

Δομές Δεδομένων & Ανάλυση Αλγορίθμων

3ο Εξάμηνο

Άσκηση Πράξης 1

(υλοποίηση Εργαστηριακού Τμήματος και
Εργαστηριακού Μαθήματος)

http://www.iee.ihu.gr/~demos/teaching_GR.html

Δημοσθένης Σταμάτης

Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων
ΔΙ.ΠΑ.Ε

Η κλάση Student

// Απλουστευμένη εκδοχή με ΑΜ, ονοματεπώνυμο, απουσίες και βαθμό

```
public class Student{  
  
    private int am;  
    private String onoma;  
    private int apousies;  
    private double vathmos;  
  
    public Student(int am,String onoma){  
        this(am, onoma, 0,0.0);  
    }  
  
    public Student(int am,String onoma,int apousies, double vathmos){  
        this.am=am;  
        this.onoma=onoma;  
        this.apousies=apousies;  
        this.vathmos=vathmos;  
    }  
  
    public void setVathmos(double vathmos){  
        this.vathmos=vathmos;  
    }  
  
    public void setApousies(int apousies){  
        this.apousies=apousies;  
    }  
}
```

Η κλάση Student

// Απλουστευμένη έκδοχή με όνομα, ΑΜ, απουσίες και βαθμό

```
public int getAM() {  
    return am;  
}  
  
public String getOnoma() {  
    return onoma;  
}  
  
public double getVathmos() {  
    return vathmos;  
}  
  
public int getApousies() {  
    return apousies;  
}  
  
public String toString() {  
    return ("AM: "+am+"\nOnoma: "+onoma+"\nVathmos: "+vathmos  
        +"\nApousies: "+apousies);  
}  
}
```

1^η άσκηση πράξης: «Εργαστηριακό Τμήμα»

```
public interface LabInterface {  
  
    public boolean insert(Student S);  
    // Εισαγωγή φοιτητή στο εργαστηριακό τμήμα  
  
    public boolean delete(Student S);  
    // Διαγραφή φοιτητή από το τμήμα  
  
    public Student search(int AM);  
    // Αναζήτηση φοιτητή με βάση το αριθμό μητρώου.  
    // Αν δεν βρεθεί επιστρέφει null  
  
    public void presentations();  
    // Εκτύπωση παρουσιολόγιου για το τμήμα  
    // (AM φοιτητή, επώνυμο και όνομα φοιτητή, αριθμός απουσιών, βαθμός)  
  
    public int succeeded();  
    // Βρίσκει και επιστρέφει το πλήθος των φοιτητών  
    // του εργαστηριακού τμήματος που πέτυχαν στο μάθημα  
  
    public int failed();  
    // Βρίσκει και επιστρέφει το πλήθος των φοιτητών  
    // του εργαστηριακού τμήματος που απέτυχαν στο μάθημα  
  
    public double averageGrade();  
    // Βρίσκει και επιστρέφει το μέσο όρο βαθμολογίας των φοιτητών  
    // του εργαστηριακού τμήματος  
  
}
```

Βλέπε
Άσκηση Πράξης 1.1.

1^η άσκηση πράξης: «Εργαστηριακό Τμήμα»

```
public class Lab implements LabInterface {  
  
    private String LabName;  
    private int capacity;  
    private int size=0;  
    private Student[] LabArray;  
  
    public Lab(String Name, int cap){  
        LabName=Name;  
        capacity=cap;  
        LabArray = new Student[capacity];  
    }  
  
    public Lab(String Name){  
        this(Name, 20);  
    }  
  
    public int size(){  
        return size;  
    }  
  
    public String getName(){  
        return LabName;  
    }  
}
```

```
public boolean insert(Student S){
// Εισαγωγή φοιτητή στο εργαστηριακό τμήμα
    if (size>=capacity)
    {   System.out.println("Lab "+LabName+" is Full");
        return false;
    }
    else
    {
        LabArray[size++]=S;
        return true;
    }
}

public boolean delete(Student S){
// Διαγραφή φοιτητή από το τμήμα
    for (int i=0; i<size; i++)
        if (LabArray[i]==S)
        {   LabArray[i]=LabArray[--size];
            return true;
        }
    return false;
}

public Student search(int AM){
// Αναζήτηση φοιτητή με βάση το αριθμό μητρώου. Αν δεν βρεθεί επιστρέφει null
    for (int i=0; i<size; i++)
        if (LabArray[i].getAM()==AM) return LabArray[i];
    return null;
}
```

```
public void presentations() {
    // Εκτύπωση παρουσιολόγιου για το τμήμα
    // (AM φοιτητή - επώνυμο και όνομα φοιτητή- αριθμός απουσιών - βαθμός)
    for (int i=0; i<size; i++)
        System.out.println(LabArray[i].getAM()+"--"+LabArray[i].getEpitheto()
            +"--"+LabArray[i].getOnoma()
            +"--"+LabArray[i].getApousies()+"--"+LabArray[i].getVathmos());
}

public int succeeded() {
    // Βρίσκει και επιστρέφει το πλήθος των φοιτητών
    // του εργαστηριακού τμήματος που πέτυχαν στο μάθημα
    int count=0;
    for (int i=0; i<size; i++)
        if ((LabArray[i].getVathmos())>=5) && (LabArray[i].getApousies())<=2)
            count++;
    return count;
}

public int failed() {
    // Βρίσκει και επιστρέφει το πλήθος των φοιτητών
    // του εργαστηριακού τμήματος που απέτυχαν στο μάθημα
    return (size-succeeded());
}
```

```
public double averageGrade() {  
    // Βρίσκει και επιστρέφει το μέσο όρο βαθμολογίας των φοιτητών  
    // του εργαστηριακού τμήματος  
    double total=0.0;  
    for (int i=0; i<size; i++)  
        total=total+LabArray[i].getVathmos();  
    return total/size;  
}
```


1^η Άσκηση Πράξης: «Εργαστηριακό μάθημα»

// Το εργαστηριακό μάθημα περιλαμβάνει έναν αριθμό από τμήματα

```
public interface LaboratoryCourseInterface {  
  
    public void printNumberOfStudents ();  
    // Εκτυπώνει το συνολικό πλήθος των φοιτητών  
    // που παρακολουθούν το εργαστηριακό μάθημα  
  
    public void printSuccessStatus ();  
    // Εκτυπώνει το πλήθος των φοιτητών που πέτυχαν-απέτυχαν στο μάθημα  
    // ανά εργαστήριο και συνολικά για το εργαστηριακό μαθημα  
  
    public void printAverageCourseGrades ();  
    // Εκτυπώνει το μέσο όρο του βαθμού των φοιτητών  
    // ανά εργαστήριο και το συνολικό μέσο όρο για το μάθημα  
  
    public void printSuccessStatistics ();  
    // Εκτυπώνει τα ποσοστά επιτυχίας και αποτυχίας ανά εργαστήριο  
    // και συνολικά για το μάθημα  
}
```

Βλέπε
Άσκηση Πράξης 1.2.

1^η Άσκηση Πράξης: «Εργαστηριακό μάθημα»

```
public class LaboratoryCourse implements LaboratoryCourseInterface {

    private String CourseName;
    private int capacity;
    private int size=0;
    private Lab[] CourseArray;

    public LaboratoryCourse(String Name, int cap){
        CourseName=Name;
        capacity=cap;
        CourseArray = new Lab[capacity];
    }

    public LaboratoryCourse(String Name){
        this(Name, 5);
    }

    public boolean insert(Lab L){
        // Εργαστηριακού τμήματος στο μάθημα
        if (size>=capacity)
        {   System.out.println("Laboratory Course "+CourseName+" is Full");
            return false;
        }
        else
        {
            CourseArray[size++]=L;
            return true;
        }
    }
}
```

```
private int numberOfStudents() {  
    // Επιστρέφει το συνολικό πλήθος των φοιτητών  
    // που παρακολουθούν το εργαστηριακό μάθημα  
    int number=0;  
    for (int i=0; i<size; i++)  
        number=number+CourseArray[i].size();  
    return number;  
}  
  
public void printNumberOfStudents() {  
    // Εκτυπώνει το συνολικό πλήθος των φοιτητών  
    // που παρακολουθούν το εργαστηριακό μάθημα  
    System.out.print("Total number of students in Laboratory Course ");  
    System.out.println(CourseName + ": " + numberOfStudents());  
}
```

```
private int[] succesStatus() {  
    // Επιστρέφει το πλήθος των φοιτητών που πέτυχαν και απέτυχαν στο  
    // μάθημα ανά εργαστήριο και συνολικά για το εργαστηριακό μαθημα  
    int pin[] = new int[2*size+2];  
    int succeeded = 0;  
    int failed = 0;  
    for (int i=0; i<size; i++) {  
        pin[i]=CourseArray[i].succeeded();  
        succeeded = succeeded + CourseArray[i].succeeded();  
    }  
    // θέσεις 0 έως size-1: φοιτητές του κάθε τμήματος που πέτυχαν  
    for (int i=0; i<size; i++) {  
        pin[i+size]=CourseArray[i].failed();  
        failed = failed + CourseArray[i].failed();  
    }  
    // θέσεις size έως 2*size-1: φοιτητές του κάθε τμήματος που απέτυχαν  
    pin[2*size]=succeeded; // Το σύνολο επιτυχόντων στη θέση 2*size  
    pin[2*size+1]=failed; // Το σύνολο αποτυχόντων στη θέση 2*size+1  
    return pin;  
}
```

```
private int[] succesStatus() {  
    // Επιστρέφει το πλήθος των φοιτητών που πέτυχαν και απέτυχαν στο  
    // μάθημα ανά εργαστήριο και συνολικά για το εργαστηριακό μαθημα  
    int pin[] = new int[2*size+2];  
    int succeeded = 0;  
    int failed = 0;  
    for (int i=0; i<size; i++) {  
        pin[i] = CourseArray[i].succeeded();  
        succeeded = succeeded + CourseArray[i].succeeded();  
    }  
    // θέσεις 0 έως size-1: φοιτητές του κάθε τμήματος που πέτυχαν  
    for (int i=0; i<size; i++) {  
        pin[i+size] = CourseArray[i].failed();  
        failed = failed + CourseArray[i].failed();  
    }  
    // θέσεις size έως 2*size-1: φοιτητές του κάθε τμήματος που απέτυχαν  
    pin[2*size] = succeeded; // Το σύνολο επιτυχόντων στη θέση 2*size  
    pin[2*size+1] = failed; // Το σύνολο αποτυχόντων στη θέση 2*size+1  
    return pin;  
}
```

```
public void printSuccessStatus() {  
    // Εκτυπώνει το πλήθος των φοιτητών που πέτυχαν-απέτυχαν στο μάθημα  
    // ανά εργαστήριο και συνολικά για το εργαστηριακό μαθημα  
    int pin[]=successStatus();  
    for (int i=0; i<size; i++) {  
        System.out.print("In Lab " + CourseArray[i].getName() + " ");  
        System.out.print(pin[i] + " students succeeded and ");  
        System.out.println(pin[i+size] + " students failed");  
    }  
    System.out.print("In the course " + CourseName + " ");  
    System.out.print(pin[2*size] + " students succeeded and ");  
    System.out.println(pin[2*size+1] + " students failed");  
}
```

```
public void printAverageCourseGrades() {  
    double total=0.0;  
    for (int i=0; i<size; i++)  
        total=total+CourseArray[i].averageGrade();  
    System.out.print("Total average grade of students in Laboratory Course ");  
    System.out.println(CourseName +": "+total/size);  
}
```

```
public void printSuccesStatistics(){
// Εκτυπώνει τα ποσοστά επιτυχίας και αποτυχίας
// ανά εργαστήριο και συνολικά για το μάθημα
    int pin[]=succesStatus();
    for (int i=0; i<size; i++) {
        System.out.println("In Lab " + CourseArray[i].getName() + " ");
        System.out.println("  Percentage of students sucess: "+
            100*(float)pin[i]/(CourseArray[i].size()));
        System.out.println("  Percentage of students failure: "+
            100*(float)pin[i+size]/(CourseArray[i].size()));
    }
    System.out.println("In the course " + CourseName + " ");
    System.out.println("  Total percentage of students sucess: "+
        100*(float)pin[2*size]/numberOfStudents());
    System.out.println("  Total percentage of students failure: "+
        100*(float)pin[2*size+1]/numberOfStudents());
}
```



```
public static void main(String[] args) {  
    Student S1 = new Student(161111, "Xaitas", "Vasilis", 'M', 20, 0, 9);  
    Student S2 = new Student(161112, "Xaitas", "Antonis", 'M', 21, 1, 8);  
    Student S3 = new Student(161113, "Xaita", "Anna", 'F', 20, 0, 8);  
    Student S4 = new Student(161114, "Xaitas", "Yannis", 'M', 21, 0, 6);  
    Student S5 = new Student(161115, "Papaioannou=", "Dimitra", 'F', 20, 2, 5);  
    Student S6 = new Student(161116, "Papaioannou=", "Anastasia", 'F', 20, 1, 6);  
    Student S7 = new Student(161117, "Papaioannou=", "Katerina", 'F', 20, 1, 4);  
    Student S8 = new Student(161118, "Antoniou=", "Thanasis", 'M', 20, 3, 9);  
    Student S9 = new Student(161118, "Antoniou=", "Kostas", 'M', 21, 2, 1);  
    Student S10 = new Student(161118, "Antoniou=", "Dimitris", 'M', 20, 1, 5);  
    Student S11 = new Student(161118, "Antoniou=", "Nikos", 'M', 20, 1, 9);  
  
    Lab T1 = new Lab("Domes1", 4);  
    Lab T2 = new Lab("Domes2");  
    Lab T3 = new Lab("Domes3");  
  
    T1.insert(S1); T1.insert(S2); T1.insert(S3); T1.insert(S4);  
    T2.insert(S5); T2.insert(S6); T2.insert(S7);  
    T3.insert(S8); T3.insert(S9); T3.insert(S10); T3.insert(S11);  
}
```

```
System.out.println(T3.search(161120));
System.out.println(T1.delete(S4));
T2.delete(S6);
T1.insert(S6);
T2.insert(S4);
T3.presentations();
System.out.println(T3.succeeded());
System.out.println(T3.failed());
System.out.println(T3.averageGrade());

LaboratoryCourse Co=new LaboratoryCourse("DataStructures");

Co.insert(T1);
Co.insert(T2);
Co.insert(T3);

Co.printNumberOfStudents();
Co.printAverageCourseGrades();
Co.printSuccessStatus();
Co.printSuccessStatistics();
}
```