



JAVA

1 SET JAVA 2020

Γιώργος Ντάκος\\A.Μ:1059569///Ετος:Γ'
29/3/2020

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ(JAVA)

Άσκηση1:

Ερώτηση 3:

Μετά την ολοκλήρωση των 2 προηγούμενων ερωτημάτων και τρέχοντας την κλάση Age αυτό που παρατηρούμε είναι ότι οι εντολές `System.out.println()` μέσα στην μέθοδο `main` με όρισμα έναν πίνακα τύπου `string` εκτελούνται(αφού η συνθήκη `if` ισχύει.)και τυπώνει τα αποτελέσματα που έχει μέσα στα εισαγωγικά και αυτά που βάζουμε εμείς ως είσοδο(δηλαδή το νούμερο της ηλικίας χάρης στο `args[0]` που δηλώνει το πρώτο στοιχείο του πίνακα `args`).Πιο αναλυτικά η `System` είναι μια βασική κλάση που περιέχει αντικείμενα και μεταβλητές για την εισαγωγή δεδομένων και για έξοδο χαρακτήρων στην οθόνη. Το `out` είναι ένα αντικείμενο που δηλώνει την έξοδο. Και η μέθοδος `println()` του αντικείμενου `out` τυπώνει στην οθόνη την φράση που περιέχουν μέσα οι αγκύλες.

Όταν δεν δώσουμε κάποια τιμή στον πίνακα `args` η εντολή `if` δεν ικανοποιείται αφού ο πίνακας δεν έχει πάρει κάποια τιμή οπότε το μήκος του είναι 0.Αρα το πρόγραμμα παραλείπει τις εντολές μέσα στο `if` και δεν κάνει απολύτως τίποτα.

Τα αποτελέσματα:

```
You are
12
  years old.
You are
13
  years old.
You are
14
  years old.
```

Ερώτηση 4:

Κάνοντας τις αλλαγές στον κώδικα παρατηρούμε τα δεδομένα μέσα στις εντολές `System.out.println()` παρουσιάζονται στο παραθύρου εξόδου σε μια γραμμή και όχι σε τρεις την μια κάτω από την άλλη. Αυτό συμβαίνει προφανώς γιατί η `print()` είναι διαφορετική από την `println()`. Το πρόθεμα `ln` σημαίνει ότι αφού τυπώσει τα δεδομένα που βρίσκονται μέσα στις παρενθέσεις θα τυπωθούν και θα αλλάξει γραμμή ενώ χωρίς το `ln` τυπώνονται απλώς τα αποτελέσματα.

Τα αποτελέσματα:

```
You are 12 years old.
```

Ερώτηση 5:

Τώρα χάρη στην εντολή `else` άμα δεν δώσουμε κάποιο όρισμα δεν θα ικανοποιηθεί η δομή `if` πάλι αλλά θα ικανοποιηθεί η δομή `else` και θα τυπώσει τα δεδομένα της `System.out.println()` που βρίσκονται μέσα στην παρένθεση.

Τα αποτελέσματα:

```
You forgot to give your age.
```

Ερώτηση 6:

Τώρα έχοντας δώσει 3 δομές συνθήκης μια `if` μια `else if` και μια `else` ανάλογα με το ποια ικανοποιείται θα τυπώνεται και το ανάλογο αποτέλεσμα. Οπότε βάζοντας 2 ορίσματα θα ικανοποιηθεί ο όρος `else`. Βάζοντας 1 όρισμα θα ικανοποιηθεί ο όρος `if` ενώ αν δεν βάλουμε κανένα όρισμα θα ικανοποιηθεί το `else if`.

Τα αποτελέσματα:

2 ορίσματα: `giorgos is 15 years old.`

1 όρισμα: `You are 13 years old.`

Κανένα όρισμα: `You forgot to give your age.`

Ερώτημα 7:

Για να δίνουμε ως πρώτο όρισμα το όνομα πρέπει να αλλάξουμε τα στοιχεία του πίνακα που βρίσκονται μέσα στις εντολές `System.out.println()`. Δηλαδή στην πρώτη εντολή `System` πρέπει να βάλουμε `args[0]` που είναι το 1^ο στοιχείο του πίνακα `args` και στο 3^ο `System` πρέπει να βάλουμε ως δεδομένα το `args[1]` που είναι το δεύτερο στοιχείο του πίνακα `args`. (Ο κώδικας με τις αλλαγές θα παρατεθεί στο τέλος στο παράρτημα.)

Ερώτημα 8:

Χάρη στο πολυμορφισμό που είναι χαρακτηριστικό του αντικειμενοστρεφούς προγραμματισμού ο τελεστής της πρόσθεσης (+) χρησιμοποιείται και για την πρόσθεση 2 αριθμών αλλά και για την ένωση 2 αλφαριθμητικών.

Τα αποτελέσματα:

```
You forgot to give your age.  
You are 21 years old.  
Giorgos is 21 years old.
```

Άσκηση 2:

Μετά την ολοκλήρωση των 2 προηγούμενων ερωτημάτων και τρέχοντας την κλάση Window αυτό που παρατηρούμε είναι ότι τυπώνονται τα δεδομένα που έχουμε βάλει στις println() που βρίσκονται μέσα στη μέθοδο main(). Δηλώνοντας ως private την εντολή size καταφέρνουμε να χρησιμοποιηθεί μόνο από το ίδιο το αντικείμενο και να μην προσπελαστεί από πουθενά έξω από αυτό. Τα Window() και Window(int x) είναι οι λεγόμενοι κατασκευαστές, δηλαδή κάποιες ειδικές μέθοδοι. Έχουν πάντα το ίδιο όνομα με το όνομα της κλάσης μας και τον χρησιμοποιούμε για να δώσει αρχικές τιμές σε μεταβλητές ενός νέου αντικειμένου και να κάνει όποιες άλλες αλλαγές χρειαστεί για να δημιουργηθεί σωστά το αντικείμενο μας. Αυτό που διαφοροποιεί τον κατασκευαστή από την μέθοδο είναι ότι 1^ο έχει το ίδιο όνομα με την κλάση μας και 2^ο δεν επιστρέφει ποτέ τιμή και δεν έχει τύπο επιστροφής όπως γίνεται με τις μεθόδους (συμπεριλαμβάνεται και ο τύπος void). Ο κατασκευαστής Window() δεν κάνει τίποτα απολύτως. Αντίθετα ο κατασκευαστής Window(int x) έχει ένα όρισμα τύπου int και αυτό που κάνει είναι να δίνει τιμές στην μεταβλητή size(size=x). Η getSize είναι μια μέθοδος που μας επιστρέφει την τιμή size. Η εντολή Window wi = new Window() (με i={1,2,3,4}) δηλώνει τη μεταβλητή wi που αποτελεί μια αναφορά προς το αντικείμενο της κλάσης Window και δημιουργεί ένα αντικείμενο της κλάσης αυτής.

Τα αποτελέσματα:

```
The size of window w1 is: 5  
The size of window w2 is: 4  
The size of window w3 is: 6  
The size of window w4 is: 5
```

Ερωτηση 4:

Πρώτα πόλλα η λέξη κλειδί `this` αναφέρεται στο τρέχον αντικείμενο για το οποίο έχει κληθεί μια μέθοδος και κάθε μέθοδος στιγμιοτύπου διαθέτει μια αναφορά με την λέξη αυτή. Έτσι αν και έχουμε πολλά αντικείμενα που δημιουργούνται από μια κλάση, κάθε φορά στην μνήμη υπάρχει μόνο ένα αντίγραφο της μεθόδου. Οπότε με τη λέξη `this`, έχουμε την δυνατότητα να κάνουμε την μέθοδο να εργάζεται για διαφορετικό αντικείμενο κάθε φορά, με αυτό που βρίσκεται εκείνη την στιγμή στην μνήμη. Δηλαδή η λέξη `this` στην ουσία αποτελεί έναν δείκτη αντικειμένων. Τρέχοντας το πρόγραμμα μας παρατηρούμε ότι η έξοδος δεν έχει αλλάξει και αυτό συμβαίνει διότι στην μνήμη δείχνει είδη στα αντικείμενα που δημιουργήσαμε. Οπότε η εισαγωγή της εντολής `this` δεν έπαιξε κάποιο ρόλο.

Τα αποτελέσματα:

```
The size of window w1 is: 5  
The size of window w2 is: 4  
The size of window w3 is: 6  
The size of window w4 is: 5
```

Ερώτηση 5:

Αλλάζουμε τον κώδικα όπως μας ζητάει η άσκηση. Τρέχουμε το πρόγραμμα και βλέπουμε ότι τα δυο αντικείμενα που δημιουργούμε αντί να πάρουν size=5 και size=4 παίρνουν και τα δυο την τιμή size=10. Αυτό συμβαίνει διότι στην μέθοδο `getSize()` έχουμε αλλάξει την τιμή του `size=10`.

Τα αποτελέσματα:

```
The size of window w1 is: 10  
The size of window w2 is: 10
```

Ερώτηση 6:

Κάνουμε τις αλλαγές που πρέπει και τρέχουμε το πρόγραμμα. Αυτή τη φορά βλέπουμε ότι οι τιμές size των αντικείμενων `w1, w2` είναι size=5 και size=4 αντίστοιχα. Αυτό συμβαίνει διότι με την `return this.size` επιστρέφουμε την τιμή size των αντικειμένων που δείχνει η μνήμη εκείνη την στιγμή και αυτά είναι αυτά που δημιουργήθηκαν με τις εντολές

```
Window w1 = new Window();
```

```
Window w2 = new Window(4);
```

Οπότε και γι' αυτό παίρνουμε και τις τιμές αυτές των size

Τα αποτελέσματα:

```
The size of window w1 is: 5  
The size of window w2 is: 4
```

Άσκηση 3:

Ερώτηση 2α:

Αφού γράψουμε τους κώδικες και τους εκτελέσουμε έχουμε τα εξής: Πρώτα από όλα οι μεταβλητές και οι μέθοδοι που έχουν χαρακτηριστεί ως `protected` σε μια κλάση μπορούν να προσπελαστούν από όλες τις κλάσεις που βρίσκονται στο ίδιο πακέτο ή από τις υποκλάσεις αυτή της κλάσης. Οπότε η εντολή `size` είναι προσπελάσιμη από την υποκλάση `MWindow` (λόγο της λέξης `extends` η κλάση `MWindow` είναι υποκλάση της `Window`). Οι κατασκευαστές `Window()` και `Window(int size)` κάνουν τα εξής :Ο 1^{ος} από αυτούς δεν έχει κάποιο όρισμα, θέτει την μεταβλητή `size` του αντικειμένου ίση με 1 και τυπώνει ένα μήνυμα μέσω της `println`. Ο 2^{ος} έχει όρισμα μια ακέραια μεταβλητή `size`. Όταν καλεστεί θέτει την τιμή της `size` ίση με αυτή που έχουμε δηλώσει εμείς στην `RunWindow` στο αντικείμενο που βρίσκεται στην μνήμη εκείνη την στιγμή και επίσης τυπώνει ένα μήνυμα. Οπότε εκτελώντας την `RunWindow`

θα κατασκευαστούν 2 αντικείμενα `w1, w2` το καθένα από τον 1^ο και 2^ο κατασκευαστή αντίστοιχα. Οπότε στην έξοδο θα τυπωθούν πρώτα τα μηνύματα των `println` που βρίσκονται μέσα στις `Window()`, `Window(int size)` και μετά θα τυπωθούν τα δεδομένα των `println` που βρίσκονται στην `main`

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
```

Ερώτηση 2β:

Πρώτα από όλα αφού η κλάση MWindow είναι υποκλάση της Window κληρονομεί όλες τις μεταβλητές και τις μεθόδους της υπερκλάσης MWindow(προφανώς όχι όσες έχουν δηλωθεί ως private). Στην MWindow έχουμε δηλώσει 2 κατασκευαστές ο πρώτος με ένα όρισμα τύπου String και ο άλλος κατασκευαστής με 2 ορίσματα ο ένας τύπου int και ο άλλος τύπου String .Ο 1ος κατασκευαστής όταν καλεστεί θέτει την τιμή size=2 και τυπώνει ένα μήνυμα σύμφωνα με το όρισμα message που θα δηλώσουμε εμείς. Επιπλέον η μεταβλητή message στην κλάση MWindow είναι δηλωμένη ως private οπότε μπορεί να προσπελαστεί μόνο από ίδιο αντικείμενο και πουθενά έξω από αυτό. Ο 2^{ος} κατασκευαστής με την εντολή super(size) καλεί τον κατασκευαστή της υπερκλάσης και τυπώνει ένα μήνυμα σύμφωνα με την είσοδο που θα βάλουμε εμείς στην message στο αντικείμενο που βρίσκεται στην μνήμη μας εκείνη την στιγμή. Οπότε τρέχουμε την RunWindow και παίρνουμε τα ίδια αποτελέσματα με το προηγούμενο ερώτημα και μετά κατασκευάζει 2 αντικείμενα της MWindow με τα ανάλογα ορίσματα, τυπώνει αυτά που πρέπει σύμφωνα με όλα τα παραπάνω και αυτά που έχει κληρονομήσει (προφανώς εννοούμε την MWindow) από την υπερκλάση της και στην συνέχεια χάρη στις println μέσα στην main τυπώνονται οι τιμές των size των αντικειμένων που έχουν οριστεί από εμάς και τους ίδιους τους κατασκευαστές της MWindow.

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
3
```

Ερώτηση 2γ:

Εμπλουτίζουμε τον κώδικα της main με τις 3 τελευταίες εντολές και τον τρέχουμε. Όπως βλέπουμε στην μέσα οι 3 τελευταίες εντολές καλούν τις μεθόδους setSize1() setSize2() της κλάσης MWindow και την μέθοδο setSize() της κλάσης Window. Πρώτα από όλα η μέθοδος setSize1() είναι τύπου void έχει ένα όρισμα τύπου int. Η λειτουργία της είναι ότι θέτει την τιμή της size σύμφωνα με το όρισμα γ που θα δηλώσουμε εμείς. Η setSize2() είναι τύπου void και αυτή και έχει ένα όρισμα z τύπου ακέραιου. Αυτό που κάνει αυτή η μέθοδος είναι ότι καλεί τη μέθοδο setSize() της υπερκλάσης Window και θέτει το x=z οπότε θα πάρει την τιμή που θα δηλώσουμε εμείς. Άρα σύμφωνα με την setSize η νέα τιμή της size θα ισούται με το άθροισμα της τωρινής μεταβλητής και της μεταβλητής z. Η μέθοδος setSize() παίρνει μια τιμή x που την ορίζουμε εμείς. Οπότε η νέα τιμή της size θα είναι ίση με το άθροισμα της τωρινής size και της x

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
3
4
6
8
```

Ερώτηση 3:

Κάνουμε τις κατάλληλες αλλαγές και τρέχουμε την main. Οι αλλαγές έγιναν διότι δηλώνοντας ως protected τη μεταβλητή size μπορεί να προσπελαστεί μόνο από όλες τις κλάσεις που βρίσκονται στο ίδιο πακέτο ή από τις υποκλάσεις της. Οπότε αφού η Window είναι η υπερκλάση της δεν μπορεί να την προσπελάσει. Οπότε ο κατασκευαστής Mwindow(int size, String message) αντί να δώσει size=3 λόγω της super, μας δίνει size=7. Μετα για τις μεθόδους την δηλώνουμε εμείς ως 4 στην μέθοδο setSize1(). Στις άλλες 2 όμως παίρνει 4 διότι έχουμε ορίσει το size=2 στο κατασκευαστή και με την εντολή protected δεν μπορεί να προσπελαστεί οπότε στην 2^η μέθοδο αντί να γίνει size=4+2 γίνεται size=2+2. Το ίδιο και για την τρίτη μέθοδο.

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
4
4
4
```

Ερώτηση 4:

Κάνουμε τις απαραίτητες αλλαγές και το τρέχουμε. Πρώτα από όλα αυτό που κάνουμε στις 4 τελευταίες εντολές της main είναι να καλέσουμε 4 μεθόδους που θα εκτελεστούν για τα 2 αντικείμενα μας w1,mw1. Το w1 εκτελείτε μόνο από την μέθοδο της Window printSize() η οποία μας τυπώνει το size του αντικειμένου μας εκείνη την στιγμή. Τώρα η μέθοδο της MWindow η printSize() κάνει και αυτή περίπου το ίδιο πράγμα με την printSize της Window άλλα το κάνει για το αντικείμενο mw1 και τυπώνει ένα λίγο διαφορετικό μήνυμα. Τώρα η μέθοδος printSize1() τυπώνει απλά το size του αντικειμένου που κληρωνομεί από την Window λόγω της εντολής super.size. Η μέθοδος printSize2() κάνει το ίδιο πράγμα με την printSize() Window χάρης στην εντολή super.printSize() που στην ουσία καλεί την μέθοδο που έχει οριστεί στην Window(υπερκλάση). Οπότε τυπώνει το size του αντικειμένου mw1 σύμφωνα με την τιμή size που παίρνει λόγω της Window.

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
4
4
4
Size=1
MSize=4
5
Size=5
```

Ερώτηση 5:

Κάνουμε τις αλλαγές μας και τρέχουμε. Παρατηρούμε μόνο μια αλλαγή στα αποτελέσματα και αυτό γιατί οι εντολές που αλλάξαμε δηλαδή τις `((Window)this).size` και `((Window)this).printSize()` τις χρησιμοποιούμε γιατί θέλουμε να αναφερθούμε στα στιγμιότυπα της υπερκλάσης.

Τα αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
4
4
4
Size=1
MSize=4
5
MSize=4
```

JAVA SET 1:

ΜΕΡΟΣ Α΄

Ερώτηση 1: Σωστές απαντήσεις είναι το C και το B και αυτό γιατί η main είναι μια μέθοδος με όρισμα έναν πίνακα οποιουδήποτε ονόματος θέλουμε εμείς τύπου String.

Ερώτηση 2: Η λάθος δήλωση είναι η C και αυτό γιατί έχουμε δηλώσει την starAmmount ως int και την βάζουμε να ισούται με τιμή double οπότε ο compiler μας πετάει σφάλμα για να αποφύγει κάποια στρογγυλοποίηση που δεν θέλουμε.

Ερώτηση 3: Οι δηλώσεις που δημιουργούν error είναι η A,D

Η Α γιατί δεν γίνεται να δηλώσουμε το πόσα στοιχεία θέλουμε να έχει ο πίνακας αν δεν δεσμεύσουμε κάποιο χώρο μνήμης που θα περιέχει τα στοιχεία του. Το D είναι λάθος γιατί η `boolean` είναι δεσμευμένη λέξη και μια μεταβλητή δεν μπορεί να πάρει το όνομα μιας δεσμευμένης λέξης. Αν ήταν δηλωμένο ως `char Boolean` τότε η πρόταση θα ήταν σωστή και αυτό γιατί η `java` ξεχωρίζει τα κεφαλαία από τα πεζά γράμματα.

Ερώτηση 4: Οι μεταβλητές που δεν είναι έγκυρες είναι η Β γιατί το όνομα μιας μεταβλητής δεν γίνεται να ξεκινάει με αριθμό και η D γιατί ένα όνομα μεταβλητής δεν γίνεται να έχει κενά μεταξύ των χαρακτήρων του.

Ερώτηση 5: Σωστή απάντηση είναι το C και αυτό γιατί στο `z=++x*++y` πρώτα αυξάνεται το `x` κατά 1 μετά αυξάνεται το `y` κατά 1 και γίνεται ο πολλαπλασιασμός μεταξύ των 2 αριθμών. Τα υπόλοιπα βγαίνουν για προφανείς λόγους.

Ερώτηση 6: Σωστή απάντηση είναι το A και αυτό γιατί στην εντολή `k=a*b +--a` γίνεται πρώτα ο πολλαπλασιασμός του `a` με το `b` μετά μειώνει κατά 1 την τιμή του `a` και την προσθέτει στο γινόμενο.

ΜΕΡΟΣ Β΄

Ερώτημα 1: Η μέθοδος `calculate()` αυτό που κάνει είναι να επιστρέφει την τιμή του όγκου ενός ορθογώνιου παραλληλόγραμμου η οποία είναι τύπου `double`.

Ερώτημα 2:Απάντηση στο κώδικα

Ερώτημα 3:Απάντηση στο κώδικα

Ερώτημα 4: Το πρόβλημα που παρατηρούμε είναι η μέθοδος αυτή όταν εκτελεστεί θα επιστρέψει μια τιμή τύπου `String` και μέσα στο σώμα της εμείς έχουμε δώσει η τιμή να είναι τύπου `double` οπότε ο `compiler` μας πετάει σφάλμα.

Ερώτημα 5: Βάζουμε `public String toString` στην κλάση `Koyti` γιατί δεν μπορεί να κάνει `override` (δηλαδή να ακυρώσει την μέθοδο της κλάσης `Object` (είναι η υπερκλάση των κλάσεων μας από την στιγμή που δεν δηλώνουμε κάποια άλλη) που ανήκει στο πακέτο `java.lang`). Τώρα το πρόβλημα λύνεται διότι η τιμή μέσα στην μέθοδο θα επιστραφεί όντως σαν `String` αφού δηλώνεται ως `String` χωρίς κανένα σφάλμα.