



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Οντοκεντρικός Προγραμματισμός

Ενότητα: Εργαστηριακές Ασκήσεις

Ιωάννης Χατζηλυγερούδης, Χρήστος Μακρής

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Εργαστηριακές Ασκήσεις JAVA

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 1

Σκοπός της άσκησης αυτής είναι:

- να δημιουργήσετε ένα νέο αρχείο Java, να εισάγετε κάποιον κώδικα, να τον μεταγλωττίσετε και να δείτε τα αποτελέσματα, με χρήση του περιβάλλοντος BlueJ,
- να μελετήσετε την είσοδο δεδομένων από τη γραμμή εντολής και
- να εξασκηθείτε στη χρήση της εντολής εξόδου στην οθόνη.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project (Project → New Project) και σώστε το (όνομα Askisi1), αφού επιλέξετε τον κατάλογο αποθήκευσης του project. Στην οθόνη θα εμφανιστεί ένα εικονίδιο εγγράφου που παριστάνει το project που μόλις δημιουργήσατε. Δημιουργήστε τώρα μία νέα κλάση στο project, κάνοντας κλικ στο κουμπί "New Class", και ονομάστε την Age. Στη συνέχεια, κάνετε διπλό κλικ πάνω στο εικονίδιο της κλάσης (που ήδη έχει εμφανιστεί στην οθόνη) οπότε και εμφανίζεται ο κώδικας της κλάσης. Αυτός είναι κώδικας που δημιουργείται αυτόματα από το BlueJ. Σβήστε ό,τι υπάρχει και αντικαταστήστε το με τον παρακάτω κώδικα:

```
public class Age
{
    public static void main(String args[])
    {
        if (args.length == 1)
        {
            System.out.println("You are ");
            System.out.println(args[0]);
            System.out.println(" years old.");
        }
    }
}
```

Σημ.: Η εντολή `System.out.println(<παράμετροι>)` της Java χρησιμοποιείται για εκτυπώσεις στην οθόνη.

- 2) Στη συνέχεια πρέπει να σώσετε την κλάση Age. Οπότε, στο παράθυρο της Age επιλέγουμε «Class → Save». Το αρχείο σώζεται σαν «Age.java». Μπορείτε να το δείτε αν πάτε στον κατάλογο που δημιουργήσατε το project. Τώρα μεταφράστε το αρχείο πατώντας την επιλογή `compile`. Αν όλα γίνουν σωστά θα εμφανιστεί το μήνυμα «Class compiled – no syntax errors» στο κάτω μέρος της οθόνης. Τώρα έχει παραχθεί το αρχείο «Age.class». Μπορείτε να το δείτε και αυτό στον κατάλογο του project.
- 3) Τώρα θα δοκιμάσετε να τρέξετε το πρόγραμμά σας. Κλείνετε το παράθυρο της κλάσης Age και πηγαίνετε στο παράθυρο του project. Εκεί, κάνοντας δεξί κλικ στην κλάση Age, επιλέγουμε το «void main(args)» οπότε ανοίγει ένα παράθυρο, όπου πρέπει να δώσετε κάποιο όρισμα τύπου String. Δώστε κάποιο όρισμα (χρειάζονται και οι αγκύλες, οπότε αν θέλετε να δώσετε το 12 θα γράψετε {"12"}). Πατήστε OK. Τι συνέβη; Επαναλάβετε δύο φορές τη διαδικασία για διαφορετικά ορίσματα. Τι γίνεται όταν δεν δοθεί η ηλικία (δηλ. δεν δοθεί όρισμα); Αποτυπώστε τα αποτελέσματα για κάθε περίπτωση στην αναφορά σας.
- 4) Αλλάξτε τις τρεις εντολές εκτύπωσης στην οθόνη με τις παρακάτω (εντοπίσατε τη διαφορά;)

```
System.out.print("You are ");
System.out.print(args[0]);
System.out.println(" years old.");
```

Ξαναμεταφράστε και τρέξτε το πρόγραμμα με όρισμα "12". Καταγράψτε το αποτέλεσμα. Τι εκτυπώνεται τώρα; Σχολιάστε.

- 5) Προσθέστε το παρακάτω τμήμα κώδικα στην κατάλληλη θέση ώστε, όταν δε δίνεται όρισμα να εκτυπώνεται ένα μήνυμα:

```
else System.out.println("You forgot to give your age.");
```

Ξαναμεταφράστε και τρέξτε το πρόγραμμα χωρίς να δώσετε όρισμα, για να διαπιστώσετε ότι λειτουργεί σωστά. Γράψτε το αποτέλεσμα.

- 6) Διαμορφώστε το σώμα της μεθόδου main ως εξής:

```
if (args.length == 1)
{
    System.out.print("You are ");
    System.out.print(args[0]);
    System.out.println(" years old.");
}
else if (args.length == 0)
{
    System.out.println("You forgot to give your age.");
}
else
{
    System.out.print(args[1]);
    System.out.print(" is ");
    System.out.print(args[0]);
    System.out.println(" years old.");
}
```

Ξαναμεταφράστε και τρέξτε το πρόγραμμα δίνοντας δύο ορίσματα αυτή τη φορά, την ηλικία και ένα όνομα, και σημειώστε το αποτέλεσμα. Βεβαιωθείτε ότι το πρόγραμμα εξακολουθεί να δουλεύει και όταν δώσουμε ένα ή κανένα όρισμα. Καταγράψτε τα αποτελέσματα.

- 7) Μπορείτε να κάνετε κάποια αλλαγή στο σώμα της main ώστε να δίνουμε πρώτα το όνομα και μετά την ηλικία; Καταγράψτε την αλλαγή.
- 8) Για να μειώσουμε τις πολλές εντολές System.out.println, χρησιμοποιείται συνήθως ο τελεστής '+'. Π.χ. οι εντολές

```
System.out.print("You are ");
System.out.print(args[0]);
System.out.println(" years old.");
```

μπορούν να αντικατασταθούν από την

```
System.out.print("You are " + args[0] + " years old. ");
```

Αντικαταστήστε και όσες από τις υπόλοιπες εντολές εκτύπωσης είναι δυνατόν να αντικατασταθούν με μια εντολή. Γράψτε το νέο πρόγραμμα. Ξαναμεταφράστε το και τρέξτε το. Βεβαιωθείτε ότι τρέχει όπως και πριν για όλες τις περιπτώσεις. Σημειώστε τα αποτελέσματα.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 2

Σκοπός της άσκησης αυτής είναι:

- a. να δημιουργήσετε ένα νέο αρχείο Java, να εισάγετε κάποιον κώδικα, να τον μεταγλωττίσετε και να δείτε τα αποτελέσματα, με χρήση του περιβάλλοντος BlueJ,
- b. να μελετήσετε τη δημιουργία στιγμιότυπων και την κλήση μεθόδων (αποστολή μηνυμάτων) μέσω αυτών και
- c. να διαπιστώσετε το ρόλο της λέξης-κλειδί "this".

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project (Project → New Project) και σώστε το (όνομα Askisi2), αφού επιλέξετε τον κατάλογο αποθήκευσης του project. Στην οθόνη θα εμφανιστεί ένα εικονίδιο εγγράφου που παριστάνει το project που μόλις δημιουργήσατε. Δημιουργήστε τώρα μία νέα κλάση στο project, κάνοντας κλικ στο κουμπί "New Class", και ονομάστε τη Window. Στη συνέχεια, κάνετε διπλό κλικ πάνω στο εικονίδιο της κλάσης (που ήδη έχει εμφανιστεί στην οθόνη) οπότε και εμφανίζεται ο κώδικας της κλάσης. Αυτός είναι κώδικας που δημιουργείται αυτόματα από το BlueJ. Σβήστε ό,τι υπάρχει και αντικαταστήστε το με τον παρακάτω κώδικα:

```
public class Window
{
    private int size=5;

    public Window ()
    {
    }

    public Window (int x)
    {
        size=x;
    }

    public int getSize()
    {
        return size;
    }

    public static void main(String args[])
    {
        Window w1 = new Window();
        Window w2 = new Window(4);
        Window w3 = new Window(6);
        Window w4 = new Window();
        System.out.println ("The size of window w1 is: " + w1.getSize());
        System.out.println ("The size of window w2 is: " + w2.getSize());
        System.out.println ("The size of window w3 is: " + w3.getSize());
        System.out.println ("The size of window w4 is: " + w4.getSize());
    }
}
```

- 2) Στη συνέχεια πρέπει να σώσετε την κλάση Window. Οπότε, στο παράθυρο της Window επιλέγουμε «Class → Save». Το αρχείο σώζεται σαν «Window.java». Μπορείτε να το δείτε αν πάτε στον κατάλογο που δημιουργήσατε το project. Τώρα μεταφράστε το αρχείο πατώντας την επιλογή compile. Αν όλα γίνουν σωστά θα εμφανιστεί το μήνυμα «Class compiled – no syntax errors» στο κάτω μέρος της οθόνης. Τώρα έχει παραχθεί το αρχείο «Window.class». Μπορείτε να το δείτε και αυτό στον κατάλογο του project.

- 3) Τώρα θα δοκιμάσετε να τρέξετε το πρόγραμμά σας. Κλείνετε το παράθυρο της κλάσης Window και πηγαίνετε στο παράθυρο του project. Εκεί, κάνοντας δεξί κλικ στην κλάση Window, επιλέγουμε το «void main(args)» οπότε ανοίγει ένα παράθυρο, όπου σας ζητά να δώσετε κάποιο όρισμα τύπου String. Μη δώσετε/γράψετε τίποτε, πατήστε OK. Αποτυπώστε τα αποτελέσματα
- 4) Χρησιμοποιείτε την εντολή "this.size = x;" αντί της "size = x;" στο δεύτερο δημιουργό, ξαναμεταφράστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε σε σχέση με τα προηγούμενα αποτελέσματα; Εξηγήστε γιατί τα αποτελέσματα είναι (ή δεν είναι) διαφορετικά. (Έπαιξε ρόλο η εισαγωγή του 'this';)
- 5) Τροποποιήστε το παραπάνω πρόγραμμα ως εξής:

```
public class Window
{
    private int size=5;

    public Window ()
    {
    }

    public Window (int x)
    {
        size=x;
    }

    public int getSize()
    {
        int size = 10;
        return size;
    }

    public static void main(String args[])
    {
        Window w1 = new Window();
        Window w2 = new Window(4);
        System.out.println ("The size of window w1 is: " + w1.getSize());
        System.out.println ("The size of window w2 is: " + w2.getSize());
    }
}
```

Μεταφράστε το και τρέξτε το. Σημειώστε τα αποτελέσματα. Τι παρατηρείτε; Εξηγήστε.

- 6) Αντικαταστήστε την εντολή "return size" με την εντολή "return this.size" στη μέθοδο 'getSize', ξαναμεταφράστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε σε σχέση με τα προηγούμενα αποτελέσματα; Εξηγήστε. (Τι ρόλο παίζει η εισαγωγή του 'this';)

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 3

Σκοπός της άσκησης αυτής είναι

- a. να μελετήσετε τη συμπεριφορά των δημιουργών και όσον αφορά την κληρονομικότητα,
- b. να μελετήσετε την κλήση και εκτέλεση μεθόδων (αποστολή μηνυμάτων) και σε σχέση με την κληρονομικότητα και
- c. να διαπιστώσετε το ρόλο της λέξης-κλειδί "super".

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project (Project → New Project) και σώστε το (όνομα Askisi3), αφού επιλέξετε τον κατάλογο αποθήκευσης του project. Στην οθόνη θα εμφανιστεί ένα εικονίδιο εγγράφου που παριστάνει το project που μόλις δημιουργήσατε.

(α) Δημιουργήστε τώρα μία κλάση στο project και ονομάστε τη **Window**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
public class Window
{
    protected int size;

    public Window()
    {
        size=1;
        System.out.println("Window size="+size);
    }
    public Window(int size)
    {
        this.size=size;
        System.out.println("Window size="+size);
    }

    public void setSize(int x)
    {size += x;}

    public void printSize()
    {System.out.println("Size=" + size);}
}
```

(β) Δημιουργήστε τώρα μία άλλη κλάση στο project και ονομάστε τη **MWindow**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
public class MWindow extends Window
{
    private String message = "No message";

    public MWindow(String message)
    {
        size = 2;
        this.message = message;
        System.out.println ("Window message = " + message);
    }

    public MWindow(int size, String message)
    {
        super(size);
        this.message = message;
        System.out.println ("Window message = " + message);
    }
}
```

```

    public void setSize1(int y)
    {size = y;}

    public void setSize2(int z)
    {super.setSize (z);}

    public void printSize()
    {System.out.println ("MSize="+size);}

    public void printSize1()
    {System.out.println (super.size);}

    public void printSize2()
    {super.printSize();}
}

```

2) Δημιουργήστε μία τρίτη κλάση στο project και ονομάστε τη **RunWindow**.

(α) Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα:

```

public class RunWindow
{
    public static void main (String[] args)
    {
        Window w1=new Window();
        Window w2=new Window(2);
        System.out.println(w1.size);
        System.out.println(w2.size);
    }
}

```

μεταφράστε την κλάση και τρέξτε τη. Αποτυπώστε τα αποτελέσματα και εξηγήστε τα.

(β) Συμπληρώστε τον παραπάνω κώδικα με τις παρακάτω προτάσεις στο τέλος της main:

```

MWindow mw1=new MWindow("First MWindow");
MWindow mw2=new MWindow(3, "Second MWindow");
System.out.println(mw1.size);
System.out.println(mw2.size);

```

Ξαναμεταφράστε και τρέξτε το πρόγραμμα (κλάση RunWindow). Αποτυπώστε τα αποτελέσματα για τις νέες προτάσεις και εξηγήστε τα.

(γ) Στη συνέχεια προσθέστε και τον παρακάτω κώδικα στο τέλος της main:

```

mw1.setSize1(4);
System.out.println(mw1.size);
mw1.setSize2(2);
System.out.println(mw1.size);
mw1.setSize(2);
System.out.println(mw1.size);

```

Ξαναμεταφράστε και τρέξτε το πρόγραμμα. Αποτυπώστε πάλι τα αποτελέσματα για τις νέες προτάσεις και εξηγήστε τα.

3) Παρεμβάλετε στον κώδικα της κλάσης MWindow, αμέσως μετά την πρόταση «private String message = "No message";», την εξής δήλωση:

```

protected int size = 7;

```


Ξαναμεταφράστε την κλάση MWindow και τρέξτε την κλάση RunWindow (δηλ. τη μέθοδο main() της κλάσης). Αποτυπώστε τα αποτελέσματα, συγκρίνετε τα με αυτά στο (2). Εξηγήστε γιατί είναι διαφορετικά (όσα είναι). Πιθανόν να χρειαστεί να αναθεωρήσετε τις εξηγήσεις που δώσατε στο (2).

- 4) Τώρα προσθέστε πάλι στο τέλος της main (κλάση RunWindow) τις παρακάτω προτάσεις:

```
w1.printSize();  
mw1.printSize();  
mw1.printSize1();  
mw1.printSize2();
```

Ξαναμεταφράστε και τρέξτε τη RunWindow. Αποτυπώστε τα νέα αποτελέσματα και εξηγήστε τα.

- 5) Αλλάξτε τις μεθόδους printSize1() και printSize2() της MWindow ως εξής (για την έννοια των νέων εκφράσεων στα ορίσματα των προτάσεων εκτύπωσης ανατρέξτε στις σελίδες 193-194 του βιβλίου “Αντικειμενοστρεφής Προγραμματισμός - Java” του Κ. Θραμπουλίδη):

```
public void printSize1()  
{System.out.println(((Window)this).size);}  
  
public void printSize2()  
{((Window)this).printSize();}
```

Ξαναμεταφράστε και τρέξτε τη RunWindow. Αποτυπώστε τα νέα αποτελέσματα και συγκρίνετε τα με τα αντίστοιχα προηγούμενα. Εξηγήστε.

- 6) Μπορείτε να υλοποιήσετε το παραπάνω πρόγραμμα σε μια κλάση (αρχείο) αντί σε τρία, όπως το κάναμε παραπάνω. Για να το κάνετε αυτό, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi3a). Δημιουργήστε μια κλάση με όνομα RunWindow (διότι αυτή περιέχει τη main και επομένως είναι η πρωτεύουσα κλάση). Αντιγράψτε τους κώδικες από τις τρεις προηγούμενες κλάσεις (αρχεία) σ’ αυτήν με τη σειρά (Window, MWindow, RunWindow). Διαγράψτε τον προσδιοριστή public από τους ορισμούς των Window και MWindow (που δεν ήταν απαραίτητος και προηγουμένως, ελλείψει main), διότι μόνο μια κλάση (η πρωτεύουσα) επιτρέπεται/πρέπει να είναι public σ’ ένα αρχείο. Μεταφράστε και τρέξτε τη κλάση RunWindow. Βεβαιωθείτε ότι παράγονται τα ίδια ακριβώς αποτελέσματα. Μια διαφορά μεταξύ των δύο υλοποιήσεων είναι ότι πλέον στην απεικόνιση του project βλέπετε μόνο μια κλάση και όχι τρεις (με τις συνδέσεις τους), όπως πριν.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 4

Σκοπός της άσκησης αυτής είναι

- να διαπιστώσετε την αναγκαιότητα και να εξασκηθείτε στη συγγραφή μεθόδων,
- να διαπιστώσετε την αναγκαιότητα και τη χρήση της "υπέρβασης μεθόδων" και
- να γνωρίσετε το πλαίσιο Συλλογών της Java (*Java Collections Framework*) και τις νέες βελτιώσεις που έχουν γίνει σε αυτό (*Generics* και επαυξημένη μορφή "for")

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi4), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

(α) Δημιουργήστε κατ' αρχήν μία κλάση στο project και ονομάστε τη **Person**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
public class Person {
    private String lastname;
    private String firstname;
    private int age;
    private boolean married;
    private float salary;

    public Person (String lastname, String firstname, int age, boolean married,
float salary) {
        this.lastname = new String(lastname);
        this.firstname = new String(firstname);
        this.age = age;
        this.married = married;
        this.salary = salary;
    }

    public String getLastName() { return lastname; }
    public String getFirstName() { return firstname; }
    public int getAge() { return age; }
    public boolean isMarried() { return married; }
    public float getSalary() { return salary; }
}
```

(β) Δημιουργήστε μία δεύτερη κλάση στο project και ονομάστε τη **MarriedPerson**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
public class MarriedPerson extends Person {
    private int children;

    public MarriedPerson(String lastname, String firstname, int age, float
salary, int children) {
        super(lastname, firstname, age, true, salary);
        this.children = children;
    }

    public int getNoOfChildren() { return children; }
}
```

(γ) Δημιουργήστε τέλος την κλάση **MyTester**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```

abstract class MyTester {
    public static void main(String[] args) {
        Person p1 = new Person("Kings", "Paul", 22, true, 1200f);
        MarriedPerson mp1 = new MarriedPerson("Tront", "Betty", 31,
        980.5f, 3);
        System.out.print(p1.getFirstName()+" "+p1.getLastName()+" is "
        +p1.getAge()+" years old, gets a "+p1.getSalary()
        +" Euros salary and is");
        if (p1.isMarried() == false)
            System.out.print(" not");
        System.out.println(" married.");
        System.out.print(mp1.getFirstName()+" "+mp1.getLastName()
        +" is " +mp1.getAge()+ " years old, gets a " + mp1.getSalary()
        +" Euros salary and is" + " married with ");
        if (mp1.getNoOfChildren() > 0)
            System.out.print(mp1.getNoOfChildren());
        else System.out.print("no");
        System.out.println(" children.");
    }
}

```

Γιατί ορίσαμε την κλάση MyTester ως abstract; Είναι απαραίτητο;

Τρέξτε την MyTester και σημειώστε τα αποτελέσματα. Σχολιάστε. Τι εκτυπώσαμε με τον παραπάνω κώδικα;

- 2) Κατασκευάστε μια μέθοδο printInfo στην κλάση Person που να εκτυπώνει όλη την πληροφορία που διαθέτουμε (το σώμα της θα είναι παρόμοιο με τον κώδικα στην MyTester που αφορά το p1). Αντικαταστήστε τώρα τη main της MyTester με τον ακόλουθο κώδικα:

```

Person p1 = new Person("Kings", "Paul", 22, true, 1200f);
MarriedPerson mp1 = new MarriedPerson("Tront", "Betty", 31, 980.5f, 3);
p1.printInfo();
mp1.printInfo();

```

Ποιά είναι τα πλεονεκτήματα αυτού του κώδικα έναντι του προηγούμενου;

Τρέξτε τώρα ξανά την MyTester και συγκρίνετε τα αποτελέσματα με αυτά που εκτυπώθηκαν πριν. Τι παρατηρείτε; Λείπει κάτι;

- 3) Υπερκαλύψτε τη μέθοδο printInfo της Person μέσα στην MarriedPerson (δηλ. γράψτε μια νέα μέθοδο με την ίδια επικεφαλίδα), ώστε να εκτυπώνει και την πληροφορία που λείπει (αν θέλετε, εξετάστε την περίπτωση να κάνετε χρήση του super, για να μην επαναλαμβάνετε ίδιο κώδικα). Τρέξτε και πάλι τη MyTester και σημειώστε τις παρατηρήσεις σας (μήπως ξεχάσατε να την ξαναμεταφράσετε;;;).

- 4) (α) Προσθέστε στην Person τις εξής μεταβλητές:

```

public static final int MALE = 0;
public static final int FEMALE = 1;
private int sex;

```

(β) Αλλάξτε κατάλληλα τους δημιουργούς στις κλάσεις Person και MarriedPerson έτσι, ώστε να αρχικοποιείται και η νέα μεταβλητή ως εξής:

```

public Person (String lastname, String firstname, int age, boolean married,
float salary, int sex) {
    this.lastname = new String(lastname);
    this.firstname = new String(firstname);
}

```

```

        this.age = age;
        this.married = married;
        this.salary = salary;
        this.sex = sex;
    }

    public MarriedPerson(String lastname, String firstname, int age, float salary, int sex, int children) {
        super(lastname, firstname, age, true, salary, sex);
        this.children = children;
    }

```

(γ) Αντικαταστήστε τώρα το σώμα της main στη MyTester με τον ακόλουθο κώδικα:

```

Person p1 = new Person("Kings", "Paul", 22, true, 1200f, Person.MALE);
MarriedPerson mp1 = new MarriedPerson("Tront", "Betty", 31, 980.5f,
Person.FEMALE, 3);
p1.printInfo();
mp1.printInfo();

```

(δ) Τρέξτε την και σημειώστε τα αποτελέσματα. Τι παρατηρείτε;

(ε) Τροποποιήστε κατάλληλα τις μεθόδους printInfo και ξανατρέξτε τη MyTester. Σημειώστε και πάλι τα αποτελέσματα.

- 5) (α) Κατασκευάστε σε κάθε κλάση για κάθε μέθοδο get ή is μια αντίστοιχη μέθοδο set (για ανάθεση τιμής στην αντίστοιχη μεταβλητή). Έτσι, μπορείτε πλέον να αλλάζετε τις τιμές των χαρακτηριστικών των στιγμιότυπων (μέσω κλήσης των μεθόδων αυτών).

(β) Αντικαταστήστε το σώμα της main στη MyTester με τον ακόλουθο κώδικα:

```

Person p1 = new Person("Kings", "Paul", 22, true, 1200f, Person.MALE);
MarriedPerson mp1 = new MarriedPerson("Tront", "Betty", 31, 980.5f,
Person.FEMALE, 3);
p1.setMarried(false);
mp1.setMarried(false);
p1.printInfo();
mp1.printInfo();
System.out.println("Is mp1 married?: "+mp1.isMarried());

```

Τρέξτε την και σημειώστε τα αποτελέσματα. Τι παρατηρείτε;

(γ) Προσοχή! Η μεταβλητή married για στιγμιότυπα της κλάσης MarriedPerson δεν πρέπει να αλλάζει! Υπερκαλύψτε κατάλληλα τη μέθοδο setMarried στην κλάση MarriedPerson για να μη συμβαίνει αυτό. Τρέξτε πάλι τη MyTester για να δείτε αν το καταφέρατε. Σημειώστε τι κάνατε.

- 6) Προσθέστε μία μέθοδο setSalary και στην κλάση MarriedPerson η οποία θα παίρνει σαν όρισμα μια μεταβλητή spouse του τύπου MarriedPerson και θα κάνει τα εξής:

(α) Θα ελέγχει το φύλο της spouse.

(β) Αν είναι αντίθετο από τη μεταβλητή sex, τότε στην τιμή της μεταβλητής salary θα προστίθεται και η αντίστοιχη τιμή του στιγμιότυπου spouse.

(γ) Αν είναι ίδιο δε θα κάνει τίποτα.

Αντικαταστήστε το σώμα της main της MyTester με τον ακόλουθο κώδικα:

```

MarriedPerson mp1 = new MarriedPerson("Tront", "Betty", 31, 980.5f,
Person.FEMALE, 3);
MarriedPerson mp2 = new MarriedPerson("Tront", "Kirk", 31, 2080f,
Person.MALE, 2);
MarriedPerson mp3 = new MarriedPerson("Tront", "Sonia", 31, 600f,
Person.FEMALE, 0);
mp1.printInfo();
mp2.printInfo();
mp3.printInfo();
mp1.setSalary(mp2);
mp1.printInfo();
mp1.setSalary(mp3);
mp1.printInfo();
mp1.setSalary(mp2.getSalary());
mp1.printInfo();

```

Τρέξτε την και σημειώστε τα αποτελέσματα. Τι παρατηρείτε;

Η μέθοδος `setSalary` που γράψατε τι σχέση έχει με την μέθοδο `setSalary` της κλάσης `Person`; Δικαιολογήστε.

- 7) Μια *Λίστα* (List) είναι μια δομή δεδομένων στην Java που επιτρέπει την ομαδοποίηση στιγμιοτύπων οποιουδήποτε τύπου. Μια υλοποίηση λίστας είναι η κλάση `ArrayList`, του πακέτου `java.util`, η οποία παρέχει μεταξύ άλλων τις μεθόδους `add`, `remove` και `get` για την προσθήκη, διαγραφή και ανάκτηση αντίστοιχα στοιχείων από τη λίστα.

(α) Αρχικά, δημιουργήστε έναν πίνακα (array) με όνομα `mpArray` και προσθέστε σε αυτόν τα δύο πρώτα στοιχεία της κλάσης `MarriedPerson` που δημιουργήσατε στο προηγούμενο ερώτημα.

Μπορείτε να προσθέσετε εκ των υστέρων και το τρίτο στιγμιότυπο; Για ποιο λόγο;

Με χρήση βρόχου `for` προσπελάστε τα στοιχεία του πίνακα και καλέστε την `printInfo()` για κάθε ένα από αυτά.

(β) Δημιουργήστε τώρα ένα στιγμιότυπο της κλάσης `ArrayList` με όνομα `mpList`.

Προσθέστε στη λίστα που μόλις δημιουργήσατε τα τρία στιγμιότυπα της κλάσης `MarriedPerson` που δημιουργήσατε στο προηγούμενο ερώτημα. Τι πλεονέκτημα παρατηρείτε σε σχέση με τη χρήση πίνακα;

(γ) Χρησιμοποιώντας βρόχο επανάληψης `while` διαπεράστε όλα τα στοιχεία της `mpList` και για καθένα καλέστε τη μέθοδο `printInfo()`. Για να το πετύχετε αυτό, δημιουργήστε ένα στιγμιότυπο `Iterator` για τη λίστα και χρησιμοποιήστε τις μεθόδους του, `hasNext` και `next`:

```

Iterator iter = mpList.iterator();

while(iter.hasNext()){
    MarriedPerson mp = iter.next();
    mp.printInfo();
}

```

Εξηγείστε το σφάλμα που βγάζει ο μεταγλωττιστής.

Διορθώστε το σφάλμα ορίζοντας ρητά τον τύπο του αντικειμένου `mp` (κάνοντας δηλαδή *cast*):

```

MarriedPerson mp = (MarriedPerson)iter.next();

```

(δ) Από τη Java 5.0 και μετά το πρόβλημα αυτό λύνεται με τη χρήση *γενικεύσεων* (generics). Με τις γενικεύσεις, μπορούμε να ορίσουμε π.χ. μια Λίστα, δίνοντας τον τύπο των περιεχομένων της ως παράμετρο:

```
List<MarriedPerson> mpList = new ArrayList<MarriedPerson>();
```

Δημιουργήστε την mpList δίνοντας τον τύπο ως παράμετρο. Επιβεβαιώστε ότι αυτή τη φορά το cast δεν είναι απαραίτητο.

(ε) Διαπεράστε τα στοιχεία της mpList χωρίς τη χρήση iterator, αλλά χρησιμοποιώντας την *επαυξημένη μορφή* της "for":

```
// obj: Αντικείμενο τύπου "Object"
// collection: Το όνομα της συλλογής που διαπερνάμε
for (Object obj : collection)
{
    // ... κώδικας

    // Παράδειγμα εκτύπωσης των αντικειμένων της συλλογής "collection"
    System.out.println("This is object: " + obj );
}
```

Υπάρχει κάποια περίπτωση που να είναι αναγκαίος ο iterator;

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 5

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε

- a. με τον τρόπο με τον οποίο η Java δέχεται δεδομένα από το χρήστη, και
- b. με τον τρόπο με τον οποίο μπορούμε να γράψουμε σε ή να διαβάσουμε από ένα αρχείο στη Java.

1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi5), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

(α) Δημιουργήστε κατ' αρχήν μία κλάση στο project και ονομάστε τη **IO_Tester**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
import java.io.*;

public class IO_Tester {

    public static int readInt() {
        byte b[] = new byte[16];
        String str;

        System.in.read(b);
        str = (new String(b)).trim();
        return Integer.parseInt(str);
    }
}
```

(β) Η μετάφραση έγινε σωστά ή ο μεταφραστής έβγαλε κάποιο μήνυμα λάθους; Στη δεύτερη περίπτωση σημειώστε το μήνυμα του μεταφραστή.

(γ) Αντικαταστήστε, τώρα, το σώμα της readInt με τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη ξανά.

```
byte b[] = new byte[16];
String str;

try {
    System.in.read(b);
    str = (new String(b)).trim();
    return Integer.parseInt(str);
} catch (IOException e) {
    System.out.println("Exception: " + e.toString());
    return 0;
}
```

Βγάζει τώρα λάθος ο μεταφραστής;

(δ) Προσθέστε τώρα στην κλάση τις ακόλουθες μεθόδους, σώστε τη και μεταφράστε τη.

```
public static float readFloat() {
    byte b[] = new byte[16];
    String str;

    try {
        System.in.read(b);
        str = (new String(b)).trim();
    }
```

```

        return Float.parseFloat(str);
    } catch (IOException e) {
        System.out.println("Exception: " + e.toString());
        return 0;
    }
}

public static String readString() {
    byte b[] = new byte[32];
    String str;

    try {
        System.in.read(b);
        str = (new String(b)).trim();
        return str;
    } catch (IOException e) {
        System.out.println("Exception: " + e.toString());
        return "";
    }
}

public static boolean readBoolean() {
    byte b[] = new byte[16];
    String str;

    try {
        System.in.read(b);
        str = (new String(b)).trim();
        Boolean bool = new Boolean(str);
        return bool.booleanValue();
    } catch (IOException e) {
        System.out.println("Exception: " + e.toString());
        return false;
    }
}

```

Αν ο μεταφραστής βγάλει κάποιο λάθος διορθώστε κατάλληλα το πρόγραμμά σας, ώστε η μετάφραση να γίνεται χωρίς κανένα λάθος.

- 2) Κατασκευάστε τη μέθοδο main στην κλάση IO_Tester, ώστε να περιέχει το παρακάτω σώμα:

```

System.out.println("Dwste enan akeraio:");
int i = readInt();
System.out.println("Dwste enan pragmatiko:");
float f = readFloat();
System.out.println("i="+i+"\tf="+f);

```

Σώστε και μεταφράστε την κλάση IO_Tester.

Τρέξτε τώρα την IO_Tester δύο φορές, δίνοντας τις τιμές '3' και '2.2' την πρώτη και 'zzz' και '9' τη δεύτερη και σημειώστε το αποτέλεσμα κάθε εκτέλεσης.

- 3) Αντικαταστήστε την readInt() με την παρακάτω και μεταφράστε την κλάση ξανά.

```

public static int readInt() {
    byte b[] = new byte[16];
    String str;

    try {
        System.in.read(b);

```



```

        str = (new String(b)).trim();
        return Integer.parseInt(str);
    } catch (IOException e) {
        System.out.println("Exception: " + e.toString());
        return 0;
    } catch (NumberFormatException e) {
        System.out.println("Exception: " + e.toString() + "\nReturned
        value: -1");
        return -1;
    }
}

```

Τρέξτε τη πάλι για τις τιμές 'zzz' και '9'. Τι παρατηρείτε;

Προσθέστε τον κατάλληλο κώδικα στη μέθοδο readFloat (παρόμοιο με αυτόν της readInt) έτσι, ώστε να αντιμετωπίζονται παρόμοια προβλήματα εξαιρέσης. Μεταφράστε το πρόγραμμα. Τρέξτε τη πάλι για τις τιμές 'zzz' και 'γγγ' και σημειώστε το αποτέλεσμα.

- 4) Αντικαταστήστε το σώμα της main με τις ακόλουθες εντολές και μεταφράστε το πρόγραμμα.

```

System.out.println("Dwste enan akeraio:");
int i = readInt();
System.out.println("Dwste enan float:");
float f = readFloat();
System.out.println("Dwste ena string:");
String s = readString();
System.out.println("Dwste mia boolean:");
boolean b = readBoolean();
System.out.println("i="+i+"\tf="+f+"\ts="+s+"\tb="+b);

```

Τρέξτε τρεις φορές το πρόγραμμα για τις ακόλουθες τιμές:

```

'4' '10' 'xch' 'True'
'number' '4.5f' 'some_text' 'true_again'
'-1' 'ff' '34' '12'

```

Σημειώστε αυτό που εκτυπώνεται στην οθόνη κάθε φορά και δικαιολογήστε το.

- 5) Προσθέστε στο τέλος της main τον ακόλουθο κώδικα, σώστε την κλάση και μεταφράστε τη.

```

FileOutputStream fil = new FileOutputStream(s);
BufferedOutputStream buf = new BufferedOutputStream(fil);
DataOutputStream dat = new DataOutputStream(buf);

```

(α) Έγινε σωστά η μετάφραση ή όχι. Αν ο μεταφραστής έβγαλε κάποιο λάθος σημειώστε το.

(β) Αντικαταστήστε τον παραπάνω κώδικα με τον ακόλουθο, σώστε την κλάση και μεταφράστε τη πάλι.

```

try {
    FileOutputStream fil = new FileOutputStream(s);
    BufferedOutputStream buf = new BufferedOutputStream(fil);
    DataOutputStream dat = new DataOutputStream(buf);
} catch (FileNotFoundException e) {}

```

Τι παρατηρείτε τώρα; Η μετάφραση έγινε σωστά ή προέκυψε κάποιο λάθος πάλι;

(γ) Αντικαταστήστε πάλι τον παραπάνω κώδικα με τον παρακάτω, σώστε την κλάση και μεταφράστε τη.

```

try {
    FileOutputStream fil = new FileOutputStream(s);
    BufferedOutputStream buf = new BufferedOutputStream(fil);
    DataOutputStream dat = new DataOutputStream(buf);
    dat.writeInt(i);
    dat.writeFloat(f);
    dat.writeBoolean(b);
    dat.close();
} catch (FileNotFoundException e) {}

```

Η μετάφραση έγινε σωστά;

(δ) Προσθέστε την ακόλουθη εντολή στο τέλος της main, σώστε την κλάση και μεταφράστε τη.

```

catch (IOException e) {}

```

Τρέξτε το πρόγραμμα για τα εξής δεδομένα και σημειώστε τα αποτελέσματα.

```
'10' '.25e-2' 'test.log' 'true'
```

(ε) Κοιτάξτε στον κατάλογο του project για το αρχείο test.log. Αν υπάρχει, ανοίξτε το και καταγράψτε τα περιεχόμενά του. Τι παρατηρείτε; Μπορείτε να βγάλετε κάποιο νόημα; Σχολιάστε.

6) (α) Προσθέστε τον παρακάτω κώδικα στο τέλος της main, σώστε την IO_Tester και μεταφράστε τη.

```

try {
    FileInputStream fil = new FileInputStream(s);
    BufferedInputStream buf = new BufferedInputStream(fil);
    DataInputStream dat = new DataInputStream(buf);
    i+=dat.readInt();
    f*=dat.readFloat();
    b=!dat.readBoolean();
    dat.close();
} catch (FileNotFoundException e) {}
catch (IOException e) {}
System.out.println("i="+i+"\tf="+f+"\ts="+s+"\tb="+b);

```

(β) Τρέξτε πάλι το πρόγραμμα, για τα ίδια με πριν δεδομένα, και σημειώστε τα καινούρια αποτελέσματα. Σχολιάστε. Τι κάνει ο κώδικας που προστέθηκε;

(γ) Προσθέστε τον ακόλουθο κώδικα στην main, σώστε και μεταφράστε την κλάση. Βεβαιωθείτε ότι η μετάφραση έγινε σωστά.

```

try {
    FileWriter fw = new FileWriter(s);
    BufferedWriter bw = new BufferedWriter(fw);
    bw.write(Integer.toString(i), 0, Integer.toString(i).length());
    bw.write(Float.toString(f), 0, Float.toString(f).length());
    bw.write(s.concat(s), 0, s.concat(s).length());
    bw.write(Boolean.toString(b), 0, Boolean.toString(b).length());
    bw.close();
} catch (IOException e) { System.out.println(e.toString()); }

```

(δ) Ανοίξτε το αρχείο test.log και καταγράψτε τα περιεχόμενά του, αφού το τρέξετε πάλι για τα ίδια δεδομένα. Τι κάνει ο παραπάνω κώδικας; Σχολιάστε.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 6

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με τις εξαιρέσεις στην Java.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi6), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

(α) Δημιουργήστε κατ' αρχήν μία κλάση στο project και ονομάστε τη **Exception_Tester**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
public class Exception_Tester {  
  
    public static void main(String args[]) {  
        int x1;  
        int x2;  
        x1 = 5;  
        x2 = 2;  
        printResults(x1, x2);  
    }  
  
    static void printResults(int a, int b) {  
        System.out.println("Add: "+(a+b));  
        System.out.println("Sub: "+(a-b));  
        System.out.println("Mul: "+(a*b));  
        System.out.println("Div: "+(a/b));  
    }  
}
```

(β) Τρέξτε τη και σημειώστε τα αποτελέσματα.

(γ) Αρχικοποιήστε τώρα τη x2 με την τιμή 0, μεταφράστε και εκτελέστε πάλι το πρόγραμμα. Τι αποτελέσματα βγάζει; Τι παρατηρείτε;

- 2) Αντικαταστήστε τη γραμμή:

```
System.out.println("Div: "+(a/b));
```

στην printResults με τον ακόλουθο κώδικα:

```
try {  
    System.out.println("Div: "+(a/b));  
}  
catch (ArithmeticException ae) {  
    System.out.println(ae.toString());  
}
```

Σώστε και μεταφράστε την κλάση.

Τρέξτε την τώρα και σημειώστε πάλι τα αποτελέσματα. Τι παρατηρείτε;

- 3) Αντικαταστήστε τον κώδικα της main με τον ακόλουθο:

```
public static void main(String args[]) {  
    Exception_Tester et = new Exception_Tester();  
    int x1;  
    int x2;  
    x1 = 5;  
    x2 = 0;
```

```

        try {
            et.printResults(x1, x2);
        }
        catch (ArithmeticException ae) {
            System.out.println(ae.toString());
        }
    }
}

```

και τον κώδικα της printResults με τον ακόλουθο

```

void printResults(int a, int b) {
    System.out.println("Add: "+(a+b));
    System.out.println("Sub: "+(a-b));
    System.out.println("Mul: "+(a*b));
    System.out.println("Div: "+(a/b));
}

```

Μεταφράστε τη και τρέξτε τη πάλι. Άλλαξαν καθόλου τα αποτελέσματα; Τι άλλαξε ο προγραμματιστής εδώ, με το χειρισμό της εξαίρεσης;

4) Αντικαταστήστε τον κώδικα της main ως εξής:

```

public static void main(String args[]) {
    Exception_Tester et = new Exception_Tester();
    int x1;
    int x2;
    x1 = 5;
    x2 = 0;
    et.printResults(x1, x2);
}

```

και τον κώδικα της printResults ως εξής:

```

void printResults(int a, int b) throws ArithmeticException {
    System.out.println("Add: "+(a+b));
    System.out.println("Sub: "+(a-b));
    System.out.println("Mul: "+(a*b));
    System.out.println("Div: "+(a/b));
}

```

Μεταφράστε και εκτελέστε το πρόγραμμα. Τι δεν πρόσεξε εδώ ο προγραμματιστής; Διορθώστε κατάλληλα τον κώδικα. Βεβαιωθείτε ότι λειτουργεί σωστά τώρα.

5) (α) Αντικαταστήστε τον κώδικα της main με τον ακόλουθο κώδικα, σώστε την κλάση και μεταφράστε τη.

```

public static void main(String args[]) {

    Exception_Tester et = new Exception_Tester();
    int x1;
    int x2;
    x1 = 5;
    x2 = 0;

    try {
        et.printResults(x1, x2);
    }
    catch (ArithmeticException ae) {
        System.out.println(ae.toString());
    }
}

```

```

        finally {
            System.out.println("The numbers are: "+x1+" "+x2);
        }
    }
}

```

Μεταφράστε την κλάση και τρέξτε τη. Σημειώστε τα αποτελέσματα.

(β) Αρχικοποιήστε την x2 με την τιμή 2 και πάλι. Μεταφράστε την κλάση, τρέξτε τη και σημειώστε τα νέα αποτελέσματα.

(γ) Σημειώστε την παρατήρησή σας σχετικά με τα αποτελέσματα στο (α) και στο (β) και την παρουσία της πρότασης finally.

6) (α) Προσθέστε στο τέλος της printResults τη γραμμή:

```
throw new ArithmeticException();
```

Τρέξτε πάλι το πρόγραμμα και σημειώστε τα καινούρια αποτελέσματα. Σχολιάστε. Τι κάνει ο κώδικας που προστέθηκε;

(β) Προσθέστε μετά την γραμμή:

```
System.out.println(ae.toString());
```

τον ακόλουθο κώδικα:

```
ae.printStackTrace();
```

Μεταφράστε και τρέξτε την κλάση. Τι παρατηρείτε;

7) Δημιουργήστε μία νέα κλάση DivideByZeroException, με τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη.

```

public class DivideByZeroException extends ArithmeticException {

    public DivideByZeroException() {}
    public DivideByZeroException(String msg) {super(msg);}
}

```

(α) Αλλάξτε τις γραμμές:

```

void printResults(int a, int b) throws ArithmeticException {

    throw new ArithmeticException();
}

```

της μεθόδου printResults, με τις εξής (αντίστοιχα):

```

void printResults(int a, int b) throws DivideByZeroException {

    throw new DivideByZeroException();
}

```

Άλλαξε κάτι στα αποτελέσματα; Δικαιολογήστε.

(β) Αλλάξτε τον ακόλουθο κώδικα της main:

```

catch (ArithmeticException ae) {
    System.out.println(ae.toString());
    ae.printStackTrace();
}

```

με τον επόμενο κώδικα:

```
catch (DivideByZeroException dbz) {  
    System.out.println(dbz.toString());  
    dbz.printStackTrace();  
}
```

Προσθέστε στην κλάση DivideByZeroException την εξής μέθοδο:

```
public String toString() {  
    return "DivideByZeroException: The denominator cannot be zero.";  
}
```

Σώστε τις κλάσεις, μεταφράστε τις, τρέξτε το πρόγραμμα και σημειώστε τα αποτελέσματα.

(γ) Αλλάξτε τώρα την τιμή της x2 από 2 σε 0 και πάλι. Σώστε το πρόγραμμα, μεταφράστε το και τρέξτε το. Τι συμβαίνει; Σχολιάστε.

(δ) Αντικαταστήστε τον κώδικα της printResults με τον ακόλουθο:

```
void printResults(int a, int b) throws DivideByZeroException {  
    System.out.println("Add: "+(a+b));  
    System.out.println("Sub: "+(a-b));  
    System.out.println("Mul: "+(a*b));  
    try {  
        System.out.println("Div: "+(a/b));  
    }  
    catch (ArithmeticException ae) {  
        throw new DivideByZeroException(ae.toString());  
    }  
}
```

Σώστε την κλάση, μεταφράστε τη και σημειώστε τα αποτελέσματα. Σχολιάστε.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 7

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με το γραφικό περιβάλλον της Java.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi7), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

Δημιουργήστε μία κλάση **MyAppBord**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```
import java.awt.*;
import java.awt.event.*;

public class MyAppBord extends Frame {
    public MyAppBord (String title) {
        super(title);
        Font f1 = new Font("Helvetica", Font.PLAIN, 10);
        setFont(f1);
        BorderLayout lay1 = new BorderLayout(3,2);
        setLayout(lay1);
        add("North", new Button("North"));
        add("South", new Button("South"));
        add("East", new Button("East"));
        add("West", new Button("West"));
        add("Center", new Button("Center"));

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent evt) {
                System.exit(0);
            }
        });
    }

    public static void main(String[] args){
        MyAppBord app2 = new MyAppBord("Application Window");
        app2.pack();
        // app2.setBounds(50, 50, 300, 300);
        app2.setVisible(true);
    }
}
```

(α) Τρέξτε τη και σημειώστε το αποτέλεσμα.

(β) Βγάλτε τα σχόλια από την main και τρέξτε ξανά. Σημειώστε το αποτέλεσμα.

- 2) Δημιουργήστε μία κλάση **MyAppButton**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη. Προσοχή: θα πρέπει να δημιουργήσετε ένα αρχείο με το όνομα icon_01.jpg στο φάκελο που βρίσκεται και το αρχείο της κλάσης.

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MyAppButton extends Frame {
    public MyAppButton (String title) {
        super(title);
        Font f1 = new Font("Tahoma", Font.PLAIN, 14);
```

```

        setFont(f1);
        FlowLayout lay1 = new FlowLayout(FlowLayout.LEFT);
        setLayout(lay1);
        add(new Button("Previous"));
        add(new Button("Stop"));
        add(new Button("Play"));
        add(new Button("Pause"));
        add(new Button("Next"));
        ImageIcon ic1 = createImageIcon("icon_01.jpg");
        add(new JButton(ic1));

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent evt) {
                System.exit(0);
            }
        });
    }

    /** Returns an ImageIcon, or null if the path was invalid. */
    protected static ImageIcon createImageIcon(String path) {
        java.net.URL imgURL = MyAppButton.class.getResource(path);
        if (imgURL != null) {
            return new ImageIcon(imgURL);
        } else {
            System.err.println("Couldn't find file: " + path);
            return null;
        }
    }

    public static void main(String[] args){
        MyAppButton app2 = new MyAppButton("Application Window");
        // GridLayout lay2 = new GridLayout(3,2); app2.setLayout(lay2);
        app2.setSize(400, 200);
        app2.setVisible(true);
    }
}

```

(α) Τρέξτε τη και δοκιμάστε να κάνετε resize το παράθυρο. Σημειώστε τι παρατηρείτε.

(β) Βγάλτε τα σχόλια από τη main, τρέξτε ξανά και δοκιμάστε να κάνετε resize το παράθυρο. Σημειώστε το αποτέλεσμα.

- 3) Δημιουργήστε μία κλάση **MyAppCheck**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MyAppCheck extends Frame {
    public MyAppCheck (String title) {
        super(title);
        setFont(new Font("Verdana", Font.BOLD, 12));
        setLayout(new GridLayout(1,3));

        Panel cboxes = new Panel();
        cboxes.setLayout(new GridLayout(0,1));
        Color c = new Color(80,120,230);
        cboxes.setBackground(c);
        Checkbox c1 = new Checkbox("Windows");
    }
}

```



```

Checkbox c2 = new Checkbox("Linux");
Checkbox c3 = new Checkbox("IRIX");
Checkbox c4 = new Checkbox("MacOS");
cboxes.add(c1);
cboxes.add(c2);
cboxes.add(c3);
cboxes.add(c4);
c2.setState(true);
c4.setState(true);

JPanel cboxgrp = new JPanel();
cboxgrp.setLayout(new GridLayout(0,1));
cboxgrp.setBackground(Color.YELLOW);
cboxgrp.setBorder(BorderFactory.createTitledBorder("Languages"));
CheckboxGroup progLang = new CheckboxGroup();
Checkbox cg1 = new Checkbox("Pascal", false, progLang);
Checkbox cg2 = new Checkbox("Java", false, progLang);
Checkbox cg3 = new Checkbox("Basic", false, progLang);
Checkbox cg4 = new Checkbox("C", false, progLang);
Checkbox cg5 = new Checkbox("C++", false, progLang);
cboxgrp.add(cg1);
cboxgrp.add(cg2);
cboxgrp.add(cg3);
cboxgrp.add(cg4);
cboxgrp.add(cg5);
cg3.setState(true);

add(cboxes);
add(cboxgrp);

addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent evt) {
        System.exit(0);
    }
});
}

public static void main(String[] args){
    MyAppCheck app2 = new MyAppCheck("Application Window");
    app2.setSize(200, 200);
    app2.setVisible(true);
}
}

```

Τρέξτε τη και σημειώστε το αποτέλεσμα. Τι παρατηρείτε για τα δύο panels; (τόσο από θέμα στιγμιότυπου όσο και εμφάνισης)

- 4) Δημιουργήστε μία κλάση **MyAppListLabel**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import java.awt.event.*;

public class MyAppListsLabel extends Frame {
    public MyAppListsLabel (String title) {
        super(title);
        setFont(new Font("Tahoma", Font.ITALIC, 10));
        setLayout(new GridLayout(2,2,10,10));

        Label lb = new Label("Choose Language");
        add(lb);
    }
}

```

```

        Choice progLang = new Choice();
        progLang.add("Pascal");
        progLang.add("Java");
        progLang.add("Basic");
        progLang.add("C");
        progLang.add("C++");
        add(progLang);

        add(new Label("Choose Operating System"));
        List l = new List(4, true);
        l.add("Windows");
        l.add("Linux");
        l.add("IRIX");
        l.add("MacOS");
        add(l);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                System.exit(0);
            }
        });
    }

    public static void main(String[] args) {
        MyAppListsLabel app2 = new MyAppListsLabel("Application Window");
        app2.setSize(200, 200);
        app2.setVisible(true);
    }
}

```

Τρέξτε τη και σημειώστε το αποτέλεσμα.

- 5) Δημιουργήστε μία κλάση **MyAppText**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MyAppText extends Frame {
    public MyAppText (String title) {
        super(title);
        setFont(new Font("Verdana", Font.BOLD, 14));
        setLayout(new BorderLayout(5,5));

        Label appName = new Label("Application name: ", Label.RIGHT);
        TextField app = new TextField(20);

        JLabel userName = new JLabel("Username: ", SwingConstants.RIGHT);
        JPasswordField user = new JPasswordField(20);

        Label passWord1 = new Label("Password awt: ");
        TextField pwd1 = new TextField(20);
        pwd1.setEchoChar('#');

        JLabel passWord2 = new JLabel("Password swing: ");
        JPasswordField pwd2 = new JPasswordField(20);

        Label comments1 = new Label("Comments awt: ", Label.RIGHT);
        TextArea areal = new TextArea("Default awt...", 5, 20,
            TextArea.SCROLLBARS_BOTH);
    }
}

```

```

JLabel comments2 = new JLabel("Comments swing: ",
    SwingConstants.RIGHT);
JTextArea area2 = new JTextArea("Default swing...",5, 20);
JScrollPane pane = new
JScrollPane(area2,ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED);

Panel MyUpPanel = new Panel();
MyUpPanel.setLayout(new GridLayout(0,2,5,5));
Panel MyDownPanel = new Panel();
MyDownPanel.setLayout(new GridLayout(0,2,5,5));

MyUpPanel.add(appName);
MyUpPanel.add(app);
MyUpPanel.add(userName);
MyUpPanel.add(user);
MyUpPanel.add(passWord1);
MyUpPanel.add(pwd1);
MyUpPanel.add(passWord2);
MyUpPanel.add(pwd2);
MyDownPanel.add(comments1);
MyDownPanel.add(area1);
MyDownPanel.add(comments2);
MyDownPanel.add(pane);

add("North", MyUpPanel);
add("South", MyDownPanel);

addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent evt) {
        System.exit(0);
    }
});
}

public static void main(String[] args){
    MyAppText app2 = new MyAppText("Application Window");
    app2.pack();
    app2.setVisible(true);
}
}

```

Τρέξτε τη και σημειώστε το αποτέλεσμα.

- 6) Δημιουργήστε μία κλάση **PLAFDemo**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import javax.swing.*;
import java.awt.*;

public class PLAFDemo
{
    public static void main(String [] args)
    {
        String laf = null;

        int choice = 0;
        if(args.length > 0)
            choice = Integer.parseInt(args[0]);

        switch(choice)

```

```

{
    case 1: laf = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";
    break;
    case 2: laf = "com.sun.java.swing.plaf.windows.WindowsLookAndFeel";
    break;
    case 3: laf = "com.sun.java.swing.plaf.mac.MacLookAndFeel";
    break;
    default: laf = "javax.swing.plaf.metal.MetalLookAndFeel";
}

try
{
    System.out.println("Setting the look and feel");
    UIManager.setLookAndFeel(laf);
} catch (Exception e)
{
    System.out.println("Could not set look and feel");
    e.printStackTrace();
}

JFrame f = new JFrame(laf);
Container content = f.getContentPane();
content.setLayout(new FlowLayout());
content.add(new JButton("OK"));
content.add(new JCheckBox("Pick me", true));
JProgressBar bar = new JProgressBar(1,10);
bar.setValue(7);
content.add(bar);
content.add(new JRadioButton("Click here", true));
content.add(new JSlider(1,10,3));
content.add(new JPasswordField("mypassword"));
String [] opts= {"Options1", "Option2", "Option3", "Option4"};
content.add(new JComboBox(opts));
f.setSize(400,100);
f.setVisible(true);
}
}

```

Τρέξτε τη 3 φορές περνώντας ως παράμετρο «1», «2», ή «3» καθώς και μια φορά χωρίς παραμέτρους. Σημειώστε τα αποτελέσματα.

- 7) Δημιουργήστε μία κλάση **CardWindow**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import java.awt.event.*;

public class CardWindow extends Frame implements ItemListener {

    Panel cards;
    Choice c = new Choice();
    final static String BUTTONPANEL = "Panel with Buttons";
    final static String TEXTPANEL = "Panel with TextField";

    public CardWindow() {
        setLayout(new BorderLayout());
        setFont(new Font("Helvetica", Font.PLAIN, 14));

        //Put the Choice in a Panel to get a nicer look.
        Panel cp = new Panel();
        c.addItem(BUTTONPANEL);
    }
}

```

```

        c.addItem(TEXTPANEL);
        c.addItemListener(this);

        cp.add(c);
        add("North", cp);

        cards = new Panel();
        cards.setLayout(new CardLayout());

        Panel p1 = new Panel();
        p1.add(new Button("Button 1"));
        p1.add(new Button("Button 2"));
        p1.add(new Button("Button 3"));

        Panel p2 = new Panel();
        p2.add(new TextField("TextField", 20));

        cards.add(BUTTONPANEL, p1);
        cards.add(TEXTPANEL, p2);
        add("Center", cards);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent evt) {
                System.exit(0);
            }
        });
    }

    public void itemStateChanged(ItemEvent evt) {

        // Οι εντολές στα σχόλια θα χρειαζονταν αν και άλλα components
        // άκουγαν ItemEvents

        // Object component = evt.getSource();
        // if (component instanceof Choice) {

            ((CardLayout)cards.getLayout()).show(cards, (String)c.getSelectedItem());

        // }
    }

    public static void main(String args[]) {
        CardWindow window = new CardWindow();
        window.setTitle("CardWindow Application");
        window.pack();
        window.setVisible(true);
    }
}

```

(α) Τρέξτε τη και σημειώστε το αποτέλεσμα.

(β) Επιχειρήστε να αλλάξετε την τιμή στη λίστα επιλογής. Σημειώστε τι παρατηρείτε.

(**Σημ.:** Αυτή η μέθοδος χρησιμοποιείται για να υλοποιήσουμε menu Tags, όπου οι οθόνες επικαλύπτονται).

8) Δημιουργήστε μία κλάση **MenuDemo**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import javax.swing.*;

```

```

import java.awt.event.*;

public class MenuDemo extends JFrame
{
    private JButton red, blue, white;
    private JProgressBar progress;

    public MenuDemo(String title)
    {
        super(title);
        Container contentPane = this.getContentPane();
        this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

        red = new JButton("Red");
        blue = new JButton("Blue");
        white = new JButton("White");

        //add the buttons to the frame
        JPanel south = new JPanel();
        south.add(red);
        south.add(blue);
        south.add(white);
        contentPane.add(south, BorderLayout.SOUTH);

        //add the progress bar
        progress = new JProgressBar(0,3);
        contentPane.add(progress, BorderLayout.NORTH);

        //register the event listener
        MenuColorChanger changer = new MenuColorChanger(contentPane, progress);
        red.addActionListener(changer);
        blue.addActionListener(changer);
        white.addActionListener(changer);

        createMenu(changer);
    }

    public void createMenu(MenuColorChanger changer)
    {
        //create a menu bar and attach it to this JFrame
        JMenuBar menuBar = new JMenuBar();
        this.setJMenuBar(menuBar);

        //create three menus and add them to the menu bar
        JMenu fileMenu = new JMenu("File");
        JMenu colorMenu = new JMenu("Color");
        JMenu helpMenu = new JMenu("Help");

        menuBar.add(fileMenu);
        menuBar.add(colorMenu);
        menuBar.add(helpMenu);

        //add three menu items to the "Color" menu
        JMenuItem redMenuItem = new JMenuItem("Red");
        JMenuItem blueMenuItem = new JMenuItem("Blue");
        JMenuItem whiteMenuItem = new JMenuItem("White");
        colorMenu.add(redMenuItem);
        colorMenu.add(blueMenuItem);
        colorMenu.add(whiteMenuItem);
        redMenuItem.addActionListener(changer);
        blueMenuItem.addActionListener(changer);
    }
}

```

```

whiteMenuItem.addActionListener(changer);

//add one menu item to the "File" menu
JMenuItem exit = new JMenuItem("Exit");
fileMenu.add(exit);
exit.addActionListener(changer);

//add one menu item and one menu to the "Help" menu
JMenuItem about = new JMenuItem("About MenuDemo...");
helpMenu.add(about);
JMenu cascade = new JMenu("Tip of the day");
helpMenu.add(cascade);
JCheckBoxMenuItem show = new JCheckBoxMenuItem("Show");
cascade.add(show);
}

public static void main(String [] args)
{
    MenuDemo f = new MenuDemo("MenuDemo");
    f.setSize(300,300);
    f.setVisible(true);
}
}

```

Δημιουργήστε μία κλάση **MenuColorChanger**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MenuColorChanger implements ActionListener
{
    private Container container;
    private JProgressBar progress;

    public MenuColorChanger(Container c, JProgressBar p)
    {
        container = c;
        progress = p;
    }

    public void actionPerformed(ActionEvent a)
    {
        String label = a.getActionCommand();
        if(label.equals("Red"))
        {
            container.setBackground(Color.RED);
            progress.setValue(1);
        }
        else if(label.equals("Blue"))
        {
            container.setBackground(Color.BLUE);
            progress.setValue(2);
        }
        else if(label.equals("White"))
        {
            container.setBackground(Color.WHITE);
            progress.setValue(3);
        }
        else if(label.equals("Exit"))
    }
}

```

```

        {
            System.exit(0);
        }
    }
}

```

(α) Τρέξτε τη MenuDemo και σημειώστε το αποτέλεσμα.

(β) Αλληλεπιδράστε με τα menu και τα buttons και περιγράψτε τη λειτουργία της εφαρμογής.

9) Δημιουργήστε μία κλάση **MyAppEv**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyAppEv extends JFrame implements ActionListener{
    private static int h=0,g=0;
    JButton b1 = new JButton("Real");
    JButton b2 = new JButton("Barcelona");

    public MyAppEv (String title) {
        super(title);

        b1.addActionListener(this);
        b2.addActionListener(this);

        JPanel myPanel = new JPanel();
        myPanel.add(b1);
        myPanel.add(b2);
        setContentPane(myPanel);
    }

    public void actionPerformed(ActionEvent e){
        Object source = e.getSource();
        if (source == b1) {
            ++g;
            setTitle("Real Madrid " + g + ":" + h + " Barcelona");
        }
        else if (source == b2) {
            ++h;
            setTitle("Real Madrid " + g + ":" + h + " Barcelona");
        }
    }

    public static void main(String[] args){

        try {

            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println("Can't set look and feel: " + e);
        }
        MyAppEv app = new MyAppEv("Real Madrid " + g + ":" + h + " Barcelona");

        WindowListener L = new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        };
    }
}

```



```

        app.addWindowListener(I);
        app.setBounds(300,200,350,250);

        app.setVisible(true);
    }
}

```

(α) Τρέξτε τη και σημειώστε το αποτέλεσμα.

(β) Πατήστε μερικές φορές τα κουμπιά και περιγράψτε τι συμβαίνει.

10) Δημιουργήστε μία κλάση **MyAppDialog**, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyAppDialog extends JFrame implements ActionListener{
    Button b = new Button("Delete");

    public MyAppDialog (String title) {
        super(title);
        JPanel myPanel = new JPanel();
        b.addActionListener(this);
        myPanel.add(b);

        setContentPane(myPanel);
    }

    public void actionPerformed(ActionEvent e){
        int answer = JOptionPane.showConfirmDialog(null, "Delete file?",
        "Deleting File", JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);
        if (answer == JOptionPane.OK_OPTION)
            JOptionPane.showMessageDialog(null,"File deleted!","File Deletion
        Information",JOptionPane.INFORMATION_MESSAGE);
        else
            JOptionPane.showMessageDialog(null,"File not deleted!","File Deletion
        Information",JOptionPane.INFORMATION_MESSAGE);    }

    public static void main(String[] args){

        try {

            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println("Can't set look and feel: " + e);
        }
        MyAppDialog app = new MyAppDialog("Application Window");

        WindowListener I = new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        };
        app.addWindowListener(I);
        app.setSize(200,200);
        app.setVisible(true);
    }
}

```

(α) Τρέξτε τη και σημειώστε το αποτέλεσμα.

(β) Πατήστε το κουμπί "Delete" και πείτε τι συμβαίνει.

(γ) Πατήστε yes και σημειώστε το αποτέλεσμα.

(δ) Τρέξτε πάλι την εφαρμογή και αυτή τη φορά πατήστε no. Σημειώστε πάλι το αποτέλεσμα.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 8

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με τις μικροεφαρμογές στην Java.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi8), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

(α) Δημιουργήστε κατ' αρχήν μία κλάση στο project και ονομάστε τη **MyApplet**. Στη συνέχεια γράψτε στην κλάση τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη:

```
import java.awt.Graphics;

public class MyApplet extends java.applet.Applet {

    int x=10, y=10;

    public void paint(Graphics g) {
        g.drawString("niyon anomhmata mh monan oyin", x, y);
        g.drawString("NIΨON ANOMHMATA MH MONAN OΨIN", x, 3*y);
    }
}
```

(β) Τρέξτε τη στον AppletViewer με ύψος και πλάτος 600 pixels και σημειώστε το αποτέλεσμα.

- 2) Αντικαταστήστε το σώμα της κλάσης με το ακόλουθο:

```
import java.awt.Graphics;
import java.awt.Font;
import java.awt.Color;

public class MyApplet extends java.applet.Applet {

    int fsize, x, y, r, g, b;
    String font;
    Font f;
    Color c;

    public void init() {
        x = 10;
        y = 20;
        font = "SansSerif";
        fsize = 24;
        f = new Font(font, Font.BOLD, fsize);
        c = Color.red;
    }

    public void paint(Graphics g) {
        g.setFont(f);
        g.setColor(c);
        g.drawString("niyon anomhmata mh monan oyin "+x+" "+y, x, y);
        g.drawString("NIΨON ANOMHMATA MH MONAN OΨIN "+x+" "+3*y, x, 3*y);
    }
}
```

Σώστε και μεταφράστε την κλάση.

Τρέξτε την τώρα, πάλι μέσω του AppletViewer και σημειώστε πάλι το αποτέλεσμα. Τι παρατηρείτε;

3) Προσθέστε την επόμενη μέθοδο στην κλάση, σώστε τη και μεταφράστε τη.

```
public void start() {
    x += 10;
    y += 50;
    x %= 500;
    y %= 500;
    java.util.Random rand = new java.util.Random();
    r = rand.nextInt(255);
    g = rand.nextInt(255);
    b = rand.nextInt(255);
    c = new Color(r, g, b);
}
```

(α) Τρέξτε πάλι τη μικροεφαρμογή. Άλλαξαν καθόλου τα αποτελέσματα;

(β) Επιλέξτε Applet-Stop. Τι συνέβη;

(γ) Εκκινήστε πάλι τη μικροεφαρμογή με Applet-Start. Άλλαξε κάτι; Σημειώστε.

4) Προσθέστε και την επόμενη μέθοδο στην κλάση, σώστε τη και μεταφράστε τη.

```
public void stop() {
    if (fsize > 10) fsize-=6; else fsize = 24;
    f = new Font(font, Font.BOLD, fsize);
}
```

(α) Τρέξτε πάλι τη μικροεφαρμογή. Τι παρατηρείτε;

(β) Σταματήστε και εκκινήστε πάλι τη μικροεφαρμογή (Applet-Stop, Applet-Start). Τι συνέβη τώρα;

5) Αντικαταστήστε όλη την κλάση με τον ακόλουθο κώδικα:

```
import java.awt.Graphics;
import java.awt.*;
import java.awt.Color;

public class MyApplet extends java.applet.Applet {

    int fsize, x, y, r, g, b;
    String font;
    Font f;
    Color c;
    Font[] fonts;
    public void init() {
        x = 20;
        y = 30;
        font = getParameter("font_name");
        if (font == null) font = "SansSerif";
        fsize = 18;
        f = new Font(font, Font.PLAIN, fsize);
        c = Color.red;
    }

    public void paint(Graphics g) {
```

```

g.setColor(c);
g.setFont(f);
g.drawString("niyon anomhmata mh monan oyin", x, 10);

GraphicsEnvironment env =
    GraphicsEnvironment.getLocalGraphicsEnvironment();
Font[] allfonts = env.getAllFonts();

for (int i=0;i<55;i++) {
    Font bigFont = allfonts[i].deriveFont(12.0f);
    g.setFont(bigFont);
    g.drawString(bigFont.getFontName(), x, y+10*i);
}
y=30;
for (int i=55;i<allfonts.length;i++) {
    Font bigFont = allfonts[i].deriveFont(12.0f);
    g.setFont(bigFont);
    g.drawString(bigFont.getFontName(), x+300, y+10*i-550);
}
}
}

```

Σώστε και μεταφράστε το πρόγραμμα.

(α) Εκτελέστε τώρα τη μικροεφαρμογή και προσθέστε μια παράμετρο με όνομα font_name και τιμή το όνομα μιας γραμματοσειράς, π.χ. Serif, Monospaced.

(β) Κλείστε την εφαρμογή και εκτελέστε τη πάλι, αλλά αυτή τη φορά αλλάξτε το όνομα της παραμέτρου από font_name σε fontname. Τι συνέβη τώρα;

6) Αντικαταστήστε όλη την κλάση με τον ακόλουθο κώδικα, σώστε την και μεταφράστε τη.

```

import java.awt.Graphics;
import java.awt.Font;
import java.awt.Color;

public class MyApplet extends java.applet.Applet {

    int fsize, x, y, r, g, b;
    String font;
    Font f;
    Color c;

    public void init() {
        x = 10;
        y = 20;
        font = "SansSerif";
        fsize = 24;
        f = new Font(font, Font.BOLD, fsize);
        c = Color.red;
    }

    public void start() {
        x += 10;
        y += 50;
        x %= 500;
        y %= 500;
        java.util.Random rand = new java.util.Random();
        r = rand.nextInt(255);
        g = rand.nextInt(255);
    }
}

```

```

        b = rand.nextInt(255);
        c = new Color(r, g, b);
    }
    public void stop() {
        if (fsize > 10) fsize-=6; else fsize = 24;
        f = new Font(font, Font.BOLD, fsize);
    }
    public void paint(Graphics g) {
        g.setFont(f);
        g.setColor(c);
        g.drawString("niyon anomhmata mh monan oyin "+x+" "+y, x, y);
        g.drawString("NIΨON ANOMHMATA MH MONAN OΨIN "+x+" "+3*y, x, 3*y);
        g.drawLine(30, 30, 90, 200);
        g.drawOval(20, 40, 15, 35);
        g.drawRect(40, 10, 60, 30);
    }
}

```

(α) Τρέξτε τη μικροεφαρμογή και σημειώστε τα αποτελέσματα.

(β) Αντικαταστήστε τις 2 τελευταίες εντολές της paint με τις ακόλουθες, σώστε την κλάση και μεταφράστε τη.

```

g.fillOval(20, 40, 15, 35);
g.fillRect(40, 10, 60, 30);

```

Τρέξτε τη μικροεφαρμογή και σημειώστε πάλι τα αποτελέσματα.

(γ) Προσθέστε στον κώδικα της paint τον ακόλουθο κώδικα, σώστε την κλάση και μεταφράστε τη.

```

Color c2=new Color(255-c.getRed(),255-c.getGreen(),255-c.getBlue());
g.setColor(c2);
g.fillArc(20, 40, 15, 35, 20, 100);
g.fillRoundRect(40, 10, 60, 30, 30, 15);

```

Τρέξτε τη μικροεφαρμογή. Τι συνέπεια είχε η πρόσθεση των παραπάνω γραμμών κώδικα;

(δ) Προσθέστε επιπλέον τον ακόλουθο κώδικα στην paint, σώστε την κλάση και μεταφράστε τη.

```

java.awt.Polygon poly = new java.awt.Polygon();
poly.addPoint(30, 30);
poly.addPoint(100, 196);
poly.addPoint(93, 209);
poly.addPoint(81, 203);
poly.addPoint(30, 30);
g.drawPolygon(poly);

```

Τρέξτε πάλι τη μικροεφαρμογή. Τι επιπλέον πρόσθεσε ο παραπάνω κώδικας στη μικροεφαρμογή;

7) Αντικαταστήστε την τελευταία εντολή της paint με τον ακόλουθο κώδικα. Σώστε την κλάση και μεταφράστε τη.

```

g.setColor(c);
g.drawPolygon(poly);
g.copyArea(0, 0, 120, 220, 150, 250);
g.clearRect(0, 0, 120, 220);

```

(α) Τρέξτε τη μικροεφαρμογή και σημειώστε το αποτέλεσμα.

(β) Σταματήστε και εκκινήστε την μικροεφαρμογή 3-4 φορές. Σημειώστε τις παρατηρήσεις σας.

(γ) Προσθέστε την επόμενη εντολή στο τέλος της paint, σώστε την κλάση και μεταφράστε τη.

```
setBackground(c2);
```

Τρέξτε τη μικροεφαρμογή. Τι συνέβη τώρα;

(δ) Αλλάξτε τις τιμές 150 και 250 στη μέθοδο copyArea σε 50 και 100. Σώστε την κλάση, μεταφράστε τη και τρέξτε πάλι τη μικροεφαρμογή. Τι παρατηρείτε;

(ε) Σταματήστε και εκκινήστε τη μικροεφαρμογή 3-4 φορές. Σημειώστε τις παρατηρήσεις σας.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 9

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με τα πακέτα και τις διασυνδέσεις στη Java.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi9), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

Στο BlueJ επιλέξτε Edit → New Package... και δημιουργήστε ένα πακέτο με το όνομα javalab. Πηγαίνετε στον κατάλογο του project και ελέγξτε αν δημιουργήθηκε ένας κατάλογος javalab.

Στο BlueJ επιλέξτε τον κατάλογο του πακέτου (με διπλό κλικ) και εκεί δημιουργήστε μία κλάση Vector η οποία θα περιέχει τον παρακάτω κώδικα:

```
package javalab;

import java.awt.*;

public class Vector {

    private Point end;

    public Vector (Point e) {
        end = e;
    }

    public Vector(Vector v) {
        end = v.getEnd();
    }

    public Vector(int x, int y) {
        end = new Point(x, y);
    }

    public Point getEnd() {
        return end;
    }

    public void move(Point e) {
        end = e;
    }

    public void setEnd(Point e) {
        end = e;
    }

    public void print() {
        System.out.print("(" + end.getX() + ", " + end.getY() + ")");
    }

    public void println() {
        System.out.println("(" + end.getX() + ", " + end.getY() + ")");
    }
}
```

Σώστε και μεταφράστε την κλάση Vector.

- 2) Πηγαίνετε, μέσα από το BlueJ, στον root κατάλογο του project, και δημιουργήστε μία νέα κλάση RunMe με τον παρακάτω κώδικα:

```
import java.util.*;
import javalab.*;

public class RunMe {
    public static void main(String[] args) {
        Vector vecs = new Vector();
    }
}
```

(α) Σώστε την κλάση και μεταγλωττίστε τη. Τι παρατηρείτε; Δικαιολογείστε.

(β) Αντικαταστήστε τον κώδικα της RunMe με τον ακόλουθο, σώστε τη και μεταφράστε τη.

```
import javalab.*;

public class RunMe {
    public static void main(String[] args) {
        java.util.Vector vecs = new java.util.Vector();
    }
}
```

Η μετάφραση έγινε τώρα κανονικά;

- 3) Αντικαταστήστε τον κώδικα της RunMe με τον ακόλουθο, σώστε τη και μεταφράστε τη.

```
import java.awt.*;
import javalab.*;

public class RunMe {
    public static void main(String[] args) {
        java.util.Vector vecs = new java.util.Vector(5);

        java.util.Random rand = new java.util.Random();

        for (int i=0;i<5;i++) {
            vecs.addElement(new Vector(new
                Point(random(rand),random(rand))));
            ((Vector)vecs.lastElement()).println();
        }

        private static int random(java.util.Random r) {
            return (r.nextBoolean())?r.nextInt(10):(-r.nextInt(10));
        }
    }
}
```

(α) Τρέξτε την και σημειώστε τα αποτελέσματα.

- 4) Πηγαίνετε τώρα στον root κατάλογο του project και δημιουργήστε ένα νέο πακέτο με όνομα my_interfaces. Πηγαίνετε σε αυτόν τον κατάλογο που μόλις δημιουργήσατε και φτιάξτε ένα νέο interface (New Class → Interface) με όνομα Operations. Ο κώδικας της διασύνδεσης θα είναι ο ακόλουθος:

```
package my_interfaces;
```

```
public interface Operations {
    public abstract void add(Operations o);
    public abstract void sub(Operations o);
    public abstract int dotProduct(Operations o);
    public abstract int crossProduct(Operations o);
}
```

(α) Σώστε τη διασύνδεση και μεταφράστε τη.

(β) Προσθέστε στην κλάση Vector την παρακάτω γραμμή.

```
import my_interfaces.*;
```

(γ) Αλλάξτε στην κλάση Vector τη δήλωση της ως εξής:

```
public class Vector implements Operations {
```

(δ) Προσθέστε τώρα στην κλάση Vector την επόμενη μέθοδο:

```
public void add(Operations o) {
    end.translate((int) ((Vector)o).getEnd().getX(), (int) ((Vector)o).getEnd().getY());
}
```

Σώστε την κλάση και μεταφράστε τη. Τι παρατηρείτε; Τι προτείνετε;

(ε) Προσθέστε τις ακόλουθες μεθόδους στην Vector:

```
public void sub(Operations o) {
    end.translate(-(int) ((Vector)o).getEnd().getX(),
        (int) ((Vector)o).getEnd().getY());
}

public int dotProduct(Operations o) {
    Point p = ((Vector)o).getEnd();
    return (int) (end.getX()*p.getX() + end.getY()*p.getY());
}

public int crossProduct(Operations o) {
    Point p = ((Vector)o).getEnd();
    return (int) (end.getX()*p.getY() - end.getY()*p.getX());
}
```

Σώστε και μεταφράστε την κλάση. Έγινε σωστά η μετάφραση τώρα;

5) Προσθέστε στην μέθοδο main της RunMe τον ακόλουθο κώδικα:

```
Vector v1, v2;
System.out.println("\nOperations:");
v1 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v2 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v1.print();
System.out.print(" + ");
v2.print();
System.out.print(" = ");
v1.add(v2);
v1.println();
v1 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v2 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
```

```

v1.print();
System.out.print(" - ");
v2.print();
System.out.print(" = ");
v1.sub(v2);
v1.println();
v1 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v2 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v1.print();
System.out.print(" * ");
v2.print();
System.out.println(" = " + v1.dotProduct(v2));
v1 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v2 = (Vector)vecs.elementAt(rand(rand, vecs.size()));
v1.print();
System.out.print(" x ");
v2.print();
System.out.println(" = " + v1.crossProduct(v2));

```

Προσθέστε επίσης στην RunMe την ακόλουθη μέθοδο:

```

private static int rand(java.util.Random r, int n) {
    return r.nextInt(n);
}

```

(α) Σώστε και μεταφράστε την κλάση.

(β) Τρέξτε την κλάση 2 φορές και σημειώστε τα αποτελέσματα.

ΕΡΓΑΣΤΗΡΙΟ JAVA

ΑΣΚΗΣΗ 10

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με τα νήματα στην Java.

- 1) Ανοίξτε το εργαλείο BlueJ, δημιουργήστε ένα νέο project και σώστε το (όνομα Askisi10), αφού επιλέξετε τον κατάλογο αποθήκευσης του project.

(α) Δημιουργήστε μία νέα Applet κλάση με όνομα **MyClock** με τον παρακάτω κώδικα:

```
import javax.swing.JApplet;
import java.awt.*;
import java.util.Date;

public class MyClock extends JApplet {

    Font f = new Font("Helvetica",Font.ITALIC, 12);
    Date d;

    public void paint(Graphics g)
    {
        d = new Date();
        g.setColor(Color.red);
        g.setFont(f);
        g.drawString(d.toString(), 25, 40);
    }
}
```

Σώστε την κλάση και μεταφράστε τη.

(β) Εκτελέστε το Applet με height=500 και width=500 και σημειώστε το αποτέλεσμα.

- 2) (α) Αλλάξτε τον κώδικα της MyClock με τον παρακάτω, σώστε την κλάση και μεταφράστε τη.

```
import javax.swing.JApplet;
import java.awt.*;
import java.util.*;
import java.text.DateFormat;

public class MyClock extends java.applet.Applet implements Runnable {

    Font f = new Font("Helvetica",Font.ITALIC, 12);
    Calendar c;
    Date d;
    DateFormat df;
    private Thread r;

    public void start() {
        if (r == null) {
            r = new Thread(this, "Digital Clock");
            r.start();
        }
    }

    public void stop() {
        if (r != null) {
            r = null;
        }
    }
}
```

```

    }

    public void run() {
        Thread curr = Thread.currentThread();
        while(r == curr) {
            repaint();
            try {r.sleep(1000);}
            catch (InterruptedException e) {}
        }
    }

    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.setFont(f);
        c = Calendar.getInstance();
        d = c.getTime();
        df = DateFormat.getDateTimeInstance(DateFormat.FULL,
            DateFormat.FULL);
        g.drawString(df.format(d), 25, 40);
    }
}

```

(β) Τρέξτε πάλι το Applet και πείτε ποια η διαφορά που παρατηρείτε.

3) (α) Φτιάξτε μία νέα κλάση **RaceThread** με τον ακόλουθο κώδικα:

```

public class TrackThread extends Thread {
    int m;

    public TrackThread(String str, int m) {
        super(str);
        this.m = m;
    }

    public void run() {
        System.out.println(getName() + " started running!");
        for (int i = 0; i < m; i+=20) {
            System.out.println(getName() + " has just passed the " + i + "
                meters.");
            try {
                sleep((long) (Math.random() * 1000));
            } catch (InterruptedException e) {}
        }
        System.out.println(getName() + " has finished the " + m + "m.
            race.");
    }
}

```

Σώστε και μεταφράστε την κλάση.

(β) Φτιάξτε μία κλάση **Runners** με τον ακόλουθο κώδικα:

```

public class Runners {
    public static void main (String[] args) {
        System.out.println("Men 200m Track with 2 runners!");
        new TrackThread("Konstantinos Kenteris", 200).start();
        new TrackThread("Ato Boldon", 200).start();
    }
}

```

Σώστε και μεταφράστε την κλάση.

(γ) Τρέξτε τη και σημειώστε τα αποτελέσματα.

(δ) Αντικαταστήστε το σώμα της `main` με τον ακόλουθο κώδικα, σώστε και μεταφράστε την κλάση.

```
System.out.println("Men 400m Hurdles Track with 3 runners!");
new TrackThread("Llewellyn Herbert", 400).start();
new TrackThread("Stephane Diagana", 400).start();
new TrackThread("Samuel Matete", 400).start();
```

(ε) Τρέξτε τη πάλι και σημειώστε τα νέα αποτελέσματα.

4) (α) Φτιάξτε μία νέα κλάση **Locker** με τον παρακάτω κώδικα σώστε τη και μεταφράστε τη:

```
public class Locker {
    private int thing;

    public int get() {
        return thing;
    }

    public void put(int thing) {
        this.thing = thing;
    }
}
```

(β) Φτιάξτε μία νέα κλάση **Storer** που να περιέχει τον παρακάτω κώδικα, σώστε τη και μεταφράστε τη.

```
public class Storer extends Thread {
    private Locker l;
    private int n;

    public Storer(Locker l, int n) {
        this.l = l;
        this.n = n;
    }

    public void run() {
        for (int i = 0; i < 10; i++) {
            l.put(i);
            System.out.println("Storer no." + n + " stored thing no." + i
+ " into the locker.");
            try {
                sleep((int)(Math.random() * 500)+100);
            } catch (InterruptedException e) { }
        }
    }
}
```

(γ) Δημιουργήστε και μία άλλη κλάση **Remover** με τον ακόλουθο κώδικα και σώστε και μεταφράστε την κλάση.

```
public class Remover extends Thread {
    private Locker l;
    private int n;
```

```

public Remover(Locker l, int n) {
    this.l = l;
    this.n = n;
}

public void run() {
    int x = 0;
    for (int i = 0; i < 10; i++) {
        x = l.get();
        System.out.println("Remover no." + n + " took the thing no." +
            x + " from the locker.");
        try {
            sleep((int)(Math.random() * 500)+100);
        } catch (InterruptedException e) { }
    }
}
}

```

(δ) Δημιουργήστε μία κλάση **LockerStory** με τον ακόλουθο κώδικα, σώστε τη και μεταφράστε τη.

```

public class LockerStory {
    public static void main(String[] args) {
        Locker l = new Locker();
        Storer s = new Storer(l, 1);
        Remover r = new Remover(l, 1);

        s.start();
        r.start();
    }
}

```

(ε) Τρέξτε την LockerStory και σημειώστε το αποτέλεσμα.

(στ) Σημειώστε, με βάση τα αποτελέσματα που πήρατε, τον τρόπο με τον οποίο ο Storer αποθηκεύει κάτι στον Locker και ο Remover το παίρνει από αυτόν. Θα είναι ο ίδιος και την επόμενη φορά ή είναι κατά κάποιο τρόπο τυχαίος;

5) (α) Αλλάξτε τον κώδικα της κλάσης Locker με τον ακόλουθο, σώστε τη και μεταφράστε τη.

```

public class Locker {
    private int thing;
    private boolean available = false;

    public synchronized int get() {
        while (available == false) {
            try {
                wait();
            } catch (InterruptedException e) { }
        }
        available = false;

        System.out.println("GET "+thing);
        notifyAll();
        return thing;
    }

    public synchronized void put(int thing) {

```

```

        while (available == true) {
            try {
                wait();
            } catch (InterruptedException e) { }
        }
        this.thing = thing;
        System.out.println("PUT "+thing);
        available = true;
        notifyAll();
    }
}

```

(β) Τρέξτε 2-3 φορές την LockerStory και σημειώστε τα αποτελέσματα.

(γ) Τι παρατηρείτε τώρα σε σχέση με τον τρόπο που ο Storer τοποθετεί και ο Remover παίρνει αντικείμενα; Σχολιάστε...

(δ) Αλλάξτε τον κώδικα της μεθόδου main της κλάσης LockerStory με τον ακόλουθο, σώστε την κλάση και μεταφράστε τη:

```

Locker l = new Locker();
Storer s1 = new Storer(l, 1);
Storer s2 = new Storer(l, 2);
Remover r1 = new Remover(l, 1);
Remover r2 = new Remover(l, 2);

s1.start();
r1.start();
s2.start();
r2.start();

```

Τρέξτε τη και σημειώστε τα αποτελέσματα.

(ε) Αλλάξτε και πάλι τον κώδικα της μεθόδου main της κλάσης LockerStory με τον ακόλουθο, σώστε την κλάση και μεταφράστε τη:

```

Locker l1 = new Locker();
Locker l2 = new Locker();
Storer s1 = new Storer(l1, 1);
Storer s2 = new Storer(l2, 2);
Remover r1 = new Remover(l1, 1);
Remover r2 = new Remover(l2, 2);

s1.start();
r1.start();
s2.start();
r2.start();

```

Τρέξτε τη και σημειώστε τα αποτελέσματα.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

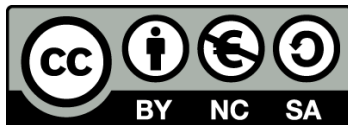
Το παρόν έργο αποτελεί την έκδοση 1.0

Σημείωμα Αναφοράς

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015. «Οντοκεντρικός Προγραμματισμός».
Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:
<https://eclass.upatras.gr/courses/CEID1105/>

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Πίνακες

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

