

ΕΡΓΑΣΤΗΡΙΟ - 4

ΘΕΜΑΤΑ:

Κλάσεις – Αντικείμενα – Δομητές/Κατασκευαστές - Μέθοδοι - Παράμετροι

Συνέχεια με μεθόδους κλάσεων (4-ασκήσεις) – χωρίς την χρήση αντικειμένων, με μεθόδους αντικειμένων και μεικτού τύπου

Άσκηση – 1^η

Πρόσθεση δύο αριθμών στην παραμετρική μέθοδο κλάσης - **add()** – τύπου επιστρεφόμενου αποτελέσματος – (τύπου *return*) και με κλήση από την *main()*.

```
class TestAdd1 {
    public static void main(String[] args) {
        int x = 5;
        int y = 2;
        int sum = add(x, y); //klisi tis methodoy - perasma parametron
        System.out.println("To athroisma tou " + x + " kai " + y + " einai = " + sum);
    }

    /* H methodos prosthesis 2 arithmvn */
    public static int add(int num1, int num2) {
        int result = num1 + num2;
        return result;
    }
}
```

Άσκηση – 2^η

Πρόσθεση δύο αριθμών στην παραμετρική μέθοδο κλάσης - **add()** – τύπου void και με κλήση από την main().

```
class TestAdd2 {
    public static void main(String[] args) {
        int x = 5;
        int y = 2;
        add(x, y);    //klisi tis methodoy - perasma parametron
    }

    /* H methodos prosthesis 2 arithmvn */
    public static void add(int num1, int num2) {
        int result = num1 + num2;
        System.out.println("To athroisma tou "+num1+" kai "+num2+" einai = "+result);
    }
}
```

Άσκηση – 3^η (χρήση αντικειμένου)

Πρόσθεση δύο αριθμών στην παραμετρική μέθοδο αντικειμένου - **add()** – τύπου επιστρεφόμενου αποτελέσματος – (τύπου return) και με κλήση από την main().

```
class AddObj {
    int ar1;
    int ar2;

    AddObj(int x, int y) {
        ar1 = x;
        ar2 = y;}

    /* H methodos prosthesis 2 arithmvn */
    public int add(int num1, int num2) {
        int result = num1 + num2;
        return result;}
}

class TestAddObj {
    public static void main(String[] args) {
        int x=5; int y=2;
        AddObj obj = new AddObj(x, y);    //dhmioyrgia antikeimenoy typou AddObj
        int sum = obj.add(5, 2);          //klisi tis methodoy - perasma parametron
        System.out.println("To athroisma tou " + x + " kai " + y + " einai = " + sum); }
}
```

Άσκηση – 4^η (χρήση αντικειμένου)

Πρόσθεση δύο αριθμών στην παραμετρική μέθοδο αντικειμένου - **add()** – τύπου void και με κλήση από την main().

```
class AddObj {
    int ar1;
    int ar2;

    AddObj(int x, int y) {
        ar1 = x;
        ar2 = y; }

    /* H methodos prosthesis 2 arithmvn */
    public void add(int num1, int num2) {
        int result = num1 + num2;
        System.out.println("To athroisma tou " +num1+" kai "+num2+" einai = "+result);}
}

class TestAddObj1 {
    public static void main(String[] args) {
        int x=5; int y=2;
        AddObj obj = new AddObj(x, y); //dhmioyrgia antikeimenoy tyrou AddObj
        obj.add(5, 2); //klisi tis methodoy - perasma parametron
    }
}
```

Άσκηση-5^η (Μεικτού τύπου: χρήση αντικειμένου και χρήση static-μεθόδου)

Στην άσκηση αυτή θα υπολογίσουμε το άθροισμα δύο αριθμών με την χρήση αντικειμένου αλλά θα υπολογίσουμε και το γινόμενο τους χωρίς την χρήση αντικειμένου (σε μέθοδο τύπου static). Η μέθοδος add() θα είναι τύπου void, και η στατική μέθοδος multiply() θα είναι τύπου return.

```
class AddObj {
    int ar1;
    int ar2;

    AddObj(int x, int y) {
        ar1 = x;
        ar2 = y; }

    /* H methodos prosthesis 2 arithmvn */
    public void add(int num1, int num2) {
        int result = num1 + num2;
        System.out.println("To athroisma tou "+num1+" kai "+num2+" einai = "+result);}
}
```

```

class TestAddObj3
{
    public static void main(String[] args) {
        int x=5; int y=2;
        AddObj obj = new AddObj(x, y); //dhmioyrgia antikeimenoy tyrou AddObj
        obj.add(5, 2); //klisi methodou
        int m = multiply(x, y);
        System.out.println("To ginomeno tou "+x+ " epi "+y+ " einai = " + m);
    }

    public static int multiply(int num1, int num2) {
        int result = num1 * num2;
        return result; }
}

```

Ασκήσεις με χρήση κλάσεων αντικειμένων και μεθόδων αντικειμένων...

Άσκηση – 6^η

Να γίνει το πρόγραμμα που δημιουργεί την κλάση **Plyntyria** που θα έχει τα παρακάτω πεδία:

- 1) Μάρκα πλυντηρίου, τύπου String,
- 2) Έτος Αγοράς, τύπου int,
- 3) Κιλά ρούχων, τύπου int,
- 4) Στροφές, τύπου int,
- 5) Διαστάσεις, τύπου String.

Το πρόγραμμα θα δημιουργεί αντικείμενο(α) του τύπου Plyntyria και θα εμφανίζει τα στοιχεία τους. Χρησιμοποιήστε τις κατάλληλες get – εντολές για την εμφάνιση των στοιχείων και ένα πλήρη δομητή.

Μια πιθανή λύση:

```

class Plyntyria {
    private String marka;
    private int etos;
    private int kila;
    private int strofes;
    private String diastaseis;

    Plyntyria (String marka, int etos, int kila, int strofes, String diastaseis){
        this.marka=marka;
        this.etos=etos;
        this.kila=kila;
        this.strofes=strofes;
        this.diastaseis=diastaseis;
    }
}

```

```

String getMarka(){
    return marka;}

int getEtos(){
    return etos;}

int getKila(){
    return kila;}

int getStrofes(){
    return strofes;}

String getDiastaseis(){
    return diastaseis;}

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

        Plyntyria plintirio = new Plyntyria("AEG - v346", 2007, 6, 1200, "85x60x58");

        System.out.println("Marka          = " + plintirio.getMarka());
        System.out.println("Etos Polisevs = " + plintirio.getEtos());
        System.out.println("kila royxvn  = " + plintirio.getKila());
        System.out.println("Strofes     = " + plintirio.getStrofes());
        System.out.println("Diastaseis  = " + plintirio.getDiastaseis());
    }
}

```

Άσκηση – 7^η

Να γίνει το πρόγραμμα οδήγησης ενός αυτοκινήτου. Το πρόγραμμα δημιουργεί την κλάση **Car** που περιέχει δύο πεδία: 1) τα χιλιόμετρα της οδήγησης (double), και 2) η ταχύτητα του αυτοκινήτου, που ξεκινά π.χ. από τα 100 χιλιόμετρα την ώρα. Στην κλάση θα περιέχονται οι παρακάτω μέθοδοι α) η μέθοδος **πάτα το γκάζι** που ανεβάζει την ταχύτητα κατά 20 χιλμ./ώρα, β) **μέγιστη ταχύτητα** που επιστρέφει την μέγιστη ταχύτητα της οδήγησης, γ) **οδήγηση** που υπολογίζει την χιλιομετρική απόσταση που διάνυσε το αυτοκίνητο και δ) **εμφάνιση στοιχείων οδήγησης** που εμφανίζει την μέγιστη ταχύτητα οδήγησης και την χιλιομετρική απόσταση που διάνυσε το αυτοκίνητο. Το πρόγραμμα δημιουργεί ένα αντικείμενο του τύπου Car (π.χ. Honda) και κάνει μια δοκιμαστική διαδρομή.

```

class Car {
    public double kilometres;
    public int speed;

    Car() {
        kilometres = 0.0;
        speed = 100;
    }
    public void pata_to_gazi() {
        speed += 20;
    }
}

```

```

    }
    public int maxSpeed() {
        return speed;
    }

    public void drive(double distance) {
        kilometres += distance;
    }

    public void showStoiheia() {
        System.out.println("Megisti Taxytita = " + speed + " km/h");
        System.out.println("Apostasi pou dianise = " + kilometres + " km");
    }
}

class TestCar {

    public static void main(String[] args) {

        Car Honda = new Car();

        //ayxisi tis taxytitas kata 25 km/h.
        Honda.pata_to_gazi();

        //ayxisi tvn hiliometrwn kata 100.0.
        Honda.drive(100.0);

        //Sets the local variable max to the value
        //returned by maxSpeed().

        int max = Honda.maxSpeed();

        //Emfanisi tvn stoiheivn odigisis
        Honda.showStoiheia();
    }
}

```

Παραλλαγές:

1) Εμπλουτισμός με μεθόδους **πάτα φρένο, ξεκίνα, σταμάτα**, κλπ. Επίσης εμπλουτισμός με περισσότερα πεδία, όπως **χρώμα αυτοκινήτου, αριθμός ταχυτήτων, αριθμός θυρών**, κλπ.

Άσκηση – 8^η

Παραλλαγή της κλάσης Employee όπου χρησιμοποιούνται αντικείμενα ως πεδία. Αυτή η τακτική εφαρμόζεται και για οποιαδήποτε άλλη κλάση που εμπεριέχει πολλά επιμέρους στοιχεία, όπως εδώ επωνυμία, διεύθυνση, κλπ. Τότε **κατασκευάζουμε διαφορετικές κλάσεις** για την επωνυμία, την διεύθυνση και ότι άλλο επιμέρους στοιχείο χρειάζεται η αρχική κλάση (εδώ η κλάση Employee). Δηλαδή στην παραλλαγή αυτή θα φτιάξουμε τέσσερις επιμέρους κλάσεις, τις:

- 1) **Name:** που ορίζει το όνομα του υπάλληλου (αντικείμενο τύπου Ονοματ/μο),
- 2) **Address:** που ορίζει την διεύθυνση του υπάλληλου (αντικείμενο τύπου Διεύθυνση),
- 3) **Employee:** που ορίζει τα στοιχεία του υπάλληλου, χρησιμοποιώντας όμως σαν μεταβλητές αντικείμενα του των τύπων Name και Address.
- 4) **TestEmployee:** η οποία ορίζει αντικείμενα του τύπου Name, Address και Employee και που εμφανίζει όλα τα στοιχεία του Υπάλληλου.

Προσοχή στις μεθόδους set και get.

```
class Name {
    String firstname;
    String lastname;

    public Name(String newFirstname, String newLastname) {
        firstname = newFirstname;
        lastname = newLastname;
    }

    public String getFirstname() {
        return firstname;
    }

    public String getLastname() {
        return lastname;
    }

    public String getFirstLast() {
        return firstname + " " + lastname;
    }
}

class Address {
    String street;
    String city;
    String state;
    String zip;

    public Address(String newStreet, String newCity, String newState, String newZip) {
        street = newStreet;
        city = newCity;
        state = newState;
        zip = newZip;
    }

    public String getStreet() {
        return street; }

    public String getCity() {
        return city;}

    public String getState() {
        return state;}

    public String getZip() {
        return zip; }
```

```

    public String getFullAddress() {
        return street + "\n" + city + ", " + state + ", " + zip; }
}

class Employee {
    //fields
    private Name myName;           //antikeimeno typou Name
    private Address myAddress;     //antikeimeno typou Address
    private String AFM;

    //constructor
    public Employee(Name n, Address a, String newAFM) {
        myName = n;
        myAddress = a;
        AFM = newAFM;
    }

    public Name getName() {
        return myName; }

    public Address getAddress() {
        return myAddress; }

    public String getAFM() {
        return AFM; }
}

class TestEmployee {
    public static void main(String[] args) {
        Name eponymia = new Name("Nikos", "Nikas");

        Address dieythinsi = new Address("Tsimiski 35", "Thessaloniki", "Greece", "56 700");

        String AFM = "123456789";

        Employee theEmployee = new Employee(eponymia, dieythinsi, AFM);

        System.out.println(theEmployee.getName().getFirstLast() + "\n" +
            theEmployee.getAddress().getFullAddress() + "\n" + theEmployee.getAFM());
    }
}

```

Άλυτη άσκηση

Να γίνει το πρόγραμμα που δημιουργεί την κλάση **Robots**. Η κλάση εκτός από τα αναγκαία πεδία όπως **όνομα, μοντέλο, χρονιά κατασκευής, ταχύτητα το λεπτό, κλπ.** Θα περιέχει και τις μεθόδους που εκκινούν και σταματούν το ρομπότ on-off (εμφάνιση απλών μηνυμάτων), αλλά και μεθόδους που το κινούν. Το πρόγραμμα για κάθε αντικείμενο του τύπου ρομπότ θα εμφανίζει όλα τα στοιχεία της διαδρομής που θα διανύσει.