

ΕΠΑΝΑΛΗΨΗ

ΠΑΡΑΔΕΙΓΜΑ

Εκφώνηση

Στην άσκηση αυτή θα σχεδιάσετε και θα αναπτύξετε (στη γλώσσα προγραμματισμού Java) ένα σύστημα διαχείρισης για μία εταιρία. Η εταιρία περιγράφεται από το όνομα και τη διεύθυνσή της. Το σύστημα θα διαχειρίζεται την εισαγωγή ενός νέου υπαλλήλου, τη διαγραφή ενός υπαλλήλου και την εκτύπωση όλων των υπαλλήλων. Οι υπάλληλοι μπορεί να είναι

- Διοικητικοί οι οποίοι έχουν όνομα, επώνυμο, Αριθμό Ταυτότητα, μισθό και τίτλο πτυχίου.
- Τεχνικοί οι οποίοι έχουν όνομα, επώνυμο, Αριθμό Ταυτότητα, μισθό και ειδικότητα.

Ο μισθός πρέπει να είναι από 900-1500.

Δεν μπορούν να εισαχθούν 2 υπάλληλοι με το ίδιο Αριθμό ταυτότητα

Η εταιρία έχει μέχρι 100 υπαλλήλους.

Το σύστημα που θα σχεδιάσετε και θα υλοποιήσετε θα πρέπει να υποστηρίζει τις ακόλουθες λειτουργίες:

1. Εισαγωγή εταιρίας (θεωρείστε ότι για το σύστημα σας θα υπάρχει μόνο μία εταιρία)
2. Εισαγωγή ενός νέου υπαλλήλου. Όταν ο χρήστης επιλέξει εισαγωγή τότε θα του ζητείται να επιλέξει τον τύπο του υπαλλήλου (1. Διοικητικός, 2. Τεχνικός) και στη συνέχεια θα ζητούνται και θα δίνονται τα ανάλογα στοιχεία. (Για ευκολία μπορείτε να ορίσετε δύο διαφορετικές μεθόδους εισαγωγής)
3. Διαγραφή Υπαλλήλου. Όταν ο χρήστης επιλέγει διαγραφή τότε θα του ζητείται ο αριθμός ταυτότητας.
4. Εκτύπωση όλων των υπαλλήλων

Να ορίσετε τις κλάσεις που χρειάζονται χρησιμοποιώντας κληρονομικότητα, πολυμορφισμό κτλ.

Να ορίσετε ένα γραφικό interface για την εισαγωγή νέου υπαλλήλου.

Η κλάση Ipaliloi

```
import java.util.Scanner;

public class Ipaliloi {

    private String onoma, eponimo, ar_tautotitas;
    private int misthos=900;
    private Scanner scanner = new Scanner(System.in);

    public Ipaliloi() { }

    public Ipaliloi(String onoma, String eponimo, String ar_tautotitas, int
misthos)
    { this.onoma = onoma; this.eponimo = eponimo;
this.ar_tautotitas = ar_tautotitas;
if ((misthos>=900) && (misthos<=1500))
    this.misthos = misthos; }

    public String getOnoma() { return onoma; }

    public void setOnoma(String onoma) { this.onoma = onoma; }

    public String getEponimo() { return eponimo; }
```

```
public void setEponimo(String eponimo) { this.eponimo = eponimo; }

    public String getAr_tautotitas() { return ar_tautotitas; }

    public void setAr_tautotitas(String ar_tautotitas) {
    this.ar_tautotitas = ar_tautotitas; }

    public int getMisthos() { return misthos; }

    public void setMisthos(int misthos) {
        if ((misthos>=900) && (misthos<=1500))
            this.misthos = misthos; }

}
```

Η κλάση Dioikitikoi

```
import java.util.Scanner;

public class Dioikitikoi extends Ipaliloi {
    private String eidikotita;

    public Dioikitikoi(){}

    public Dioikitikoi(String onoma, String eponimo, String ar_tautotitas, int
    misthos, String eidikotita)
    { super(onoma, eponimo, ar_tautotitas, misthos);
        this.eidikotita = eidikotita; }

    public String getEidikotita() { return eidikotita; }
    public void setEidikotita(String eidikotita) { this.eidikotita = eidikotita; }

    public void Create_dik(int gui) throws InterruptedException {
        if (gui == 1) {
            Gui g = new Gui(1);
            setOnoma(g.getOnoma());
            setEponimo(g.getEponimo());
            setAr_tautotitas(g.getAr_tautotitas());
            setMisthos(g.getMisthos());
            setEidikotita(g.getEidikotita());
        }
    }
}
```

```
else { Scanner scanner = new Scanner(System.in);
    //Onoma input
    System.out.println("\nDoste onoma dioikitikou ipalilou :");
    setOnoma(scanner.nextLine());
    //input eponimo
    System.out.println("\nDoste eponimo dioikitikou ipalilou :");
    setEponimo(scanner.nextLine());
    //input arithmou tautotias
    System.out.println("\nDoste arithmo tautotitas dioikitikou ipalilou :");
    setAr_tautotitas(scanner.nextLine());
    //input eidikotitas
    System.out.println("\nDoste eidikotita dioikitikou ipalilou :");
    setEidikotita(scanner.nextLine());
    //input misthou
    System.out.println("\nDoste mistho dioikitikou ipalilou :");
    setMisthos(scanner.nextInt());
    while(getMisthos()<900 || getMisthos()>1500)
    { System.out.println("\nMi epitreptos misthos. Doste mistho dioikitikou
    ipalilou :");
        setMisthos(scanner.nextInt());
    }
} } }
```

Η κλάση Texnikoi

```
import java.util.Scanner;

public class Texnikoi extends Ipaliloi {
    private String titlos_ptixiou;

    public Texnikoi(){}

    public Texnikoi(String onoma, String eponimo, String ar_tautotitas, int misthos, String
titlos_ptixiou)
{
    super(onoma, eponimo, ar_tautotitas, misthos);
    this.titlos_ptixiou = titlos_ptixiou; }

    public String getTitlos_ptixiou() { return titlos_ptixiou; }

    public void setTitlos_ptixiou(String titlos_ptixiou) {
        this.titlos_ptixiou = titlos_ptixiou; }

    public void Create_texn(int gui) throws InterruptedException {
        if (gui == 1) {
            Gui g = new Gui(2);
            setOnoma(g.getOnoma());
            System.out.println("to onoma einai :" + g.getOnoma());
            setEponimo(g.getEponimo());
            setAr_tautotitas(g.getAr_tautotitas());
            setMisthos(g.getMisthos());
            setTitlos_ptixiou(g.getTitlos_ptixiou());
        }
    }
}
```

```
else { Scanner scanner = new Scanner(System.in);
    //Onoma input
    System.out.println("\nDoste onoma texnikou ipalilou :");
    setOnoma(scanner.nextLine());
    //input eponimo
    System.out.println("\nDoste eponimo texnikou ipalilou :");
    setEponimo(scanner.nextLine());
    //input arithmou tautotias
    System.out.println("\nDoste arithmo tautotitas texnikou ipalilou :");
    setAr_tautotitas(scanner.nextLine());
    //input ptixio tenikou
    System.out.println("\nDoste titlo ptixiou texnikou ipalilou :");
    setTitlos_ptixiou(scanner.nextLine());
    //input misthou
    System.out.println("\nDoste mistho texnikou ipalilou :");
    setMisthos(scanner.nextInt());

    while(getMisthos()<900 || getMisthos()>1500) {
        System.out.println("\nMi epitreptos misthos. Doste mistho dioikitikou
ipalilou :");
        setMisthos(scanner.nextInt());
    }
}}
```

Η κλάση Etairia

```
import java.util.Scanner;

public class Etairia {

    private String onoma;

    public Etairia() {
    }

    Public Etairia(String onoma) {
        this.onoma = onoma;    }

    public String getOnoma() {
        return onoma;
    }
}
```

```
public void setOnoma(String onoma)
{
    this.onoma = onoma;
}

public void eisagwgi_eterias() {
    Scanner scanner = new
Scanner(System.in);
    System.out.println("Doste
onoma eterias :");
    this.onoma = scanner.nextLine();
    System.out.println("I eteria exei
onoma :" + this.onoma + "\n");
}
}
```

Η κλάση GUI

```
import javax.swing.JPanel;
import javax.swing.BorderFactory;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Gui {

    private JFrame frame;
    private JPanel panel;
    private JTextField tfText2, tfText3, tfText4,
    tfText5, tfText6, tfText7;
    private JButton btBut0;
    private JLabel lbLabel0;
    private JLabel p1, p2, p3, p4, p5, p6;
    private static String onoma, eponimo,
    ar_tautotitas, eidikotita, titlos_ptixiou;
    private int misthos, close_flg;
```

```
public Gui(int id) throws InterruptedException {

    frame = new JFrame("Dimiourgia Ipalilou");

    panel = new JPanel(new GridLayout(10, 0));

    p1 = new JLabel("Onoma");
    panel.add(p1);
    tfText2 = new JTextField();
    tfText2.setName("Onoma");
    panel.add(tfText2);

    p2 = new JLabel("Epwnymo");
    panel.add(p2);
    tfText3 = new JTextField();
    tfText3.setName("Epwnymo");
    panel.add(tfText3);

    p3 = new JLabel("AT");
    panel.add(p3);
    tfText4 = new JTextField();
    tfText4.setName("AT");
    panel.add(tfText4);

    p4 = new JLabel("Misthos");
    panel.add(p4);
    tfText5 = new JTextField();
    tfText5.setName("Misthos");
    panel.add(tfText5);
```

Η κλάση GUI

```
if (id == 1) {  
    p5 = new JLabel("Eidikotita");  
    panel.add(p5);  
    tfText6 = new JTextField();  
    tfText6.setName("Eidikotita");  
    panel.add(tfText6);  
} else {  
    p6 = new JLabel("Titlos ptixiou");  
    panel.add(p6);  
    tfText7 = new JTextField();  
    tfText7.setName("Titlos ptixiou");  
    panel.add(tfText7);  
}
```

```
btBut0 = new JButton("Eisagwgi Stoixeiwn");  
  
btBut0.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        onoma = tfText2.getText();  
        eponimo = tfText3.getText();  
        ar_tautotitas = tfText4.getText();  
        misthos =  
Integer.parseInt(tfText5.getText());  
  
        if (id == 1) { // gia dikoiitiko  
            eidikotita = tfText6.getText();  
        } else { //gia texniko  
            titlos_ptixiou = tfText7.getText();  
        }  
        close_flag = 1;  
        frame.dispose();  
    }  
});
```

Η κλάση GUI

```
panel.add(btBut0);

panel.setVisible(true);
frame.add(panel);
frame.setSize(400, 400);
frame.pack();
frame.setVisible(true);

//xrisimopoioume sleep giati an den iparxei, o kwdikas
trexei xwris na exei parei akoma times
//apo to GUI me apotelesma otan paei i Main sto
ArrayList na vazei NULL.
Thread.sleep(25000);
}

public static String getOnoma() { return onoma; }

public static void setOnoma(String onoma2) { onoma =
onoma2; }

public static String getEponimo() { return eponimo; }

public static String getAr_tautotitas() { return
ar_tautotitas; }
```

```
public static void setAr_tautotitas(String ar_tautotitas2) {
    ar_tautotitas = ar_tautotitas2;
}

public static void setEponimo(String eponimo2) {
    eponimo = eponimo2; }

public static String getIdikotita() {
    return eidikotita; }

public static void setIdikotita(String eidikotita2) {
    eidikotita = eidikotita2; }

public static String getIditlos_ptixiou() {
    return titlos_ptixiou; }

public static void setIditlos_ptixiou(String titlos_ptixiou2) {
    titlos_ptixiou = titlos_ptixiou2;
}

public int getMisthos() {
    return misthos;
}

public void setIditlos_ptixiou(int misthos2) {
    misthos = misthos2;
}
}
```

Η κλάση Main

```
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;
import java.util.Scanner;
import javax.swing.*;
import java.io.File; // Import the File class
import java.io.FileWriter;
import java.io.IOException; // Import the IOException class to handle
errors
import java.util.logging.Level;
import java.util.logging.Logger;

public class Main {

    private static int id, ak, plithos = 0;
    private static String ar_taut;
    private static char ch;
    private static Scanner scanner;
    private static List<Ipaliloi> listIpaliloi;

    public static void main(String[] args) throws InterruptedException {

        listIpaliloi = new ArrayList<Ipaliloi>();
        listIpaliloi.clear();
        scanner = new Scanner(System.in);
        Etairia et = new Etairia();

        et.eisagwgi_eterias();
```

```
while (true) {
    System.out.println("\nDoste (1) gia dimiourgia ipalilou \nDoste
(2) diagrafh ipalilou \nDoste (3) gia emfanisi ipalilon");
    ch = scanner.next().charAt(0);
    switch (ch) {
        case '1':
            if (plithos <= 100) {
                Create_Ipal();
            }
            break;
        case '2':
            Remove_Ipal();
            break;
        case '3':
            Print_All();
            break;
        default:
            System.out.println("Lathos eisagwgh timhs\n\n");
    }
}
}
```

Η κλάση Main

```
public static void Create_Ipal() throws  
InterruptedException {  
    while (true) {  
        System.out.println("Doste (1) gia  
dimiourgia texnikou ipalilou kai (2) gia  
dimiourgia dioikitikou ipalilou :");  
        id = scanner.nextInt();  
        if (id == 1) {  
            Texnikoi tex = new Texnikoi();  
            System.out.println("Doste (1) gia  
dimiourgia me interface kai (2) xoris interface  
:");  
            ak = scanner.nextInt();  
            tex.Create_texn(ak);  
            listIpaliloi.add(tex);  
            break;  
        }  
    }  
}
```

```
if (id == 2) {  
    Dioikitikoi dik = new  
Dioikitikoi();  
    System.out.println("Doste (1)  
gia dimiourgia me interface kai (2) xoris  
interface :");  
    ak = scanner.nextInt();  
    dik.Create_dik(ak);  
    listIpaliloi.add(dik);  
    break;  
} else {  
    System.out.println("Dosate  
lathos epilogi");  
    continue;  
}  
}
```

Η κλάση Main

```
public static void Remove_Ipal() {
    scanner = new Scanner(System.in);
    System.out.println("Doste arithmo tautotitas pou
thelete na diagraftei");
    ar_taut = scanner.nextLine();
    //ar_taut = scanner.nextLine();
    System.out.println("Isoute me : " + ar_taut);
    for (int counter = 0; counter < listIpaliloi.size();
counter++) {
        System.out.println((listIpaliloi.get(counter)).getAr_taut
otitas());
        if
(Objects.equals((listIpaliloi.get(counter)).getAr_tautoti
tas(), ar_taut) == true) {
            listIpaliloi.remove(counter);
            System.out.println("Diagraftike me epitixia");
        }
    }
}
```

```
public static void Print_All() {
    //dimiourgia arxeiou
    try {
        File myObj = new File("ipaliloi.txt");
        if (myObj.exists() && myObj.isFile()) {
            myObj.delete();
        }
        if (myObj.createNewFile()) {
            System.out.println("File created: " +
myObj.getName());
        } else {
            System.out.println("File already exists.");
        }
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}
```

Η Κλάση Main

```
//emfanizei ipallilous kai vazei ta stixia tous se ena arxeio
for (int counter = 0; counter < listIpaliloi.size(); counter++) {
try {
    FileWriter myWriter = new FileWriter("ipaliloi.txt");
    if (listIpaliloi.get(counter) instanceof Texnikos)
    {
        myWriter.write(listIpaliloi.get(counter).getOnoma()+" "
+listIpaliloi.get(counter).getEponimo()+" "
+listIpaliloi.get(counter).getAr_tautotitas()+" "
+listIpaliloi.get(counter).getMisthos()+" "+listIpaliloi.get(counter).
getTitlos_ptixiou() +" "+ "Texnikos");
    }
    else if (listIpaliloi.get(counter) instanceof Dioikitikos)
    {
        myWriter.write(listIpaliloi.get(counter).getOnoma()+" "
+listIpaliloi.get(counter).getEponimo()+" "
+listIpaliloi.get(counter).getAr_tautotitas()+" "
+listIpaliloi.get(counter).getMisthos()+" "+listIpaliloi.get(counter).
getEidikotita() +" "+ "Dioikitikos");
    }
    myWriter.close();
}
}
```

```
catch (IOException ex) {
    System.out.println("An error occurred.");
}
}
System.out.print("\n");
}
}
```

Να Υλοποιήσετε την equals

Να Υλοποιήσετε την equals στις κλάσεις Ipalliloi, Dioikitikoi, Texnikoi και να επιτρέπεται την σύγκριση αντικειμένων διαφορετικών κλάσεων με βάση των αριθμό ταυτότητας.

Η κλάση IPalliloi

```
import java.util.Scanner;

public class Ipaliloi {

    private String onoma, eponimo,
ar_tautotitas, type;
    private int misthos;
```

```
public Boolean equals(object obj){
if (obj.getClass() == this.getClass()) {
return obj.ar_tautotitas. equals(
    this.ar_tautotitas);}
else if ((obj instanceof Dioikitikoi) ||
        (obj instanceof Texnikoi )) {
return this.ar_tautotitas. equals( obj.
getAr_tautotitas());}
else{ return False;}
}
```

Η κλάση Dioikitikoi

```
public class Dioikitikoi extends  
Ipalliloi {  
    private String eidikotita;
```

```
public Boolean equals(object obj){  
    if (obj.getClass() == this.getClass())  
        {  
        return obj. getAr_tautotitas().  
            equals( this. getAr_tautotitas());}  
    else if ((obj instanceof Ipalliloi)  
        || (obj instanceof Texnikoi )) {  
        return this. getAr_tautotitas().  
            equals( obj. getAr_tautotitas());}  
    else{ return False;}  
    }
```

Η κλάση Texnikoi

```
public class Texnikoi extends  
Ipalliloi {  
    private String titlos_ptixiou;
```

```
public Boolean equals(object obj){  
if (obj.getClass() == this.getClass())  
    {  
return obj. getAr_tautotitas().  
    equals( this. getAr_tautotitas());}  
else if ((obj instanceof Ipalliloi)  
    || (obj instanceof Dioikitikoi )) {  
return this. getAr_tautotitas().  
    equals( obj. getAr_tautotitas());}  
else{ return False;}  
}
```

ΆΛΛΟΣ ΟΡΙΣΜΟΣ ΤΗΣ Equal

- Να υλοποιήσετε την equal ελέγχοντας όλα τα πεδία.
- **Τι πρόβλημα θα υπάρξει αν επιτρέψουμε την σύγκριση αντικειμένων διαφορετικών κλάσεων;**

Η κλάση IPalliloi

```
import java.util.Scanner;

public class Ipalliloi {

    private String onoma, eponimo,
ar_tautotitas, type;
    private int misthos;
```

```
public Boolean equals(object obj){
if (obj instanceof Ipalliloi) {
return (obj. getAr_tautotitas().equals(
    this.ar_tautotitas)) &&
    (obj.getOnoma().equals(this.onoma))
    && (obj.getEponimo(). equals(
    this.eponimo) )&& (obj.
    getMisthos()==this. misthos);
}
else False;
}
```

Η κλάση Dioikitikoi

```
public class Dioikitikoi extends  
Ipaliloi {
```

```
    private String eidikotita;
```

```
public Boolean equals(object obj){  
    if (obj.getClass() == this.getClass())  
        {  
        return super.equals(obj) &&  
            obj.eidikotita. equals( this.  
                eidikotita);}  
  
    else if ((obj instanceof Ipalliloi)  
        ||      (obj instanceof Texnikoi  
        ))  
  
        return super.equals(obj);  
    else{ return False;}  
}
```

Η κλάση Texnikoi

```
public class Texnikoi extends  
Ipaliloi {  
    private String titlos_ptixiou;
```

```
public Boolean equals(object obj){  
if (obj.getClass() == this.getClass())  
    {  
return super.equals(obj) && obj.  
    titlos_ptixiou. equals( this.  
    titlos_ptixiou);}  
else if ((obj instanceof Ipalliloi)  
    || (obj instanceof  
    Dioikitikoi ))  
        return super.equals(obj);  
else{ return False;}  
}
```

main

```
Ipalliloi y=new Ipallilioi("Kostas", "Papadopoulos", "P987394",1200);
```

```
Dioikitikoi d=new Dioikitikoi("Kostas", "Papadopoulos",  
"P987394",1200,"Manager");
```

```
Texnikoi t=new Texnikoi("Kostas", "Papadopoulos",  
"P987394",1200,"CS");
```

```
Boolean b1=y.equals(d); // True
```

```
Boolean b2=d.equals(y); // True
```

```
Boolean b3=t.equals(d); // True
```

```
Boolean b4=d.equals(t); // True
```

```
Boolean b5=y.equals(t); // True
```

```
Boolean b6=t.equals(y); // True
```

Main – τι θα γίνονταν αν έλειπαν τα bold

```
Ipalliloi y=new Ipallilioi("Kostas", "Papadopoulos", "P987394",1200);
```

```
Dioikitikoi d=new Dioikitikoi("Kostas", "Papadopoulos",  
"P987394",1200,"Manager");
```

```
Texnikoi t=new Texnikoi("Kostas", "Papadopoulos", "P987394",1200,"CS");
```

```
Boolean b1=y.equals(d); // True
```

```
Boolean b2=d.equals(y); // False
```

```
Boolean b3=t.equals(d); // False
```

```
Boolean b4=d.equals(t); // False
```

```
Boolean b5=y.equals(t); // True
```

```
Boolean b6=t.equals(y); // False
```

Άρα η υλοποίηση θα ήταν λάθος

ΠΑΡΑΔΕΙΓΜΑ interface

- Να ορίσετε ένα interface το οποίο θα περιέχει την μέθοδο compareto. Να το υλοποιήσετε για τις κλάσεις Ιρalliloi, Τεχνικοί, Δικοικητικοί. Η μέθοδος θα επιστρέφει 0 αν είναι δύο αντικείμενα ίσα, -1 αν το πρώτο αντικείμενο είναι μικρότερο του δεύτερου και 1 αν το πρώτο αντικείμενο είναι μεγαλύτερο του δεύτερου.

Δύο Ιρalliloi είναι ίσοι αν έχουν το ίδιο μισθό. Όποιος έχει μεγαλύτερο μισθό θεωρείται μεγαλύτερος.

Οι Τεχνικοί, Δικοικητικοί είναι μεγαλύτεροι του Ιρalliloi.

Οι Δικοικητικοί είναι μεγαλύτεροι του Τεχνικοί.

Δύο Δικοικητικοί είναι ίσοι αν έχουν το ίδιο μισθό. Όποιος έχει μεγαλύτερο μισθό θεωρείται μεγαλύτερος.

Δύο Τεχνικοί είναι ίσοι αν έχουν το ίδιο μισθό. Όποιος έχει μεγαλύτερο μισθό θεωρείται μεγαλύτερος.

```
interface MyCompare
{
    /* compiler will treat them as: * public abstract void method1(); */
    public int compareEmployee(object obj);
}
```

```
public class Ipalliloi implements MyCompare{ .....
public int compareEmployee(object obj){
if (obj.getClass()==this.getClass()){ // δύο υπάλληλοι
    if (obj.misthos==this.misthos) return 0;
    else if (this.misthos<obj.misthos) return -1;
    else return 1;    }
else if ((obj instanceof Dioikitikoi) || (obj instanceof Texnikoi))
    return -1;
    else System.exit(0);
}
```

```
interface MyCompare
{
    /* compiler will treat them as: * public abstract void method1(); */
    public int compareEmployee(object obj);
}
```

```
public class Dioikitikoi extends Ipalliloi implements MyCompare{ .....
public int compareEmployee(object obj){
    if (obj.getClass()==this.getClass()){ // δύο Δικαιητικοι
        if (obj.misthos==this.misthos) return 0;
        else if (this.misthos<obj.misthos) return -1;
        else return 1;    }
    else if ((obj instanceof Ipalliloi) || (obj instanceof Texnikoi))
        return 1;
    else System.exit(0);
}
```

```
interface MyCompare
{
    /* compiler will treat them as: * public abstract void method1(); */
    public int compareEmployee(object obj);
}
```

```
public class Texnikoi extends Ipalliloi implements MyCompare{ .....
public int compareEmployee(object obj){
if (obj.getClass()==this.getClass()){ // δύο Δικαιητικοι
    if (obj.misthos==this.misthos) return 0;
    else if (this.misthos<obj.misthos) return -1;
    else return 1; }
else if (obj instanceof Ipalliloi) return 1;
    else if (obj instanceof Dioikitikoi)) return -1;
    else System.exit(0);
}
```

Εφαρμογή compareto

Να γράψετε την main ώστε να διαβάσει 100 Ipalliloι ή Dikoitikoι ή Τεχνικοι. Να κάνετε ταξινόμησει.

ΠΑΡΑΔΕΙΓΜΑ abstract class

- Σε ένα πανεπιστήμιο έχουμε φοιτητές οι οποίοι μπορεί να είναι προπτυχιακοί, μεταπτυχιακοί, Διδακτορικοί, Erasmus.
- Όλοι οι φοιτητές έχουν όνομα επώνυμο, αμ, τηλεφωνο. Επίσης χρειάζεται set και get για κάθε ένα από τα προηγούμενα χαρακτηριστικά και μέθοδος printInfo() η οποία πρώτα τυπώνει την κατηγορία του φοιτητή και μετά τα υπολοιπα στοιχεία.
- Οι προπτυχιακοί έχουν ένα επιπλέον πεδίο τμήμα
- Οι μεταπτυχιακοί έχουν ένα επιπλέον πεδίο τίτλος μεταπτυχιακού
- Οι διδακτορικοί έχουν ένα επιπλέον πεδίο αριθμός δημοσιεύσεων
- Οι Erasmus έχουν ένα επιπλέον πεδίο χώρα προέλευσης.
- Υπάρχει και η κλάση σχολή που έχει μια λίστα με τους φοιτητές της σχολής

Ενδεικτική λύση

```
public abstract class Student{  
    private String fname,lname,am,phone;  
    public void setFname(String f){fname=f;}  
    public void setLname(String l){lname=l;}  
    public void setPhone(String p){phone=p;}  
    public void setAm(String a){am=a;}  
    public String getFname(){return fname;}  
    .....  
    public abstract void printInfo();  
}
```

Ενδεικτική λύση

```
public class UnderGraduate extends Student{
private String programmeStudy;
public UnderGaduate(String f, String l, String a, String p, String ps){
setFname(f); setLname(l)..... programmeStudy=ps;
}
public void setProgrammaStudy(String p){programmeStudy=p;}
public String getProgrammaStudy (){return programmeStudy;}

public void printInfo(){
System.out.println("undergraduate student"+" "+getFname()+" "+getLname()+"
"+.....+" "+ programmeStudy);
}
}
```

Ενδεικτική λύση

```
public class Graduate extends Student{
private String title;
public Graduate(String f, String l, String a, String p, String t){
setFname(f); setLname(l)..... title=t;
}

public void setTitle(String t){title=t;}
public String getTitle (){return title;}

public void printInfo(){
System.out.println("graduate student"+ " "+getFname()+" "+getLname()+" "+.....+" "+ title);
}
}
```

Ενδεικτική λύση

```
public class Phd extends Student{
private int num;
public Phd(String f, String l, String a, String p, int n){
setFname(f); setLname(l)..... num=n;
}

public void setNum(int n){num=n;}
public int getNum (){return num;}

public void printInfo(){
System.out.println("graduate student"+ " "+getFname()+" "+getLname()+" "+.....+"
"+ num);
}
}
```

Ενδεικτική λύση

```
public class Erasmus extends Student{  
private String country;  
public Erasmus(String f, String l, String a, String p, String c){  
setFname(f); setLname(l)..... country=c;  
}
```

```
public void setCountry(String c){country =c;}  
public String getCountry (){return country;}
```

```
public void printInfo(){  
System.out.println("graduate student"+ " "+getFname()+" "+getLname()+" "+.....+" "+  
country);  
}  
}
```

Ενδεικτική λύση

```
public class Department{
HashMap<String,Student> myMap = new HashMap<String,Student>();
public Department() {
For(int i=0;i<1000;i++){
system.out.println("dose epilog 1-undergraduate 2-graduate 3-phd 3-Erasmus:");
//diabasma epilogis
if ( epilog==1){// undergraduate student diabazoume ta stoixeia
myMap.put(am, new Undergraduate(.....));
if ( epilog==2){// graduate student diabazoume ta stoixeia
myMap.put(am, new graduate(.....));
If .....
}
}
```

Παράδειγμα Exception

Να γράψετε ένα πρόγραμμα το οποίο θα διαβάζει ένα βαθμό από 0 έως 10. Αν δοθεί αριθμός εκτός ορίων θα παράγεται ένα exception τύπου `gradeException` το οποίο θα πρέπει να ορίσετε.

Ενδεικτική λύση

```
class gradeException extends Exception
{
    private int num;

    // default constructor
    MyException() { }

    // parametrized constructor
    MyException(String str) { super("out of limits" +str); num=Integer.parseInt(str); System.out.println(num);}
}

public class mymain{

    public static void main(String[] args)
    {
        try {
            Scanner scan = new Scanner(System.in);
            System.out.print("Enter any number: ");

            // This method reads the number provided using keyboard
            int num = scan.nextInt();
            if ((num < 0) || (num>10))
            {
                gradeException me = new gradeException(num);
                throw me;
            }
        } //end of try

        catch (MyException e) {
            e.printStackTrace();
        } }}}
```

Παράδειγμα

```
class A { ... };  
class B extends A { ...  
//μεθοδος που δεν υπάρχει στην A  
void printClassB(){...} .....};  
B[] bArray = new B[10];  
A[] aArray = bArray;  
A x = new A();  
if ( ... )  
    x = new B();  
bArray[5] = x;
```

1)Είναι σωστό το παρακάτω;

```
A[] aArray = bArray;
```

Είναι σωστό γιατί είναι upcasting

2)Πότε υπάρχει πρόβλημα στην τελευταία γραμμή;

Αν δεν εκτελεστεί το if-body έχουμε downcasting που μπορεί να οδηγήσει σε λάθος(σε περίπτωση που εκτελέσουμε `bArray[5].printClassB();`)

ΠΑΡΑΔΕΙΓΜΑ

```
class A{.....}  
class B extends A{.....}  
class C extends A{.....}  
class D extends C{.....}  
A x=new D();  
x=new C();
```

- Ο στατικός τύπος της x είναι ο A ενώ ο δυναμικός είναι ο C.