

ΕΡΓΑΣΤΗΡΙΟ - 3

ΘΕΜΑΤΑ:

Κλάσεις – Αντικείμενα – Δομητές/Κατασκευαστές - Μέθοδοι - Παράμετροι

Προσοχή!!! Να εκτελεστούν πρώτα όλες οι παραλλαγές της Box (ασκήσεις 1,2,3,4)

Άσκηση – 1^η

Να γραφεί πρόγραμμα που δημιουργεί 2 αντικείμενα τύπου Box, χωρίς δομητή και σε μία μέθοδο τύπου void() υπολογίζει και εμφανίζει τον όγκο του κάθε Box. Ο ορισμός των τιμών (χαρακτηριστικά) του κάθε κουτιού θα γίνει μέσα στο κυρίως πρόγραμμα.

Το πρόγραμμα:

```
class Box {
double width; double height; double depth;

void volume() {
System.out.println("Ogkos = " + (width * height * depth));
}}

class BoxDemo1 {
public static void main(String args[]) {
Box mybox1 = new Box();
Box mybox2 = new Box();

mybox1.width = 10; mybox1.height = 20; mybox1.depth = 15;
mybox2.width = 3; mybox2.height = 6; mybox2.depth = 9;
mybox1.volume(); mybox2.volume();
}}
```

Άσκηση – 2^η

Να γραφεί πρόγραμμα που δημιουργεί 2 αντικείμενα τύπου Box, χωρίς δομητή και σε μία μέθοδο τύπου return() υπολογίζει τον όγκο του Box. Ο ορισμός των τιμών (χαρακτηριστικά) του κάθε κουτιού καθώς και η εμφάνιση των αποτελεσμάτων (όγκων των – κουτιών) θα γίνει μέσα στο κυρίως πρόγραμμα.

To πρόγραμμα:

```
class Box {
double width; double height; double depth;

    double volume() {
        return width * height * depth;
    }
}

class BoxDemo2 {
    public static void main(String args[]) {
        Box mybox1 = new Box();
        Box mybox2 = new Box();

        mybox1.width = 10; mybox1.height = 20; mybox1.depth = 15;
        mybox2.width = 3; mybox2.height = 6; mybox2.depth = 9;

        double vol;
        vol = mybox1.volume();
        System.out.println("Volume is " + vol);

        vol = mybox2.volume();
        System.out.println("Volume is " + vol);
    }
}
```

Άσκηση – 3^η

Να γραφεί πρόγραμμα, παραλλαγή της άσκησης 1, που δημιουργεί 2 αντικείμενα τύπου Box, αλλά με πλήρη δομητή και σε μία μέθοδο τύπου void() υπολογίζει και εμφανίζει τον όγκο του κάθε Box. Ο ορισμός των τιμών (χαρακτηριστικά) του κάθε κουτιού θα γίνει μέσα στο κυρίως πρόγραμμα στον ορισμό του κάθε αντικειμένου-Box (τελεστής new).

To πρόγραμμα:

```
class Box {
double width; double height; double depth;

    Box(double w, double h ,double d) {
        width = w;
    }
}
```

```
height = h;  
depth = d; }
```

```
void volume() {  
    System.out.println("Ogkos = " + (width * height * depth));  
}
```

```
class BoxDemo3 {  
    public static void main(String args[]) {  
        Box mybox1 = new Box(10, 20, 15);  
        Box mybox2 = new Box(3, 6, 9);  
  
        mybox1.volume(); mybox2.volume();  
    }
```

Άσκηση – 4^η

Να γραφεί πρόγραμμα, παραλλαγή της άσκησης 2, που δημιουργεί 2 αντικείμενα τύπου Box, αλλά με πλήρη δομητή και σε μία μέθοδο τύπου return() υπολογίζει τον όγκο του κάθε Box. Ο ορισμός των τιμών (χαρακτηριστικά) του κάθε κουτιού θα γίνει στην new και η εμφάνιση των αποτελεσμάτων (όγκων των – κουτιών) θα γίνει μέσα στο κυρίως πρόγραμμα.

Το πρόγραμμα:

```
class Box  
{  
    double width; double height; double depth;  
  
    Box(double w, double h, double d) {  
        width = w;  
        height = h;  
        depth = d;}  
    double volume() {return width * height * depth; }  
}  
class BoxDemo4 {  
    public static void main(String args[]) {  
        Box mybox1 = new Box(10, 20, 15);  
        Box mybox2 = new Box(3, 6, 9);  
  
        double vol;
```

```
    vol = mybox1.volume();
    System.out.println("Volume is " + vol);

    vol = mybox2.volume();
    System.out.println("Volume is " + vol);
}}
```

Άσκηση – 5^η

α) Να γραφεί πρόγραμμα Java που δημιουργεί την κλάση Circle, με τις απαραίτητες μεθόδους:

setRadius(): που καταχωρεί τιμή ακτίνας στα αντικείμενα τύπου Circle,

getRadius(): που επιστρέφει την ακτίνα,

getDiameter(): που επιστρέφει τη διάμετρο (2 x ακτίνα) και την

getArea(): που επιστρέφει το εμβαδό του κύκλου.

Το πρόγραμμα θα δημιουργήσει ένα αντικείμενο τύπου Circle με ακτίνα (radius) = 100 και θα εμφανίσει τα αποτελέσματα (ακτίνα, διάμετρο και εμβαδό του κύκλου).

To πρόγραμμα:

```
class Circle
{
    private int radius = 0;

    void setRadius(int y) {radius = y;}
    int getRadius() { return(radius);}

    int getDiameter() {return(2 * radius);}

    double getArea() {return((int)(3.14159 * radius * radius));}
}}

class CircleDemo {
    public static void main(String[] args) {
        Circle kyklos = new Circle();
        kyklos.setRadius(100);
        System.out.println("Aktina toy Kykloy = " + kyklos.getRadius());
        System.out.println("Diametros toy Kykloy = " + kyklos.getDiameter());
        System.out.println("Emvadon toy Kykloy = " + kyklos.getArea());
    }
}
```

β) Παραλλαγή της άσκησης-5 :

Να γραφεί το ίδιο πρόγραμμα αλλά με τη χρήση δομητή – κατασκευαστή.

Άσκηση – 6^η

Να γραφεί πρόγραμμα Java που υλοποιεί την κλάση υπάλληλος (περιγράφει τα στοιχεία ενός υπαλλήλου). Το πρόγραμμα θα περιέχει δύο ξεχωριστές κλάσεις τις:

- 1) **class Employee:** που περιέχει τα πεδία μεταβλητές: `FirstName`, `LastName`, και `Salary`, τον αρχικό (default) δομητή-κατασκευαστή, και τις μεθόδους:
 - setFirstName():** που καταχωρεί το όνομα του υπάλληλου,
 - getFirstName():** που επιστρέφει το όνομα του υπάλληλου,
 - setLastName():** που καταχωρεί το επίθετο του υπάλληλου,
 - getLastName():** που επιστρέφει το επίθετο του υπάλληλου,
 - setSalary():** που καταχωρεί τον μισθό του υπάλληλου,
 - getSalary():** που επιστρέφει τον μισθό του υπάλληλου.

- 2) **class TestEmployee,** η οποία:
 - α) δημιουργεί ένα αντικείμενο του τύπου `Employee`,
 - β) αρχικοποιεί τα στοιχεία του υπάλληλου με κλήσεις στις κατάλληλες **set** – μεθόδους
 - γ) λαμβάνει τα στοιχεία του υπάλληλου με κλήση στις κατάλληλες **get** – μεθόδους και
 - δ) εμφανίζει τα στοιχεία του υπάλληλου

Το πρόγραμμα:

```
class Employee {  
  
    private String FirstName;  
    private String LastName;  
    private float Salary;  
  
    public Employee() {  
        FirstName = "";  
        LastName = "";  
        Salary = 0.0f; }  
  
    public void setFirstName(String n){  
        FirstName = n; }  
  
    public String getFirstName() {  
        return FirstName; }  
  
    public void setLastName(String n){  
        LastName = n; }  
  
    public String getLastName() {  
        return LastName; }  
  
    public void setSalary(float Euro) {  
        Salary = Euro; }  
  
    public float getSalary() {  
        return Salary;}  
}
```

```

class TestEmployee {
  public static void main(String[] args){

    Employee ypallilos = new Employee();

    String name, epitheto;
    float misthos;

    ypallilos.setFirstName("Nikos");
    ypallilos.setLastName("Nikas");
    ypallilos.setSalary(1600.0f);

    name = ypallilos.getFirstName();
    epitheto = ypallilos.getLastName();
    misthos = ypallilos.getSalary();

    System.out.println("Onoma    = " + name);
    System.out.println("Epitheto  = " + epitheto);
    System.out.println("Misthos   = " + misthos);
  }}

```

Παραλλαγή της 6^{ης} άσκησης (Employee):

A) Μια παραλλαγή της άσκησης είναι να χρησιμοποιήσετε ένα πλήρη δομητή - κατασκευαστή. Τότε οι **set methods** δεν είναι απαραίτητοι, αλλά είναι απαραίτητες οι μέθοδοι επιστροφής των στοιχείων του υπάλληλου - **get methods**.

Το πρόγραμμα:

```

class Employee {

  private String FirstName;
  private String LastName;
  private float Salary;

  //arxikos constructor
  public Employee() {
    FirstName = "";
    LastName = "";
    Salary = 0.0f;
  }

  //pliris constructor
  public Employee(String First, String Last, float salary) {
    FirstName = First;
    LastName = Last;
    Salary = salary; }

  public String getFirstName() {
    return FirstName;
  }

  public String getLastName() {
    return LastName;
  }
}

```

```
public float getSalary() {  
    return Salary;  
}
```

```
class TestEmployee {  
    public static void main(String[] args){  
  
        Employee ypallilos = new Employee("Nikos", "Nikas", 1600.00f);  
  
        String name, epitheto;  
        float misthos;  
  
        name = ypallilos.getFirstName();  
        epitheto = ypallilos.getLastName();  
        misthos = ypallilos.getSalary();  
  
        System.out.println("Όνομα    = " + name);  
        System.out.println("Επίθετο  = " + epitheto);  
        System.out.println("Μισθος  = " + misthos);  
    }  
}
```

Άσκηση – 7^η

Στην παρακάτω άσκηση θα χειριστούμε τον τραπεζικό λογαριασμό ενός πελάτη. Στην επίλυση του προβλήματος θα χρειαστούμε σαν δεδομένα :

- Τον αριθμό λογαριασμού
- Το όνομα του λογαριασμού
- Το τρέχον υπόλοιπο του λογαριασμού

Ο Δομητής – Constructor θα δημιουργήσει το αντικείμενο και θα αρχικοποιήσει τα τρία αυτά πεδία - μεταβλητές.

Οι μέθοδοι που θα δημιουργήσουμε θα :

- Δέχονται μία κατάθεση
- Θα επεξεργάζεται ένα τσεκ
- Θα χειρίζεται το τρέχον υπόλοιπο

Στην παρακάτω πρώτη μορφή δεν θα περιλάβουμε τις μεθόδους. Είναι μια πρώτη κατασκευή της κλάσης. Στο κομμάτι προγράμματος υπάρχει και ο έλεγχος της κλάσης.

```
class CheckingAccount {  
    // μεταβλητές του αντικειμένου  
    String accountNumber;  
    String accountHolder;  
    int balance;
```

```

//δομητές
CheckingAccount( String accNumber, String holder, int start ) {
    accountNumber = accNumber ;
    accountHolder = holder ;
    balance      = start ;
}
}
class CheckingAccountTester {
    public static void main( String[] args ) {
        CheckingAccount account1 = new CheckingAccount( "123", "Bob", 100 );
        System.out.println(account1.accountNumber+" "+account1.accountHolder+" "+account1.balance);
    }
}

```

Οι προτεινόμενοι μέθοδοι θα έχουν την παρακάτω μορφή :

```

class CheckingAccount {
    // μεταβλητές του αντικειμένου
    String accountNumber;
    String accountHolder;
    int balance;

    // δομητές - constructors
    CheckingAccount( String accNumber, String holder, int start) {
        accountNumber = accNumber ;
        accountHolder = holder ;
        balance      = start ;
    }

    // επιστροφή του υπολοίπου
    int currentBalance() {
        return balance ;
    }

    // χειρισμός της κατάθεσης
    void processDeposit(int amount) {
        balance = balance + amount ;
    }

    // χειρισμός του check
    void processCheck( int amount ) {
        balance = balance - amount;
    }
}

```

Ένας έλεγχος των ανωτέρω μεθόδων είναι και ο παρακάτω :

```

class CheckingAccountTester {
    public static void main(String[] args) {
        CheckingAccount account1 = new CheckingAccount( "123", "Bob", 100 );
        System.out.println( account1.currentBalance() );
        account1.processDeposit( 2000 );
        account1.processCheck( 1500 );
        System.out.println( account1.currentBalance() ); }
}

```

Μία ακόμη μέθοδος θα μπορούσε να είναι και η παρακάτω που εμφανίζει τα αποτελέσματα και για τις τρεις μεθόδους.

```

void display() {
    System.out.println( accountNumber + "\t" + accountHolder + "\t" + balance );
}

```



```
}
```

Ο έλεγχος της display() θα μπορούσε να γίνει με τον κώδικα :

```
class CheckingAccountTester {  
  public static void main( String[] args ) {  
    CheckingAccount account1 = new CheckingAccount( "123", "Bob", 100 );  
    account1.display() ;  
    account1.processDeposit( 2000 );  
    account1.processCheck( 1500 );  
    account1.display() ;  
  }  
}
```