

```
import java.util.ArrayList;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // 2η εργασία (Λόγω αρχής υποκατάστασης, βάζω στο ArrayList μόνο αναφορές Student)
```

```
        ArrayList<Student> students = new ArrayList<>();
```

```
        //Εισαγωγή φοιτητή με αλληλεπίδραση με τον χρήστη
```

```
        boolean more = true;
```

```
        while(more){
```

```
            //Μέθοδος που εμφανίζει παράθυρο για εισαγωγή δεδομένων
```

```
            //Η μέθοδος είναι static οπότε δεν χρειάστηκε να φτιάξω κλάση.
```

```
            String name = JOptionPane.showInputDialog("Enter name: ");
```

```
            String id = JOptionPane.showInputDialog("Enter id: ");
```

```
            String answer = JOptionPane.showInputDialog("Type of Student: stud, grad, phd ");
```

```
            //Αρχή της υποκατάστασης. Δηλώνω μια μεταβλητή, ώστε να βάλω μια student.add στο τέλος
```

```
            Student student = null;
```

```
            //Τι θα γίνει αν δεν δηλώσει ο χρήστης και δεν μπει μέσα στις if; Δηλώνω null για αρχικοποίηση.
```

```
            //Με else, τότε θα έμπαινε στην τελευταία οπότε δεν θα χρειαζόταν αρχικοποίηση
```

```
            //Όταν συγκρίνω αντικείμενα κλάσεων, δεν μπορώ να χρησιμοποιήσω ==. Βλέπει μόνο αν είναι ίσες οι διευθύνσεις. Το ==  
            MONO σε String
```

```
            // Τι θα είναι ίσα στα αντικείμενα; Τα αντικείμενα έχουν πολλές μεταβλητές μέσα
```

```
            //equalsIgnoreCase = ανεξάρτητα με το αν είναι κεφαλαία ή μικρά
```

```
            //Εισάγω UnderGradStudent την νέα κλάση
```

```
            if(answer.equals("stud")){
```

```
                String yearText = JOptionPane.showInputDialog("Enter year: ");
```

```
                //Μετατρέπω ένα κείμενο που έχει ακέραιο αριθμό και τον μετατρέπει ως μεταβλητή τύπου ακέραιο
```

```
                int year = Integer.parseInt(yearText);
```

```
                student = new UnderGradStudent(name, id, year);
```

```
            }
```

```
            if(answer.equals("grad")){
```

```
                String supervisor = JOptionPane.showInputDialog("Supervisor: ");
```

```
                student = new GraduateStudent(name, id, supervisor);
```

```
            }
```

```
            if(answer.equals("phd")){
```

```
                String thesis = JOptionPane.showInputDialog("Thesis: ");
```

```
                student = new PhDStudent(name, id, thesis);
```

```
            }
```

```
            students.add(student);
```

```
            String another = JOptionPane.showInputDialog("Another student: yes, no");
```

```
            if(another.equals("no")){
```

```
                more = false;
```

```
            }
```

```
        }
```

//students = όνομα δομής δεδομένων που θέλω να τυπώσω.

//Student = τύπος δεδομένων μέσα στην δομή δεδομένων

//s = είναι αυθαίρετο όνομα. Είναι σαν το students.get(i)

```
for(Student s: students){  
    s.printlnInfo(); /** ΠΟΛΥΜΟΡΦΙΚΗ ΚΛΗΣΗ **/  
}
```

```
for(Student s: students){  
    s.printType();  
}
```

// //Καταχωρώ φοιτητές (δημιουργώ τα αντικείμενα και θέτω την αναφορά της ArrayList στο αντικείμενο φοιτητή)

// //OCP -> Εισάγω κλάσεις αλλά δεν πειράζω το πρόγραμμα της main πχ σε if ή for (ανοικτή κλειστή σχεδίαση)

// students.add(new Student("John", "ics19047"));

// students.add(new GraduateStudent("Mary", "mai20012", "Roberts"));

// students.add(new Student("Babis", "iis20113"));

// students.add(new PhDStudent("Helen", "phd123", "Software Quality"));

// for(int i=0 ; i<students.size(); i++){

// //ΔΥΝΑΜΙΚΗ ΔΙΑΣΥΝΔΕΣΗ. Ποιά printlnInfo είναι; --> ΠΟΛΥΜΟΡΦΙΣΜΟΣ (ανάλογα με τι είναι ο αποδέκτης, επιλέγεται το printlnInfo του και αποκρίνεται ανάλογα)

// students.get(i).printlnInfo();

// }

// //students = όνομα δομής δεδομένων που θέλω να τυπώσω.

// //Student = τύπος δεδομένων μέσα στην δομή δεδομένων

// //student = είναι αυθαίρετο όνομα. Είναι σαν το students.get(i)

// for(Student student: students){

// student.printlnInfo();

// }

//

// Student s1 = new Student("John", "ics20132");

//

// //Βάζω έναν GraduateStudent ως Student λόγω αρχής υποκατάστασης

// Student s2 = new GraduateStudent("Mary", "mai20098", "Nikolaou");

//

// // Στατική διασύνδεση με τον printlnInfo

// s1.printlnInfo();

//

// // ΔΥΝΑΜΙΚΗ ΔΙΑΣΥΝΔΕΣΗ ή ΚΑΘΥΣΤΕΡΗΜΕΝΗ ΔΙΑΣΥΝΔΕΣΗ. Την ώρα που διαβάζει, δεν ξέρει ποια μέθοδος θα κληθεί, την ώρα που τρέχει τον κώδικα αποφασίζει ποια μέθοδος θα κληθεί.

// s2.printlnInfo();

//

// //GraduateStudent gs1 = new GraduateStudent("Mary", "mai20098", "Nikolaou");

// //gs1.printlnInfo();

}

}