

## ΠΑΡΑΔΕΙΓΜΑ #1

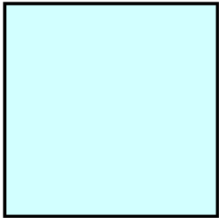
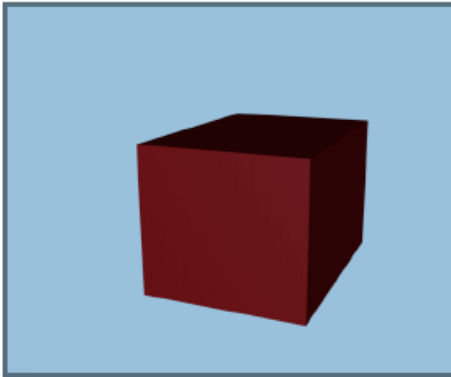
- Να δημιουργήσετε μια **κλάση**, η οποία να αντιπροσωπεύει έναν κύκλο.
  - Συγκεκριμένα, να αποτελείται από μια **μεταβλητή-μέλος (ιδιότητα)**, που θα αντιπροσωπεύει την (κινητής υποδιαστολής διπλής ακρίβειας) ακτίνα του, και
  - από μια **συνάρτηση-μέλος (μέθοδο)**, που θα υπολογίζει και επιστρέφει την περίμετρό του (κινητής υποδιαστολής διπλής ακρίβειας).
- Επιπλέον να δημιουργήσετε 2 **constructors**:
  - ο ένας θα είναι default, και θα αρχικοποιεί την ακτίνα στην τιμή 1.
  - ο δεύτερος, θα έχει μια παράμετρο σαν «είσοδο», η τιμή της οποίας θα αρχικοποιεί την ακτίνα.
- και έναν **destructor**, ο οποίος θα εμφανίζει το μήνυμα: "ΚΑΤΑΣΤΡΟΦΗΚΥΚΛΟΥ"

Να γράψετε ένα πρόγραμμα, στο οποίο:

- Να δημιουργήσετε 2 **αντικείμενα** της κλάσης κύκλου:
  - ένα χωρίς να καθορίσετε καμιά αρχική τιμή για την ακτίνα, και
  - ένα με αρχική τιμή ακτίνας το 4
- Στη συνέχεια, το πρόγραμμά σας να υπολογίζει την περίμετρο του καθενός κύκλου και να εμφανίζει τις περιμέτρους σε ξεχωριστές γραμμές με ανάλογα μηνύματα.

Προσέξτε ότι η άσκηση δεν ζητά να δώσει ο χρήστης τιμή για την ακτίνα του κύκλου, οπότε θα ισχύει η τιμή της ακτίνας που δίνουν οι κατασκευαστές (constructors) της κλάσης.

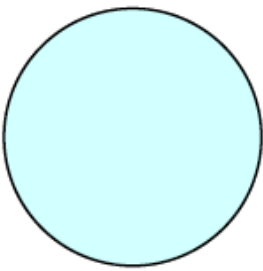
## ΠΑΡΑΔΕΙΓΜΑ #2

- Να δημιουργήσετε μια **κλάση**, η οποία να αντιπροσωπεύει ένα τετράγωνο. Συγκεκριμένα, να αποτελείται
  - από μια **μεταβλητή-μέλος (ιδιότητα)**, που θα αντιπροσωπεύει το μήκος της πλευράς του (κινητής υποδιαστολής διπλής ακρίβειας), η οποία να μπορεί να προσπελαστεί απ' ευθείας από τους απογόνους του και
  - από μια **συνάρτηση-μέλος (μέθοδο)**, που θα υπολογίζει και επιστρέφει την περίμετρό του.
  - Επιπλέον να δημιουργήσετε έναν **constructor**, ο οποίος θα έχει μια παράμετρο σαν «είσοδο», η τιμή της οποίας θα αρχικοποιεί το μήκος της πλευράς. Η παράμετρος να έχει εξ' ορισμού τιμή το 1.
  - και έναν **destructor**, ο οποίος θα εμφανίζει το μήνυμα: "ΚΑΤΑΣΤΡΟΦΗΤΕΤΡΑΓΟΝΟΥ"
- 
- 
- Να δημιουργήσετε μια ακόμη **κλάση**, η οποία να αντιπροσωπεύει ένα παραλληλεπίπεδο (ορθογώνιο) και να προέρχεται από την κλάση (να έχει σαν βασική της κλάση) την κλάση του τετραγώνου. Συγκεκριμένα, να αποτελείται
  - από μια **μεταβλητή-μέλος (ιδιότητα)**, που θα αντιπροσωπεύει το ύψος του (κινητής υποδιαστολής διπλής ακρίβειας), και
  - από μια **συνάρτηση-μέλος (μέθοδο)**, που θα υπολογίζει και θα επιστρέφει τον όγκο του (εμβαδό βάσης x ύψος).
  - Επιπλέον να δημιουργήσετε έναν **constructor**, ο οποίος θα έχει 2 παραμέτρους σαν «είσοδο», η 1<sup>η</sup> θα αρχικοποιεί το μήκος της πλευράς της βάσης του, και η 2<sup>η</sup> το ύψος του παραλληλεπίπεδου. Η 2<sup>η</sup> παράμετρος να έχουν εξ' ορισμού τιμή το 1.
  - και έναν **destructor**, ο οποίος θα εμφανίζει το μήνυμα: "ΚΑΤΑΣΤΡΟΦΗΠΑΡΑΛΛΗΛΕΠΙΠΕΔΟΥ"
- 

Να γράψετε ένα πρόγραμμα, στο οποίο:

- Να δημιουργήσετε ένα **αντικείμενο** της κλάσης παραλληλεπίπεδου, με αρχική τιμή μήκους πλευράς βάσης το 5 και αρχική τιμή του ύψους το 10.
- Στη συνέχεια, το πρόγραμμά σας να υπολογίζει (χωρίς ο χρήστης να δώσει καμία τιμή από το πληκτρολόγιο):
  - την περίμετρο της βάσης του παραλληλεπίπεδου καθώς και
  - τον όγκο του παραλληλεπίπεδουκαι να τα εμφανίζει στην οθόνη σε ξεχωριστές γραμμές με ανάλογα μηνύματα.

### ΠΑΡΑΔΕΙΓΜΑ #3

- Να δημιουργήσετε μια **κλάση**, η οποία να αντιπροσωπεύει έναν κύκλο. Συγκεκριμένα, να αποτελείται
    - από μια **μεταβλητή-μέλος (ιδιότητα)**, που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας), η οποία να μπορεί να προσπελαστεί απ' ευθείας από τους απογόνους του και
    - από μια **συνάρτηση-μέλος (μέθοδο)**, που θα υπολογίζει και επιστρέφει την περίμετρό του.
  - Επιπλέον να δημιουργήσετε έναν **constructor**, ο οποίος θα έχει μια παράμετρο σαν «είσοδο», η τιμή της οποίας θα αρχικοποιεί την ακτίνα. Η παράμετρος να έχει εξ' ορισμού τιμή το 1.
  - και έναν **destructor**, ο οποίος θα εμφανίζει το μήνυμα: "ΚΑΤΑΣΤΡΟΦΗΚΥΚΛΟΥ"
- 
- 
- Να δημιουργήσετε μια ακόμη **κλάση**, η οποία να αντιπροσωπεύει έναν κύλινδρο και να προέρχεται από την κλάση (να έχει σαν βασική της κλάση) την κλάση του κύκλου. Συγκεκριμένα, να αποτελείται
    - από μια **μεταβλητή-μέλος (ιδιότητα)**, που θα αντιπροσωπεύει το ύψος του (κινητής υποδιαστολής διπλής ακρίβειας),
    - από μια **μεταβλητή-μέλος (ιδιότητα)**, που να αντιπροσωπεύει την πυκνότητα του υλικού του (κινητής υποδιαστολής διπλής ακρίβειας), και
    - από μια **συνάρτηση-μέλος (μέθοδο)**, που θα υπολογίζει και επιστρέφει την μάζα του (όγκος x πυκνότητα), όπου  $\text{όγκος} = \text{εμβαδό βάσης} \times \text{ύψος}$ .
  - Επιπλέον να δημιουργήσετε έναν **constructor**, ο οποίος θα έχει 3 παραμέτρους σαν «είσοδο», η 1<sup>η</sup> θα αρχικοποιεί την ακτίνα, η 2<sup>η</sup> το ύψος και η 3<sup>η</sup> την πυκνότητα του κυλίνδρου. Η 2<sup>η</sup> και η 3<sup>η</sup> παράμετρος να έχουν εξ' ορισμού τιμή το 1.
  - και έναν **destructor**, ο οποίος θα εμφανίζει το μήνυμα: "ΚΑΤΑΣΤΡΟΦΗΚΥΛΙΝΔΡΟΥ"



Να γράψετε ένα πρόγραμμα, στο οποίο:

- Να δημιουργήσετε ένα **αντικείμενο** της κλάσης κυλίνδρου, με αρχική τιμή ακτίνας το 20, πυκνότητας το 2 και αρχική τιμή του ύψους το 5.
- Στη συνέχεια, το πρόγραμμά σας να υπολογίζει (χωρίς ο χρήστης να δώσει καμιά τιμή από το πληκτρολόγιο):
  - την περίμετρο της βάσης του κυλίνδρου καθώς και
  - την μάζα του κυλίνδρουκαι να τα εμφανίζει στην οθόνη σε ξεχωριστές γραμμές με ανάλογα μηνύματα.

#### ΠΑΡΑΔΕΙΓΜΑ #4

- Να δημιουργήσετε μια **κλάση κύκλου**, η οποία θα αποτελείται:
  - από μια **μεταβλητή-μέλος (ιδιότητα)** που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)
  - από έναν **constructor** χωρίς παραμέτρους, ο οποίος αρχικοποιεί την ακτίνα με την τιμή 1,
  - έναν **destructor**, που να μην κάνει τίποτα
  - από μια **συνάρτηση-μέλος (μέθοδο)** που να μπορεί να αλλάξει την τιμή της ακτίνας και να ονομάζεται `set_aktina` (θα δέχεται σαν "είσοδο" έναν κινητής υποδιαστολής διπλής ακρίβειας αριθμό, ο οποίος θα αποτελεί την τιμή της ακτίνας)
  - από μια **συνάρτηση-μέλος (μέθοδο)**, χωρίς παραμέτρους, που να επιστρέφει το εμβαδόν του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται `give_embado`.
- Να γράψετε ένα πρόγραμμα που:
  - να δηλώνει έναν **πίνακα (διάταξη) τεσσάρων αντικειμένων** τύπου κύκλου
  - κατόπιν, επαναληπτικά:
    - να ζητά 4 φορές από τον χρήστη την τιμή της ακτίνας ισάριθμων κύκλων
    - και μέσω της συνάρτησης-μέλους της κλάσης κύκλου να "περνά" αυτή την τιμή σε καθένα από τα 4 αντικείμενα.
  - Στη συνέχεια, να υπολογίζει επαναληπτικά και να εμφανίζει το εμβαδόν του κάθε αντικειμένου-κύκλου.

## ΠΑΡΑΔΕΙΓΜΑ #5

---

Να γράψετε ένα πρόγραμμα:

- Στο οποίο να δηλώσετε **έναν** δείκτη κατάλληλο να «δείξει» σε δεδομένα τύπου κινητής υποδιαστολής διπλής ακρίβειας
- Στη συνέχεια το πρόγραμμα να **ρωτήσει** τον χρήστη **πόσους** αριθμούς θέλει να προσθέσει και να **δεσμεύσει** μνήμη αρκετή για την αποθήκευσή τους, χρησιμοποιώντας τον παραπάνω δείκτη
- Έπειτα, να ζητά **επαναληπτικά** από τον χρήστη και να διαβάζει τους αριθμούς και να τους αποθηκεύει σε διαδοχικές θέσεις της δεσμευμένης μνήμης
- Τέλος να υπολογίζει το άθροισμά τους σε μια χωριστή μεταβλητή. Επιπλέον να το εμφανίζει στην οθόνη με ανάλογο μήνυμα
- Η δεσμευμένη **μνήμη να αποδεσμευτεί** πριν τελειώσει το πρόγραμμα.



## ΠΑΡΑΔΕΙΓΜΑ #6

Θέλουμε να δημιουργήσουμε **μια κλάση πολύπλευρου**. Προσέξτε ότι θέλουμε η κλάση να είναι «ελαστική», δηλαδή να δημιουργούμε από αυτή αντικείμενα (δηλαδή πολύπλευρα) με οσεσδήποτε πλευρές θέλουμε κάθε φορά. Δηλαδή ένα πολύπλευρό μας να είναι τετράπλευρο, ένα άλλο να είναι πεντάπλευρο, κλπ.

Επομένως, για κάθε πολύπλευρό μας θέλουμε να αποθηκεύονται 2 «πράγματα»: πόσες πλευρές έχει το πολύπλευρο και τα μήκη των πλευρών (κάθε πλευρά θα έχει το δικό της μήκος).

Για παράδειγμα, αν το πολύπλευρο έχει 4 πλευρές θα χρειαστεί να αποθηκεύσουμε 4 μήκη, αν έχει 5 πλευρές θα αποθηκεύσουμε 5 μήκη, κοκ. Καταλαβαίνετε ότι αφού το πλήθος των πλευρών δεν μας είναι εκ των προτέρων γνωστό θα χρησιμοποιήσουμε δείκτη και δυναμική δέσμευση μνήμης για να αποθηκεύσουμε τα μήκη των πλευρών.

Επομένως, η κλάση θα έχει 2 **μεταβλητές-μέλη (ιδιότητες)**:

- έναν ακέραιο που θα αντιπροσωπεύει τον αριθμό (το πλήθος) των πλευρών του και
- από έναν **δείκτη** τύπου `double` που θα χρησιμοποιούμε για να αποθηκεύσουμε τα μήκη των πλευρών (πραγματικοί).

Θέλουμε, όταν δημιουργούμε πολύπλευρα (δηλαδή αντικείμενα της κλάσης), αρχικά αυτά να μην έχουν καθόλου πλευρές (άρα πλήθος πλευρών ίσο με 0) και φυσικά ο δείκτης της κλάσης να έχει την τιμή `NULL`. Φτιάξτε έναν **κατασκευαστή** (constructor) που να κάνει τα παραπάνω.

Θέλουμε όταν ένα πολύπλευρο (δηλαδή αντικείμενο της κλάσης) διαγράφεται από την μνήμη να απελευθερώνεται η μνήμη στην οποία δείχνει ο δείκτης της κλάσης. Φτιάξτε έναν **καταστροφέα** (destructor) που κάνει αυτό.

Αφού όταν δημιουργούμε ένα νέο αντικείμενο (πολύπλευρο) αυτό δεν έχει αρχικά καθόλου πλευρές, θα πρέπει να μπορούμε να το αλλάξουμε αυτό μέσα στο πρόγραμμά μας καλώντας μια συνάρτηση-μέλος της κλάσης.

Δηλαδή, θέλουμε να υπάρχει μια συνάρτηση-μέλος με μια παράμετρο («είσοδο») που να αντιπροσωπεύει το πλήθος των πλευρών του πολύπλευρου.

Η συνάρτηση αυτή, όταν την καλούμε, **θα αλλάζει την τιμή της μεταβλητής-μέλους που καθορίζει το πλήθος των πλευρών** στην τιμή που γράψαμε μέσα στις παρενθέσεις της συνάρτησης.

Επιπλέον, **θα δεσμεύει μνήμη** για όσες πλευρές χρειάζεται, με τη βοήθεια του δείκτη της κλάσης.

Θέλουμε να μπορούμε **να αλλάζουμε το μήκος κάποιας πλευράς του πολύπλευρου**, καλώντας μια συνάρτηση-μέλος της κλάσης. Πώς θα το κάνουμε αυτό; Θα «λέμε» στο υπολογιστή: για την πλευρά «τάδε» κάνε το μήκος «τόσο».

Επομένως, να φτιάξετε μια συνάρτηση-μέλος που να έχει 2 παραμέτρους («εισόδους»). Η μία να είναι ο αύξοντας αριθμός της πλευράς (δηλαδή ποιας πλευράς θα αλλάξουμε το μήκος;) και η δεύτερη να είναι το μήκος της πλευράς αυτής. Η συνάρτηση να αλλάζει το μήκος της πλευράς, με τη βοήθεια του δείκτη της κλάσης.

Θέλουμε να μπορούμε **να βρούμε το μήκος της περιμέτρου** του πολύπλευρου μας, δηλαδή το άθροισμα των μηκών των πλευρών του. Να φτιάξετε μια συνάρτηση-μέλος που να υπολογίζει και να επιστρέφει το μήκος της περιμέτρου του πολύπλευρου.

Και τώρα θα δοκιμάσετε την κλάση που φτιάξατε, μέσα από ένα πρόγραμμα.

- Να γράψετε ένα πρόγραμμα που:
  - αρχικά να γίνεται δήλωση ενός **αντικείμενου** της κλάσης πολύπλευρου
  - να ζητά από τον χρήστη τον αριθμό (πλήθος) των πλευρών ενός πολύπλευρου και να τον διαβάσει, ώστε να κάνει την ανάλογη ρύθμιση στο παραπάνω αντικείμενο, καλώντας την ανάλογη συνάρτηση.
  - να ζητά και να διαβάσει επαναληπτικά τα μήκη των πλευρών (π.χ. της 1<sup>ης</sup> πλευράς, της 2<sup>ης</sup>, κλπ), ώστε να τα αποθηκεύει στο παραπάνω αντικείμενο, με τη βοήθεια της ανάλογης συνάρτησης-μέλους
  - να υπολογίζει και να εμφανίζει στην οθόνη την περίμετρο του πολύπλευρου

## ΠΑΡΑΔΕΙΓΜΑ #7

- Να δημιουργήσετε μια **κλάση κύκλου**, η οποία θα αποτελείται:
  - από μια **μεταβλητή-μέλος (ιδιότητα)** που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)
  - από έναν **constructor** με μια παράμετρο, η οποία θα καθορίζει την τιμή της ακτίνας. Η προκαθορισμένη τιμή της ακτίνας να είναι το 1,
  - έναν **destructor**, που να εμφανίζει απλώς το μήνυμα ότι ο κύκλος καταστράφηκε
  - από μια **συνάρτηση-μέλος (μέθοδο)**, με μία παράμετρο, που να αλλάζει την ακτίνα του κύκλου και να ονομάζεται `set_aktina`.
  - από μια **συνάρτηση-μέλος (μέθοδο)**, χωρίς παραμέτρους, που να επιστρέφει το εμβαδόν του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται `give_embado`.
- Να γράψετε ένα πρόγραμμα που:
  - να δημιουργεί ένα **νέο αντικείμενο** κύκλου (με αρχική τιμή ακτίνας 5) **δεσμεύοντας μνήμη**. Να χρησιμοποιήσετε **δείκτη** τύπου "κλάσης κύκλου".
  - στη συνέχεια, να ζητά από τον χρήστη μια νέα τιμή για την ακτίνα του κύκλου, να τη διαβάσει και να καλεί την κατάλληλη συνάρτηση της κλάσης ώστε να την αλλάξει (την ακτίνα) για το αντικείμενο που δημιούργησε παραπάνω
  - να υπολογίζει (καλώντας την κατάλληλη συνάρτηση της κλάσης) και να εμφανίζει το εμβαδόν του κύκλου
  - **να αποδεσμεύει** τη δεσμευμένη μνήμη.

## ΠΑΡΑΔΕΙΓΜΑ #8

Να δημιουργήσετε μια *κλάση κύκλου*, η οποία θα αποτελείται:

- από μια **μεταβλητή-μέλος (ιδιότητα)** που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)
- από έναν **constructor** με μια παράμετρο, η οποία θα καθορίζει την τιμή της ακτίνας. Η προκαθορισμένη τιμή της ακτίνας να είναι το 1,
- έναν **destructor**, που να μην κάνει τίποτα. Θα ρωτήσετε: «Να μην κάνει τίποτα; Τότε γιατί να τον φτιάξουμε;» Και η απάντηση είναι: «Μα για να δείξετε ότι ξέρετε πως φτιάχνουμε έναν destructor!»
- από μια **συνάρτηση-μέλος (μέθοδο)**, χωρίς παραμέτρους, που να επιστρέφει το εμβαδόν του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται `give_embado`.
- από μια **συνάρτηση υπερφόρτωσης του τελεστή \***, η οποία θα αναλαμβάνει να αυξήσει ή μειώσει την ακτίνα του κύκλου κατά το δεκαδικό ποσοστό που θα καθορίσει ο χρήστης. Π.χ. με την εντολή:  
$$\text{Κύκλος2} = \text{Κύκλος1} * n;$$
ο Κύκλος2 θα πρέπει να έχει ακτίνα  $n$ -πλάσια από τον Κύκλο1. *Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.*
- από μια **συνάρτηση υπερφόρτωσης του τελεστή +**, η οποία θα αναλαμβάνει να προσθέσει τις ακτίνες δύο κύκλων. Π.χ. με την εντολή:  
$$\text{Κύκλος3} = \text{Κύκλος1} + \text{Κύκλος2};$$
ο Κύκλος3 θα πρέπει να έχει ακτίνα ίση με το άθροισμα των ακτίνων του Κύκλος1 και του Κύκλος2. *Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.*

Να γράψετε ένα πρόγραμμα που:

- να δημιουργεί 3 αντικείμενα της κλάσης κύκλου, το ένα με αρχική τιμή ακτίνας 10, ενώ τα άλλα δύο χωρίς αρχική τιμή ακτίνας (οπότε θα πάρει την προκαθορισμένη 1)
- να υπολογίζει και εμφανίζει το εμβαδόν του 1<sup>ου</sup> κύκλου.
- Έπειτα, να ρωτά τον χρήστη το ποσοστό αύξησης ή μείωσης της ακτίνας και να το διαβάζει
- να **πολλαπλασιάζει** τον 1<sup>ο</sup> κύκλο επί το ποσοστό αύξησης/μείωσης και να αποθηκεύει το αποτέλεσμα στον 2<sup>ο</sup> κύκλο
- να υπολογίζει και να εμφανίζει το εμβαδόν του 2<sup>ου</sup> κύκλου.
- να **προσθέτει** τον 1<sup>ο</sup> κύκλο με τον 2<sup>ο</sup> κύκλο και να αποθηκεύει το αποτέλεσμα στον 3<sup>ο</sup> κύκλο.
- να υπολογίζει και να εμφανίζει το εμβαδόν του 3<sup>ου</sup> κύκλου.



## ΠΑΡΑΔΕΙΓΜΑ #9

- Να δημιουργήσετε μια **κλάση κύκλου**, η οποία θα αποτελείται:
  - από μια **μεταβλητή-μέλος (ιδιότητα)** που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)
  - από έναν **constructor** με μια παράμετρο, η οποία θα καθορίζει την τιμή της ακτίνας. Η προκαθορισμένη τιμή της ακτίνας να είναι το 1,
  - έναν **destructor**, που να μην κάνει τίποτα. Θα ρωτήσετε: «Να μην κάνει τίποτα; Τότε γιατί να τον φτιάξουμε;» Και η απάντηση είναι: «Μα για να δείξετε ότι ξέρετε πως φτιάχνουμε έναν destructor!»
  - από μια **συνάρτηση-μέλος (μέθοδο)**, χωρίς παραμέτρους, που να επιστρέφει το εμβαδόν του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται `give_embado`.
  - από μια, φιλική προς την κλάση κύκλου, **συνάρτηση υπερφόρτωσης του τελεστή \***, η οποία θα αναλαμβάνει να αυξήσει ή μειώσει την ακτίνα του κύκλου κατά το δεκαδικό ποσοστό που θα καθορίσει ο χρήστης. Π.χ. με την εντολή:  
$$\text{Κύκλος2} = \text{Κύκλος1} * n;$$
ο Κύκλος2 θα πρέπει να έχει ακτίνα n-πλάσια από τον Κύκλο1.
  - από μια, φιλική προς την κλάση κύκλου, **συνάρτηση υπερφόρτωσης του τελεστή +**, η οποία θα αναλαμβάνει να προσθέσει τις ακτίνες δύο κύκλων. Π.χ. με την εντολή:  
$$\text{Κύκλος3} = \text{Κύκλος1} + \text{Κύκλος2};$$
ο Κύκλος3 θα πρέπει να έχει ακτίνα ίση με το άθροισμα των ακτίνων του Κύκλος1 και του Κύκλος2.
- Να γράψετε ένα πρόγραμμα που:
  - να δημιουργεί 3 αντικείμενα της κλάσης κύκλου, το ένα με αρχική τιμή ακτίνας 10, ενώ τα άλλα δύο χωρίς αρχική τιμή ακτίνας (οπότε θα πάρει την προκαθορισμένη 1)
  - να υπολογίζει και εμφανίζει το εμβαδόν του 1<sup>ου</sup> κύκλου.
  - Έπειτα, να ρωτά τον χρήστη το ποσοστό αύξησης ή μείωσης της ακτίνας και να το διαβάσει
  - να **πολλαπλασιάζει** τον 1<sup>ο</sup> κύκλο επί το ποσοστό αύξησης/μείωσης και να αποθηκεύει το αποτέλεσμα στον 2<sup>ο</sup> κύκλο
  - να υπολογίζει και να εμφανίζει το εμβαδόν του 2<sup>ου</sup> κύκλου.
  - να **προσθέτει** τον 1<sup>ο</sup> κύκλο με τον 2<sup>ο</sup> κύκλο και να αποθηκεύει το αποτέλεσμα στον 3<sup>ο</sup> κύκλο.
  - να υπολογίζει και να εμφανίζει το εμβαδόν του 3<sup>ου</sup> κύκλου.

## ΠΑΡΑΔΕΙΓΜΑ #10

- Να δημιουργήσετε μια *κλάση παραλληλογράμμου*, η οποία θα αποτελείται:
  - από δύο μεταβλητές-μέλη (*ιδιότητες*) που θα αντιπροσωπεύουν το μήκος και το πλάτος του (κινητής υποδιαστολής διπλής ακρίβειας)
  - από έναν **constructor** με δύο παραμέτρους, η οποίες θα καθορίζουν τις τιμές του μήκους και του πλάτους
  - από μια **συνάρτηση υπερφόρτωσης του τελεστή +**, η οποία θα αναλαμβάνει να προσθέσει τα μήκη δύο παραλληλογράμμων μεταξύ τους και τα πλάτη μεταξύ τους. Π.χ. με την εντολή:  
 $\text{Παραλληλόγραμμο3} = \text{Παραλληλόγραμμο1} + \text{Παραλληλόγραμμο2};$   
το Παραλληλόγραμμο3 θα πρέπει να έχει μήκος ίσο με το άθροισμα των μηκών του Παραλληλόγραμμο1 και του Παραλληλόγραμμο2. Το ίδιο να ισχύει για το πλάτος. **Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.**
  - από μια **συνάρτηση υπερφόρτωσης του τελεστή ==**, η οποία θα αναλαμβάνει να συγκρίνει τα μήκη δύο παραλληλογράμμων μεταξύ τους και τα πλάτη μεταξύ τους. Π.χ. με την εντολή:  
 $\text{Παραλληλόγραμμο1} == \text{Παραλληλόγραμμο2}$   
στην περίπτωση που τα μήκη και πλάτη των δύο παραλληλογράμμων είναι ίσα, να επιστρέφει «αληθές» αλλιώς να επιστρέφει «ψευδές». **Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.**
  - από μια, φιλική προς την κλάση κύκλου, **συνάρτηση υπερφόρτωσης του τελεστή >>**, η οποία θα αναλαμβάνει να διαβάσει από το πληκτρολόγιο τις διαστάσεις ενός παραλληλογράμμου στη μορφή: μήκος x πλάτος. Π.χ.: 3 x 4
  - από μια, φιλική προς την κλάση κύκλου, **συνάρτηση υπερφόρτωσης του τελεστή <<**, η οποία θα αναλαμβάνει να εμφανίσει στην οθόνη τις διαστάσεις ενός παραλληλογράμμου στη μορφή: μήκος x πλάτος. Π.χ.: 3 x 4
- Να γράψετε ένα πρόγραμμα που:
  - να δημιουργεί 3 αντικείμενα της κλάσης παραλληλογράμμου, όλα με αρχική τιμή μήκους 0 και πλάτους 0.
  - να ζητά από τον χρήστη να δώσει νέες διαστάσεις για το 1<sup>ο</sup> και το 2<sup>ο</sup> παραλληλόγραμμο
  - να προσθέτει το 1<sup>ο</sup> παραλληλόγραμμο με το 2<sup>ο</sup> παραλληλόγραμμο και να αποθηκεύει το αποτέλεσμα στο 3<sup>ο</sup> παραλληλόγραμμο.
  - να εμφανίζει τις διαστάσεις του 3<sup>ου</sup> παραλληλόγραμμου στην οθόνη
  - να συγκρίνει το 1<sup>ο</sup> παραλληλόγραμμο με το 2<sup>ο</sup> παραλληλόγραμμο για ισότητα και να εμφανίζει μήνυμα για το αν είναι ίδια ή όχι

## ΠΑΡΑΔΕΙΓΜΑ #11

- Να δημιουργήσετε μια *κλάση κύκλου*, η οποία θα αποτελείται:
  - από μια *μεταβλητή-μέλος (ιδιότητα)* που θα αντιπροσωπεύει την ακτίνα του (κινητής υποδιαστολής διπλής ακρίβειας)
  - από έναν *constructor* με μια παράμετρο, η οποία θα καθορίζει την τιμή της ακτίνας.
  - από μια *συνάρτηση-μέλος (μέθοδο)*, χωρίς παραμέτρους, που να επιστρέφει την ακτίνα του κύκλου (κινητής υποδιαστολής διπλής ακρίβειας) και να ονομάζεται `give_aktina`.
  - από μια *συνάρτηση υπερφόρτωσης του τελεστή προθέματος ++*, η οποία θα αναλαμβάνει να αυξήσει την ακτίνα ενός κύκλου κατά 1. Π.χ. με την εντολή:  
`Κύκλος2 = ++Κύκλος1;`  
ο `Κύκλος1` θα πρέπει να έχει ακτίνα αυξημένη κατά 1. Την ίδια τιμή ακτίνας πρέπει να έχει και ο `Κύκλος3`. *Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.*
  - από μια *συνάρτηση υπερφόρτωσης του τελεστή επιθέματος ++*, η οποία θα αναλαμβάνει να αυξήσει την ακτίνα ενός κύκλου κατά 1. Π.χ. με την εντολή:  
`Κύκλος 3 = Κύκλος2++;`  
ο `Κύκλος2` θα πρέπει να έχει ακτίνα αυξημένη κατά 1. Όμως ο `Κύκλος3` πρέπει να έχει την τιμή ακτίνας που είχε ο `Κύκλος2` πριν την αύξηση. *Η συνάρτηση αυτή να είναι συνάρτηση-μέλος της κλάσης.*
- Να γράψετε ένα πρόγραμμα που:
  - να δημιουργεί 3 αντικείμενα της κλάσης κύκλου, το ένα με αρχική τιμή ακτίνας 10, το δεύτερο με αρχική τιμή ακτίνας 10 επίσης, ενώ το τρίτο με αρχική τιμή ακτίνας 0.
  - να αυξάνει τον 1<sup>ο</sup> κύκλο κατά 1, με τον τελεστή *προθέματος ++*, και να αποθηκεύει το αποτέλεσμα στον 3<sup>ο</sup> κύκλο.
  - να εμφανίζει την ακτίνα του 1<sup>ου</sup> και του 3<sup>ου</sup> κύκλου.
  - να αυξάνει τον 2<sup>ο</sup> κύκλο κατά 1, με τον τελεστή *επιθέματος ++*, και να αποθηκεύει το αποτέλεσμα στον 3<sup>ο</sup> κύκλο.
  - να εμφανίζει την ακτίνα του 2<sup>ου</sup> και του 3<sup>ου</sup> κύκλου.