchCircles
```

```
Set oCircs5 = oSketch4.SketchCircles
```

```
Dim oCirc5 As SketchCircle
```

```
Set oCirc5 = oCircs5.AddByCenterRadius(oSkPnts4(1), b)
```

```
Dim oCircs6 As SketchCircles
```

```
Set oCircs6 = oSketch4.SketchCircles
```

```
Dim oCirc6 As SketchCircle
```

```
Set oCirc6 = oCircs6.AddByCenterRadius(oSkPnts4(1), r)
```

```
Dim oProfile4 As Profile
```

```
Set oProfile4 = oSketch4.Profiles.AddForSolid
```

```
Dim oExtrusion3 As ExtrudeFeature
```

```
Set oExtrude3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile4, p, kPositiveExtentDirection, kJoinOperation)
```

```
'part 3 flantza katw
```

```
Dim fk As Double
```

```
Dim fl As Double
```

```
fk = Cdbl(TextBox19.Text)
```

```
fl = Cdbl(TextBox20.Text)
```

```
fk = fk / 10
```

```
fl = fl / 10
```

```
fk = fk / 2
```

```
Dim oSketchf5 As PlanarSketch
```

```
Set oSketchf5 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
Dim oSkPntsf5 As SketchPoints
```

```
Set oSkPntsf5 = oSketchf5.SketchPoints
```

```
Call oSkPntsf5.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf9 As SketchCircles
```

```
Set oCircsf9 = oSketchf5.SketchCircles
```

```
Dim oCircf9 As SketchCircle
```

```
Set oCircf9 = oCircsf9.AddByCenterRadius(oSkPntsf5(1), r)
```

```
Dim oCircsf10 As SketchCircles
```

```
Set oCircsf10 = oSketchf5.SketchCircles
```

```
Dim oCircf10 As SketchCircle
```

```
Set oCircf10 = oCircsf10.AddByCenterRadius(oSkPntsf5(1), fk)
```

```
Dim oProfilef5 As Profile
```

```
Set oProfilef5 = oSketchf5.Profiles.AddForSolid
```

```
Set oextrudef5 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef5, fl, kPositiveExtentDirection, kJoinOperation)
```

```
'part 3 flantza panw
```

```
Dim fm As Double
```

```
Dim fn As Double
```

```
Dim fo As Double
```

```
fn = Cdbl(TextBox21.Text)
```

```
fm = Cdbl(TextBox22.Text)
```

```
fm = fm / 10
```

```
fn = fn / 10
```

```
fo = e + a + p
```

```
fn = fn / 2
```

```
Dim oWPlane3 As WorkPlane
```



```

Set oWPlane3 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ
Plane"), fo)

Dim oSketchf6 As PlanarSketch

Set oSketchf6 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It
em(6))

Dim oSkPntsf6 As SketchPoints

Set oSkPntsf6 = oSketchf6.SketchPoints

Call oSkPntsf6.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf11 As SketchCircles

Set oCircsf11 = oSketchf6.SketchCircles

Dim oCircf11 As SketchCircle

Set oCircf11 = oCircsf11.AddByCenterRadius(oSkPntsf6(1), r)

Dim oCircsf12 As SketchCircles

Set oCircsf12 = oSketchf6.SketchCircles

Dim oCircf12 As SketchCircle

Set oCircf12 = oCircsf12.AddByCenterRadius(oSkPntsf6(1), fn)

Dim oProfilef6 As Profile

Set oProfilef6 = oSketchf6.Profiles.AddForSolid

Set oextrudef6 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
oProfilef6, fm, kNegativeExtentDirection, kJoinOperation)

oWPlane1.Visible = False
oWPlane31.Visible = False
oWPlane3.Visible = False

End Sub

Κώδικας του button «Κλείσιμο»

Private Sub CommandButton2_Click()

```

Unload Me

Unload UserForm7

Unload UserForm6

End Sub

Κώδικας του userform 5

```
Private Sub UserForm_Initialize()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 2.4
```

```
End Sub
```

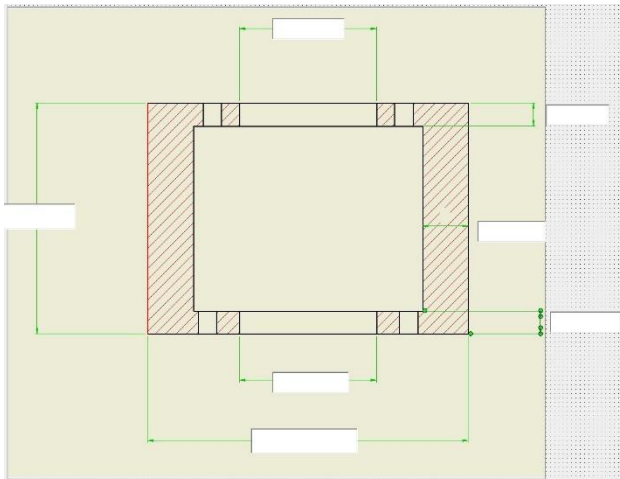
```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

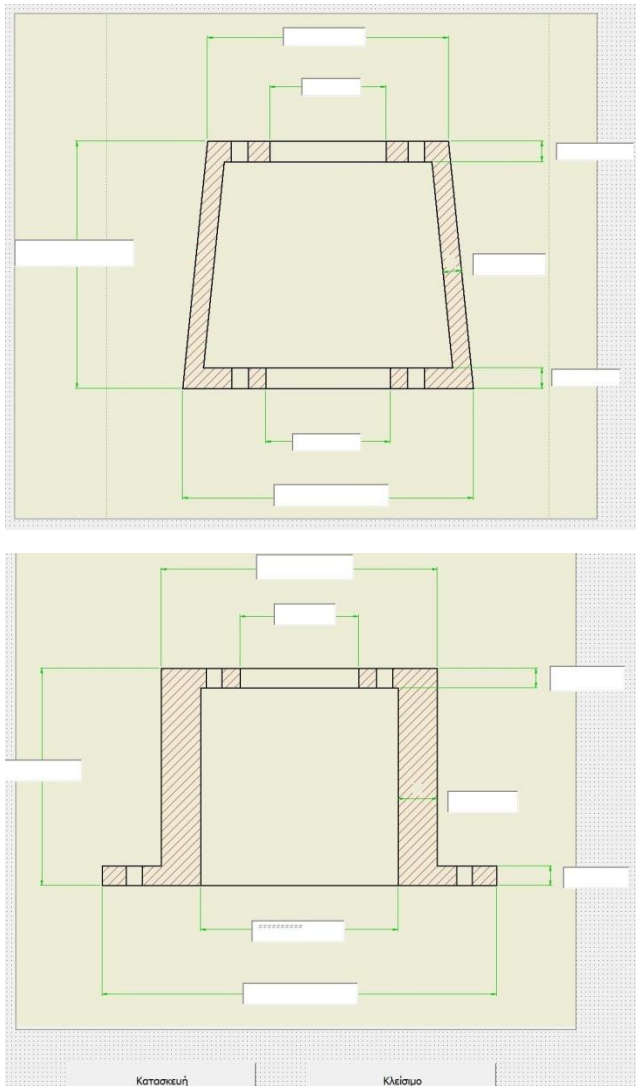
```
TextBox61.Text = TextBox2.Text
```

```
TextBox68.Text = TextBox2.Text
```

```
End Sub
```

Userform 6





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

Dim oTG As TransientGeometry

Set oTG = ThisApplication.TransientGeometry

'PART A CYLINDER

Dim a As Double

Dim b As Double

Dim c As Double

Dim d As Double

a = Cdbl(TextBox1.Text)

b = Cdbl(TextBox2.Text)

c = Cdbl(TextBox3.Text)

a = a / 10

b = b / 10

c = c / 10

b = b / 2

d = b - c

Dim oSketch1 As PlanarSketch

Set oSketch1 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPnts1 As SketchPoints

Set oSkPnts1 = oSketch1.SketchPoints

```
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs1 As SketchCircles
```

```
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = Cdbl(TextBox17.Text)
```

fa = fa / 10

fc = fc / 10

fc = fc / 2

Dim oSketchf1 As PlanarSketch

Set oSketchf1 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPntsf1 As SketchPoints

Set oSkPntsf1 = oSketchf1.SketchPoints

Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf1 As SketchCircles

Set oCircsf1 = oSketchf1.SketchCircles

Dim oCircf1 As SketchCircle

Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)

Dim oCircsf2 As SketchCircles

Set oCircsf2 = oSketchf1.SketchCircles

Dim oCircf2 As SketchCircle

Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)

Dim oProfilef1 As Profile

Set oProfilef1 = oSketchf1.Profiles.AddForSolid

Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_

oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)

'part 1 flantza panw

Dim fd As Double

Dim fb As Double

fd = CDbI(TextBox5.Text)

fb = CDbI(TextBox4.Text)

fd = fd / 10

fb = fb / 10

fd = fd / 2

Dim oWPlane1 As WorkPlane

Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), a)

Dim oSketchf2 As PlanarSketch

Set oSketchf2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf2 As SketchPoints

Set oSkPntsf2 = oSketchf2.SketchPoints

Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf3 As SketchCircles

```
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
```

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = CDbI(TextBox6.Text)
```



```
l = CDbI(TextBox9.Text)
```

```
m = CDbI(TextBox8.Text)
```

```
j = j / 10
```

```
l = l / 10
```

```
m = m / 10
```

```
l = l / 2
```

```
mi = b - m
```

```
n = (b - l) / Sqr((j) ^ 2 + (b - l) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch2 As PlanarSketch
```

```
Set oSketch2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
Dim oSketch3 As PlanarSketch
```

```
Set oSketch3 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts2 As SketchPoints
```

```
Set oSkPnts2 = oSketch2.SketchPoints
```

```
Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oSkPnts3 As SketchPoints
Set oSkPnts3 = oSketch3.SketchPoints
Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircs3 As SketchCircles
Set oCircs3 = oSketch2.SketchCircles
Dim oCirc3 As SketchCircle
Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)
```

```
Dim oCircs4 As SketchCircles
Set oCircs4 = oSketch3.SketchCircles
Dim oCirc4 As SketchCircle
Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)
```

'Create a profile

```
Dim oProfile2 As Profile
Set oProfile2 = oSketch2.Profiles.AddForSolid
Dim oProfile3 As Profile
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

'Create an extrusion

```
Dim oExtrusion2 As ExtrudeFeature
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)

Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)
```

```
b = ((b * 2) * 10)
```

```
TextBox8.Value = b
```

```
'part 2 katw flantza
```

```
Dim fe As Double
```

```
Dim ff As Double
```

```
fe = CDbI(TextBox10.Text)
```

```
ff = CDbI(TextBox11.Text)
```

```
fe = fe / 10
```

```
ff = ff / 10
```

```
fe = fe / 2
```

```
Dim oSketchf3 As PlanarSketch
```

```
Set oSketchf3 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
Dim oSkPntsf3 As SketchPoints
```

```
Set oSkPntsf3 = oSketchf3.SketchPoints
```

```
Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf5 As SketchCircles
```

```
Set oCircsf5 = oSketchf3.SketchCircles
```

```
Dim oCircf5 As SketchCircle
```

```
Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)
```

```
Dim oCircsf6 As SketchCircles
```

```
Set oCircsf6 = oSketchf3.SketchCircles
```

```
Dim oCircf6 As SketchCircle
```

```
Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)
```

```
Dim oProfilef3 As Profile
```

```
Set oProfilef3 = oSketchf3.Profiles.AddForSolid
```

```
Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```

```
'part 2 panw flantza
```

```
Dim fg As Double
```

```
Dim fh As Double
```

```
Dim fi As Double
```

```
Dim fj As Double
```

```
fg = Cdbl(TextBox12.Text)
```

```
fh = Cdbl(TextBox13.Text)
```

```
fg = fg / 10
```

fh = fh / 10

fj = l - m

fg = fg / 2

fi = a + j

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi)

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf4 As SketchPoints

Set oSkPntsf4 = oSketchf4.SketchPoints

Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf7 As SketchCircles

Set oCircsf7 = oSketchf4.SketchCircles

Dim oCircf7 As SketchCircle

Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj)

Dim oCircsf8 As SketchCircles

Set oCircsf8 = oSketchf4.SketchCircles

```
Dim oCircf8 As SketchCircle
```

```
Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg)
```

```
Dim oProfilef4 As Profile
```

```
Set oProfilef4 = oSketchf4.Profiles.AddForSolid
```

```
Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef4, fh, kNegativeExtentDirection, kJoinOperation, n)
```

'part 3

```
Dim p As Double
```

```
Dim q As Double
```

```
Dim r As Double
```

```
p = Cdbl(TextBox18.Text)
```

```
q = Cdbl(TextBox14.Text)
```

```
p = p / 10
```

```
q = q / 10
```

$r = l - q$

Dim oSketch4 As PlanarSketch

Set oSketch4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPnts4 As SketchPoints

Set oSkPnts4 = oSketch4.SketchPoints

Call oSkPnts4.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs5 As SketchCircles

Set oCircs5 = oSketch4.SketchCircles

Dim oCirc5 As SketchCircle

Set oCirc5 = oCircs5.AddByCenterRadius(oSkPnts4(1), l)

Dim oCircs6 As SketchCircles

Set oCircs6 = oSketch4.SketchCircles

Dim oCirc6 As SketchCircle

Set oCirc6 = oCircs6.AddByCenterRadius(oSkPnts4(1), r)

Dim oProfile4 As Profile

Set oProfile4 = oSketch4.Profiles.AddForSolid

Dim oExtrusion3 As ExtrudeFeature

Set oextrude3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile4, p, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza katw

Dim fk As Double

Dim fl As Double

fk = Cdbl(TextBox19.Text)

fl = Cdbl(TextBox20.Text)

fk = fk / 10

fl = fl / 10

fk = fk / 2

Dim oSketchf5 As PlanarSketch

Set oSketchf5 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf5 As SketchPoints

Set oSkPntsf5 = oSketchf5.SketchPoints

Call oSkPntsf5.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf9 As SketchCircles

Set oCircsf9 = oSketchf5.SketchCircles

Dim oCircf9 As SketchCircle

Set oCircf9 = oCircsf9.AddByCenterRadius(oSkPntsf5(1), r)

Dim oCircsf10 As SketchCircles

Set oCircsf10 = oSketchf5.SketchCircles

Dim oCircf10 As SketchCircle

Set oCircf10 = oCircsf10.AddByCenterRadius(oSkPntsf5(1), fk)

Dim oProfilef5 As Profile

Set oProfilef5 = oSketchf5.Profiles.AddForSolid

Set oextrudef5 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef5, fl, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza panw

Dim fm As Double

Dim fn As Double

Dim fo As Double

fn = Cdbl(TextBox21.Text)

fm = Cdbl(TextBox22.Text)

$fm = fm / 10$

$fn = fn / 10$

$fo = fi + p$

$fn = fn / 2$

Dim oWPlane3 As WorkPlane

Set oWPlane3 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo)

Dim oSketchf6 As PlanarSketch

Set oSketchf6 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf6 As SketchPoints

Set oSkPntsf6 = oSketchf6.SketchPoints

Call oSkPntsf6.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf11 As SketchCircles

Set oCircsf11 = oSketchf6.SketchCircles

Dim oCircf11 As SketchCircle

Set oCircf11 = oCircsf11.AddByCenterRadius(oSkPntsf6(1), r)

Dim oCircsf12 As SketchCircles

Set oCircsf12 = oSketchf6.SketchCircles

```
Dim oCircf12 As SketchCircle
```

```
Set oCircf12 = oCircsf12.AddByCenterRadius(oSkPntsf6(1), fn)
```

```
Dim oProfilef6 As Profile
```

```
Set oProfilef6 = oSketchf6.Profiles.AddForSolid
```

```
Set oextrudef6 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef6, fm, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane2.Visible = False
```

```
oWPlane3.Visible = False
```

```
End Sub
```

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm7
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του userform 6

```
Private Sub UserForm_Initialize()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 2.4
```

End Sub

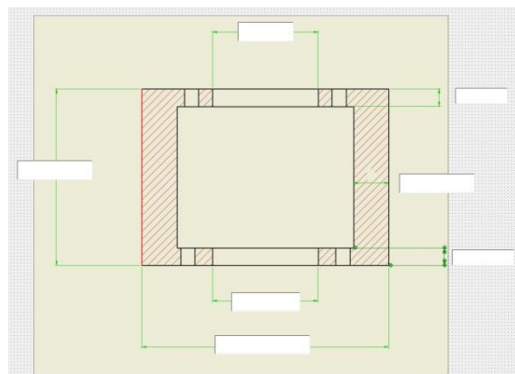
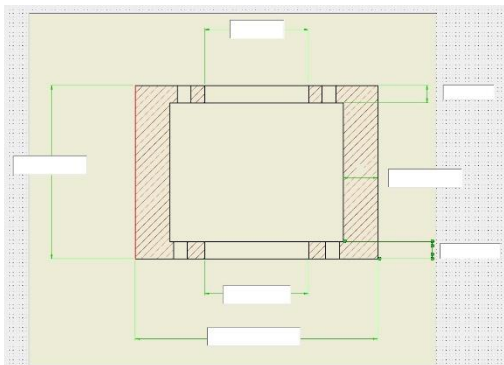
```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

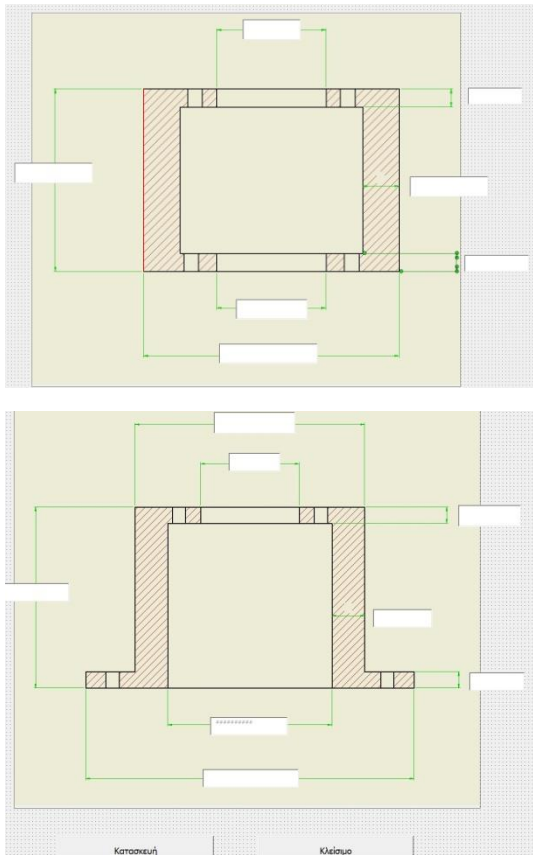
```
    TextBox7.Text = TextBox2.Text
```

```
    TextBox61.Text = TextBox9.Text
```

End Sub

Userform 7





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

'PART A CYLINDER

Dim a As Double

Dim b As Double

Dim c As Double

Dim d As Double

a = Cdbl(TextBox1.Text)

b = Cdbl(TextBox2.Text)

c = Cdbl(TextBox3.Text)

a = a / 10

b = b / 10

c = c / 10

b = b / 2

d = b - c

Dim oSketch1 As PlanarSketch

Set oSketch1 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPnts1 As SketchPoints

Set oSkPnts1 = oSketch1.SketchPoints

Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs1 As SketchCircles

Set oCircs1 = oSketch1.SketchCircles

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = Cdbl(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

```
fc = fc / 2
```

```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf1 As SketchCircles
```

```
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
```

```
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
```

```
Dim oCircsf2 As SketchCircles
```

```
Set oCircsf2 = oSketchf1.SketchCircles
```

```
Dim oCircf2 As SketchCircle
```

```
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
```

```
Dim oProfilef1 As Profile
```

```
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```

```
'part 1 flantza panw
```



```
Dim fd As Double
```

```
Dim fb As Double
```

```
fd = Cdbl(TextBox5.Text)
```

```
fb = Cdbl(TextBox4.Text)
```

```
fd = fd / 10
```

```
fb = fb / 10
```

```
fd = fd / 2
```

```
Dim oWPlane1 As WorkPlane
```

```
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
```

```
Set oSketchf2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf2 As SketchPoints
```

```
Set oSkPntsf2 = oSketchf2.SketchPoints
```

```
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf3 As SketchCircles
```

```
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
```

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim e As Double
```

```
Dim f As Double
```

```
Dim g As Double
```

```
e = Cdbl(TextBox64.Text)
```

```
f = Cdbl(TextBox65.Text)
```

```
e = e / 10
```

```
f = f / 10
```

$g = b - f$

Dim oSketch41 As PlanarSketch

Set oSketch41 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPnts41 As SketchPoints

Set oSkPnts41 = oSketch41.SketchPoints

Call oSkPnts41.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs51 As SketchCircles

Set oCircs51 = oSketch41.SketchCircles

Dim oCirc51 As SketchCircle

Set oCirc51 = oCircs51.AddByCenterRadius(oSkPnts41(1), b)

Dim oCircs61 As SketchCircles

Set oCircs61 = oSketch41.SketchCircles

Dim oCirc61 As SketchCircle

Set oCirc61 = oCircs61.AddByCenterRadius(oSkPnts41(1), g)

Dim oProfile41 As Profile

Set oProfile41 = oSketch41.Profiles.AddForSolid

```
Dim oExtrusion31 As ExtrudeFeature
```

```
Set oextrude31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile41, e, kPositiveExtentDirection, kJoinOperation)
```

```
'part 2 flantza katw
```

```
Dim fg As Double
```

```
Dim fh As Double
```

```
fg = CDbI(TextBox63.Text)
```

```
fh = CDbI(TextBox62.Text)
```

```
fg = fg / 10
```

```
fh = fh / 10
```

```
fh = fh / 2
```

```
Dim oSketchf51 As PlanarSketch
```

```
Set oSketchf51 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf51 As SketchPoints
```

```
Set oSkPntsf51 = oSketchf51.SketchPoints
```

```
Call oSkPntsf51.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

Dim oCircsf91 As SketchCircles

Set oCircsf91 = oSketchf51.SketchCircles

Dim oCircf91 As SketchCircle

Set oCircf91 = oCircsf91.AddByCenterRadius(oSkPntsf51(1), g)

Dim oCircsf101 As SketchCircles

Set oCircsf101 = oSketchf51.SketchCircles

Dim oCircf101 As SketchCircle

Set oCircf101 = oCircsf101.AddByCenterRadius(oSkPntsf51(1), fh)

Dim oProfilef51 As Profile

Set oProfilef51 = oSketchf51.Profiles.AddForSolid

Set oextrudef51 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef51, fg, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza panw

Dim fi As Double

Dim fj As Double

Dim fki As Double

```
fi = Cdbl(TextBox66.Text)
```

```
fj = Cdbl(TextBox67.Text)
```

```
fi = fi / 10
```

```
fj = fj / 10
```

```
fki = e + a
```

```
fj = fj / 2
```

```
Dim oWPlane31 As WorkPlane
```

```
Set oWPlane31 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fki)
```

```
Dim oSketchf61 As PlanarSketch
```

```
Set oSketchf61 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(5))
```

```
Dim oSkPntsf61 As SketchPoints
```

```
Set oSkPntsf61 = oSketchf61.SketchPoints
```

```
Call oSkPntsf61.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf111 As SketchCircles
```

```
Set oCircsf111 = oSketchf61.SketchCircles
```

```
Dim oCircf111 As SketchCircle
```

```
Set oCircf111 = oCircsf111.AddByCenterRadius(oSkPntsf61(1), g)
```

```
Dim oCircsf121 As SketchCircles
```

```
Set oCircsf121 = oSketchf61.SketchCircles
```

```
Dim oCircf121 As SketchCircle
```

```
Set oCircf121 = oCircsf121.AddByCenterRadius(oSkPntsf61(1), fj)
```

```
Dim oProfilef61 As Profile
```

```
Set oProfilef61 = oSketchf61.Profiles.AddForSolid
```

```
Set oextrudef61 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef61, fi, kNegativeExtentDirection, kJoinOperation)
```

'part 3

```
Dim p As Double
```

```
Dim q As Double
```

```
Dim r As Double
```

```
p = Cdbl(TextBox18.Text)
```

```
q = Cdbl(TextBox14.Text)
```

```
p = p / 10
```

```
q = q / 10
```

$r = b - q$

Dim oSketch4 As PlanarSketch

Set oSketch4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPnts4 As SketchPoints

Set oSkPnts4 = oSketch4.SketchPoints

Call oSkPnts4.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs5 As SketchCircles

Set oCircs5 = oSketch4.SketchCircles

Dim oCirc5 As SketchCircle

Set oCirc5 = oCircs5.AddByCenterRadius(oSkPnts4(1), b)

Dim oCircs6 As SketchCircles

Set oCircs6 = oSketch4.SketchCircles

Dim oCirc6 As SketchCircle

Set oCirc6 = oCircs6.AddByCenterRadius(oSkPnts4(1), r)

Dim oProfile4 As Profile

Set oProfile4 = oSketch4.Profiles.AddForSolid

Dim oExtrusion3 As ExtrudeFeature

Set oextrude3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_

oProfile4, p, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza katw

Dim fk As Double

Dim fl As Double

fk = CDbI(TextBox19.Text)

fl = CDbI(TextBox20.Text)

fk = fk / 10

fl = fl / 10

fk = fk / 2

Dim oSketchf5 As PlanarSketch

Set oSketchf5 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf5 As SketchPoints

Set oSkPntsf5 = oSketchf5.SketchPoints

Call oSkPntsf5.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf9 As SketchCircles

Set oCircsf9 = oSketchf5.SketchCircles

Dim oCircf9 As SketchCircle

Set oCircf9 = oCircsf9.AddByCenterRadius(oSkPntsf5(1), r)

Dim oCircsf10 As SketchCircles

Set oCircsf10 = oSketchf5.SketchCircles

Dim oCircf10 As SketchCircle

Set oCircf10 = oCircsf10.AddByCenterRadius(oSkPntsf5(1), fk)

Dim oProfilef5 As Profile

Set oProfilef5 = oSketchf5.Profiles.AddForSolid

Set oextrudef5 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef5, fl, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza panw

Dim fm As Double

Dim fn As Double

Dim fo As Double

fn = Cdbl(TextBox21.Text)

fm = Cdbl(TextBox22.Text)

$fm = fm / 10$

$fn = fn / 10$

$fo = e + a + p$

$fn = fn / 2$

Dim oWPlane3 As WorkPlane

Set oWPlane3 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo)

Dim oSketchf6 As PlanarSketch

Set oSketchf6 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf6 As SketchPoints

Set oSkPntsf6 = oSketchf6.SketchPoints

Call oSkPntsf6.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf11 As SketchCircles

Set oCircsf11 = oSketchf6.SketchCircles

Dim oCircf11 As SketchCircle

Set oCircf11 = oCircsf11.AddByCenterRadius(oSkPntsf6(1), r)

Dim oCircsf12 As SketchCircles

```
Set oCirscf12 = oSketchf6.SketchCircles
```

```
Dim oCircf12 As SketchCircle
```

```
Set oCircf12 = oCirscf12.AddByCenterRadius(oSkPntsf6(1), fn)
```

```
Dim oProfilef6 As Profile
```

```
Set oProfilef6 = oSketchf6.Profiles.AddForSolid
```

```
Set oextrudef6 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef6, fm, kNegativeExtentDirection, kJoinOperation)
```

```
'part 4
```

```
Dim p2 As Double
```

```
Dim q2 As Double
```

```
Dim r2 As Double
```

```
p2 = CDbI(TextBox28.Text)
```

```
q2 = CDbI(TextBox25.Text)
```

```
p2 = p2 / 10
```

```
q2 = q2 / 10
```

```
r2 = b - q2
```

```
Dim oSketch42 As PlanarSketch
```

```
Set oSketch42 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(6))
```

```
Dim oSkPnts42 As SketchPoints
```

```
Set oSkPnts42 = oSketch42.SketchPoints
```

```
Call oSkPnts42.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs52 As SketchCircles
```

```
Set oCircs52 = oSketch42.SketchCircles
```

```
Dim oCirc52 As SketchCircle
```

```
Set oCirc52 = oCircs52.AddByCenterRadius(oSkPnts42(1), b)
```

```
Dim oCircs62 As SketchCircles
```

```
Set oCircs62 = oSketch42.SketchCircles
```

```
Dim oCirc62 As SketchCircle
```

```
Set oCirc62 = oCircs62.AddByCenterRadius(oSkPnts42(1), r2)
```

```
Dim oProfile42 As Profile
```

```
Set oProfile42 = oSketch42.Profiles.AddForSolid
```

```
Dim oExtrusion32 As ExtrudeFeature
```

```
Set oextrude32 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile42, p2, kPositiveExtentDirection, kJoinOperation)
```

```
'part 4 flantza katw
```

```
Dim fk2 As Double
```

```
Dim fl2 As Double
```

```
fk2 = CDbI(TextBox23.Text)
```

```
fl2 = CDbI(TextBox24.Text)
```

```
fk2 = fk2 / 10
```

```
fl2 = fl2 / 10
```

```
fk2 = fk2 / 2
```

```
Dim oSketchf52 As PlanarSketch
```

```
Set oSketchf52 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
Dim oSkPntsf52 As SketchPoints
```

```
Set oSkPntsf52 = oSketchf52.SketchPoints
```

```
Call oSkPntsf52.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf92 As SketchCircles
```

```
Set oCircsf92 = oSketchf52.SketchCircles
```

Dim oCircf92 As SketchCircle

Set oCircf92 = oCircsf92.AddByCenterRadius(oSkPntsf52(1), r2)

Dim oCircsf102 As SketchCircles

Set oCircsf102 = oSketchf52.SketchCircles

Dim oCircf102 As SketchCircle

Set oCircf102 = oCircsf102.AddByCenterRadius(oSkPntsf52(1), fk2)

Dim oProfilef52 As Profile

Set oProfilef52 = oSketchf52.Profiles.AddForSolid

Set oextrudef52 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef52, fl2, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza panw

Dim fm2 As Double

Dim fn2 As Double

Dim fo2 As Double

fn2 = CDbI(TextBox27.Text)

fm2 = CDbI(TextBox26.Text)

fm2 = fm2 / 10

fn2 = fn2 / 10

fo2 = e + a + p + p2

fn2 = fn2 / 2

Dim oWPlane32 As WorkPlane

Set oWPlane32 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo2)

Dim oSketchf62 As PlanarSketch

Set oSketchf62 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf62 As SketchPoints

Set oSkPntsf62 = oSketchf62.SketchPoints

Call oSkPntsf62.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf112 As SketchCircles

Set oCircsf112 = oSketchf62.SketchCircles

Dim oCircf112 As SketchCircle

Set oCircf112 = oCircsf112.AddByCenterRadius(oSkPntsf62(1), r2)

Dim oCircsf122 As SketchCircles

Set oCircsf122 = oSketchf62.SketchCircles

Dim oCircf122 As SketchCircle

Set oCircf122 = oCircsf122.AddByCenterRadius(oSkPntsf62(1), fn2)


```
Dim oProfilef62 As Profile
```

```
Set oProfilef62 = oSketchf62.Profiles.AddForSolid
```

```
Set oextrudef62 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef62, fm2, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane31.Visible = False
```

```
oWPlane3.Visible = False
```

```
oWPlane32.Visible = False
```

```
End Sub
```

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm11
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του userform 7

```
Private Sub UserForm_Initialize()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 3
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As  
MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal  
ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

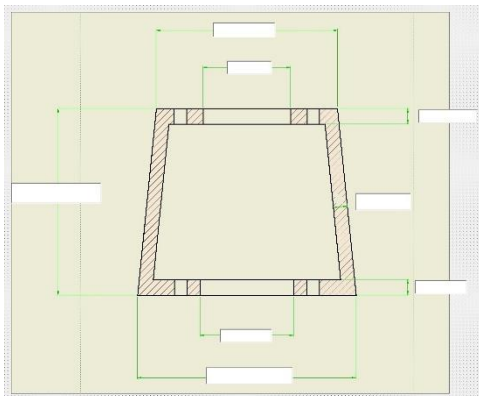
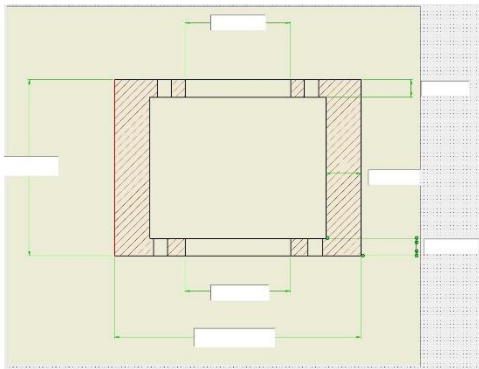
```
TextBox61.Text = TextBox2.Text
```

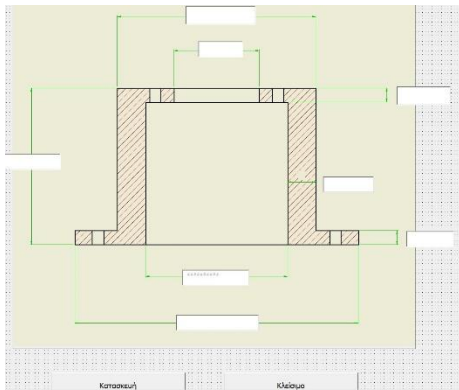
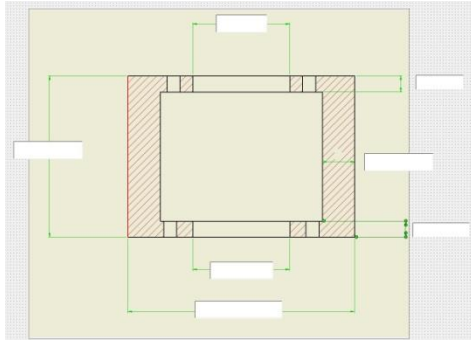
```
TextBox68.Text = TextBox2.Text
```

```
TextBox69.Text = TextBox2.Text
```

```
End Sub
```

Userform 8





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

'PART A CYLINDER

Dim a As Double

Dim b As Double

Dim c As Double

Dim d As Double

a = Cdbl(TextBox1.Text)

b = Cdbl(TextBox2.Text)

c = Cdbl(TextBox3.Text)

a = a / 10

b = b / 10

c = c / 10

b = b / 2

d = b - c

Dim oSketch1 As PlanarSketch

Set oSketch1 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPnts1 As SketchPoints

Set oSkPnts1 = oSketch1.SketchPoints

Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs1 As SketchCircles

Set oCircs1 = oSketch1.SketchCircles

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = Cdbl(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

```
fc = fc / 2
```

```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf1 As SketchCircles
```

```
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
```

```
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
```

```
Dim oCircsf2 As SketchCircles
```

```
Set oCircsf2 = oSketchf1.SketchCircles
```

```
Dim oCircf2 As SketchCircle
```

```
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
```

```
Dim oProfilef1 As Profile
```

```
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```

```
'part 1 flantza panw
```

```
Dim fd As Double
```

```
Dim fb As Double
```

```
fd = Cdbl(TextBox5.Text)
```

```
fb = Cdbl(TextBox4.Text)
```

```
fd = fd / 10
```

```
fb = fb / 10
```

```
fd = fd / 2
```

```
Dim oWPlane1 As WorkPlane
```

```
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
```

```
Set oSketchf2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf2 As SketchPoints
```

```
Set oSkPntsf2 = oSketchf2.SketchPoints
```

```
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf3 As SketchCircles
```

```
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
```

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim e As Double
```

```
Dim f As Double
```

```
Dim g As Double
```

```
e = Cdbl(TextBox64.Text)
```

```
f = Cdbl(TextBox65.Text)
```

```
e = e / 10
```

```
f = f / 10
```


$g = b - f$

Dim oSketch41 As PlanarSketch

Set oSketch41 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPnts41 As SketchPoints

Set oSkPnts41 = oSketch41.SketchPoints

Call oSkPnts41.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs51 As SketchCircles

Set oCircs51 = oSketch41.SketchCircles

Dim oCirc51 As SketchCircle

Set oCirc51 = oCircs51.AddByCenterRadius(oSkPnts41(1), b)

Dim oCircs61 As SketchCircles

Set oCircs61 = oSketch41.SketchCircles

Dim oCirc61 As SketchCircle

Set oCirc61 = oCircs61.AddByCenterRadius(oSkPnts41(1), g)

Dim oProfile41 As Profile

Set oProfile41 = oSketch41.Profiles.AddForSolid

```
Dim oExtrusion31 As ExtrudeFeature
```

```
Set oextrude31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile41, e, kPositiveExtentDirection, kJoinOperation)
```

```
'part 2 flantza katw
```

```
Dim fg As Double
```

```
Dim fh As Double
```

```
fg = CDbI(TextBox63.Text)
```

```
fh = CDbI(TextBox62.Text)
```

```
fg = fg / 10
```

```
fh = fh / 10
```

```
fh = fh / 2
```

```
Dim oSketchf51 As PlanarSketch
```

```
Set oSketchf51 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf51 As SketchPoints
```

```
Set oSkPntsf51 = oSketchf51.SketchPoints
```

```
Call oSkPntsf51.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

Dim oCircsf91 As SketchCircles

Set oCircsf91 = oSketchf51.SketchCircles

Dim oCircf91 As SketchCircle

Set oCircf91 = oCircsf91.AddByCenterRadius(oSkPntsf51(1), g)

Dim oCircsf101 As SketchCircles

Set oCircsf101 = oSketchf51.SketchCircles

Dim oCircf101 As SketchCircle

Set oCircf101 = oCircsf101.AddByCenterRadius(oSkPntsf51(1), fh)

Dim oProfilef51 As Profile

Set oProfilef51 = oSketchf51.Profiles.AddForSolid

Set oextrudef51 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef51, fg, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza panw

Dim fi As Double

Dim fj As Double

Dim fki As Double

```
fi = Cdbl(TextBox66.Text)
```

```
fj = Cdbl(TextBox67.Text)
```

```
fi = fi / 10
```

```
fj = fj / 10
```

```
fki = e + a
```

```
fj = fj / 2
```

```
Dim oWPlane31 As WorkPlane
```

```
Set oWPlane31 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fki)
```

```
Dim oSketchf61 As PlanarSketch
```

```
Set oSketchf61 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(5))
```

```
Dim oSkPntsf61 As SketchPoints
```

```
Set oSkPntsf61 = oSketchf61.SketchPoints
```

```
Call oSkPntsf61.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf111 As SketchCircles
```

```
Set oCircsf111 = oSketchf61.SketchCircles
```

```
Dim oCircf111 As SketchCircle
```

```
Set oCircf111 = oCircsf111.AddByCenterRadius(oSkPntsf61(1), g)
```

```
Dim oCircsf121 As SketchCircles
```

```
Set oCircsf121 = oSketchf61.SketchCircles
```

```
Dim oCircf121 As SketchCircle
```

```
Set oCircf121 = oCircsf121.AddByCenterRadius(oSkPntsf61(1), fj)
```

```
Dim oProfilef61 As Profile
```

```
Set oProfilef61 = oSketchf61.Profiles.AddForSolid
```

```
Set oextrudef61 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef61, fi, kNegativeExtentDirection, kJoinOperation)
```

```
'part 3
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = Cdbl(TextBox6.Text)
```

```
l = Cdbl(TextBox9.Text)
```

```
m = Cdbl(TextBox8.Text)
```

```
j = j / 10
```

```
l = l / 10
```

```
m = m / 10
```

```
l = l / 2
```

```
mi = b - m
```

$$n = (b - l) / \text{Sqr}((j) ^ 2 + (b - l) ^ 2)$$

'create workplane FOR CONY PART 3

Dim oSketch2 As PlanarSketch

Set oSketch2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSketch3 As PlanarSketch

Set oSketch3 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

'Create 3 points...one for the center of the circle and 2 for the line

Dim oSkPnts2 As SketchPoints

Set oSkPnts2 = oSketch2.SketchPoints

Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oSkPnts3 As SketchPoints

Set oSkPnts3 = oSketch3.SketchPoints

Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircs3 As SketchCircles

Set oCircs3 = oSketch2.SketchCircles

Dim oCirc3 As SketchCircle

Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)

```
Dim oCircs4 As SketchCircles
Set oCircs4 = oSketch3.SketchCircles
Dim oCirc4 As SketchCircle
Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)
```

```
'Create a profile
```

```
Dim oProfile2 As Profile
Set oProfile2 = oSketch2.Profiles.AddForSolid
Dim oProfile3 As Profile
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

```
'Create an extrusion
```

```
Dim oExtrusion2 As ExtrudeFeature
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)
```

```
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)
```

```
b = ((b * 2) * 10)
    TextBox8.Value = b
```

```
'part 3 katw flantza
```

```
Dim fe As Double
Dim ff As Double
fe = Cdbl(TextBox10.Text)
ff = Cdbl(TextBox11.Text)
```

fe = fe / 10

ff = ff / 10

fe = fe / 2

Dim oSketchf3 As PlanarSketch

Set oSketchf3 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf3 As SketchPoints

Set oSkPntsf3 = oSketchf3.SketchPoints

Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf5 As SketchCircles

Set oCircsf5 = oSketchf3.SketchCircles

Dim oCircf5 As SketchCircle

Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)

Dim oCircsf6 As SketchCircles

Set oCircsf6 = oSketchf3.SketchCircles

Dim oCircf6 As SketchCircle

Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)

Dim oProfilef3 As Profile


```
Set oProfile3 = oSketchf3.Profiles.AddForSolid
```

```
Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```

```
'part 3 panw flantza
```

```
Dim fg2 As Double
```

```
Dim fh2 As Double
```

```
Dim fi2 As Double
```

```
Dim fj2 As Double
```

```
fg2 = CDb1(TextBox12.Text)
```

```
fh2 = CDb1(TextBox13.Text)
```

```
fg2 = fg2 / 10
```

```
fh2 = fh2 / 10
```

```
fj2 = l - m
```

```
fg2 = fg2 / 2
```

```
fi2 = e + a + j
```

```
Dim oWPlane2 As WorkPlane
```

```
Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fi2)
```

```
Dim oSketchf4 As PlanarSketch
```

```
Set oSketchf4 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(6))
```

```
Dim oSkPntsf4 As SketchPoints
```

```
Set oSkPntsf4 = oSketchf4.SketchPoints
```

```
Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf7 As SketchCircles
```

```
Set oCircsf7 = oSketchf4.SketchCircles
```

```
Dim oCircf7 As SketchCircle
```

```
Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj2)
```

```
Dim oCircsf8 As SketchCircles
```

```
Set oCircsf8 = oSketchf4.SketchCircles
```

```
Dim oCircf8 As SketchCircle
```

```
Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg2)
```

```
Dim oProfilef4 As Profile
```

```
Set oProfilef4 = oSketchf4.Profiles.AddForSolid
```

```
Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef4, fh2, kNegativeExtentDirection, kJoinOperation, n)
```

'part 4

Dim p2 As Double

Dim q2 As Double

Dim r2 As Double

p2 = Cdbl(TextBox28.Text)

q2 = Cdbl(TextBox25.Text)

p2 = p2 / 10

q2 = q2 / 10

r2 = 1 - q2

Dim oSketch42 As PlanarSketch

Set oSketch42 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPnts42 As SketchPoints

Set oSkPnts42 = oSketch42.SketchPoints

Call oSkPnts42.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs52 As SketchCircles

Set oCircs52 = oSketch42.SketchCircles

Dim oCirc52 As SketchCircle

Set oCirc52 = oCircs52.AddByCenterRadius(oSkPnts42(1), l)

Dim oCircs62 As SketchCircles

Set oCircs62 = oSketch42.SketchCircles

Dim oCirc62 As SketchCircle

Set oCirc62 = oCircs62.AddByCenterRadius(oSkPnts42(1), r2)

Dim oProfile42 As Profile

Set oProfile42 = oSketch42.Profiles.AddForSolid

Dim oExtrusion32 As ExtrudeFeature

Set oextrude32 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile42, p2, kPositiveExtentDirection, kJoinOperation)

'part 4 flantza katw

Dim fk2 As Double

Dim fl2 As Double

fk2 = Cdbl(TextBox23.Text)

fl2 = Cdbl(TextBox24.Text)

fk2 = fk2 / 10

fl2 = fl2 / 10

fk2 = fk2 / 2

Dim oSketchf52 As PlanarSketch

Set oSketchf52 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf52 As SketchPoints

Set oSkPntsf52 = oSketchf52.SketchPoints

Call oSkPntsf52.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf92 As SketchCircles

Set oCircsf92 = oSketchf52.SketchCircles

Dim oCircf92 As SketchCircle

Set oCircf92 = oCircsf92.AddByCenterRadius(oSkPntsf52(1), r2)

Dim oCircsf102 As SketchCircles

Set oCircsf102 = oSketchf52.SketchCircles

Dim oCircf102 As SketchCircle

Set oCircf102 = oCircsf102.AddByCenterRadius(oSkPntsf52(1), fk2)

Dim oProfilef52 As Profile

Set oProfilef52 = oSketchf52.Profiles.AddForSolid

```
Set oextrudef52 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef52, fl2, kPositiveExtentDirection, kJoinOperation)
```

'part 4 flantza panw

```
Dim fm2 As Double
```

```
Dim fn2 As Double
```

```
Dim fo2 As Double
```

```
fn2 = Cdbl(TextBox27.Text)
```

```
fm2 = Cdbl(TextBox26.Text)
```

```
fm2 = fm2 / 10
```

```
fn2 = fn2 / 10
```

```
fo2 = e + a + j + p2
```

```
fn2 = fn2 / 2
```

```
Dim oWPlane32 As WorkPlane
```

```
Set oWPlane32 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fo2)
```

```
Dim oSketchf62 As PlanarSketch
```

```
Set oSketchf62 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(7))
```

```
Dim oSkPntsf62 As SketchPoints
Set oSkPntsf62 = oSketchf62.SketchPoints
Call oSkPntsf62.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircsf112 As SketchCircles
Set oCircsf112 = oSketchf62.SketchCircles
Dim oCircf112 As SketchCircle
Set oCircf112 = oCircsf112.AddByCenterRadius(oSkPntsf62(1), r2)
```

```
Dim oCircsf122 As SketchCircles
Set oCircsf122 = oSketchf62.SketchCircles
Dim oCircf122 As SketchCircle
Set oCircf122 = oCircsf122.AddByCenterRadius(oSkPntsf62(1), fn2)
```

```
Dim oProfilef62 As Profile
Set oProfilef62 = oSketchf62.Profiles.AddForSolid
```

```
Set oextrudef62 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef62, fm2, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane31.Visible = False
```

```
oWPlane3.Visible = False
```

```
oWPlane32.Visible = False
```

```
End Sub
```

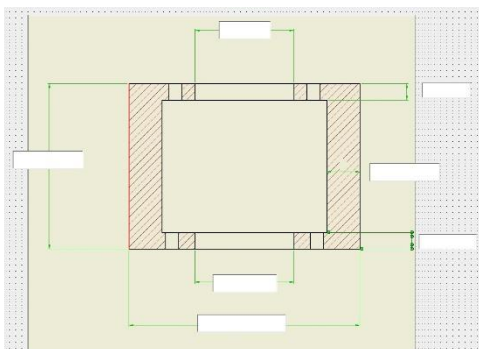
Κώδικας του button «Κλείσιμο»

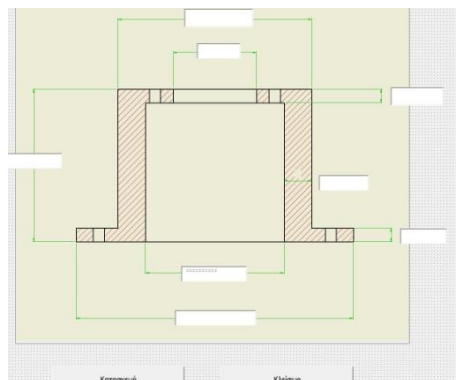
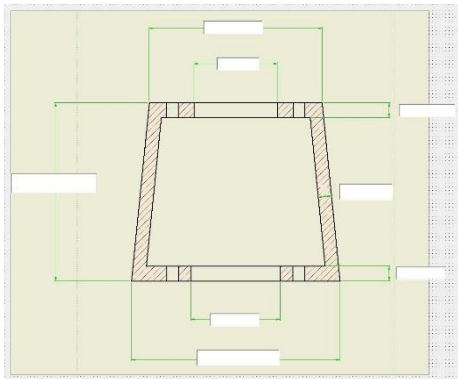
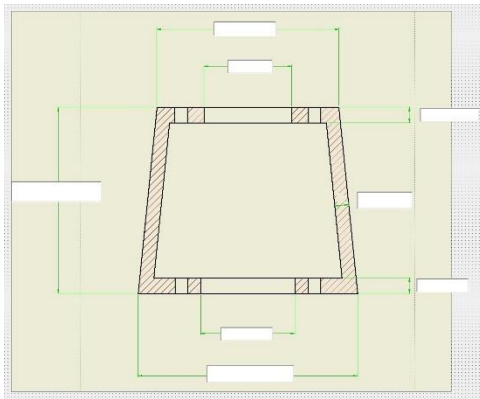
```
Private Sub CommandButton2_Click()  
  
Unload Me  
  
Unload UserForm11  
  
Unload UserForm6  
  
End Sub
```

Κώδικας του Userform 8

```
Private Sub UserForm_Initialize()  
  
Me.ScrollBars = fmScrollBarsVertical  
  
Me.ScrollHeight = Me.InsideHeight * 3  
  
End Sub  
  
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As  
MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal  
ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)  
  
TextBox61.Text = TextBox2.Text  
  
TextBox7.Text = TextBox2.Text  
  
TextBox69.Text = TextBox2.Text  
  
End Sub
```

Userform 9





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
Set oTG = ThisApplication.TransientGeometry
```

```
'PART A CYLINDER
```

```
Dim a As Double
```

```
Dim b As Double
```

```
Dim c As Double
```

```
Dim d As Double
```

```
a = Cdbl(TextBox1.Text)
```

```
b = Cdbl(TextBox2.Text)
```

```
c = Cdbl(TextBox3.Text)
```

```
a = a / 10
```

```
b = b / 10
```

```
c = c / 10
```

```
b = b / 2
```

```
d = b - c
```

```
Dim oSketch1 As PlanarSketch
```

```
Set oSketch1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(2))
```

```
Dim oSkPnts1 As SketchPoints
Set oSkPnts1 = oSketch1.SketchPoints
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs1 As SketchCircles
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

'Part flantza bashs

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = CDb1(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

```
fc = fc / 2
```

```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf1 As SketchCircles
```

```
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
```

```
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
```

```
Dim oCircsf2 As SketchCircles
```

```
Set oCircsf2 = oSketchf1.SketchCircles
```

```
Dim oCircf2 As SketchCircle
```

```
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
```

```
Dim oProfilef1 As Profile
```

```
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```

```
'part 1 flantza panw
```

```
Dim fd As Double
```

```
Dim fb As Double
```

```
fd = Cdbl(TextBox5.Text)
```

```
fb = Cdbl(TextBox4.Text)
```

```
fd = fd / 10
```

```
fb = fb / 10
```

```
fd = fd / 2
```

```
Dim oWPlane1 As WorkPlane
```

```
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
```

```
Set oSketchf2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf2 As SketchPoints
```

```
Set oSkPntsf2 = oSketchf2.SketchPoints
```

```
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf3 As SketchCircles
```

```
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
```

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = CDb1(TextBoX6.Text)
l = CDb1(TextBoX9.Text)
m = CDb1(TextBoX8.Text)
```

```
j = j / 10
l = l / 10
m = m / 10
l = l / 2
mi = b - m
```

```
n = (b - l) / Sqr((j) ^ 2 + (b - l) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch2 As PlanarSketch
```

```
Set oSketch2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
Dim oSketch3 As PlanarSketch
```

```
Set oSketch3 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts2 As SketchPoints
```

```
Set oSkPnts2 = oSketch2.SketchPoints
Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)
Dim oSkPnts3 As SketchPoints
Set oSkPnts3 = oSketch3.SketchPoints
Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircs3 As SketchCircles
Set oCircs3 = oSketch2.SketchCircles
Dim oCirc3 As SketchCircle
Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)
```

```
Dim oCircs4 As SketchCircles
Set oCircs4 = oSketch3.SketchCircles
Dim oCirc4 As SketchCircle
Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)
```

'Create a profile

```
Dim oProfile2 As Profile
Set oProfile2 = oSketch2.Profiles.AddForSolid
Dim oProfile3 As Profile
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

'Create an extrusion

```
Dim oExtrusion2 As ExtrudeFeature
Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)

Set oExtrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
```


oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)

b = ((b * 2) * 10)

TextBox8.Value = b

'part 2 katw flantza

Dim fe As Double

Dim ff As Double

fe = CDbI(TextBox10.Text)

ff = CDbI(TextBox11.Text)

fe = fe / 10

ff = ff / 10

fe = fe / 2

Dim oSketchf3 As PlanarSketch

Set oSketchf3 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf3 As SketchPoints

Set oSkPntsf3 = oSketchf3.SketchPoints

Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

```
Dim oCircsf5 As SketchCircles
Set oCircsf5 = oSketchf3.SketchCircles
Dim oCircf5 As SketchCircle
Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)
```

```
Dim oCircsf6 As SketchCircles
Set oCircsf6 = oSketchf3.SketchCircles
Dim oCircf6 As SketchCircle
Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)
```

```
Dim oProfilef3 As Profile
Set oProfilef3 = oSketchf3.Profiles.AddForSolid
```

```
Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```

'part 2 panw flantza

```
Dim fg2 As Double
Dim fh2 As Double
Dim fi2 As Double
Dim fj2 As Double
fg2 = CDbI(TextBox12.Text)
fh2 = CDbI(TextBox13.Text)
```

$fg2 = fg2 / 10$

$fh2 = fh2 / 10$

$fj2 = l - m$

$fg2 = fg2 / 2$

$fi2 = a + j$

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi2)

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf4 As SketchPoints

Set oSkPntsf4 = oSketchf4.SketchPoints

Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf7 As SketchCircles

Set oCircsf7 = oSketchf4.SketchCircles

Dim oCircf7 As SketchCircle

Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj2)

```
Dim oCircsf8 As SketchCircles
```

```
Set oCircsf8 = oSketchf4.SketchCircles
```

```
Dim oCircf8 As SketchCircle
```

```
Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg2)
```

```
Dim oProfilef4 As Profile
```

```
Set oProfilef4 = oSketchf4.Profiles.AddForSolid
```

```
Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef4, fh2, kNegativeExtentDirection, kJoinOperation, n)
```

```
'part 3
```

```
Dim j1 As Double
```

```
Dim l1 As Double
```

```
Dim m1 As Double
```

```
Dim n1 As Double
```

```
Dim o1 As Double
```

```
j1 = Cdbl(TextBox61.Text)
```

```
l1 = Cdbl(TextBox91.Text)
```

```
m1 = Cdbl(TextBox81.Text)
```

```
j1 = j1 / 10
```

```
l1 = l1 / 10
```

```
m1 = m1 / 10
```

```
l1 = l1 / 2
```

```
mi1 = l - m1
```

$n1 = (l - l1) / \text{Sqr}((j1)^2 + (l - l1)^2)$

'create workplane FOR CONY PART 3

Dim oSketch21 As PlanarSketch

Set oSketch21 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSketch31 As PlanarSketch

Set oSketch31 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

'Create 3 points...one for the center of the circle and 2 for the line

Dim oSkPnts21 As SketchPoints

Set oSkPnts21 = oSketch21.SketchPoints

Call oSkPnts21.Add(oTG.CreatePoint2d(0, 0), False)

Dim oSkPnts31 As SketchPoints

Set oSkPnts31 = oSketch31.SketchPoints

Call oSkPnts31.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircs31 As SketchCircles

Set oCircs31 = oSketch21.SketchCircles

Dim oCirc31 As SketchCircle

Set oCirc31 = oCircs31.AddByCenterRadius(oSkPnts21(1), l)

```
Dim oCircs41 As SketchCircles
Set oCircs41 = oSketch31.SketchCircles
Dim oCirc41 As SketchCircle
Set oCirc41 = oCircs41.AddByCenterRadius(oSkPnts31(1), mi1)
```

'Create a profile

```
Dim oProfile21 As Profile
Set oProfile21 = oSketch21.Profiles.AddForSolid
Dim oProfile31 As Profile
Set oProfile31 = oSketch31.Profiles.AddForSolid
```

'Create an extrusion

```
Dim oExtrusion21 As ExtrudeFeature
Set oExtrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile21, j1, kPositiveExtentDirection, kJoinOperation, -n1)
Set oExtrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile31, j1, kPositiveExtentDirection, kCutOperation, -n1)
```

'part 3 katw flantza

```
Dim fe1 As Double
Dim ff1 As Double
fe1 = CDb1(TextBox101.Text)
ff1 = CDb1(TextBox111.Text)
fe1 = fe1 / 10
```

ff1 = ff1 / 10

fe1 = fe1 / 2

Dim oSketchf31 As PlanarSketch

Set oSketchf31 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf31 As SketchPoints

Set oSkPntsf31 = oSketchf31.SketchPoints

Call oSkPntsf31.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf51 As SketchCircles

Set oCircsf51 = oSketchf31.SketchCircles

Dim oCircf51 As SketchCircle

Set oCircf51 = oCircsf51.AddByCenterRadius(oSkPntsf31(1), mi1)

Dim oCircsf61 As SketchCircles

Set oCircsf61 = oSketchf31.SketchCircles

Dim oCircf61 As SketchCircle

Set oCircf61 = oCircsf61.AddByCenterRadius(oSkPntsf31(1), fe1)

Dim oProfilef31 As Profile

Set oProfilef31 = oSketchf31.Profiles.AddForSolid

```
Set oextrudef31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef31, ff1, kPositiveExtentDirection, kJoinOperation, -n1)
```

'part 3 panw flantza

```
Dim fg1 As Double
```

```
Dim fh1 As Double
```

```
Dim fi1 As Double
```

```
Dim fj1 As Double
```

```
fg1 = Cdbl(TextBox121.Text)
```

```
fh1 = Cdbl(TextBox131.Text)
```

```
fg1 = fg1 / 10
```

```
fh1 = fh1 / 10
```

```
fj1 = l1 - m1
```

```
fg1 = fg1 / 2
```

```
fi1 = a + j + j1
```

```
Dim oWPlane21 As WorkPlane
```

```
Set oWPlane21 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fi1)
```

```
Dim oSketchf41 As PlanarSketch
```



```
Set oSketchf41 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(6))
```

```
Dim oSkPntsf41 As SketchPoints
```

```
Set oSkPntsf41 = oSketchf41.SketchPoints
```

```
Call oSkPntsf41.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf71 As SketchCircles
```

```
Set oCircsf71 = oSketchf41.SketchCircles
```

```
Dim oCircf71 As SketchCircle
```

```
Set oCircf71 = oCircsf71.AddByCenterRadius(oSkPntsf41(1), fj1)
```

```
Dim oCircsf81 As SketchCircles
```

```
Set oCircsf81 = oSketchf41.SketchCircles
```

```
Dim oCircf81 As SketchCircle
```

```
Set oCircf81 = oCircsf81.AddByCenterRadius(oSkPntsf41(1), fg1)
```

```
Dim oProfilef41 As Profile
```

```
Set oProfilef41 = oSketchf41.Profiles.AddForSolid
```

```
Set oextrudef41 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef41, fh1, kNegativeExtentDirection, kJoinOperation, n1)
```

```
'part 4
```

Dim p2 As Double

Dim q2 As Double

Dim r2 As Double

p2 = Cdbl(TextBox28.Text)

q2 = Cdbl(TextBox25.Text)

p2 = p2 / 10

q2 = q2 / 10

r2 = l1 - q2

Dim oSketch42 As PlanarSketch

Set oSketch42 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPnts42 As SketchPoints

Set oSkPnts42 = oSketch42.SketchPoints

Call oSkPnts42.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs52 As SketchCircles

Set oCircs52 = oSketch42.SketchCircles

Dim oCirc52 As SketchCircle

Set oCirc52 = oCircs52.AddByCenterRadius(oSkPnts42(1), l1)

```
Dim oCircs62 As SketchCircles
```

```
Set oCircs62 = oSketch42.SketchCircles
```

```
Dim oCirc62 As SketchCircle
```

```
Set oCirc62 = oCircs62.AddByCenterRadius(oSkPnts42(1), r2)
```

```
Dim oProfile42 As Profile
```

```
Set oProfile42 = oSketch42.Profiles.AddForSolid
```

```
Dim oExtrusion32 As ExtrudeFeature
```

```
Set oExtrude32 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile42, p2, kPositiveExtentDirection, kJoinOperation)
```

```
'part 4 flantza katw
```

```
Dim fk2 As Double
```

```
Dim fl2 As Double
```

```
fk2 = Cdbl(TextBox23.Text)
```

```
fl2 = Cdbl(TextBox24.Text)
```

```
fk2 = fk2 / 10
```

```
fl2 = fl2 / 10
```

fk2 = fk2 / 2

Dim oSketchf52 As PlanarSketch

Set oSketchf52 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf52 As SketchPoints

Set oSkPntsf52 = oSketchf52.SketchPoints

Call oSkPntsf52.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf92 As SketchCircles

Set oCircsf92 = oSketchf52.SketchCircles

Dim oCircf92 As SketchCircle

Set oCircf92 = oCircsf92.AddByCenterRadius(oSkPntsf52(1), r2)

Dim oCircsf102 As SketchCircles

Set oCircsf102 = oSketchf52.SketchCircles

Dim oCircf102 As SketchCircle

Set oCircf102 = oCircsf102.AddByCenterRadius(oSkPntsf52(1), fk2)

Dim oProfilef52 As Profile

Set oProfilef52 = oSketchf52.Profiles.AddForSolid

```
Set oextrudef52 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef52, fl2, kPositiveExtentDirection, kJoinOperation)
```

```
'part 4 flantza panw
```

```
Dim fm2 As Double
```

```
Dim fn2 As Double
```

```
Dim fo2 As Double
```

```
fn2 = Cdbl(TextBox27.Text)
```

```
fm2 = Cdbl(TextBox26.Text)
```

```
fm2 = fm2 / 10
```

```
fn2 = fn2 / 10
```

```
fo2 = a + j + j1 + p2
```

```
fn2 = fn2 / 2
```

```
Dim oWPlane32 As WorkPlane
```

```
Set oWPlane32 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), fo2)
```

```
Dim oSketchf62 As PlanarSketch
```

```
Set oSketchf62 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(7))
```

```
Dim oSkPntsf62 As SketchPoints
Set oSkPntsf62 = oSketchf62.SketchPoints
Call oSkPntsf62.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircsf112 As SketchCircles
Set oCircsf112 = oSketchf62.SketchCircles
Dim oCircf112 As SketchCircle
Set oCircf112 = oCircsf112.AddByCenterRadius(oSkPntsf62(1), r2)
```

```
Dim oCircsf122 As SketchCircles
Set oCircsf122 = oSketchf62.SketchCircles
Dim oCircf122 As SketchCircle
Set oCircf122 = oCircsf122.AddByCenterRadius(oSkPntsf62(1), fn2)
```

```
Dim oProfilef62 As Profile
Set oProfilef62 = oSketchf62.Profiles.AddForSolid
```

```
Set oextrudef62 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef62, fm2, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
oWPlane21.Visible = False
oWPlane2.Visible = False
oWPlane32.Visible = False
```

End Sub

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm11
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του Userform 9

```
Private Sub UserForm_Initialize()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 3
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

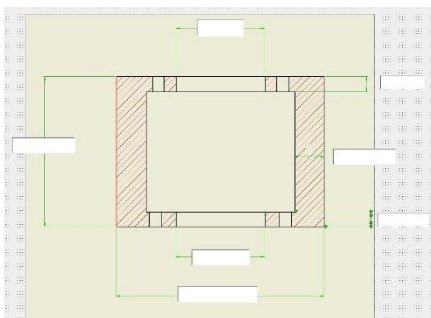
```
TextBox7.Text = TextBox2.Text
```

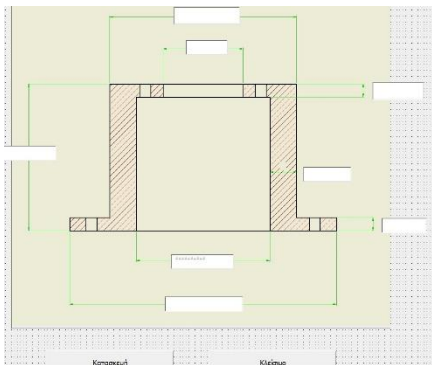
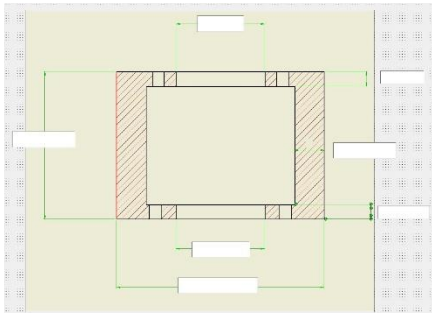
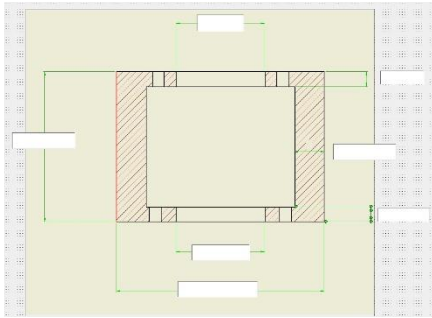
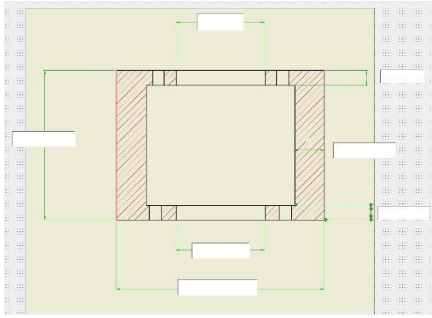
```
TextBox71.Text = TextBox9.Text
```

```
TextBox69.Text = TextBox91.Text
```

```
End Sub
```

Userform 10





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
    kMetricSystemOfMeasure, kISO_DraftingStandard))
```



```
'Set a reference to the component definition  
Dim oCompDef As PartComponentDefinition  
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry  
Set oTG = ThisApplication.TransientGeometry
```

```
'PART A CYLINDER
```

```
Dim a As Double
```

```
Dim b As Double
```

```
Dim c As Double
```

```
Dim d As Double
```

```
a = Cdbl(TextBox1.Text)
```

```
b = Cdbl(TextBox2.Text)
```

```
c = Cdbl(TextBox3.Text)
```

```
a = a / 10
```

```
b = b / 10
```

```
c = c / 10
```

```
b = b / 2
```

```
d = b - c
```

```
Dim oSketch1 As PlanarSketch
```

```
Set oSketch1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(2))
```

```
Dim oSkPnts1 As SketchPoints
```

```
Set oSkPnts1 = oSketch1.SketchPoints
```

```
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs1 As SketchCircles
```

```
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oextrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

Dim fc As Double

fa = CDb1(TextBox15.Text)

fc = CDb1(TextBox17.Text)

fa = fa / 10

fc = fc / 10

fc = fc / 2

Dim oSketchf1 As PlanarSketch

Set oSketchf1 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPntsf1 As SketchPoints

Set oSkPntsf1 = oSketchf1.SketchPoints

Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf1 As SketchCircles

Set oCircsf1 = oSketchf1.SketchCircles

Dim oCircf1 As SketchCircle

Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)

Dim oCircsf2 As SketchCircles

Set oCircsf2 = oSketchf1.SketchCircles

Dim oCircf2 As SketchCircle

Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, fa, kPositiveExtentDirection, kJoinOperation)
```

```
'part 1 flantza panw
```

```
Dim fd As Double
```

```
Dim fb As Double
```

```
fd = Cdbl(TextBox5.Text)
```

```
fb = Cdbl(TextBox4.Text)
```

```
fd = fd / 10
```

```
fb = fb / 10
```

```
fd = fd / 2
```

```
Dim oWPlane1 As WorkPlane
```

```
Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ  
Plane"), a)
```

```
Dim oSketchf2 As PlanarSketch
```

```
Set oSketchf2 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf2 As SketchPoints
Set oSkPntsf2 = oSketchf2.SketchPoints
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf3 As SketchCircles
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
TextBox60.Value = d
```

'part 2

Dim e As Double

Dim f As Double

Dim g As Double

e = Cdbl(TextBox64.Text)

f = Cdbl(TextBox65.Text)

e = e / 10

f = f / 10

g = b - f

Dim oSketch41 As PlanarSketch

Set oSketch41 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPnts41 As SketchPoints

Set oSkPnts41 = oSketch41.SketchPoints

Call oSkPnts41.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs51 As SketchCircles

Set oCircs51 = oSketch41.SketchCircles

Dim oCirc51 As SketchCircle

Set oCirc51 = oCircs51.AddByCenterRadius(oSkPnts41(1), b)

```
Dim oCircs61 As SketchCircles
Set oCircs61 = oSketch41.SketchCircles
```

```
Dim oCirc61 As SketchCircle
Set oCirc61 = oCircs61.AddByCenterRadius(oSkPnts41(1), g)
```

```
Dim oProfile41 As Profile
Set oProfile41 = oSketch41.Profiles.AddForSolid
```

```
Dim oExtrusion31 As ExtrudeFeature
Set oExtrude31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile41, e, kPositiveExtentDirection, kJoinOperation)
```

'part 2 flantza katw

```
Dim fg As Double
Dim fh As Double

fg = Cdbl(TextBox63.Text)
fh = Cdbl(TextBox62.Text)
```

```
fg = fg / 10
```

```
fh = fh / 10
```

```
fh = fh / 2
```

```
Dim oSketchf51 As PlanarSketch
```

```
Set oSketchf51 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf51 As SketchPoints
```

```
Set oSkPntsf51 = oSketchf51.SketchPoints
```

```
Call oSkPntsf51.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf91 As SketchCircles
```

```
Set oCircsf91 = oSketchf51.SketchCircles
```

```
Dim oCircf91 As SketchCircle
```

```
Set oCircf91 = oCircsf91.AddByCenterRadius(oSkPntsf51(1), g)
```

```
Dim oCircsf101 As SketchCircles
```

```
Set oCircsf101 = oSketchf51.SketchCircles
```

```
Dim oCircf101 As SketchCircle
```

```
Set oCircf101 = oCircsf101.AddByCenterRadius(oSkPntsf51(1), fh)
```

```
Dim oProfilef51 As Profile
```

```
Set oProfilef51 = oSketchf51.Profiles.AddForSolid
```

```
Set oextrudef51 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef51, fg, kPositiveExtentDirection, kJoinOperation)
```


'part 2 flantza panw

Dim fi As Double

Dim fj As Double

Dim fki As Double

fi = Cdbl(TextBox66.Text)

fj = Cdbl(TextBox67.Text)

fi = fi / 10

fj = fj / 10

fki = e + a

fj = fj / 2

Dim oWPlane31 As WorkPlane

Set oWPlane31 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fki)

Dim oSketchf61 As PlanarSketch

Set oSketchf61 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf61 As SketchPoints

```
Set oSkPntsf61 = oSketchf61.SketchPoints
Call oSkPntsf61.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf111 As SketchCircles
Set oCircsf111 = oSketchf61.SketchCircles
Dim oCircf111 As SketchCircle
Set oCircf111 = oCircsf111.AddByCenterRadius(oSkPntsf61(1), g)
```

```
Dim oCircsf121 As SketchCircles
Set oCircsf121 = oSketchf61.SketchCircles
Dim oCircf121 As SketchCircle
Set oCircf121 = oCircsf121.AddByCenterRadius(oSkPntsf61(1), fj)
```

```
Dim oProfilef61 As Profile
Set oProfilef61 = oSketchf61.Profiles.AddForSolid
```

```
Set oextrudef61 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef61, fi, kNegativeExtentDirection, kJoinOperation)
```

```
'part 3
```

```
Dim p As Double
Dim q As Double
Dim r As Double
```

p = Cdbl(TextBox18.Text)

q = Cdbl(TextBox14.Text)

p = p / 10

q = q / 10

r = b - q

Dim oSketch4 As PlanarSketch

Set oSketch4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPnts4 As SketchPoints

Set oSkPnts4 = oSketch4.SketchPoints

Call oSkPnts4.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs5 As SketchCircles

Set oCircs5 = oSketch4.SketchCircles

Dim oCirc5 As SketchCircle

Set oCirc5 = oCircs5.AddByCenterRadius(oSkPnts4(1), b)

Dim oCircs6 As SketchCircles

Set oCircs6 = oSketch4.SketchCircles

Dim oCirc6 As SketchCircle

Set oCirc6 = oCircs6.AddByCenterRadius(oSkPnts4(1), r)

Dim oProfile4 As Profile

Set oProfile4 = oSketch4.Profiles.AddForSolid

Dim oExtrusion3 As ExtrudeFeature

Set oextrude3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile4, p, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza katw

Dim fk As Double

Dim fl As Double

fk = CDb1(TextBox19.Text)

fl = CDb1(TextBox20.Text)

fk = fk / 10

fl = fl / 10

fk = fk / 2

Dim oSketchf5 As PlanarSketch

```
Set oSketchf5 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
Dim oSkPntsf5 As SketchPoints  
Set oSkPntsf5 = oSketchf5.SketchPoints  
Call oSkPntsf5.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle  
Dim oCircsf9 As SketchCircles  
Set oCircsf9 = oSketchf5.SketchCircles  
Dim oCircf9 As SketchCircle  
Set oCircf9 = oCircsf9.AddByCenterRadius(oSkPntsf5(1), r)
```

```
Dim oCircsf10 As SketchCircles  
Set oCircsf10 = oSketchf5.SketchCircles  
Dim oCircf10 As SketchCircle  
Set oCircf10 = oCircsf10.AddByCenterRadius(oSkPntsf5(1), fk)
```

```
Dim oProfilef5 As Profile  
Set oProfilef5 = oSketchf5.Profiles.AddForSolid
```

```
Set oextrudef5 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef5, fl, kPositiveExtentDirection, kJoinOperation)
```

'part 3 flantza panw

Dim fm As Double

Dim fn As Double

Dim fo As Double

fn = Cdbl(TextBox21.Text)

fm = Cdbl(TextBox22.Text)

fm = fm / 10

fn = fn / 10

fo = e + a + p

fn = fn / 2

Dim oWPlane3 As WorkPlane

Set oWPlane3 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo)

Dim oSketchf6 As PlanarSketch

Set oSketchf6 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf6 As SketchPoints

Set oSkPntsf6 = oSketchf6.SketchPoints

Call oSkPntsf6.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf11 As SketchCircles

Set oCircsf11 = oSketchf6.SketchCircles

Dim oCircf11 As SketchCircle

Set oCircf11 = oCircsf11.AddByCenterRadius(oSkPntsf6(1), r)

Dim oCircsf12 As SketchCircles

Set oCircsf12 = oSketchf6.SketchCircles

Dim oCircf12 As SketchCircle

Set oCircf12 = oCircsf12.AddByCenterRadius(oSkPntsf6(1), fn)

Dim oProfilef6 As Profile

Set oProfilef6 = oSketchf6.Profiles.AddForSolid

Set oextrudef6 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef6, fm, kNegativeExtentDirection, kJoinOperation)

'part 4

Dim p2 As Double

Dim q2 As Double

Dim r2 As Double

p2 = Cdbl(TextBox28.Text)

q2 = Cdbl(TextBox25.Text)

p2 = p2 / 10

$q2 = q2 / 10$

$r2 = b - q2$

Dim oSketch42 As PlanarSketch

Set oSketch42 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPnts42 As SketchPoints

Set oSkPnts42 = oSketch42.SketchPoints

Call oSkPnts42.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs52 As SketchCircles

Set oCircs52 = oSketch42.SketchCircles

Dim oCirc52 As SketchCircle

Set oCirc52 = oCircs52.AddByCenterRadius(oSkPnts42(1), b)

Dim oCircs62 As SketchCircles

Set oCircs62 = oSketch42.SketchCircles

Dim oCirc62 As SketchCircle

Set oCirc62 = oCircs62.AddByCenterRadius(oSkPnts42(1), r2)

Dim oProfile42 As Profile

Set oProfile42 = oSketch42.Profiles.AddForSolid

Dim oExtrusion32 As ExtrudeFeature

Set oextrude32 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile42, p2, kPositiveExtentDirection, kJoinOperation)

'part 4 flantza katw

Dim fk2 As Double

Dim fl2 As Double

fk2 = Cdbl(TextBox23.Text)

fl2 = Cdbl(TextBox24.Text)

fk2 = fk2 / 10

fl2 = fl2 / 10

fk2 = fk2 / 2

Dim oSketchf52 As PlanarSketch

Set oSketchf52 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf52 As SketchPoints

```
Set oSkPntsf52 = oSketchf52.SketchPoints
Call oSkPntsf52.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf92 As SketchCircles
```

```
Set oCircsf92 = oSketchf52.SketchCircles
```

```
Dim oCircf92 As SketchCircle
```

```
Set oCircf92 = oCircsf92.AddByCenterRadius(oSkPntsf52(1), r2)
```

```
Dim oCircsf102 As SketchCircles
```

```
Set oCircsf102 = oSketchf52.SketchCircles
```

```
Dim oCircf102 As SketchCircle
```

```
Set oCircf102 = oCircsf102.AddByCenterRadius(oSkPntsf52(1), fk2)
```

```
Dim oProfilef52 As Profile
```

```
Set oProfilef52 = oSketchf52.Profiles.AddForSolid
```

```
Set oextrudef52 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef52, fl2, kPositiveExtentDirection, kJoinOperation)
```

```
'part 4 flantza panw
```

```
Dim fm2 As Double
```

```
Dim fn2 As Double
```

Dim fo2 As Double

fn2 = Cdbl(TextBox27.Text)

fm2 = Cdbl(TextBox26.Text)

fm2 = fm2 / 10

fn2 = fn2 / 10

fo2 = e + a + p + p2

fn2 = fn2 / 2

Dim oWPlane32 As WorkPlane

Set oWPlane32 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo2)

Dim oSketchf62 As PlanarSketch

Set oSketchf62 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf62 As SketchPoints

Set oSkPntsf62 = oSketchf62.SketchPoints

Call oSkPntsf62.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf112 As SketchCircles

Set oCircsf112 = oSketchf62.SketchCircles

Dim oCircf112 As SketchCircle

```
Set oCircf112 = oCircsf112.AddByCenterRadius(oSkPntsf62(1), r2)
```

```
Dim oCircsf122 As SketchCircles
```

```
Set oCircsf122 = oSketchf62.SketchCircles
```

```
Dim oCircf122 As SketchCircle
```

```
Set oCircf122 = oCircsf122.AddByCenterRadius(oSkPntsf62(1), fn2)
```

```
Dim oProfilef62 As Profile
```

```
Set oProfilef62 = oSketchf62.Profiles.AddForSolid
```

```
Set oextrudef62 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef62, fm2, kNegativeExtentDirection, kJoinOperation)
```

'part 5

```
Dim p21 As Double
```

```
Dim q21 As Double
```

```
Dim r21 As Double
```

```
p21 = CDb1(TextBox281.Text)
```

```
q21 = CDb1(TextBox251.Text)
```

```
p21 = p21 / 10
```

```
q21 = q21 / 10
```

r21 = b - q21

Dim oSketch421 As PlanarSketch

Set oSketch421 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPnts421 As SketchPoints

Set oSkPnts421 = oSketch421.SketchPoints

Call oSkPnts421.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs521 As SketchCircles

Set oCircs521 = oSketch421.SketchCircles

Dim oCirc521 As SketchCircle

Set oCirc521 = oCircs521.AddByCenterRadius(oSkPnts421(1), b)

Dim oCircs621 As SketchCircles

Set oCircs621 = oSketch421.SketchCircles

Dim oCirc621 As SketchCircle

Set oCirc621 = oCircs621.AddByCenterRadius(oSkPnts421(1), r21)

Dim oProfile421 As Profile

Set oProfile421 = oSketch421.Profiles.AddForSolid

Dim oExtrusion321 As ExtrudeFeature

Set oextrude321 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_

oProfile421, p21, kPositiveExtentDirection, kJoinOperation)

'part 5 flantza katw

Dim fk21 As Double

Dim fl21 As Double

fk21 = Cdbl(TextBox231.Text)

fl21 = Cdbl(TextBox241.Text)

fk21 = fk21 / 10

fl21 = fl21 / 10

fk21 = fk21 / 2

Dim oSketchf521 As PlanarSketch

Set oSketchf521 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf521 As SketchPoints

Set oSkPntsf521 = oSketchf521.SketchPoints

Call oSkPntsf521.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf921 As SketchCircles

Set oCircsf921 = oSketchf521.SketchCircles

Dim oCircf921 As SketchCircle

Set oCircf921 = oCircsf921.AddByCenterRadius(oSkPntsf521(1), r21)

Dim oCircsf1021 As SketchCircles

Set oCircsf1021 = oSketchf521.SketchCircles

Dim oCircf1021 As SketchCircle

Set oCircf1021 = oCircsf1021.AddByCenterRadius(oSkPntsf521(1), fk21)

Dim oProfilef521 As Profile

Set oProfilef521 = oSketchf521.Profiles.AddForSolid

Set oextrudef521 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef521, fl21, kPositiveExtentDirection, kJoinOperation)

'part 5 flantza panw

Dim fm21 As Double

Dim fn21 As Double

Dim fo21 As Double

fn21 = Cdbl(TextBox271.Text)

fm21 = Cdbl(TextBox261.Text)

fm21 = fm21 / 10

fn21 = fn21 / 10

fo21 = e + a + p + p2 + p21

fn21 = fn21 / 2

Dim oWPlane321 As WorkPlane

Set oWPlane321 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo21)

Dim oSketchf621 As PlanarSketch

Set oSketchf621 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(8))

Dim oSkPntsf621 As SketchPoints

Set oSkPntsf621 = oSketchf621.SketchPoints

Call oSkPntsf621.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf1121 As SketchCircles

Set oCircsf1121 = oSketchf621.SketchCircles

Dim oCircf1121 As SketchCircle

Set oCircf1121 = oCircsf1121.AddByCenterRadius(oSkPntsf621(1), r21)

Dim oCircsf1221 As SketchCircles


```

Set oCircsf1221 = oSketchf621.SketchCircles
Dim oCircf1221 As SketchCircle
Set oCircf1221 = oCircsf1221.AddByCenterRadius(oSkPntsf621(1), fn21)

Dim oProfilef621 As Profile
Set oProfilef621 = oSketchf621.Profiles.AddForSolid

Set oextrudef621 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef621, fm21, kNegativeExtentDirection, kJoinOperation)

oWPlane1.Visible = False
oWPlane31.Visible = False
oWPlane3.Visible = False
oWPlane32.Visible = False
oWPlane321.Visible = False
End Sub

```

Κώδικας του button «Κλείσιμο»

```

Private Sub CommandButton2_Click()
Unload Me
Unload UserForm14
Unload UserForm6

End Sub

```

Κωδικας του Userform 10

```

Private Sub UserForm_Initialize()

```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 4.2
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

```
TextBox61.Text = TextBox2.Text
```

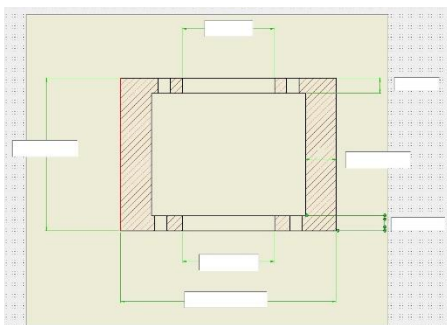
```
TextBox68.Text = TextBox2.Text
```

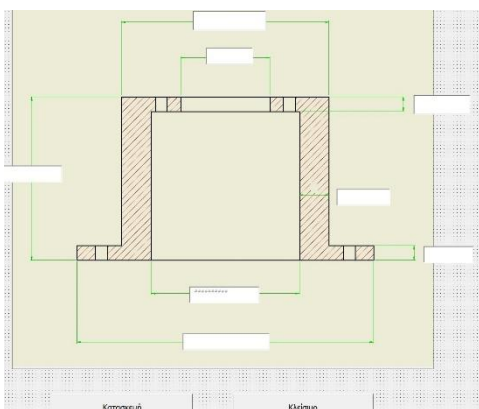
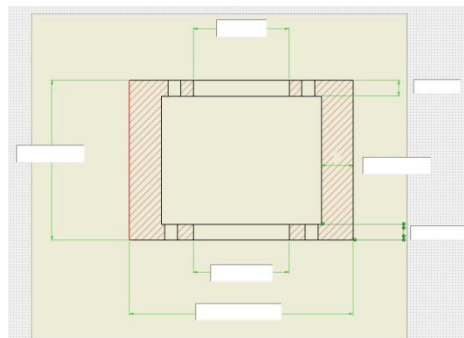
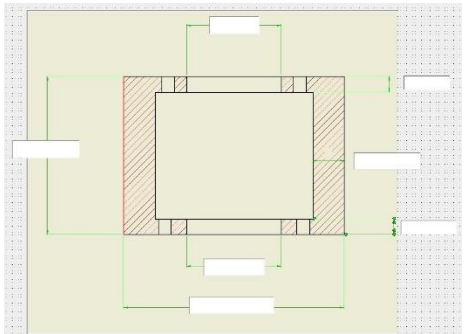
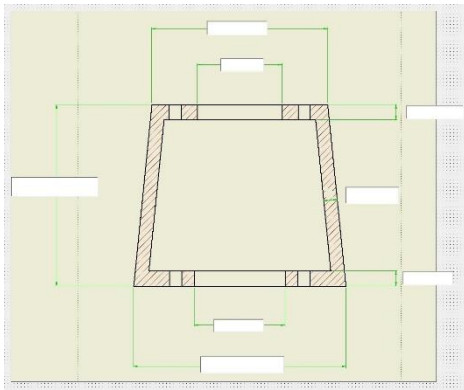
```
TextBox69.Text = TextBox2.Text
```

```
TextBox691.Text = TextBox2.Text
```

```
End Sub
```

Userform 11





Κώδικας του button «Κατασκευή»

Private Sub CommandButton1_Click()

'Create a new Part document

Dim oPartDoc As PartDocument

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _  
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

```
'PART A CYLINDER
```

```
Dim a As Double
```

```
Dim b As Double
```

```
Dim c As Double
```

```
Dim d As Double
```

```
a = Cdbl(TextBox1.Text)
```

```
b = Cdbl(TextBox2.Text)
```

```
c = Cdbl(TextBox3.Text)
```

```
a = a / 10
```

```
b = b / 10
```

```
c = c / 10
```

```
b = b / 2
```

```
d = b - c
```

```
Dim oSketch1 As PlanarSketch
```

```
Set oSketch1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(2))
```

```
Dim oSkPnts1 As SketchPoints
```

```
Set oSkPnts1 = oSketch1.SketchPoints
```

```
Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs1 As SketchCircles
```

```
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oextrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

'Part flantza bashs

Dim fa As Double

Dim fc As Double

fa = CDbI(TextBox15.Text)

fc = CDbI(TextBox17.Text)

fa = fa / 10

fc = fc / 10

fc = fc / 2

Dim oSketchf1 As PlanarSketch

Set oSketchf1 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPntsf1 As SketchPoints

Set oSkPntsf1 = oSketchf1.SketchPoints

Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf1 As SketchCircles

Set oCircsf1 = oSketchf1.SketchCircles

Dim oCircf1 As SketchCircle

Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)

Dim oCircsf2 As SketchCircles

Set oCircsf2 = oSketchf1.SketchCircles

Dim oCircf2 As SketchCircle

Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)

Dim oProfilef1 As Profile

Set oProfilef1 = oSketchf1.Profiles.AddForSolid

Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)

'part 1 flantza panw

Dim fd As Double

Dim fb As Double

fd = Cdbl(TextBox5.Text)

fb = Cdbl(TextBox4.Text)

fd = fd / 10

fb = fb / 10

fd = fd / 2

Dim oWPlane1 As WorkPlane

Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ
Plane"), a)

Dim oSketchf2 As PlanarSketch

```
Set oSketchf2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

```
Dim oSkPntsf2 As SketchPoints
```

```
Set oSkPntsf2 = oSketchf2.SketchPoints
```

```
Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf3 As SketchCircles
```

```
Set oCircsf3 = oSketchf2.SketchCircles
```

```
Dim oCircf3 As SketchCircle
```

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```


Dim e As Double

Dim f As Double

Dim g As Double

e = Cdbl(TextBox64.Text)

f = Cdbl(TextBox65.Text)

e = e / 10

f = f / 10

g = b - f

Dim oSketch41 As PlanarSketch

Set oSketch41 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPnts41 As SketchPoints

Set oSkPnts41 = oSketch41.SketchPoints

Call oSkPnts41.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs51 As SketchCircles

Set oCircs51 = oSketch41.SketchCircles

Dim oCirc51 As SketchCircle

Set oCirc51 = oCircs51.AddByCenterRadius(oSkPnts41(1), b)

Dim oCircs61 As SketchCircles

Set oCircs61 = oSketch41.SketchCircles

Dim oCirc61 As SketchCircle

Set oCirc61 = oCircs61.AddByCenterRadius(oSkPnts41(1), g)

Dim oProfile41 As Profile

Set oProfile41 = oSketch41.Profiles.AddForSolid

Dim oExtrusion31 As ExtrudeFeature

Set oextrude31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile41, e, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza katw

Dim fg As Double

Dim fh As Double

fg = Cdbl(TextBox63.Text)

fh = Cdbl(TextBox62.Text)

fg = fg / 10

fh = fh / 10

fh = fh / 2

```
Dim oSketchf51 As PlanarSketch
```

```
Set oSketchf51 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
Dim oSkPntsf51 As SketchPoints
```

```
Set oSkPntsf51 = oSketchf51.SketchPoints
```

```
Call oSkPntsf51.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf91 As SketchCircles
```

```
Set oCircsf91 = oSketchf51.SketchCircles
```

```
Dim oCircf91 As SketchCircle
```

```
Set oCircf91 = oCircsf91.AddByCenterRadius(oSkPntsf51(1), g)
```

```
Dim oCircsf101 As SketchCircles
```

```
Set oCircsf101 = oSketchf51.SketchCircles
```

```
Dim oCircf101 As SketchCircle
```

```
Set oCircf101 = oCircsf101.AddByCenterRadius(oSkPntsf51(1), fh)
```

```
Dim oProfilef51 As Profile
```

```
Set oProfilef51 = oSketchf51.Profiles.AddForSolid
```

```
Set oextrudef51 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
```

oProfilef51, fg, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza panw

Dim fi As Double

Dim fj As Double

Dim fki As Double

fi = Cdbl(TextBox66.Text)

fj = Cdbl(TextBox67.Text)

fi = fi / 10

fj = fj / 10

fki = e + a

fj = fj / 2

Dim oWPlane31 As WorkPlane

Set oWPlane31 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fki)

Dim oSketchf61 As PlanarSketch

Set oSketchf61 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

```
Dim oSkPntsf61 As SketchPoints
Set oSkPntsf61 = oSketchf61.SketchPoints
Call oSkPntsf61.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircsf111 As SketchCircles
Set oCircsf111 = oSketchf61.SketchCircles
Dim oCircf111 As SketchCircle
Set oCircf111 = oCircsf111.AddByCenterRadius(oSkPntsf61(1), g)
```

```
Dim oCircsf121 As SketchCircles
Set oCircsf121 = oSketchf61.SketchCircles
Dim oCircf121 As SketchCircle
Set oCircf121 = oCircsf121.AddByCenterRadius(oSkPntsf61(1), fj)
```

```
Dim oProfilef61 As Profile
Set oProfilef61 = oSketchf61.Profiles.AddForSolid
```

```
Set oextrudef61 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef61, fi, kNegativeExtentDirection, kJoinOperation)
```

'part 3

Dim e1 As Double

Dim f1 As Double

Dim g1 As Double

e1 = Cdbl(TextBox641.Text)

f1 = Cdbl(TextBox651.Text)

e1 = e1 / 10

f1 = f1 / 10

g1 = b - f1

Dim oSketch411 As PlanarSketch

Set oSketch411 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPnts411 As SketchPoints

Set oSkPnts411 = oSketch411.SketchPoints

Call oSkPnts411.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs511 As SketchCircles

Set oCircs511 = oSketch411.SketchCircles

Dim oCirc511 As SketchCircle

Set oCirc511 = oCircs511.AddByCenterRadius(oSkPnts411(1), b)

Dim oCircs611 As SketchCircles

Set oCircs611 = oSketch411.SketchCircles

Dim oCirc611 As SketchCircle

Set oCirc611 = oCircs611.AddByCenterRadius(oSkPnts411(1), g1)

Dim oProfile411 As Profile

Set oProfile411 = oSketch411.Profiles.AddForSolid

Dim oExtrution311 As ExtrudeFeature

Set oextrude311 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile411, e1, kPositiveExtentDirection, kJoinOperation)

'part 3 flantza katw

Dim fg1 As Double

Dim fh1 As Double

fg1 = Cdbl(TextBox631.Text)

fh1 = Cdbl(TextBox621.Text)

fg1 = fg1 / 10

fh1 = fh1 / 10

fh1 = fh1 / 2

```
Dim oSketchf511 As PlanarSketch
```

```
Set oSketchf511 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
Dim oSkPntsf511 As SketchPoints
```

```
Set oSkPntsf511 = oSketchf511.SketchPoints
```

```
Call oSkPntsf511.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf911 As SketchCircles
```

```
Set oCircsf911 = oSketchf511.SketchCircles
```

```
Dim oCircf911 As SketchCircle
```

```
Set oCircf911 = oCircsf911.AddByCenterRadius(oSkPntsf511(1), g1)
```

```
Dim oCircsf1011 As SketchCircles
```

```
Set oCircsf1011 = oSketchf511.SketchCircles
```

```
Dim oCircf1011 As SketchCircle
```

```
Set oCircf1011 = oCircsf1011.AddByCenterRadius(oSkPntsf511(1), fh1)
```

```
Dim oProfilef511 As Profile
```

```
Set oProfilef511 = oSketchf511.Profiles.AddForSolid
```

```
Set oextrudef511 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef511, fg1, kPositiveExtentDirection, kJoinOperation)
```


'part 3 flantza panw

Dim fi1 As Double

Dim fj1 As Double

Dim fki1 As Double

fi1 = Cdbl(TextBox661.Text)

fj1 = Cdbl(TextBox671.Text)

fi1 = fi1 / 10

fj1 = fj1 / 10

fki1 = e + a + e1

fj1 = fj1 / 2

Dim oWPlane311 As WorkPlane

Set oWPlane311 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fki1)

Dim oSketchf611 As PlanarSketch

Set oSketchf611 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf611 As SketchPoints

```
Set oSkPntsf611 = oSketchf611.SketchPoints
Call oSkPntsf611.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf1111 As SketchCircles
```

```
Set oCircsf1111 = oSketchf611.SketchCircles
```

```
Dim oCircf1111 As SketchCircle
```

```
Set oCircf1111 = oCircsf1111.AddByCenterRadius(oSkPntsf611(1), g1)
```

```
Dim oCircsf1211 As SketchCircles
```

```
Set oCircsf1211 = oSketchf611.SketchCircles
```

```
Dim oCircf1211 As SketchCircle
```

```
Set oCircf1211 = oCircsf1211.AddByCenterRadius(oSkPntsf611(1), fj1)
```

```
Dim oProfilef611 As Profile
```

```
Set oProfilef611 = oSketchf611.Profiles.AddForSolid
```

```
Set oextrudef611 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef611, fi1, kNegativeExtentDirection, kJoinOperation)
```

```
'part 4
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = Cdbl(TextBox6.Text)
```

```
l = CDbI(TextBox9.Text)
```

```
m = CDbI(TextBox8.Text)
```

```
j = j / 10
```

```
l = l / 10
```

```
m = m / 10
```

```
l = l / 2
```

```
mi = b - m
```

```
n = (b - l) / Sqr((j) ^ 2 + (b - l) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch2 As PlanarSketch
```

```
Set oSketch2 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
Dim oSketch3 As PlanarSketch
```

```
Set oSketch3 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts2 As SketchPoints
```

```
Set oSkPnts2 = oSketch2.SketchPoints
```

```
Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oSkPnts3 As SketchPoints
Set oSkPnts3 = oSketch3.SketchPoints
Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircs3 As SketchCircles
Set oCircs3 = oSketch2.SketchCircles
Dim oCirc3 As SketchCircle
Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)
```

```
Dim oCircs4 As SketchCircles
Set oCircs4 = oSketch3.SketchCircles
Dim oCirc4 As SketchCircle
Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)
```

'Create a profile

```
Dim oProfile2 As Profile
Set oProfile2 = oSketch2.Profiles.AddForSolid
Dim oProfile3 As Profile
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

'Create an extrusion

```
Dim oExtrusion2 As ExtrudeFeature
Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)

Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)
```

```
b = ((b * 2) * 10)
```

```
TextBox8.Value = b
```

```
'part 4 katw flantza
```

```
Dim fe As Double
```

```
Dim ff As Double
```

```
fe = CDbI(TextBox10.Text)
```

```
ff = CDbI(TextBox11.Text)
```

```
fe = fe / 10
```

```
ff = ff / 10
```

```
fe = fe / 2
```

```
Dim oSketchf3 As PlanarSketch
```

```
Set oSketchf3 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
Dim oSkPntsf3 As SketchPoints
```

```
Set oSkPntsf3 = oSketchf3.SketchPoints
```

```
Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf5 As SketchCircles
```

```
Set oCircsf5 = oSketchf3.SketchCircles
```

Dim oCircf5 As SketchCircle

Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)

Dim oCircsf6 As SketchCircles

Set oCircsf6 = oSketchf3.SketchCircles

Dim oCircf6 As SketchCircle

Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)

Dim oProfilef3 As Profile

Set oProfilef3 = oSketchf3.Profiles.AddForSolid

Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef3, ff, kPositiveExtentDirection, kJoinOperation, -n)

'part 4 panw flantza

Dim fg2 As Double

Dim fh2 As Double

Dim fi2 As Double

Dim fj2 As Double

fg2 = CDbI(TextBox12.Text)

fh2 = CDbI(TextBox13.Text)

fg2 = fg2 / 10

fh2 = fh2 / 10

fj2 = l - m

fg2 = fg2 / 2

fi2 = e + a + e1 + j

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi2)

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf4 As SketchPoints

Set oSkPntsf4 = oSketchf4.SketchPoints

Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf7 As SketchCircles

Set oCircsf7 = oSketchf4.SketchCircles

Dim oCircf7 As SketchCircle

Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj2)

Dim oCircsf8 As SketchCircles

Set oCircsf8 = oSketchf4.SketchCircles

```
Dim oCircf8 As SketchCircle
```

```
Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg2)
```

```
Dim oProfilef4 As Profile
```

```
Set oProfilef4 = oSketchf4.Profiles.AddForSolid
```

```
Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef4, fh2, kNegativeExtentDirection, kJoinOperation, n)
```

```
'part 5
```

```
Dim p21 As Double
```

```
Dim q21 As Double
```

```
Dim r21 As Double
```

```
p21 = Cdbl(TextBox281.Text)
```

```
q21 = Cdbl(TextBox251.Text)
```

```
p21 = p21 / 10
```

```
q21 = q21 / 10
```

```
r21 = l - q21
```

```
Dim oSketch421 As PlanarSketch
```



```
Set oSketch421 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(7))
```

```
Dim oSkPnts421 As SketchPoints
```

```
Set oSkPnts421 = oSketch421.SketchPoints
```

```
Call oSkPnts421.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs521 As SketchCircles
```

```
Set oCircs521 = oSketch421.SketchCircles
```

```
Dim oCirc521 As SketchCircle
```

```
Set oCirc521 = oCircs521.AddByCenterRadius(oSkPnts421(1), l)
```

```
Dim oCircs621 As SketchCircles
```

```
Set oCircs621 = oSketch421.SketchCircles
```

```
Dim oCirc621 As SketchCircle
```

```
Set oCirc621 = oCircs621.AddByCenterRadius(oSkPnts421(1), r21)
```

```
Dim oProfile421 As Profile
```

```
Set oProfile421 = oSketch421.Profiles.AddForSolid
```

```
Dim oExtrusion321 As ExtrudeFeature
```

```
Set oextrude321 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile421, p21, kPositiveExtentDirection, kJoinOperation)
```

```
'part 5 flantza katw
```

```
Dim fk21 As Double
```

```
Dim fl21 As Double
```

```
fk21 = CDbI(TextBox231.Text)
```

```
fl21 = CDbI(TextBox241.Text)
```

```
fk21 = fk21 / 10
```

```
fl21 = fl21 / 10
```

```
fk21 = fk21 / 2
```

```
Dim oSketchf521 As PlanarSketch
```

```
Set oSketchf521 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))
```

```
Dim oSkPntsf521 As SketchPoints
```

```
Set oSkPntsf521 = oSketchf521.SketchPoints
```

```
Call oSkPntsf521.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf921 As SketchCircles
```

```
Set oCircsf921 = oSketchf521.SketchCircles
```

```
Dim oCircf921 As SketchCircle
```

```
Set oCircf921 = oCircsf921.AddByCenterRadius(oSkPntsf521(1), r21)
```

```
Dim oCircsf1021 As SketchCircles
```

```
Set oCircsf1021 = oSketchf521.SketchCircles
```

```
Dim oCircf1021 As SketchCircle
```

```
Set oCircf1021 = oCircsf1021.AddByCenterRadius(oSkPntsf521(1), fk21)
```

```
Dim oProfilef521 As Profile
```

```
Set oProfilef521 = oSketchf521.Profiles.AddForSolid
```

```
Set oextrudef521 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef521, fl21, kPositiveExtentDirection, kJoinOperation)
```

'part 5 flantza panw

```
Dim fm21 As Double
```

```
Dim fn21 As Double
```

```
Dim fo21 As Double
```

```
fn21 = CDbI(TextBox271.Text)
```

```
fm21 = CDbI(TextBox261.Text)
```

```
fm21 = fm21 / 10
```

```
fn21 = fn21 / 10
```

```
fo21 = e + a + e1 + j + p21
```

fn21 = fn21 / 2

Dim oWPlane321 As WorkPlane

Set oWPlane321 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo21)

Dim oSketchf621 As PlanarSketch

Set oSketchf621 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(8))

Dim oSkPntsf621 As SketchPoints

Set oSkPntsf621 = oSketchf621.SketchPoints

Call oSkPntsf621.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf1121 As SketchCircles

Set oCircsf1121 = oSketchf621.SketchCircles

Dim oCircf1121 As SketchCircle

Set oCircf1121 = oCircsf1121.AddByCenterRadius(oSkPntsf621(1), r21)

Dim oCircsf1221 As SketchCircles

Set oCircsf1221 = oSketchf621.SketchCircles

Dim oCircf1221 As SketchCircle

Set oCircf1221 = oCircsf1221.AddByCenterRadius(oSkPntsf621(1), fn21)

Dim oProfilef621 As Profile

```
Set oProfile621 = oSketchf621.Profiles.AddForSolid
```

```
Set oextrudef621 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef621, fm21, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane31.Visible = False
```

```
oWPlane311.Visible = False
```

```
oWPlane2.Visible = False
```

```
oWPlane321.Visible = False
```

```
End Sub
```

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm14
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του userform 11

```
Private Sub UserForm_Initialize()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 4.2
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As  
MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal  
ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

```
TextBox61.Text = TextBox2.Text
```

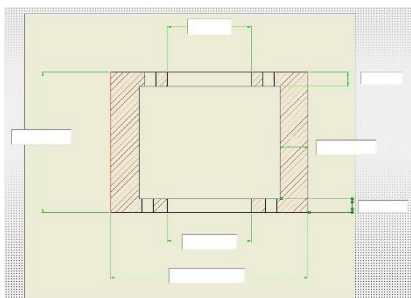
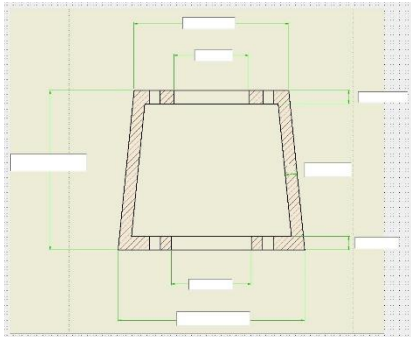
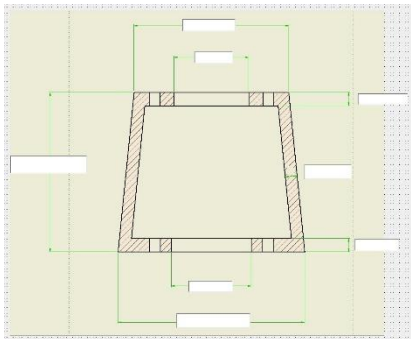
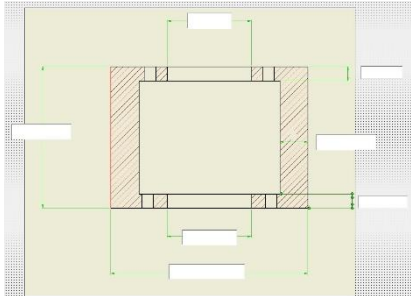
```
TextBox611.Text = TextBox2.Text
```

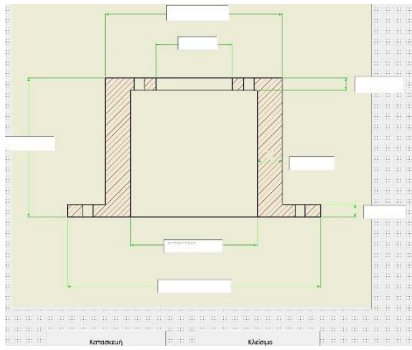
TextBox7.Text = TextBox2.Text

TextBox691.Text = TextBox9.Text

End Sub

Userform 12





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

```
'PART A CYLINDER
```

```
Dim a As Double
```

```
Dim b As Double
```

```
Dim c As Double
```

Dim d As Double

a = Cdbl(TextBox1.Text)

b = Cdbl(TextBox2.Text)

c = Cdbl(TextBox3.Text)

a = a / 10

b = b / 10

c = c / 10

b = b / 2

d = b - c

Dim oSketch1 As PlanarSketch

Set oSketch1 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPnts1 As SketchPoints

Set oSkPnts1 = oSketch1.SketchPoints

Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs1 As SketchCircles

Set oCircs1 = oSketch1.SketchCircles

Dim oCirc1 As SketchCircle

Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)

Dim oCircs2 As SketchCircles

Set oCircs2 = oSketch1.SketchCircles


```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = CDbI(TextBox15.Text)
```

```
fc = CDbI(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

```
fc = fc / 2
```

```
Dim oSketchf1 As PlanarSketch
```

```
Set oSketchf1 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(2))
```

```
Dim oSkPntsf1 As SketchPoints
```

```
Set oSkPntsf1 = oSketchf1.SketchPoints
```

```
Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircsf1 As SketchCircles
```

```
Set oCircsf1 = oSketchf1.SketchCircles
```

```
Dim oCircf1 As SketchCircle
```

```
Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)
```

```
Dim oCircsf2 As SketchCircles
```

```
Set oCircsf2 = oSketchf1.SketchCircles
```

```
Dim oCircf2 As SketchCircle
```

```
Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)
```

```
Dim oProfilef1 As Profile
```

```
Set oProfilef1 = oSketchf1.Profiles.AddForSolid
```

```
Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)
```

```
'part 1 flantza panw
```

```
Dim fd As Double
```

```
Dim fb As Double
```

```
fd = Cdbl(TextBox5.Text)
```

```
fb = Cdbl(TextBox4.Text)
```

fd = fd / 10

fb = fb / 10

fd = fd / 2

Dim oWPlane1 As WorkPlane

Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), a)

Dim oSketchf2 As PlanarSketch

Set oSketchf2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf2 As SketchPoints

Set oSkPntsf2 = oSketchf2.SketchPoints

Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf3 As SketchCircles

Set oCircsf3 = oSketchf2.SketchCircles

Dim oCircf3 As SketchCircle

Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)

Dim oCircsf4 As SketchCircles

Set oCircsf4 = oSketchf2.SketchCircles

Dim oCircf4 As SketchCircle

Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)

```
Dim oProfile2 As Profile
```

```
Set oProfile2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim e1 As Double
```

```
Dim f1 As Double
```

```
Dim g1 As Double
```

```
e1 = Cdbl(TextBox641.Text)
```

```
f1 = Cdbl(TextBox651.Text)
```

```
e1 = e1 / 10
```

```
f1 = f1 / 10
```

```
g1 = b - f1
```

```
Dim oSketch411 As PlanarSketch
```

```
Set oSketch411 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(4))
```

Dim oSkPnts411 As SketchPoints

Set oSkPnts411 = oSketch411.SketchPoints

Call oSkPnts411.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs511 As SketchCircles

Set oCircs511 = oSketch411.SketchCircles

Dim oCirc511 As SketchCircle

Set oCirc511 = oCircs511.AddByCenterRadius(oSkPnts411(1), b)

Dim oCircs611 As SketchCircles

Set oCircs611 = oSketch411.SketchCircles

Dim oCirc611 As SketchCircle

Set oCirc611 = oCircs611.AddByCenterRadius(oSkPnts411(1), g1)

Dim oProfile411 As Profile

Set oProfile411 = oSketch411.Profiles.AddForSolid

Dim oExtrusion311 As ExtrudeFeature

Set oextrude311 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile411, e1, kPositiveExtentDirection, kJoinOperation)

'part 2 flantza katw

Dim fg1 As Double

Dim fh1 As Double

```
fg1 = CDb1(TextBox631.Text)
```

```
fh1 = CDb1(TextBox621.Text)
```

```
fg1 = fg1 / 10
```

```
fh1 = fh1 / 10
```

```
fh1 = fh1 / 2
```

```
Dim oSketchf511 As PlanarSketch
```

```
Set oSketchf511 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))
```

```
Dim oSkPntsf511 As SketchPoints
```

```
Set oSkPntsf511 = oSketchf511.SketchPoints
```

```
Call oSkPntsf511.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf911 As SketchCircles
```

```
Set oCircsf911 = oSketchf511.SketchCircles
```

```
Dim oCircf911 As SketchCircle
```

```
Set oCircf911 = oCircsf911.AddByCenterRadius(oSkPntsf511(1), g1)
```

```
Dim oCircsf1011 As SketchCircles
```

```
Set oCircsf1011 = oSketchf511.SketchCircles
```

```
Dim oCircf1011 As SketchCircle
```

```
Set oCircf1011 = oCircsf1011.AddByCenterRadius(oSkPntsf511(1), fh1)
```

```
Dim oProfilef511 As Profile
```

```
Set oProfilef511 = oSketchf511.Profiles.AddForSolid
```

```
Set oextrudef511 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef511, fg1, kPositiveExtentDirection, kJoinOperation)
```

```
'part 2 flantza panw
```

```
Dim fi1 As Double
```

```
Dim fj1 As Double
```

```
Dim fki1 As Double
```

```
fi1 = CDb1(TextBox661.Text)
```

```
fj1 = CDb1(TextBox671.Text)
```

```
fi1 = fi1 / 10
```

```
fj1 = fj1 / 10
```

```
fki1 = a + e1
```

```
fj1 = fj1 / 2
```

Dim oWPlane311 As WorkPlane

Set oWPlane311 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fki1)

Dim oSketchf611 As PlanarSketch

Set oSketchf611 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf611 As SketchPoints

Set oSkPntsf611 = oSketchf611.SketchPoints

Call oSkPntsf611.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf1111 As SketchCircles

Set oCircsf1111 = oSketchf611.SketchCircles

Dim oCircf1111 As SketchCircle

Set oCircf1111 = oCircsf1111.AddByCenterRadius(oSkPntsf611(1), g1)

Dim oCircsf1211 As SketchCircles

Set oCircsf1211 = oSketchf611.SketchCircles

Dim oCircf1211 As SketchCircle

Set oCircf1211 = oCircsf1211.AddByCenterRadius(oSkPntsf611(1), fj1)

Dim oProfilef611 As Profile

Set oProfilef611 = oSketchf611.Profiles.AddForSolid


```
Set oextrudef611 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef611, fi1, kNegativeExtentDirection, kJoinOperation)
```

```
'part 3
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = CDb1(TextBox6.Text)
```

```
l = CDb1(TextBox9.Text)
```

```
m = CDb1(TextBox8.Text)
```

```
j = j / 10
```

```
l = l / 10
```

```
m = m / 10
```

```
l = l / 2
```

```
mi = b - m
```

```
n = (b - l) / Sqr((j) ^ 2 + (b - l) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch2 As PlanarSketch
```

```
Set oSketch2 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(5))
```

Dim oSketch3 As PlanarSketch

Set oSketch3 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

'Create 3 points...one for the center of the circle and 2 for the line

Dim oSkPnts2 As SketchPoints

Set oSkPnts2 = oSketch2.SketchPoints

Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oSkPnts3 As SketchPoints

Set oSkPnts3 = oSketch3.SketchPoints

Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircs3 As SketchCircles

Set oCircs3 = oSketch2.SketchCircles

Dim oCirc3 As SketchCircle

Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)

Dim oCircs4 As SketchCircles

Set oCircs4 = oSketch3.SketchCircles

Dim oCirc4 As SketchCircle

Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)

'Create a profile

Dim oProfile2 As Profile

Set oProfile2 = oSketch2.Profiles.AddForSolid

```
Dim oProfile3 As Profile
```

```
Set oProfile3 = oSketch3.Profiles.AddForSolid
```

```
'Create an extrusion
```

```
Dim oExtrusion2 As ExtrudeFeature
```

```
Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)
```

```
Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)
```

```
b = ((b * 2) * 10)
```

```
TextBox8.Value = b
```

```
'part 3 katw flantza
```

```
Dim fe As Double
```

```
Dim ff As Double
```

```
fe = Cdbl(TextBox10.Text)
```

```
ff = Cdbl(TextBox11.Text)
```

```
fe = fe / 10
```

```
ff = ff / 10
```

```
fe = fe / 2
```

```
Dim oSketchf3 As PlanarSketch
```

```
Set oSketchf3 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(5))
```

```
Dim oSkPntsf3 As SketchPoints
Set oSkPntsf3 = oSketchf3.SketchPoints
Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircsf5 As SketchCircles
Set oCircsf5 = oSketchf3.SketchCircles
Dim oCircf5 As SketchCircle
Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)
```

```
Dim oCircsf6 As SketchCircles
Set oCircsf6 = oSketchf3.SketchCircles
Dim oCircf6 As SketchCircle
Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)
```

```
Dim oProfilef3 As Profile
Set oProfilef3 = oSketchf3.Profiles.AddForSolid
```

```
Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```

'part 3 panw flantza

Dim fg2 As Double

Dim fh2 As Double

Dim fi2 As Double

Dim fj2 As Double

fg2 = Cdbl(TextBox12.Text)

fh2 = Cdbl(TextBox13.Text)

fg2 = fg2 / 10

fh2 = fh2 / 10

fj2 = l - m

fg2 = fg2 / 2

fi2 = a + e1 + j

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi2)

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf4 As SketchPoints

Set oSkPntsf4 = oSketchf4.SketchPoints

Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf7 As SketchCircles

Set oCircsf7 = oSketchf4.SketchCircles

Dim oCircf7 As SketchCircle

Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj2)

Dim oCircsf8 As SketchCircles

Set oCircsf8 = oSketchf4.SketchCircles

Dim oCircf8 As SketchCircle

Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg2)

Dim oProfilef4 As Profile

Set oProfilef4 = oSketchf4.Profiles.AddForSolid

Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef4, fh2, kNegativeExtentDirection, kJoinOperation, n)

'part 4

Dim j1 As Double

Dim l1 As Double

Dim m1 As Double

Dim n1 As Double

Dim o1 As Double

```
j1 = CDb1(TextBox61.Text)
l1 = CDb1(TextBox91.Text)
m1 = CDb1(TextBox81.Text)
```

```
j1 = j1 / 10
l1 = l1 / 10
m1 = m1 / 10
l1 = l1 / 2
mi1 = l - m1
```

```
n1 = (l - l1) / Sqr((j1) ^ 2 + (l - l1) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch21 As PlanarSketch
```

```
Set oSketch21 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
Dim oSketch31 As PlanarSketch
```

```
Set oSketch31 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts21 As SketchPoints
```

```
Set oSkPnts21 = oSketch21.SketchPoints
```

Call oSkPnts21.Add(oTG.CreatePoint2d(0, 0), False)

Dim oSkPnts31 As SketchPoints

Set oSkPnts31 = oSketch31.SketchPoints

Call oSkPnts31.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircs31 As SketchCircles

Set oCircs31 = oSketch21.SketchCircles

Dim oCirc31 As SketchCircle

Set oCirc31 = oCircs31.AddByCenterRadius(oSkPnts21(1), l)

Dim oCircs41 As SketchCircles

Set oCircs41 = oSketch31.SketchCircles

Dim oCirc41 As SketchCircle

Set oCirc41 = oCircs41.AddByCenterRadius(oSkPnts31(1), mi1)

'Create a profile

Dim oProfile21 As Profile

Set oProfile21 = oSketch21.Profiles.AddForSolid

Dim oProfile31 As Profile

Set oProfile31 = oSketch31.Profiles.AddForSolid

'Create an extrusion

Dim oExtrusion21 As ExtrudeFeature

Set oextrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile21, j1, kPositiveExtentDirection, kJoinOperation, -n1)

Set oextrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile31, j1, kPositiveExtentDirection, kCutOperation, -n1)

'part 4 katw flantza

Dim fe1 As Double

Dim ff1 As Double

fe1 = CDb1(TextBox101.Text)

ff1 = CDb1(TextBox111.Text)

fe1 = fe1 / 10

ff1 = ff1 / 10

fe1 = fe1 / 2

Dim oSketchf31 As PlanarSketch

Set oSketchf31 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf31 As SketchPoints

Set oSkPntsf31 = oSketchf31.SketchPoints

Call oSkPntsf31.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf51 As SketchCircles

Set oCircsf51 = oSketchf31.SketchCircles

Dim oCircf51 As SketchCircle

```
Set oCircf51 = oCircsf51.AddByCenterRadius(oSkPntsf31(1), mi1)
```

```
Dim oCircsf61 As SketchCircles
```

```
Set oCircsf61 = oSketchf31.SketchCircles
```

```
Dim oCircf61 As SketchCircle
```

```
Set oCircf61 = oCircsf61.AddByCenterRadius(oSkPntsf31(1), fe1)
```

```
Dim oProfilef31 As Profile
```

```
Set oProfilef31 = oSketchf31.Profiles.AddForSolid
```

```
Set oextrudef31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef31, ff1, kPositiveExtentDirection, kJoinOperation, -n1)
```

```
'part 4 panw flantza
```

```
Dim fg12 As Double
```

```
Dim fh12 As Double
```

```
Dim fi12 As Double
```

```
Dim fj12 As Double
```

```
fg12 = CDb1(TextBox121.Text)
```

```
fh12 = CDb1(TextBox131.Text)
```

```
fg12 = fg12 / 10
```

```
fh12 = fh12 / 10
```

$fj12 = l1 - m1$

$fg12 = fg12 / 2$

$fi12 = a + e1 + j + j1$

Dim oWPlane21 As WorkPlane

Set oWPlane21 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi12)

Dim oSketchf41 As PlanarSketch

Set oSketchf41 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf41 As SketchPoints

Set oSkPntsf41 = oSketchf41.SketchPoints

Call oSkPntsf41.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf71 As SketchCircles

Set oCircsf71 = oSketchf41.SketchCircles

Dim oCircf71 As SketchCircle

Set oCircf71 = oCircsf71.AddByCenterRadius(oSkPntsf41(1), fj12)

Dim oCircsf81 As SketchCircles

Set oCircsf81 = oSketchf41.SketchCircles

Dim oCircf81 As SketchCircle

```
Set oCircf81 = oCircof81.AddByCenterRadius(oSkPntsf41(1), fg12)
```

```
Dim oProfilef41 As Profile
```

```
Set oProfilef41 = oSketchf41.Profiles.AddForSolid
```

```
Set oextrudef41 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_  
oProfilef41, fh12, kNegativeExtentDirection, kJoinOperation, n1)
```

```
'part 5
```

```
Dim p21 As Double
```

```
Dim q21 As Double
```

```
Dim r21 As Double
```

```
p21 = Cdbl(TextBox281.Text)
```

```
q21 = Cdbl(TextBox251.Text)
```

```
p21 = p21 / 10
```

```
q21 = q21 / 10
```

```
r21 = l1 - q21
```

```
Dim oSketch421 As PlanarSketch
```

```
Set oSketch421 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(7))
```

```
Dim oSkPnts421 As SketchPoints
```

```
Set oSkPnts421 = oSketch421.SketchPoints
```

```
Call oSkPnts421.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs521 As SketchCircles
```

```
Set oCircs521 = oSketch421.SketchCircles
```

```
Dim oCirc521 As SketchCircle
```

```
Set oCirc521 = oCircs521.AddByCenterRadius(oSkPnts421(1), l1)
```

```
Dim oCircs621 As SketchCircles
```

```
Set oCircs621 = oSketch421.SketchCircles
```

```
Dim oCirc621 As SketchCircle
```

```
Set oCirc621 = oCircs621.AddByCenterRadius(oSkPnts421(1), r21)
```

```
Dim oProfile421 As Profile
```

```
Set oProfile421 = oSketch421.Profiles.AddForSolid
```

```
Dim oExtrusion321 As ExtrudeFeature
```

```
Set oextrude321 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfile421, p21, kPositiveExtentDirection, kJoinOperation)
```

```
'part 5 flantza katw
```

```
Dim fk21 As Double
```

```
Dim fl21 As Double
```

```
fk21 = CDbI(TextBox231.Text)
```

```
fl21 = CDbI(TextBox241.Text)
```

```
fk21 = fk21 / 10
```

```
fl21 = fl21 / 10
```

```
fk21 = fk21 / 2
```

```
Dim oSketchf521 As PlanarSketch
```

```
Set oSketchf521 =
```

```
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))
```

```
Dim oSkPntsf521 As SketchPoints
```

```
Set oSkPntsf521 = oSketchf521.SketchPoints
```

```
Call oSkPntsf521.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCirclf921 As SketchCircles
```

```
Set oCirclf921 = oSketchf521.SketchCircles
```

```
Dim oCircf921 As SketchCircle
```

```
Set oCircf921 = oCircsf921.AddByCenterRadius(oSkPntsf521(1), r21)
```

```
Dim oCircsf1021 As SketchCircles
```

```
Set oCircsf1021 = oSketchf521.SketchCircles
```

```
Dim oCircf1021 As SketchCircle
```

```
Set oCircf1021 = oCircsf1021.AddByCenterRadius(oSkPntsf521(1), fk21)
```

```
Dim oProfilef521 As Profile
```

```
Set oProfilef521 = oSketchf521.Profiles.AddForSolid
```

```
Set oextrudef521 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef521, fl21, kPositiveExtentDirection, kJoinOperation)
```

'part 5 flantza panw

```
Dim fm21 As Double
```

```
Dim fn21 As Double
```

```
Dim fo21 As Double
```

```
fn21 = CDbI(TextBox271.Text)
```

```
fm21 = CDbI(TextBox261.Text)
```

```
fm21 = fm21 / 10
```

```
fn21 = fn21 / 10
```

```
fo21 = a + e1 + j + j1 + p21
```

fn21 = fn21 / 2

Dim oWPlane321 As WorkPlane

Set oWPlane321 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo21)

Dim oSketchf621 As PlanarSketch

Set oSketchf621 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(8))

Dim oSkPntsf621 As SketchPoints

Set oSkPntsf621 = oSketchf621.SketchPoints

Call oSkPntsf621.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf1121 As SketchCircles

Set oCircsf1121 = oSketchf621.SketchCircles

Dim oCircf1121 As SketchCircle

Set oCircf1121 = oCircsf1121.AddByCenterRadius(oSkPntsf621(1), r21)

Dim oCircsf1221 As SketchCircles

Set oCircsf1221 = oSketchf621.SketchCircles

Dim oCircf1221 As SketchCircle

Set oCircf1221 = oCircsf1221.AddByCenterRadius(oSkPntsf621(1), fn21)

Dim oProfilef621 As Profile


```
Set oProfilef621 = oSketchf621.Profiles.AddForSolid
```

```
Set oextrudef621 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef621, fm21, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane21.Visible = False
```

```
oWPlane311.Visible = False
```

```
oWPlane2.Visible = False
```

```
oWPlane321.Visible = False
```

```
End Sub
```

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm14
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του userform 12

```
Private Sub UserForm_Activate()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 4.2
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

```
    TextBox611.Text = TextBox2.Text
```

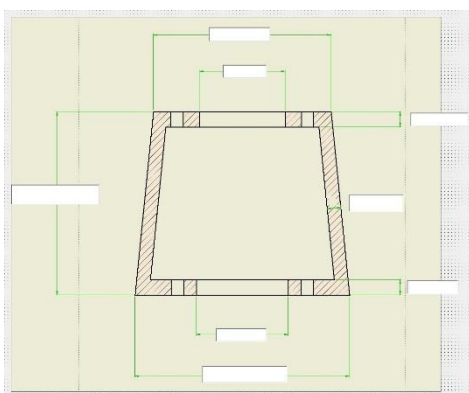
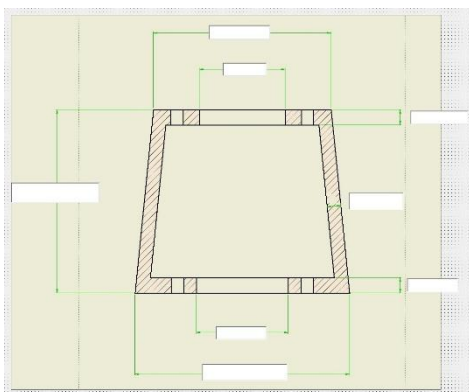
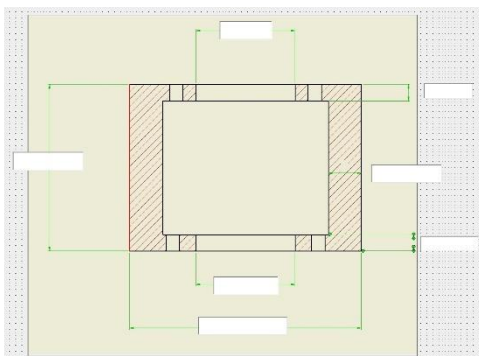
```
    TextBox7.Text = TextBox2.Text
```

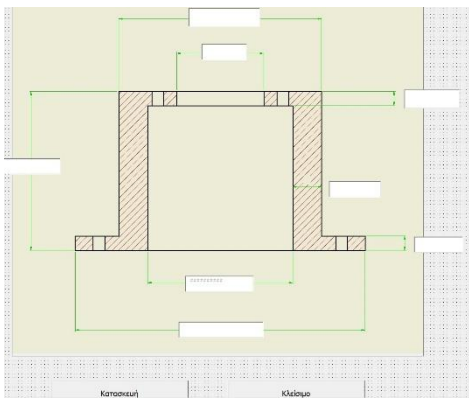
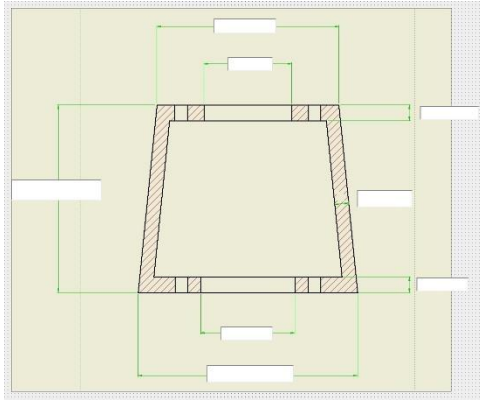
```
    TextBox71.Text = TextBox9.Text
```

```
    TextBox691.Text = TextBox91.Text
```

```
End Sub
```

Userform 13





Κώδικας του button «Κατασκευή»

```
Private Sub CommandButton1_Click()
```

```
'Create a new Part document
```

```
Dim oPartDoc As PartDocument
```

```
Set oPartDoc = ThisApplication.Documents.Add(kPartDocumentObject, _
```

```
    ThisApplication.FileManager.GetTemplateFile(kPartDocumentObject,  
kMetricSystemOfMeasure, kISO_DraftingStandard))
```

```
'Set a reference to the component definition
```

```
Dim oCompDef As PartComponentDefinition
```

```
Set oCompDef = oPartDoc.ComponentDefinition
```

```
Dim oTG As TransientGeometry
```

```
Set oTG = ThisApplication.TransientGeometry
```

'PART A CYLINDER

Dim a As Double

Dim b As Double

Dim c As Double

Dim d As Double

a = Cdbl(TextBox1.Text)

b = Cdbl(TextBox2.Text)

c = Cdbl(TextBox3.Text)

a = a / 10

b = b / 10

c = c / 10

b = b / 2

d = b - c

Dim oSketch1 As PlanarSketch

Set oSketch1 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPnts1 As SketchPoints

Set oSkPnts1 = oSketch1.SketchPoints

Call oSkPnts1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircs1 As SketchCircles

```
Set oCircs1 = oSketch1.SketchCircles
```

```
Dim oCirc1 As SketchCircle
```

```
Set oCirc1 = oCircs1.AddByCenterRadius(oSkPnts1(1), b)
```

```
Dim oCircs2 As SketchCircles
```

```
Set oCircs2 = oSketch1.SketchCircles
```

```
Dim oCirc2 As SketchCircle
```

```
Set oCirc2 = oCircs2.AddByCenterRadius(oSkPnts1(1), d)
```

```
Dim oProfile1 As Profile
```

```
Set oProfile1 = oSketch1.Profiles.AddForSolid
```

```
Dim oExtrusion1 As ExtrudeFeature
```

```
Set oExtrude1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile1, a, kPositiveExtentDirection, kJoinOperation)
```

```
'Part flantza bashs
```

```
Dim fa As Double
```

```
Dim fc As Double
```

```
fa = Cdbl(TextBox15.Text)
```

```
fc = Cdbl(TextBox17.Text)
```

```
fa = fa / 10
```

```
fc = fc / 10
```

$fc = fc / 2$

Dim oSketchf1 As PlanarSketch

Set oSketchf1 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(2))

Dim oSkPntsf1 As SketchPoints

Set oSkPntsf1 = oSketchf1.SketchPoints

Call oSkPntsf1.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf1 As SketchCircles

Set oCircsf1 = oSketchf1.SketchCircles

Dim oCircf1 As SketchCircle

Set oCircf1 = oCircsf1.AddByCenterRadius(oSkPntsf1(1), fc)

Dim oCircsf2 As SketchCircles

Set oCircsf2 = oSketchf1.SketchCircles

Dim oCircf2 As SketchCircle

Set oCircf2 = oCircsf2.AddByCenterRadius(oSkPntsf1(1), b)

Dim oProfilef1 As Profile

Set oProfilef1 = oSketchf1.Profiles.AddForSolid

Set oextrudef1 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef1, fa, kPositiveExtentDirection, kJoinOperation)

'part 1 flantza panw

Dim fd As Double

Dim fb As Double

fd = Cdbl(TextBox5.Text)

fb = Cdbl(TextBox4.Text)

fd = fd / 10

fb = fb / 10

fd = fd / 2

Dim oWPlane1 As WorkPlane

Set oWPlane1 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), a)

Dim oSketchf2 As PlanarSketch

Set oSketchf2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf2 As SketchPoints

Set oSkPntsf2 = oSketchf2.SketchPoints

Call oSkPntsf2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oCircsf3 As SketchCircles

Set oCircsf3 = oSketchf2.SketchCircles

Dim oCircf3 As SketchCircle

```
Set oCircf3 = oCircsf3.AddByCenterRadius(oSkPntsf2(1), d)
```

```
Dim oCircsf4 As SketchCircles
```

```
Set oCircsf4 = oSketchf2.SketchCircles
```

```
Dim oCircf4 As SketchCircle
```

```
Set oCircf4 = oCircsf4.AddByCenterRadius(oSkPntsf2(1), fd)
```

```
Dim oProfilef2 As Profile
```

```
Set oProfilef2 = oSketchf2.Profiles.AddForSolid
```

```
Set oextrudef2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef2, fb, kNegativeExtentDirection, kJoinOperation)
```

```
d = ((d * 2) * 10)
```

```
TextBox60.Value = d
```

```
'part 2
```

```
Dim j As Double
```

```
Dim l As Double
```

```
Dim m As Double
```

```
Dim n As Double
```

```
Dim o As Double
```

```
j = Cdbl(TextBox6.Text)
```

```
l = Cdbl(TextBox9.Text)
```

```
m = Cdbl(TextBox8.Text)
```


$j = j / 10$

$l = l / 10$

$m = m / 10$

$l = l / 2$

$m_i = b - m$

$n = (b - l) / \text{Sqr}((j)^2 + (b - l)^2)$

'create workplane FOR CONY PART 3

Dim oSketch2 As PlanarSketch

Set oSketch2 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSketch3 As PlanarSketch

Set oSketch3 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

'Create 3 points...one for the center of the circle and 2 for the line

Dim oSkPnts2 As SketchPoints

Set oSkPnts2 = oSketch2.SketchPoints

Call oSkPnts2.Add(oTG.CreatePoint2d(0, 0), False)

Dim oSkPnts3 As SketchPoints

Set oSkPnts3 = oSketch3.SketchPoints

Call oSkPnts3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircs3 As SketchCircles

Set oCircs3 = oSketch2.SketchCircles

Dim oCirc3 As SketchCircle

Set oCirc3 = oCircs3.AddByCenterRadius(oSkPnts2(1), b)

Dim oCircs4 As SketchCircles

Set oCircs4 = oSketch3.SketchCircles

Dim oCirc4 As SketchCircle

Set oCirc4 = oCircs4.AddByCenterRadius(oSkPnts3(1), mi)

'Create a profile

Dim oProfile2 As Profile

Set oProfile2 = oSketch2.Profiles.AddForSolid

Dim oProfile3 As Profile

Set oProfile3 = oSketch3.Profiles.AddForSolid

'Create an extrusion

Dim oExtrusion2 As ExtrudeFeature

Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile2, j, kPositiveExtentDirection, kJoinOperation, -n)

Set oextrude2 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile3, j, kPositiveExtentDirection, kCutOperation, -n)

b = ((b * 2) * 10)

TextBox8.Value = b

'part 2 katw flantza

Dim fe As Double

Dim ff As Double

fe = CDb1(TextBox10.Text)

ff = CDb1(TextBox11.Text)

fe = fe / 10

ff = ff / 10

fe = fe / 2

Dim oSketchf3 As PlanarSketch

Set oSketchf3 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(4))

Dim oSkPntsf3 As SketchPoints

Set oSkPntsf3 = oSketchf3.SketchPoints

Call oSkPntsf3.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf5 As SketchCircles

Set oCircsf5 = oSketchf3.SketchCircles

Dim oCircf5 As SketchCircle

Set oCircf5 = oCircsf5.AddByCenterRadius(oSkPntsf3(1), mi)

Dim oCircsf6 As SketchCircles

```
Set oCircsf6 = oSketchf3.SketchCircles
```

```
Dim oCircf6 As SketchCircle
```

```
Set oCircf6 = oCircsf6.AddByCenterRadius(oSkPntsf3(1), fe)
```

```
Dim oProfilef3 As Profile
```

```
Set oProfilef3 = oSketchf3.Profiles.AddForSolid
```

```
Set oextrudef3 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef3, ff, kPositiveExtentDirection, kJoinOperation, -n)
```

```
'part 2 panw flantza
```

```
Dim fg2 As Double
```

```
Dim fh2 As Double
```

```
Dim fi2 As Double
```

```
Dim fj2 As Double
```

```
fg2 = CDbI(TextBox12.Text)
```

```
fh2 = CDbI(TextBox13.Text)
```

```
fg2 = fg2 / 10
```

```
fh2 = fh2 / 10
```

```
fj2 = l - m
```

```
fg2 = fg2 / 2
```

fi2 = a + j

Dim oWPlane2 As WorkPlane

Set oWPlane2 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi2)

Dim oSketchf4 As PlanarSketch

Set oSketchf4 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))

Dim oSkPntsf4 As SketchPoints

Set oSkPntsf4 = oSketchf4.SketchPoints

Call oSkPntsf4.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf7 As SketchCircles

Set oCircsf7 = oSketchf4.SketchCircles

Dim oCircf7 As SketchCircle

Set oCircf7 = oCircsf7.AddByCenterRadius(oSkPntsf4(1), fj2)

Dim oCircsf8 As SketchCircles

Set oCircsf8 = oSketchf4.SketchCircles

Dim oCircf8 As SketchCircle

Set oCircf8 = oCircsf8.AddByCenterRadius(oSkPntsf4(1), fg2)

Dim oProfilef4 As Profile

```
Set oProfile4 = oSketchf4.Profiles.AddForSolid
```

```
Set oextrudef4 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfile4, fh2, kNegativeExtentDirection, kJoinOperation, n)
```

```
'part 3
```

```
Dim j1 As Double
```

```
Dim l1 As Double
```

```
Dim m1 As Double
```

```
Dim n1 As Double
```

```
Dim o1 As Double
```

```
j1 = CDb1(TextBox61.Text)
```

```
l1 = CDb1(TextBox91.Text)
```

```
m1 = CDb1(TextBox81.Text)
```

```
j1 = j1 / 10
```

```
l1 = l1 / 10
```

```
m1 = m1 / 10
```

```
l1 = l1 / 2
```

```
mi1 = l - m1
```

```
n1 = (l - l1) / Sqr((j1) ^ 2 + (l - l1) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch21 As PlanarSketch
```

```
Set oSketch21 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
Dim oSketch31 As PlanarSketch
```

```
Set oSketch31 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(5))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts21 As SketchPoints
```

```
Set oSkPnts21 = oSketch21.SketchPoints
```

```
Call oSkPnts21.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oSkPnts31 As SketchPoints
```

```
Set oSkPnts31 = oSketch31.SketchPoints
```

```
Call oSkPnts31.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircs31 As SketchCircles
```

```
Set oCircs31 = oSketch21.SketchCircles
```

```
Dim oCirc31 As SketchCircle
```

```
Set oCirc31 = oCircs31.AddByCenterRadius(oSkPnts21(1), I)
```

```
Dim oCircs41 As SketchCircles
```

```
Set oCircs41 = oSketch31.SketchCircles
```

```
Dim oCirc41 As SketchCircle
```

```
Set oCirc41 = oCircs41.AddByCenterRadius(oSkPnts31(1), mi1)
```

'Create a profile

Dim oProfile21 As Profile

Set oProfile21 = oSketch21.Profiles.AddForSolid

Dim oProfile31 As Profile

Set oProfile31 = oSketch31.Profiles.AddForSolid

'Create an extrusion

Dim oExtrusion21 As ExtrudeFeature

Set oextrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile21, j1, kPositiveExtentDirection, kJoinOperation, -n1)

Set oextrude21 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfile31, j1, kPositiveExtentDirection, kCutOperation, -n1)

'part 3 katw flantza

Dim fe1 As Double

Dim ff1 As Double

fe1 = CDb1(TextBox101.Text)

ff1 = CDb1(TextBox111.Text)

fe1 = fe1 / 10

ff1 = ff1 / 10

fe1 = fe1 / 2

Dim oSketchf31 As PlanarSketch


```
Set oSketchf31 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.It  
em(5))
```

```
Dim oSkPntsf31 As SketchPoints
```

```
Set oSkPntsf31 = oSketchf31.SketchPoints
```

```
Call oSkPntsf31.Add(oTG.CreatePoint2d(0, 0), False)
```

```
'Sketch a circle
```

```
Dim oCircsf51 As SketchCircles
```

```
Set oCircsf51 = oSketchf31.SketchCircles
```

```
Dim oCircf51 As SketchCircle
```

```
Set oCircf51 = oCircsf51.AddByCenterRadius(oSkPntsf31(1), mi1)
```

```
Dim oCircsf61 As SketchCircles
```

```
Set oCircsf61 = oSketchf31.SketchCircles
```

```
Dim oCircf61 As SketchCircle
```

```
Set oCircf61 = oCircsf61.AddByCenterRadius(oSkPntsf31(1), fe1)
```

```
Dim oProfilef31 As Profile
```

```
Set oProfilef31 = oSketchf31.Profiles.AddForSolid
```

```
Set oextrudef31 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
oProfilef31, ff1, kPositiveExtentDirection, kJoinOperation, -n1)
```

'part 3 panw flantza

Dim fg1 As Double

Dim fh1 As Double

Dim fi1 As Double

Dim fj1 As Double

fg1 = Cdbl(TextBox121.Text)

fh1 = Cdbl(TextBox131.Text)

fg1 = fg1 / 10

fh1 = fh1 / 10

fj1 = l1 - m1

fg1 = fg1 / 2

fi1 = a + j + j1

Dim oWPlane21 As WorkPlane

Set oWPlane21 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi1)

Dim oSketchf41 As PlanarSketch

Set oSketchf41 =

oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

```
Dim oSkPntsf41 As SketchPoints
Set oSkPntsf41 = oSketchf41.SketchPoints
Call oSkPntsf41.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircsf71 As SketchCircles
Set oCircsf71 = oSketchf41.SketchCircles
Dim oCircf71 As SketchCircle
Set oCircf71 = oCircsf71.AddByCenterRadius(oSkPntsf41(1), fj1)
```

```
Dim oCircsf81 As SketchCircles
Set oCircsf81 = oSketchf41.SketchCircles
Dim oCircf81 As SketchCircle
Set oCircf81 = oCircsf81.AddByCenterRadius(oSkPntsf41(1), fg1)
```

```
Dim oProfilef41 As Profile
Set oProfilef41 = oSketchf41.Profiles.AddForSolid
```

```
Set oextrudef41 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfilef41, fh1, kNegativeExtentDirection, kJoinOperation, n1)
```

'part 4

```
Dim j2 As Double
Dim l2 As Double
Dim m2 As Double
Dim n2 As Double
Dim o2 As Double
```

```
j2 = CDb1(TextBox62.Text)
l2 = CDb1(TextBox92.Text)
m2 = CDb1(TextBox82.Text)
```

```
j2 = j2 / 10
l2 = l2 / 10
m2 = m2 / 10
l2 = l2 / 2
mi2 = l1 - m2
```

```
n2 = (l1 - l2) / Sqr((j2) ^ 2 + (l1 - l2) ^ 2)
```

```
'create workplane FOR CONY PART 3
```

```
Dim oSketch22 As PlanarSketch
```

```
Set oSketch22 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Ite
m(6))
```

```
Dim oSketch32 As PlanarSketch
```

```
Set oSketch32 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Ite
m(6))
```

```
'Create 3 points...one for the center of the circle and 2 for the line
```

```
Dim oSkPnts22 As SketchPoints
```

```
Set oSkPnts22 = oSketch22.SketchPoints
Call oSkPnts22.Add(oTG.CreatePoint2d(0, 0), False)
Dim oSkPnts32 As SketchPoints
Set oSkPnts32 = oSketch32.SketchPoints
Call oSkPnts32.Add(oTG.CreatePoint2d(0, 0), False)
```

'Sketch a circle

```
Dim oCircs32 As SketchCircles
Set oCircs32 = oSketch22.SketchCircles
Dim oCirc32 As SketchCircle
Set oCirc32 = oCircs32.AddByCenterRadius(oSkPnts22(1), l1)
```

```
Dim oCircs42 As SketchCircles
Set oCircs42 = oSketch32.SketchCircles
Dim oCirc42 As SketchCircle
Set oCirc42 = oCircs42.AddByCenterRadius(oSkPnts32(1), mi2)
```

'Create a profile

```
Dim oProfile22 As Profile
Set oProfile22 = oSketch22.Profiles.AddForSolid
Dim oProfile32 As Profile
Set oProfile32 = oSketch32.Profiles.AddForSolid
```

'Create an extrusion

```
Dim oExtrusion22 As ExtrudeFeature
Set oextrude22 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
    oProfile22, j2, kPositiveExtentDirection, kJoinOperation, -n2)
```

```
Set oextrude22 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _
```

oProfile32, j2, kPositiveExtentDirection, kCutOperation, -n2)

'part 4 katw flantza

Dim fe2 As Double

Dim ff2 As Double

fe2 = Cdbl(TextBox102.Text)

ff2 = Cdbl(TextBox112.Text)

fe2 = fe2 / 10

ff2 = ff2 / 10

fe2 = fe2 / 2

Dim oSketchf32 As PlanarSketch

Set oSketchf32 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(6))

Dim oSkPntsf32 As SketchPoints

Set oSkPntsf32 = oSketchf32.SketchPoints

Call oSkPntsf32.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf52 As SketchCircles

Set oCircsf52 = oSketchf32.SketchCircles

Dim oCircf52 As SketchCircle

Set oCircf52 = oCircsf52.AddByCenterRadius(oSkPntsf32(1), mi2)

Dim oCircsf62 As SketchCircles

Set oCircsf62 = oSketchf32.SketchCircles

Dim oCircf62 As SketchCircle

Set oCircf62 = oCircsf62.AddByCenterRadius(oSkPntsf32(1), fe2)

Dim oProfilef32 As Profile

Set oProfilef32 = oSketchf32.Profiles.AddForSolid

Set oextrudef32 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef32, ff2, kPositiveExtentDirection, kJoinOperation, -n2)

'part 4 panw flantza

Dim fg22 As Double

Dim fh22 As Double

Dim fi22 As Double

Dim fj22 As Double

fg22 = Cdbl(TextBox122.Text)

fh22 = Cdbl(TextBox132.Text)

fg22 = fg22 / 10

fh22 = fh22 / 10

fj22 = l2 - m2

fg22 = fg22 / 2

fi22 = a + j + j1 + j2

Dim oWPlane22 As WorkPlane

Set oWPlane22 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fi22)

Dim oSketchf42 As PlanarSketch

Set oSketchf42 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf42 As SketchPoints

Set oSkPntsf42 = oSketchf42.SketchPoints

Call oSkPntsf42.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf72 As SketchCircles

Set oCircsf72 = oSketchf42.SketchCircles

Dim oCircf72 As SketchCircle

Set oCircf72 = oCircsf72.AddByCenterRadius(oSkPntsf42(1), fj22)

Dim oCircsf82 As SketchCircles

Set oCircsf82 = oSketchf42.SketchCircles

Dim oCircf82 As SketchCircle

Set oCircf82 = oCircsf82.AddByCenterRadius(oSkPntsf42(1), fg22)

Dim oProfilef42 As Profile

Set oProfilef42 = oSketchf42.Profiles.AddForSolid

Set oextrudef42 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef42, fh22, kNegativeExtentDirection, kJoinOperation, n2)

'part 5

Dim p21 As Double

Dim q21 As Double

Dim r21 As Double

p21 = CDbI(TextBox281.Text)

q21 = CDbI(TextBox251.Text)

p21 = p21 / 10

q21 = q21 / 10

r21 = l2 - q21

```
Dim oSketch421 As PlanarSketch
```

```
Set oSketch421 =  
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))
```

```
Dim oSkPnts421 As SketchPoints
```

```
Set oSkPnts421 = oSketch421.SketchPoints
```

```
Call oSkPnts421.Add(oTG.CreatePoint2d(0, 0), False)
```

```
Dim oCircs521 As SketchCircles
```

```
Set oCircs521 = oSketch421.SketchCircles
```

```
Dim oCirc521 As SketchCircle
```

```
Set oCirc521 = oCircs521.AddByCenterRadius(oSkPnts421(1), l2)
```

```
Dim oCircs621 As SketchCircles
```

```
Set oCircs621 = oSketch421.SketchCircles
```

```
Dim oCirc621 As SketchCircle
```

```
Set oCirc621 = oCircs621.AddByCenterRadius(oSkPnts421(1), r21)
```

```
Dim oProfile421 As Profile
```

```
Set oProfile421 = oSketch421.Profiles.AddForSolid
```

```
Dim oExtrusion321 As ExtrudeFeature
```

```
Set oextrude321 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(  
_ oProfile421, p21, kPositiveExtentDirection, kJoinOperation)
```

'part 5 flantza katw

Dim fk21 As Double

Dim fl21 As Double

fk21 = Cdbl(TextBox231.Text)

fl21 = Cdbl(TextBox241.Text)

fk21 = fk21 / 10

fl21 = fl21 / 10

fk21 = fk21 / 2

Dim oSketchf521 As PlanarSketch

Set oSketchf521 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(7))

Dim oSkPntsf521 As SketchPoints

Set oSkPntsf521 = oSketchf521.SketchPoints

Call oSkPntsf521.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf921 As SketchCircles

Set oCircsf921 = oSketchf521.SketchCircles

Dim oCircf921 As SketchCircle

Set oCircf921 = oCircsf921.AddByCenterRadius(oSkPntsf521(1), r21)

Dim oCircsf1021 As SketchCircles

Set oCircsf1021 = oSketchf521.SketchCircles

Dim oCircf1021 As SketchCircle

Set oCircf1021 = oCircsf1021.AddByCenterRadius(oSkPntsf521(1), fk21)

Dim oProfilef521 As Profile

Set oProfilef521 = oSketchf521.Profiles.AddForSolid

Set oextrudef521 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent(_
oProfilef521, fl21, kPositiveExtentDirection, kJoinOperation)

'part 5 flantza panw

Dim fm21 As Double

Dim fn21 As Double

Dim fo21 As Double

fn21 = Cdbl(TextBox271.Text)

fm21 = Cdbl(TextBox261.Text)

fm21 = fm21 / 10

fn21 = fn21 / 10

fo21 = a + j + j1 + j2 + p21

fn21 = fn21 / 2

Dim oWPlane321 As WorkPlane

Set oWPlane321 = oCompDef.WorkPlanes.AddByPlaneAndOffset(oCompDef.WorkPlanes("XZ Plane"), fo21)

Dim oSketchf621 As PlanarSketch

Set oSketchf621 =
oPartDoc.ComponentDefinition.Sketches.Add(oPartDoc.ComponentDefinition.WorkPlanes.Item(8))

Dim oSkPntsf621 As SketchPoints

Set oSkPntsf621 = oSketchf621.SketchPoints

Call oSkPntsf621.Add(oTG.CreatePoint2d(0, 0), False)

'Sketch a circle

Dim oCircsf1121 As SketchCircles

Set oCircsf1121 = oSketchf621.SketchCircles

Dim oCircf1121 As SketchCircle

Set oCircf1121 = oCircsf1121.AddByCenterRadius(oSkPntsf621(1), r21)

Dim oCircsf1221 As SketchCircles

Set oCircsf1221 = oSketchf621.SketchCircles

Dim oCircf1221 As SketchCircle

Set oCircf1221 = oCircsf1221.AddByCenterRadius(oSkPntsf621(1), fn21)

```
Dim oProfilef621 As Profile
```

```
Set oProfilef621 = oSketchf621.Profiles.AddForSolid
```

```
Set oextrudef621 = oCompDef.Features.ExtrudeFeatures.AddByDistanceExtent( _  
    oProfilef621, fm21, kNegativeExtentDirection, kJoinOperation)
```

```
oWPlane1.Visible = False
```

```
oWPlane21.Visible = False
```

```
oWPlane22.Visible = False
```

```
oWPlane2.Visible = False
```

```
oWPlane321.Visible = False
```

```
End Sub
```

Κώδικας του button «Κλείσιμο»

```
Private Sub CommandButton2_Click()
```

```
Unload Me
```

```
Unload UserForm14
```

```
Unload UserForm6
```

```
End Sub
```

Κώδικας του Useform 13

```
Private Sub UserForm_Activate()
```

```
Me.ScrollBars = fmScrollBarsVertical
```

```
Me.ScrollHeight = Me.InsideHeight * 4.2
```

```
End Sub
```

```
Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As  
MSForms.fmScrollAction, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal  
ActualDx As MSForms.ReturnSingle, ByVal ActualDy As MSForms.ReturnSingle)
```

```
TextBox7.Text = TextBox2.Text
```

```
TextBox71.Text = TextBox9.Text
```

```
TextBox72.Text = TextBox91.Text
```

```
TextBox691.Text = TextBox92.Text
```

```
End Sub
```