

Εργαστηριακές Ασκήσεις JAVA

Έτος: 2021-2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Βασιλόπουλος Ιωάννης
Βογιατζάκη Ελένη
Καλογερόπουλος Νικήτας - Ρήγας
Κουτσομητρόπουλος Δημήτριος
Μακρής Χρήστος
Χατζηλυγερούδης Ιωάννης

Περιεχόμενα

1	Άσκηση 1: Εισαγωγή	2
2	Άσκηση 2: Κλάσεις - Μέθοδοι - Στιγμιότυπα	5
3	Άσκηση 3: Κληρονομικότητα	7
4	Άσκηση 4: Συλλογές - Υπέρβαση Μεθόδων	11
5	Άσκηση 5: Χειρισμός I/O Streams	18
6	Άσκηση 6: Εξαιρέσεις	23
7	Άσκηση 7: Γραφικά	28
8	Άσκηση 8: Πακέτα-Διασυνδέσεις	39
9	Άσκηση 9: Νήματα	43

1 Άσκηση 1: Εισαγωγή

Σκοπός της άσκησης αυτής είναι:

- α'. να δημιουργήσετε ένα νέο αρχείο JAVA
- β'. να εισάγετε κάποιον κώδικα, να τον μεταγλωττίσετε και να δείτε να τα αποτελέσματα
- γ'. να εξοικειωθείτε με την είσοδο δεδομένων από την γραμμή εντολής
- δ'. να εξοικειωθείτε με συνθήκες ελέγχου και στην χρήση εντολής εξόδου στην οθόνη.

1) Να ανοίξετε το εργαλείο BlueJ, αφού δημιουργήσετε ένα νέο project (Project→ New Project), να το αποθηκεύσετε στον επιθυμητό κατάλογο με το όνομα "Askisi1". Στον κατάλογο που επιλέξατε θα υπάρχει ένας νέος, που θα αντιστοιχεί στο project που μόλις δημιουργήθηκε. Στη συνέχεια, να δημιουργήσετε μια νέα κλάση και να την ονομάσετε Age επιλέγοντας το πλήκτρο "New Class". Αφού, κάνετε διπλό κλικ στο εικονίδιο που εμφανίστηκε το οποίο αντιστοιχεί στην κλάση που δημιουργήσατε πιο πριν και διαγράψετε το περιεχόμενο του αρχείου, που δημιουργήθηκε αυτόματα από το BlueJ, να γράψετε 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.");
        }
    }
}
```

Σημείωση 1: Η εντολή `System.out.println('παράμετροι')` της Java χρησιμοποιούνται για εκτυπώσεις στην οθόνη.

2) Στη συνέχεια να αποθηκεύσετε τις αλλαγές που έγιναν στην κλάση Age, επιλέγοντας στο παράθυρο της Class→ Save ή τον συνδυασμό πλήκτρων "Ctrl + s". Το αρχείο θα αποθηκευτεί με το όνομα Age.Java και θα πρέπει να μεταφραστεί πατώντας το πλήκτρο compile. Αν δεν υπάρχει κάποιο σφάλμα και γίνουν όλα σωστά, στο κάτω μέρος θα εμφανιστεί το μήνυμα "Class compiled – no syntax errors". Στον κατάλογο του project, θα πρέπει μετά τη μεταγλώττιση να μπορείτε να δείτε το αρχείο "Age.class".

3) Αφού κλείσετε το παράθυρο της κλάσης Age, στο παράθυρο του project να κάνετε δεξί κλικ στην κλάση Age και να επιλέξετε το "void main(args)". Στο νέο παράθυρο που εμφανίζεται, να δώσετε κάποιο όρισμα τύπου String. Άρα, αν θέλετε να δώσετε τον αριθμό 29 θα πρέπει να

γράψετε {"29"}, αυστηρά με αυτή την δομή, γιατί και οι αγκύλες αλλά και τα εισαγωγικά είναι απαραίτητα. Τέλος, πατήστε OK. Να δοκιμάσετε άλλες 2 περιπτώσεις και να αναφέρετε στην αναφορά σας τις παρατηρήσεις σας. Τι θα συμβεί αν δε δοθεί κάποιο όρισμα;

4) Να αλλάξετε τώρα τις εντολές εκτύπωσης με τις παρακάτω και να αποτυπώσετε στην αναφορά σας τις παρατηρήσεις σας, αφού ξανά μεταφράσετε και τρέξετε το πρόγραμμα με τα ίδια ορίσματα. Για ποίο λόγο παρατηρείτε τις διαφορές αυτές;

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

Σημείωση 2: Στο παράθυρο του Project αν μετά τις αλλαγές δεν γίνει μετάφραση, θα παρατηρήσετε ότι το εικονίδιο της κλάσης είναι διαφορετικό και έχει γκριζες γραμμές. Αντίστοιχα, σε περίπτωση κάποιου σφάλματος στον κώδικα, πάλι το εικονίδιο της κλάσης που έχει το σφάλμα θα έχει κόκκινες γραμμές.

5) Προσθέστε, τώρα, την παρακάτω γραμμή κώδικα και ελέγξτε τη συμπεριφορά του προγράμματος όταν δε δίνεται είσοδος, αφού ξανά μεταφράσετε το νέο πρόγραμμα. Τι παρατηρείτε;

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

6) Διαμορφώστε το σώμα της μεθόδου main με το παρακάτω τμήμα κώδικα και αφού ξανά μεταφράσετε το πρόγραμμα να δώσετε δύο ορίσματα, το πρώτο θα εκφράζει την ηλικία και το δεύτερο το όνομα. Να ελέγξετε την περίπτωση που δίνεται κανένα όρισμα και την περίπτωση που δίνεται μόνο η ηλικία ως όρισμα. Καταγράψετε τα αποτελέσματα στην αναφορά σας.

```
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.");  
}
```

Ποια αλλαγή θα κάνατε στο νέο σώμα της `main`, ώστε το πρώτο όρισμα να αντιστοιχεί στο όνομα και το δεύτερο στην ηλικία. Να συμπεριλάβετε την αλλαγή αυτή στην αναφορά σας.

7) Για να μειωθεί το πλήθος των εντολών `System.out.print` και `System.out.println`, χρησιμοποιείται συνήθως ο τελεστής `+`. Για παράδειγμα, οι εντολές

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

μπορούν εύκολα να αντικατασταθούν από την εντολή

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

Να αντικαταστήσετε και τις υπόλοιπες εντολές εκτύπωσης με μία εντολή και αφού ξανά μεταφράσετε το πρόγραμμα βεβαιωθείτε ότι λειτουργεί σωστά για όλες τις περιπτώσεις. Σημειώστε τα αποτελέσματα στην αναφορά σας.

Σημείωση 3: Θα παρατηρήσατε ότι μετά από κάθε αλλαγή που έγινε, έπρεπε να ξανά γίνει μεταγλώττιση του προγράμματος.

Σημείωση 4: Θα παρατηρήσατε, επίσης, ότι μετά από κάθε εκτέλεση του προγράμματος, το μήνυμα που εμφανιζόταν στο τερματικό του BlueJ (BlueJ: Terminal Window - "Project Name") από την προηγούμενη εκτέλεση παραμένει. Αν θέλετε να καθαρίσετε την οθόνη του τερματικού επιλέγετε Options → Clear.

2 Άσκηση 2: Κλάσεις - Μέθοδοι - Στιγμιότυπα

Σκοπός της άσκησης αυτής είναι η εξοικείωση με την γλώσσα προγραμματισμού JAVA και με το περιβάλλον προγραμματισμού. Ειδικότερα:

- α'. να δημιουργήσετε ένα νέο αρχείο JAVA
- β'. να εισάγετε κάποιον κώδικα και να τον μεταγλωττίσετε
- γ'. να μελετήσετε την δημιουργία στιγμιότυπων και την κλήση μεθόδων και
- δ'. να διαπιστώσετε το ρόλο της λέξης-κλειδί "this".

1) Δημιουργήστε ένα νέο Project με όνομα Askisi2 όπως υποδείχθηκε στην πρώτη άσκηση. Δημιουργήστε μία νέα κλάση με όνομα Window και αντικαταστήστε τον αυτόματα δημιουργημένο κώδικα με τον παρακάτω.

```
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 ());
    }
}
```

Αφού αποθηκεύσετε τις αλλαγές σας να μεταγλωττίσετε το πρόγραμμα σας και να το εκτελέσετε χωρίς ορίσματα. Δηλαδή, στο παράθυρο που ζητά να δώσετε κάποιο όρισμα τύπου String εσείς δε θα γράψετε/δώσετε κάτι και θα πατήσετε OK. Να αποτυπώσετε τα αποτελέσματα στην αναφορά σας και να τα εξηγήσετε.

Σημείωση 1: Θα παρατηρήσετε στον κατάλογο του project ότι κατά την αποθήκευση της κλάσης δημιουργείται το αρχείο "Window.java", ενώ, αφού πατήσετε "compile" εμφανίζεται το μήνυμα Class compiled – no syntax errors στο κάτω μέρος και το αρχείο "Window.class" παράγεται στον κατάλογο του project

2) Να αντικαταστήσετε την εντολή `size = x;` με `this.size = x;`. Μεταφράστε, εκτελέστε το πρόγραμμα και να καταγράψετε τα αποτελέσματα. Έπειτα, να αντικαταστήσετε τον δημιουργό που φαίνεται παρακάτω:

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

με τον εξής:

```
public Window (int size){
    this.size=size;
}
```

Ξανά μεταγλωττίστε το πρόγραμμα σας και εκτελέστε το και ξανά σημειώσατε τα μηνύματα που εμφανίζονται. Αφαιρέστε τώρα το `this` από τον δημιουργό και επαναλάβετε την διαδικασία μετάφρασης και εκτέλεσης. Τι παρατηρείτε; Είναι τα αποτελέσματα διαφορετικά και σε ποια/ες περιπτώσεις; Ποίος ο ρόλος, τελικά, του `this`;

3) Να αλλάξετε τον κώδικα της κλάσης `Window` στον παρακάτω:

```
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());
    }
}
```

Να μεταφράσετε τις αλλαγές και να το τρέξετε. Τι παρατηρείτε για τα αποτελέσματα; Τι θεωρείτε ότι επηρέασε τα αποτελέσματα;

6) Στην μέθοδο `getSize` να αντικαταστήσετε την εντολή `return size` με την `return this.size`. Ξαναμεταφράστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε σε σχέση με τα προηγούμενα αποτελέσματα; Ποίος, τελικά, είναι ο ρόλος του `this`;

3 Άσκηση 3: Κληρονομικότητα

Σκοπός της άσκησης αυτής είναι:

- α'. η μελέτη δημιουργών σε σχέση με την κληρονομικότητα, βασικό χαρακτηριστικό του αντικειμενοστραφή προγραμματισμού
- β'. η κλήση και εκτέλεση μεθόδων, όσον αφορά την κληρονομικότητα
- γ'. να διαπιστώσετε το ρόλο της λέξης-κλειδί "super" και
- δ'. να εξοικειωθείτε με μερικούς τροποποιητές πρόσβασης (access modifiers) της γλώσσας JAVA.

1) Δημιουργήστε ένα νέο Project με όνομα Askisi3 όπως υποδείχθηκε στην πρώτη άσκηση. Δημιουργήστε μία νέα κλάση με όνομα Window και αντικαταστήστε τον αυτόματα δημιουργημένο κώδικα με τον παρακάτω, να την αποθηκεύσετε και να την μεταφράσετε:

```
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);
    }
}
```

Στην συνέχεια δημιουργήστε μία νέα κλάση με το όνομα MWindow. Γράψτε τον παρακάτω κώδικα, σώστε τη και μεταφράστε την.

```
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();}
}
```


Σημείωση 1: Θα παρατηρήσετε ότι μετά την μεταγλώττιση των δυο παραπάνω κλάσεων αν προσπαθήσετε να τρέξετε κάτι δεν θα μπορέσετε, αφού main δεν υπάρχει και συνεπώς, δεν υπάρχει το σημείο αρχής εκτέλεσης του προγράμματος.

2) Στο σημείο αυτό θα δημιουργηθεί η κλάση που θα αναλάβει την διαδικασία εκτέλεσης του προγράμματος και θα μπορεί να χρησιμοποιήσει τις επιμέρους κλάσεις που δημιουργήθηκαν πιο πάνω.

A) Να δημιουργήσετε, λοιπόν, μία κλάση με όνομα RunWindow, που θα περιλαμβάνει τον παρακάτω κώδικα, να την μεταφράσετε, να την τρέξετε, να αποτυπώσετε τα αποτελέσματα στην αναφορά σας και να τα εξηγήσετε.

```
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 );
    }
}
```

B) Στο τέλος της main να προσθέσετε τις παρακάτω 4 γραμμές, και αφού επαναλάβετε την μετάφραση της κλάσης, να αποτυπώσετε και να εξηγήσετε τα αποτελέσματα.

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

Γ) Προσθέστε ακόμα και τον παρακάτω κώδικα:

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

Ξανά μεταφράστε και τρέξτε το πρόγραμμα. Να αποτυπώσετε και να εξηγήσετε τα αποτελέσματά σας για τις νέες προσθήκες.

3) Παρεμβάλετε στον κώδικα της κλάσης MWindow, αμέσως μετά την πρόταση «private String message = "No message;"», την δήλωση «protected int size = 7;». Να ξανά μεταφράσετε την κλάση MWindow και να τρέξετε την κλάση RunWindow. Να αποτυπώσετε τα αποτελέσματά σας και να τα συγκρίνετε με αυτά των παραπάνω ερωτημάτων; Τι παρατηρείτε τώρα; Εξηγήστε.

4) Να προσθέσετε πάλι στο τέλος της `main` στην κλάση `RunWindow` τις παρακάτω προτάσεις. Αφού, κάνετε τις απαραίτητες διαδικασίες, να αποτυπώσετε τα αποτελέσματα σας στην αναφορά και να τα εξηγήσετε.

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

5) Τέλος, να αλλάξετε τις μεθόδους `printSize1()` και `printSize2()` της κλάσης `MWindow`, όπως φαίνονται παρακάτω και να συγκρίνετε και να αιτιολογήσετε τα αποτελέσματα με τα αντίστοιχα προηγούμενα.

*Σημείωση 2: Για να κατανοήσετε την έννοια των νέων εκφράσεων που παρατηρείτε στα ορίσματα των προτάσεων εμφάνισης των παρακάτω 2 μεθόδων να αναζητήσετε πληροφορίες για την έννοια του *Type Casting*, που ουσιαστικά μετατρέπει ο προγραμματιστής, δεδομένα κάποιου τύπου σε άλλον.*

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

Σημείωση 3: Η παραπάνω υλοποίηση θα μπορούσε να γίνει και σε ένα αρχείο συνολικά, αντί για 3 διαφορετικά. Πλέον, όμως στην απεικόνιση του αρχείου θα βλέπετε μόνο μία κλάση χωρίς τις διασυνδέσεις, όπως πριν. Επίσης, σχεδιαστικά θα υπάρχει ο κίνδυνος, ο κώδικας σας να γίνει δύσκολα διαχειρίσιμος ειδικά σε μεγάλα project πολλών ατόμων και για αυτό η χρήση πολλαπλών κλάσεων σε ένα αρχείο πρέπει να χρησιμοποιείται με σύνεση.

Αρχικά, θα πρέπει να δημιουργηθεί μία κλάση με όνομα `RunWindow`, διότι αυτή η κλάση περιέχει την `main` και άρα είναι η πρωτεύουσα κλάση. Στη συνέχεια, θα πρέπει να τοποθετηθεί ο κώδικας της `Window`, της `MWindow` και της `RunWindow` με αυστηρά αυτή τη σειρά. Από τις κλάσεις `Window` και `MWindow` θα πρέπει να αφαιρεθεί ο προσδιοριστής `public` αφού μόνο μία κλάση μπορεί να είναι δημόσια σε ένα αρχείο. Τέλος, θα πρέπει να μεταφράσετε το πρόγραμμα και αφού το τρέξετε θα λάβετε τα ίδια αποτελέσματα, αν όλα έγιναν σωστά.

6) Στο σημείο αυτό θα ασχοληθούμε με τον τελεστή `"instanceof"`. Ο τελεστής αυτός χρη-

σιμοποιείται για τον έλεγχο ενός αντικειμένου σε σχέση με κάποιο συγκεκριμένο τύπο. Δηλαδή είναι ένας τελεστής σύγκρισης, που συγκρίνει ένα στιγμιότυπο με ένα προκαθορισμένο τύπο. Αν το στιγμιότυπο έχει τον ίδιο τύπο με τον προκαθορισμένο τότε επιστρέφει True αλλιώς σε κάθε άλλη περίπτωση επιστρέφει false.

Να δημιουργήσετε ένα νέο Project με όνομα askisi3instance και να δημιουργήσετε τις εξής κλάσεις όπως φαίνονται παρακάτω:

A) Window.java

```
public class Window
{
    public Window(){}
    public void printtest(){System.out.println("HI! I am a simple window");}
}
```

B) MWindow.java

```
public class MWindow extends Window
{
    public MWindow(){}
    public void printtest(){System.out.println("HI! I am a Mwindow");}
}
```

Γ) RunWindow.java

```
public class RunWindow
{
    public static void main(String[] ar){
        Window test1 = new Window();
        MWindow test2 = new MWindow();

        System.out.println("——Class Method Printing——");
        test1.printtest();
        test2.printtest();
        System.out.println("——Instance Of Printing——");
        if(test1 instanceof Window){
            test1.printtest();
        }
        if(test1 instanceof MWindow){
            test1.printtest();
        }
        if(test2 instanceof Window){
            test2.printtest();
        }
        if(test2 instanceof MWindow){
            test2.printtest();
        }
    }
}
```

Μεταφράστε και τις 3 κλάσεις και αφού τρέξετε το πρόγραμμα να αποτυπώσετε τις παρατηρήσεις σας στην αναφορά σας. Τελικά, είναι σωστή η χρήση του τελεστή instanceof για απλές διαδικασίες ή μπορεί να προκαλέσει προβλήματα;

4 Άσκηση 4: Συλλογές - Υπέρβαση Μεθόδων

Σκοπός της άσκησης αυτής είναι:

- α'. να διαπιστώσετε την αναγκαιότητα και να εξασκηθείτε στην συγγραφή μεθόδων,
- β'. να διαπιστώσετε την αναγκαιότητα και την χρήση της "υπέρβασης μεθόδων" και
- γ'. να γνωρίσετε το πλαίσιο Συλλογών της γλώσσας JAVA και τις νέες βελτιώσεις που έχουν γίνει σε αυτό (Generics και επαυξημένη μορφή της δομής επανάληψης "for").

1) Δημιουργήστε ένα νέο Project με όνομα Askisi4 όπως υποδείχθηκε στην πρώτη άσκηση. Δημιουργήστε μία νέα κλάση με όνομα Robot και αντικαταστήστε τον αυτόματα δημιουργημένο κώδικα με τον παρακάτω, να την αποθηκεύσετε και να την μεταφράσετε:

```
public class Robot {
    private String name;
    private String manufacturer;
    private int days_of_usage;
    private boolean online;
    private float costs;

    public Robot(String name,String manufacturer ,int days_of_usage ,boolean online , float costs){
        this.name = new String(name);
        this.manufacturer = new String(manufacturer);
        this.days_of_usage=days_of_usage;
        this.online=online;
        this.costs=costs;
    }

    public String getName() {return name;}
    public String getManufacturer() {return manufacturer;}
    public boolean isOnline() {return online;}
    public float getCosts() {return costs;}
    public int getDays_of_usage() {return days_of_usage;}
}
```

Στην συνέχεια, δημιουργήστε μία νέα κλάση με το όνομα RobVehicle. Γράψτε τον παρακάτω κώδικα, σώστε τη και μεταφράστε την.

```
public class RobVehicle extends Robot {
    private int miles;
    public RobVehicle(String name,String manufacturer ,int days_of_usage , float costs ,int miles){
        super(name, manufacturer , days_of_usage , true , costs );
        this.miles=miles;
    }

    public int getMiles() {return miles;}
}
```

Τέλος, δημιουργήστε ακόμα μια νέα κλάση με όνομα Tester, που θα αποτελέσει την πρωτεύουσα κλάση του project και θα αναλάβει την είσοδο στο πρόγραμμα και δημιουργία των διαφόρων αντικειμένων.

```

abstract public class Tester {
    public static void main(String [] args){
        Robot r1 = new Robot("CRX", "FANUC", 15, true, 12.2 f);
        RobVehicle rv1 = new RobVehicle("test1", "ABB", 2, 25.4 f, 23) ;

        System.out.print("Robot "+r1.getName()+" of "+r1.getManufacturer()+" is");
        if (!r1.isOnline()){System.out.print(" not");}
        System.out.println(" online for "+r1.getDays_of_usage()+
"days with costs equals to "+r1.getCosts()+"M");

        System.out.print("Robotic Vehicle "+rv1.getName()+
" of "+rv1.getManufacturer()+" is online for "
+r1.getDays_of_usage()+
" days with costs equals to "+
r1.getCosts()+"M");
        if(rv1.getMiles(>0){System.out.println(" with traveled distance "+rv1.getMiles()+" miles");}
        else {System.out.println(" 0 miles traveled distance");}

    }
}

```

Γιατί ορίστηκε η Tester ως abstract; Είναι απαραίτητο; Να την τρέξετε και να καταγράψετε τα αποτελέσματα στην αναφορά σας. Τι εκτυπώθηκε με τον παραπάνω κώδικα;

2) Να κατασκευάσετε μία μέθοδο printinfo της κλάσης Robot που θα εκτυπώνει την διαθέσιμη πληροφορία. Και στη συνέχεια να αλλάξετε τον κώδικα της main στον εξής:

```

public static void main(String [] args){
    Robot r1 = new Robot("CRX", "FANUC", 15, true, 12.2 f);
    RobVehicle rv1 = new RobVehicle("test1", "ABB", 2, 25.4 f, 23) ;

    r1.printinfo();
    rv1.printinfo();

}

```

Τι παρατηρείτε; Λείπει ποσότητα πληροφορίας, συγκριτικά με το πρώτο ερώτημα; Τι πλεονέκτημα έχει ο τρόπος υλοποίησης αυτός σε σχέση με αυτόν του πρώτου ερωτήματος; Θεωρείτε ότι ο πρώτος είναι καλύτερος τρόπος προγραμματισμού σε σχέση με τον δεύτερο; Αιτιολογήσετε.

3) Υπερκαλύψτε τώρα την μέθοδο printinfo της κλάσης Robot με μία νέα printinfo που θα τοποθετήσετε στην RobVehicle, για να εμφανίσετε την πληροφορία που λείπει. Δοκιμάστε να χρησιμοποιήσετε και την λέξη-κλειδί super. Όπως, μετά από κάθε αλλαγή να μεταφράσετε τον κωδικά σας και να ξανά τρέξετε την κλάση Tester.

Σημείωση 1: Annotations στη Java.

Τα annotations είναι μία μορφή μεταδεδομένων που περιγράφουν πληροφορία για το πρόγραμμα, χωρίς να επηρεάζουν το πρόγραμμα το ίδιο σε θέμα λειτουργικότητας. Μπορεί να παρέχουν:

1. Πληροφορία στον μεταγλωττιστή, ώστε να εντοπίζει σφάλματα ή να περιορίζει προειδοποιήσεις ή να ενημερώνει-εμφανίζει προειδοποιήσεις για συγκεκριμένα θέματα.
2. Οδηγίες κατά την διάρκεια της μεταγλώττισης, ώστε να παράγεται απαραίτητος κώδικας ή αρχεία που τυχόν χρειάζονται για τη λειτουργία του προγράμματος.
3. Πληροφορία στην ροή εκτέλεσης.

Θα σταθούμε στην πρώτη κατηγορία annotations, στην οποία από την 5 έκδοση της Java υποστηρίζονται 3 βασικές ενσωματωμένες επιλογές.

1. `@Override`, τοποθετείται πάνω από μία μέθοδο που θα υπερκαλύψει μία της υπέρ-κλάσης της. Αν η μέθοδος αυτή δεν υπάρχει στην υπέρ-κλάση τότε θα εμφανίσει μήνυμα λάθους
2. `@Deprecated`, τοποθετείται από τον προγραμματιστή πάνω από μία κλάση, μέθοδο ή πεδίο και εμφανίζει κατά την διάρκεια μεταγλώττισης την αντίστοιχη προειδοποίηση
3. `@SuppressWarnings`, τοποθετείται πάνω από μία μέθοδο και απαγορεύει στον μεταγλωττιστή να εμφανίζει προειδοποιήσεις. Όπως, είναι φυσικό, δεν αποσιωπά τα λάθη, αλλά μόνο τυχόν προειδοποιήσεις.

Τέλος, η Java επιτρέπει και την δημιουργία Annotation από τον χρήστη.

Δοκιμάστε να χρησιμοποιήσετε στην υπερκάλυψη μεθόδων, που κάνατε στο 3ο ερώτημα το annotation ”@Override” στο κατάλληλο σημείο.

4) A. Να προσθέσετε τις εξής δηλώσεις στην κλάση Robot.

```
public static final int FLAWED = 0;
public static final int NONFLAWED = 1;
private int condition;
```

B. Να αλλάξετε τον δημιουργό της κλάσης RobVehicle στον παρακάτω:

```
public RobVehicle(String name, String manufacturer, int days_of_usage, float costs,
int miles, int condition){
    super(name, manufacturer, days_of_usage, true, costs, condition);
    this.miles=miles;
}
```

Γ. Να αλλάξετε τον δημιουργό της κλάσης Robot στον παρακάτω:

```
public Robot(String name, String manufacturer, int days_of_usage, boolean online,
float costs, int condition){
    this.name = new String(name);
    this.manufacturer = new String(manufacturer);
    this.days_of_usage=days_of_usage;
    this.online=online;
    this.costs=costs;
    this.condition=condition;
}
```

Δ. Να αλλάξετε την main της κλάσης Tester στην παρακάτω:

```
public static void main(String [] args){
    Robot r1 = new Robot("CRX", "FANUC", 15, true, 12.2f, Robot.NONFLAWED);
    RobVehicle rv1 = new RobVehicle("test1", "ABB", 2, 25.4f, 23, Robot.NONFLAWED);

    r1.printlnfo();
    rv1.printlnfo();

}
```

Ε. Να τρέξετε και να σημειώσετε τα νέα αποτελέσματα; Τι παρατηρείτε;

ΣΤ. Να τροποποιήσετε κατάλληλα τις μεθόδους printlnfo, ώστε να εμφανίζουν και τι νέα πληροφορία και αποτυπώστε τα αποτελέσματα και τις παρατηρήσεις σας στην αναφορά σας;

5) Α. Κατασκευάστε get και set μεθόδους για κάθε ιδιότητα κάθε κλάσης. Χαρακτηριστικό των μεθόδων get είναι ότι επιστρέφουν την τιμή της αντίστοιχης ιδιότητας ενώ των set είναι η αλλαγή της τιμής της ιδιότητας για το αναφερόμενο στιγμιότυπο της κλάσης.

Β. Να αλλάξετε την main της κλάσης Tester στην παρακάτω:

```
public static void main(String [] args){
    Robot r1 = new Robot("CRX", "FANUC", 15, true, 12.2f, Robot.NONFLAWED);
    RobVehicle rv1 = new RobVehicle("test1", "ABB", 2, 25.4f, 23, Robot.NONFLAWED);
    r1.printlnfo();
    rv1.printlnfo();
    r1.setOnline(false);
    rv1.setOnline(false);
    r1.printlnfo();
    rv1.printlnfo();
}
```

Αφού μεταφράσετε και τρέξετε το κώδικα, αποτυπώστε τα αποτελέσματά σας στην αναφορά.

Γ. Θα παρατηρήσατε ότι η `setOnline` αλλάζει την ιδιότητα `Online` της υποκλάσης `RobVehicle`, πράγμα που δεν είναι επιθυμητό. Άρα θα πρέπει να γίνει υπέρβαση της μεθόδου αυτής στην υποκλάση. Να δημιουργήσετε, λοιπόν την νέα μέθοδο αυτή και να εκτελέσετε το πρόγραμμα με την αλλαγή αυτή; Τι παρατηρείτε;

6) Να προσθέσετε μία μέθοδο `setDays_of_usage(RobVehicle rob)` στην κλάση `RobVehicle`, που θα χρησιμοποιείται όταν θέλουμε 2 ρομποτικά οχήματα να δουλεύουν ταυτόχρονα. Θα δέχεται ως όρισμα ένα αντικείμενο τύπου `RobVehicle` και στη συνέχεια:

A) Αρχικά, θα ελέγχει, αν η κατάσταση του οχήματος είναι ικανή για εργασία, δηλαδή ίση με 1 και για τα 2 οχήματα.

B) Αν η παραπάνω συνθήκη είναι αληθής τότε θα μετατρέπει τις ημέρες χρήσης (`days of use`) των οχημάτων ίσες με την μέγιστη από τα 2 οχήματα και για τα 2 αντικείμενα.

Γ) Αν κάποιο απο τα δύο είναι ακατάλληλο για χρήση (`FLAWED`), τότε δε θα κάνει απολύτως τίποτα ή μπορεί αν επιθυμείτε να εμφανίζει κάποιο απλό μήνυμα.

Δ) Τοποθετήστε στο σώμα της `main` τον παρακάτω κώδικα.

```
Robot r1 = new Robot("CRX", "FANUC", 15, true, 12.2f, Robot.NONFLAWED);
RobVehicle rv1 = new RobVehicle("test1", "ABB", 2, 25.4f, 23, Robot.NONFLAWED) ;
RobVehicle rv2 = new RobVehicle("test2", "ABB", 150, 0.4f, 45, Robot.FLAWED) ;

r1.printlnfo();
rv1.printlnfo();
r1.setOnline(false);
rv1.setOnline(false);
r1.printlnfo();
rv1.printlnfo();
rv1.setDays_of_usage(rv2);
rv2.printlnfo();
rv1.printlnfo();
rv1.setDays_of_usage(r1);
rv2.printlnfo();
rv1.printlnfo();
rv1.setDays_of_usage(rv2.getDays_of_usage());
rv2.printlnfo();
rv1.printlnfo();
```

Τι παρατηρείτε μόλις προσπαθήσετε να τρέξετε τον κώδικα; Αιτιολογήστε. Διορθώστε το λάθος αφαιρώντας τις κατάλληλες γραμμές κώδικα (μέθοδο και τις δύο συναρτήσεις εκτύπωσης που χρησιμοποιούνται για έλεγχο). Ποία η σχέση μεταξύ των 2 `setDays_of_usage` στις κλάσεις `Robot` και `RobVehicle`;

Σημείωση 2: Μια λίστα είναι μια δομή δεδομένων στην Java που επιτρέπει την ομαδοποίηση στιγμιοτύπων οποιουδήποτε τύπου. Μια υλοποίηση της δομής αυτής προσφέρει η κλάση `ArrayList`, του πακέτου `java.util`, η οποία μεταξύ άλλων προσφέρει μεθόδους προσθήκης (`add`), διαγραφής (`remove`) και ανάκτησης (`get`) στιγμιοτύπων από την λίστα.

7) Α. Αρχικά, δημιουργήστε ένα πίνακα (`array`) με όνομα `robarray` και προσθέστε τα αντικείμενα `r1`, `rv1`. Με μία δομή επανάληψης `for` εμφανίστε τα αποτελέσματα με την χρήση της `println()` μεθόδου και για τα 2 στιγμιότυπα. Προσπαθήστε τώρα να προσθέσετε και το τελευταίο αντικείμενο στον πίνακα. Τι παρατηρείτε;

Β. Δημιουργήστε τώρα ένα στιγμιότυπο της κλάσης `ArrayList` με όνομα `roblast` και προσθέστε και τα 3 στιγμιότυπα του παραδείγματος. Τι παρατηρείτε τώρα;

Γ. Να εμφανίσετε τα στοιχεία των αντικειμένων της λίστας με 2 τρόπους. Αρχικά, με την χρήση μίας δομής επανάληψης και των μεθόδων `roblast.size()`, που επιστρέφει το μέγεθος της λίστας και της `get(i)` μεθόδου της λίστας, που επιστρέφει το περιεχόμενο της λίστας στην θέση `i`. Έπειτα, με την χρήση της κλάσης `Iterator`, αφού, δημιουργήσετε ένα στιγμιότυπο της κλάσης αυτής και με τις μεθόδους `hasNext()` και `next()` να προσπελάσετε τα περιεχόμενα και να καλέσετε την μέθοδο `println()`, όπως φαίνεται παρακάτω:

```
Iterator iter =roblast.iterator();
while (iter.hasNext()){
    Robot rob = iter.next();
    rob.println();
}
```

Τι παρατηρείτε; Εξηγήστε το σφάλμα του μεταγλωττιστή και διορθώστε το αντικαθιστώντας κατάλληλα την εξής εντολή, η οποία δηλώνει ρητά τον τύπο του αντικειμένου κάνοντας δηλαδή `type casting`:

```
Robot rob =(Robot) iter.next();
```

Δ. Από την έκδοση Java 5.0 το πρόβλημα αυτό λύνεται με την χρήση γενικεύσεων (`generics`). Με τις γενικεύσεις μπορούμε να ορίσουμε μια δομή, π.χ. μία λίστα δίνοντας τον τύπο των περιεχομένων ως παράμετρο

```
List<Robot> test = new ArrayList<Robot>();
```

Και να καθορίσετε αυστηρά τον τύπο του `Iterator` που θα χρησιμοποιήσετε.

```
Iterator <Robot> iter = test.iterator();
```

Δημιουργήστε την λίστα `test` δίνοντας τον τύπο ως παράμετρο. Και επιβεβαιώστε ότι το `cast` πια δεν είναι απαραίτητο. Ε. Διαπεράστε τα στοιχεία της `roblast` με την χρήση της επαυξημένης μορφής της `for`:

```
for (Object obj : collection){  
    obj.doSomethingFunctionHere();  
}
```

Πότε είναι απαραίτητη η χρήση της κλάσης `iterator`;

5 Άσκηση 5: Χειρισμός I/O Streams

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με τον τρόπο που:

- α'. η JAVA δέχεται δεδομένα από τον χρήστη
- β'. μπορούμε να γράψουμε και να διαβάσουμε δεδομένα σε και από ένα αρχείο, αντίστοιχα στην JAVA.

1) Δημιουργήστε ένα νέο Project με όνομα Askisi5 όπως υποδείχθηκε στην πρώτη άσκηση.

A. Δημιουργήστε μία νέα κλάση με όνομα io_tester και αντικαταστήστε τον αυτόματα δημιουργημένο κώδικα με τον παρακάτω, σώστε την και μεταφράστε την:

```
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);
    }
}
```

B. Έγινε η μεταγλώττιση σωστά ή υπήρξε κάποιο μήνυμα σφάλματος; Στη δεύτερη περίπτωση να παραθέσετε στην αναφορά σας το σφάλμα και να κάνετε μία πρώτη υπόθεση για την αιτία του.

Γ. Να αντικαταστήσετε το σώμα της παραπάνω μεθόδου με το εξής τμήμα κώδικα:

```
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);
    return 0;
}
```

Το σφάλμα διορθώθηκε;

Δ. Να υλοποιήσετε και τις παρακάτω μεθόδους, να αποθηκεύσετε και να μεταγλωττίσετε το πρόγραμμα,

```

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);
        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);
        return "";}
}

public static boolean readBoolean() {
    byte b[] = new byte[16];
    String str;
    try {
        System.in.read(b);
        str = (new String(b)).trim();
        Boolean bool = Boolean.parseBoolean(str);
        return bool.booleanValue();
    } catch (IOException e) {
        System.out.println("Exception: " + e);
        return false;}
}

```

Διορθώστε τυχόν λάθος του μεταφραστή αν και όπου είναι απαραίτητο αυτό.

2) Να κατασκευάσετε την main μέθοδο στην κλάση IO_Tester, με σώμα όπως φαίνεται παρακάτω.

```

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);

```

Στη συνέχεια, σώστε και μεταφράστε την κλάση και ελέγξτε την σωστή λειτουργία της, δίνοντας τις τιμές '23' και '29,5' σε πρώτη φάση και σε δεύτερη τις τιμές 'ZZZ' και '9'.

3) Να αντικαταστήσετε την `readInt` με την παρακάτω και να ξανά μεταφράσετε την κλάση.

```
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);
        return 0;
    } catch (NumberFormatException e) {
        System.out.println("Exception: " + e + "\nReturned value: -1");
        return -1;
    }
}
```

Επαναλάβετε τον έλεγχο για τις τιμές 'ZZZ' και '9'. Τι παρατηρείτε; Να ενημερώσετε κατά αντιστοιχία και την μέθοδο `readFloat` με κώδικα παρόμοιο της `readInt`, ώστε να αντιμετωπίζονται παρόμοια προβλήματα εξαιρέσεων και τώρα να ελέγξετε για τις τιμές 'XXX' και 'YYY'.

4) Αντικαταστήστε το σώμα της `main` με τις ακόλουθες εντολές:

```
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);
```

Μεταφράστε και ελέγξτε το πρόγραμμα στις εξής 3 περιπτώσεις:

Περίπτωση 1:	'10'	'4'	'asd'	'True'
Περίπτωση 2:	'number'	'4.5f'	'input'	'true_not_again'
Περίπτωση 3:	'-1'	'ff'	'29'	'30'

Σημειώστε και δικαιολογήστε το μήνυμα που εμφανίζετε κάθε φορά.

5) Να προσθέσετε στο τέλος της main τις εξής γραμμές κώδικα. Μεταφράστε το πρόγραμμα και σημειώστε τις παρατηρήσεις σας.

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

A. Έβγαλε κατά τη μεταγλώττιση κάποιο σφάλμα;

B. Να αντικαταστήσετε τις γραμμές που μόλις προσθέσατε με τις παρακάτω, σώστε και μεταφράστε:

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

Τι παρατηρείτε; Έγινε η μετάφραση σωστά ή εμφανίστηκε πάλι κάποιο σφάλμα;

Γ. Αντικαταστήστε πάλι τον κώδικα με τον παρακάτω, σώστε και μεταφράστε.

```
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) {}
```

Αυτή τη φορά έγινε η μετάφραση σωστά;

Δ. Προσθέστε στο τέλος και την εξής εντολή:

```
catch (IOException e) {}
```

Μεταφράστε, εκτελέστε το πρόγραμμα με τις εξής εισόδους και σημειώστε τα αποτελέσματα σας: '10' '.25e-2' 'test.log' 'tRUe'

E. Κοιτάξτε στον κατάλογο του project για το αρχείο με όνομα test.log. Αν υπάρχει, ανοίξτε το και καταγράψτε τα περιεχόμενά του. Τι παρατηρείτε; Να αιτιολογήσετε την απάντησή σας.

6) Α. Να προσθέσετε τον παρακάτω κώδικα στην main της κλάσης IO_Tester, να την αποθηκεύσετε και να την μεταφράσετε. Αφού τρέξετε πάλι το πρόγραμμα να περιγράψετε την λειτουργία του κώδικα που μόλις προσθέσατε.

```
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. Τέλος, προσθέστε τον ακόλουθο κώδικα στην main και αφού τον μεταφράσετε και εκτελέσετε, να περιγράψετε την λειτουργία του.

```
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()); }
```

Γ. Ανοίξτε τώρα το αρχείο text.log και καταγράψετε τα περιεχόμενα του, αφού το εκτελέσετε για τα ίδια δεδομένα. Σχολιάστε τις παρατηρήσεις σας.

Σημείωση 3: Στην JAVA εκτός από την κλάση `DataInputStream` για ανάγνωση απο το πληκτρολόγιο υπάρχουν και άλλες αντίστοιχες κλάσεις, κάθε μία με τα πλεονεκτήματα και τα μειονεκτήματά της. Μερικές απο αυτές είναι:

- η [Scanner Class](#),
- η [Console Class](#),
- η [InputStreamReader Class](#).

6 Άσκηση 6: Εξαιρέσεις

Σκοπός της άσκησης αυτής είναι να εξοικειωθείτε με:

- α'. τις ετοιμές εξαιρέσεις της JAVA και
- β'. τη δημιουργία νέων εξαιρέσεων από το χρήστη.

1) Δημιουργήστε ένα νέο Project με όνομα Askisi6 όπως υποδείχθηκε στην πρώτη άσκηση.

A. Δημιουργήστε τώρα μια κλάση στο project με όνομα Exception_Tester. Και γράψτε στην κλάση τον παρακάτω κώδικα. Αφού μεταφράσετε την κλάση, να καταγράψετε τα αποτελέσματα σας.

```
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));  
    }  
}
```

B. Αρχικοποιήστε τώρα την την μεταβλητή $x2$ με 0. Μεταφράστε και εκτελέστε πάλι το πρόγραμμα. Τι αποτέλεσμα βγάζει;

2) Στην μέθοδο printResults να αντικαταστήσετε την εντολή System.out.println("Div: "+(a/b)); με το ακόλουθο τμήμα κώδικα:

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

Σώστε και μεταφράστε την κλάση. Τρέξτε την και σημειώστε πάλι τα αποτελέσματα σας. Τι παρατηρείτε;

3) Αντικαταστήστε τώρα τον κώδικα της main μεθόδου με τον παρακάτω:

```
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);
    }
}
```

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 την κλάση και ξανά τρέξτε την. Άλλαξαν τα αποτελέσματα; Τι άλλαξε ο προγραμματιστής εδώ στην διαχείριση της εξαίρεσης;

4) Αντικαταστήστε τώρα τον κώδικα της main μεθόδου με τον παρακάτω:

```
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 με τον ακόλουθο:

```
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 των κώδικα και βεβαιωθείτε ότι λειτουργεί σωστά.

5) Αντικαταστήστε ξανά τον κώδικα της main μεθόδου με τον παρακάτω:

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

A. Μεταφράστε και τρέξτε τη. Σημειώστε τα αποτελέσματα σας στην αναφορά.

B. Αρχικοποιήστε την $x2$ με την τιμή 2 και πάλι. Μεταφράστε και τρέξτε την κλάση με τις νέες αλλαγές. Σημειώστε τις παρατηρήσεις και τα αποτελέσματα στην αναφορά.

Γ. Να σημειώσετε και την παρατήρηση σας σχετικά με τα αποτελέσματα στο A. και στο B. σχετικά και με την παρουσία της πρότασης finally.

6) A. Να προσθέσετε τώρα στο τέλος της printResults τη γραμμή:

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

Μεταφράστε την κλάση και τρέξτε την. Σημειώστε τα καινούρια αποτελέσματα και σχολιάστε τα. Τι τελικά, κάνει η γραμμή που προστέθηκε;

B. Να αλλάξετε το 'catch block' στο εξής:

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

Μεταφράστε και τρέξτε και ελέγξτε την αλλαγή. Τι κάνει η νέα γραμμή που προστέθηκε;

7) Τώρα θα δημιουργήσουμε μία νέα κλάση, που θα εκφράζει μία νέα εξαίρεση, με συμπεριφορά ορισμένη από τον προγραμματιστή. Ονομάστε λοιπόν την νέα κλάση `DivideByZeroEx` και διαμορφώστε το σώμα της ως εξής:

```
public class DivideByZeroEx extends ArithmeticException {  
    public DivideByZeroEx() {}  
    public DivideByZeroEx(String msg) {super(msg);}  
}
```

A. Στη συνέχεια αλλάζτε τις γραμμές:

```
void printResults(int a, int b) throws ArithmeticException {  
    throw new ArithmeticException();
```

της μεθόδου `printResults` με τις εξής γραμμές, αντίστοιχα:

```
void printResults(int a, int b) throws DivideByZeroEx {  
    throw new DivideByZeroEx();
```

Παρατηρείτε κάποια αλλαγή στα αποτελέσματα; Δικαιολογήστε.

B. Προσθέστε στην κλάση `DivideByZeroEx` την μέθοδο `toString`, όπως φαίνεται παρακάτω:

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

και στη συνέχεια να αλλάξετε τον ακόλουθο κώδικα της `main`

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

στον εξής:

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

Σώστε, μεταφράστε και τρέξτε το πρόγραμμα και να σημειώσετε τα αποτελέσματα σας στην αναφορά.

Γ. Να αλλάξετε την τιμή της $x2$ μεταβλητής από 2 σε 0 και πάλι. Αφού κάνετε τις κατάλληλες διαδικασίες μεταγλώττισης - αποθήκευσης να εκτελέσετε το πρόγραμμα και να αποτυπώσετε τα αποτελέσματα στην αναφορά. Τι συμβαίνει τώρα;

Δ. Αντικαταστήστε την `printResults` με την παρακάτω και αφού εκτελέσετε σωστά το πρόγραμμα, σημειώστε και σχολιάστε τα αποτελέσματα.

```
void printResults(int a, int b) throws DivideByZeroEx {
    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 DivideByZeroEx(ae.toString());
    }
}
```

7 Άσκηση 7: Γραφικά

Σκοπός της άσκησης αυτής είναι η εξοικείωσή σας με το γραφικό περιβάλλον της JAVA.

1) Αρχικά, ανοίξτε το εργαλείο BlueJ και να δημιουργήσετε ένα νέο Project με όνομα askisi7. Αφού δημιουργήσετε μια κλάση με το όνομα MyAppBord, να εισάγετε τον ακόλουθο κώδικα και να την μεταφράσετε.

```
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);
    }
}
```

A. Τρέξτε τη και σημειώστε τα αποτελέσματα.

B. Αφαιρέστε τα σχόλια από τον κώδικα ξανά εκτελέστε το πρόγραμμα και σημειώστε τις παρατηρήσεις σας. Ποία η λειτουργία της μεθόδου setBounds;

2) Να δημιουργήσετε μία νέα κλάση με όνομα MyAppButton και το εξής περιεχόμενο:

```

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);
    }
}

```

A. Να την τρέξετε και να σημειώσετε τις παρατηρήσεις σας.

B. Δοκιμάστε τώρα να τοποθετήσετε στον κατάλογο του project ένα αρχείο εικόνας με όνομα icon_01.jpg. Δοκιμάστε τώρα να την ξανά τρέξετε και να σημειώσετε τα αποτελέσματα σας στην αναφορά.

Γ. Αφαιρέστε τα σχόλια απο την main και δοκιμάστε να αλλάξετε τις διαστάσεις του παραθύρου (resize). Σημειώστε και πάλι τα αποτελέσματα σας.

3) Τώρα δημιουργήστε μία κλάση με το όνομα MyAppCheck, με περιεχόμενο:

```
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);
    }
}
```

Μεταφράστε την, τρέξτε τη και σημειώστε το αποτέλεσμα. Τι παρατηρείτε για τα δύο panels, τόσο από θέμα στιγμιότυπου όσο και από εμφάνισης;

4) Δημιουργήστε μία κλάση με όνομα MyAppListsLabel, εισάγετε τον ακόλουθο κώδικα, αποθηκεύστε και μεταφράστε.

```
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 τα αποτελέσματά σας.

5) Δημιουργήστε μία κλάση με όνομα MyAppText, εισάγετε τον ακόλουθο κώδικα, αποθηκεύστε και μεταφράστε.


```

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);
        JTextField user = new JTextField(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 area1 = 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 τα αποτελέσματα σας.

6) Δημιουργήστε μία κλάση με το όνομα PLAFDemo είάγετε τον παρακάτω κώδικα, σώστε την και μεταφράστε την.

```
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);
    }
}
```

Να την τρέξετε χρησιμοποιώντας τις επιλογές '1', '2' και '3'. Σημειώστε τα αποτελέσματα.

7) Δημιουργήστε τώρα μία νέα κλάση CardWindow, εισάγετε τον ακόλουθο κώδικα και μεταφράστε την.

```
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) {
//if more components listened ItemEvents
//the followed commented lines would be necessary
// Object component = evt.getSource();
// if (component instanceof Choice) {
        ((CardLayout)cards.getLayout()).show(cards, (String)c.getSelectedItemAt());
// }
    }
    public static void main(String args[]) {
        CardWindow window = new CardWindow();
        window.setTitle("CardWindow Application");
        window.pack();
        window.setVisible(true);
    }
}
```

Να την τρέξετε και να σημειώσετε τα αποτελέσματα. Να αλλάξετε την τιμή στην λίστα επιλογής

και να καταγράψετε την παρατήρησή σας.

8) Δημιουργείστε μία κλάση με το όνομα MenuColorChanger, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε την.

```
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);
        }
    }
}
```

Δημιουργείστε μία κλάση με το όνομα MenuDemo, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε την.

```
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);
    }
}
```

A. Τρέξτε την MenuDemo και σημειώστε το αποτέλεσμα.

B. Αλληλεπιδράστε με τα Menu και τα Buttons και περιγράψτε την λειτουργία της εφαρμογής.

9) Δημιουργείστε μία κλάση με το όνομα MyAppEv, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε την.

```
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(L);
        app.setBounds(300,200,350,250);
        app.setVisible(true);
    }
}
```

A. Τρέξτε τη και σημειώστε το αποτέλεσμα.

B. Πατήστε μερικές φορές και περιγράψτε τι συμβαίνει.

10) Τέλος, δημιουργήστε μία κλάση με το όνομα MyAppDialog, εισάγετε τον ακόλουθο κώδικα, σώστε τη και μεταφράστε την.

```
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);
    }
}
```

A. Τρέξτε την και σημειώστε το αποτέλεσμα.

B. Πατήστε το κουμπί Delete και πείτε τι συμβαίνει.

Γ. Πατήστε yes και σημειώστε το αποτέλεσμα.

Δ. Τρέξτε πάλι την εφαρμογή και αυτή τη φορά επιλέξτε no. Σημειώστε πάλι το αποτέλεσμα.

8 Άσκηση 8: Πακέτα-Διασυνδέσεις

Σκοπός της άσκησης αυτής είναι η εξοικείωσή σας με τις διασυνδέσεις και τα πακέτα της JAVA.

1) Αρχικά, ανοίξτε το εργαλείο BlueJ και να δημιουργήσετε ένα νέο Project με όνομα askisi9. Στη συνέχεια, στο εργαλείο BlueJ να επιλέξετε Edit → New Package... και έτσι να δημιουργήσετε ένα νέο πακέτο με το όνομα javalab. Ελέγξτε τώρα αν δημιουργήθηκε ο αντίστοιχος κατάλογος μέσα στον κατάλογο του project. Επιλέξτε τον κατάλογο του πακέτου με διπλό κλικ και εκεί δημιουργήστε την παρακάτω κλάση με όνομα Vector.

```
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 μέσω του BlueJ στον αρχικό (root) κατάλογο του project και δημιουργήστε την κλάση RunMe με το εξής σώμα.


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

A. Σώστε την κλάση και μεταφράστε την. Τι παρατηρείτε; Δικαιολογήστε.

B. Αντικαταστήστε τον κώδικα της RunMe με τον εξής. Έγινε τώρα η μετάφραση κανονικά;

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

3) Αντικαταστήστε τον κώδικα της RunMe με τον εξής.

```
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));
    }
}
```

Τρέξτε τη και σημειώστε τα αποτελέσματα.

4) Πηγαίνετε τώρα στον root κατάλογο του project δημιουργήστε ένα νέο πακέτο με όνομα my_interfaces. Μεταβείτε στον νέο κατάλογο μέσω του περιβάλλοντος του BlueJ και εκεί

δημιουργήστε ένα νέο interface (New Class \rightarrow Interface), με τον ακόλουθο κώδικα και όνομα Operations.

```
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);
}
```

A. Σώστε τη και μεταφράστε την.

B. Προσθέστε στην κλάση Vector την παρακάτω γραμμή.

```
import my_interfaces.*;
```

Γ. Αλλάξτε την δήλωση της κλάσης Vector στην παρακάτω:

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

Δ. Προσθέστε την εξής 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;

E. Προσθέστε και τις παρακάτω μεθόδους:

```
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());
}
```

Έγινε τώρα η μετάφραση σωστά;

5) Προσθέστε στην μέθοδο 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));
```

Και την εξής μέθοδο:

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

A. Σώστε και μεταφράστε την κλάση.

B. Τρέξτε τη κλάση 2 φορές και σημειώστε τα αποτελέσματα σας.

9 Άσκηση 9: Νήματα

Σημαντικό!: Οι μικροεφαρμογές - applets εντάχθηκαν στην JAVA το 1995 και αποτελούν προγράμματα που τρέχουν από τον φυλλομετρητή και στη συνέχεια μέσα στο Java Virtual Machine - JVM σε μία διαδικασία ώστε να διαχωριστεί από τον φυλλομετρητή. Όμως, το 2013 οι πιο διάσημοι φυλλομετρητές ξεκίνησαν σταδιακά να αποβάλλουν την υποκείμενη τεχνολογία που χρησιμοποιούσαν οι μικροεφαρμογές. Μέσα στο χρονικό διάστημα 2015 - 2017 έγινε αδύνατον να λειτουργήσουν και συνεπώς στην 9η έκδοση της Java έγινε απόσβεση της κλάσης.

Άρα για την υλοποίηση της παρακάτω άσκησης θα πρέπει να χρησιμοποιηθεί η κατάλληλη έκδοση του [BlueJ](https://www.bluej.org/)[1]. Για το λόγο αυτό να εγκαταστήσετε την έκδοση 3.1.7, που είναι η τελευταία έκδοση που υποστηρίζει τις τεχνολογίες που πραγματεύεται η παρούσα άσκηση. Να επιλέξετε προς εγκατάσταση την πλήρη έκδοση που περιλαμβάνει και τον κατάλληλο μεταγλωττιστή.

[1]<https://www.bluej.org/versions.html>

Σκοπός της άσκησης αυτής είναι η εξοικείωσή σας με τα νήματα στην JAVA.

1) Αρχικά, ανοίξτε το εργαλείο BlueJ και να δημιουργήσετε ένα νέο Project με όνομα askisi10. Αφού δημιουργήσετε μια μικροεφαρμογή - applet με το όνομα MyClock, να εισάγετε τον ακόλουθο κώδικα και να την μεταφράσετε.

```
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);  
    }  
}
```

Να εκτελέσετε το Applet με `height = 500` και `width = 500` και να σημειώσετε τα αποτελέσματά σας.

2) Να αλλάξετε τον κώδικα της παραπάνω κλάσης όπως φαίνεται παρακάτω. Αποθηκεύστε τη και μεταφράστε την. Τέλος, τρέξτε την ξανά και αναφέρετε ποια διαφορά παρατηρείτε.

```
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.getDateInstance(DateFormat.FULL,
            DateFormat.FULL);
        g.drawString(df.format(d), 25, 40);
    }
}
```

3) Α. Δημιουργήστε μία νέα κλάση με όνομα RaceThread

```
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.");
    }
}
```

Β. Φτιάξτε και μία κλάση με όνομα 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();
    }
}
```

Γ. Τρέξτε την και σημειώστε τα αποτελέσματα.

Δ. Αντικαταστήστε το σώμα της main στην κλάση Runner με τις εξής εντολές.

```
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();
```

Να τρέξετε την νέα πια κλάση και να αναφέρετε τις παρατηρήσεις σας.

4) Α. Φτιάξτε και μία κλάση με όνομα Locker με τον ακόλουθο 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 και μία κλάση με όνομα Storer με τον ακόλουθο 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. Φτιάξτε και μία κλάση με όνομα Remover με τον ακόλουθο κώδικα σώστε τη και μεταφράστε τη.

```
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. Φτιάξτε και μία κλάση με όνομα 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(1, 1);
        Remover r = new Remover(1, 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 θα είναι ο ίδιος και την επόμενη φορά ή είναι κατά κάποιον τρόπο τυχαίος;

5) Α. Αλλάξτε τον κώδικα της 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 τη 2-3 φορές την LockerStory και σημειώστε τα αποτελέσματα σας.

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. Αφού σώσετε και μεταφράσετε την κλάση, να περιγράψετε τα αποτελέσματα.

```
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 ();
```

Ε. Αλλάξτε τον κώδικα της μεθόδου main της κλάσης LockerStory με τον ακόλουθο. Αφού σώσετε και μεταφράσετε την κλάση, να περιγράψετε τα αποτελέσματα.

```
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 ();
```