



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
UNIVERSITY OF PIRAEUS

# Αντικειμενοστρεφής Προγραμματισμός

## Ασκήσεις

### 28/5/2024

Εαρινό Εξάμηνο 2024  
Κούτσικας Χρήστος



# 1<sup>η</sup> άσκηση

```
public class Student{  
    private String name;  
    private double[] grades;  
  
    public Student (String name){}  
  
    public void addGrade(double grade) {}  
  
    public double getAvgGrade() {}  
}
```

α) Συμπληρώστε τον κώδικα στον constructor Student.

Arrays





# 1<sup>η</sup> άσκηση

```
public class Student{  
    private String name;  
    private double[] grades;  
  
    public Student (String name){  
        this.name = name;  
    }  
  
    public void addGrade(double grade) {}  
  
    public double getAvgGrade() {}  
}
```

α) Συμπληρώστε τον κώδικα στον constructor Student.

**Τι γίνεται με τον πίνακα των βαθμών;**

Πρέπει να δημιουργηθεί στη δήλωση του;  
Πρέπει να δημιουργηθεί στον constructor;  
Υπάρχει περίπτωση να εισαχθεί εξωτερικά από την κλάση, π.χ. με άλλο constructor;  
Ποιο πρέπει να είναι το μέγεθος του;



# 1<sup>η</sup> άσκηση

```
public class Student{
    private String name;
    private double[] grades;
    private int numGrades=0;

    public Student (String name){
        this.name = name;
        grades = new double[100];
    }

    public void addGrade(double grade) {}

    public double getAvgGrade() {}
}
```

α) Συμπληρώστε τον κώδικα στον constructor Student.

## Μια προσέγγιση:

Δημιουργείται στον constructor με ένα αρχικό μέγεθος 100  
Χρησιμοποιούμε τη μεταβλητή numGrades για να ξέρουμε  
ποιο είναι το πραγματικό πλήθος των βαθμών



# 1<sup>η</sup> άσκηση

```
public class Student{
    private String name;
    private double[] grades;
    private int numGrades=0;

    public Student (String name){
        this.name = name;
        grades = new double[100];
    }

    public void addGrade(double grade) {}

    public double getAvgGrade() {}
}
```

β) Με τη μέθοδο **addGrade** εισάγεται ο βαθμός μιας εξέτασης του φοιτητή. Συμπληρώστε τον κώδικα της μεθόδου.



# 1<sup>η</sup> άσκηση

```
public class Student{
    private String name;
    private double[] grades;
    private int numGrades=0;

    public Student (String name){
        this.name = name;
        grades = new double[100];
    }

    public void addGrade(double grade) {
        if (numGrades<100) {
            grades [numGrades]=grade;
            numGrades++;
        }
    }

    public double getAvgGrade() {}
}
```

β) Με τη μέθοδο **addGrade** εισάγεται ο βαθμός μιας εξέτασης του φοιτητή. Συμπληρώστε τον κώδικα της μεθόδου.



# 1<sup>η</sup> άσκηση

```
public class Student{
    private String name;
    private double[] grades;
    private int numGrades=0;

    public Student (String name){
        this.name = name;
        grades = new double[100];
    }

    public void addGrade(double grade) {
        if (numGrades<100) {
            grades[numGrades]=grade;
            numGrades++;
        }
    }

    public double getAvgGrade() {}
}
```

γ) Με τη μέθοδο **getAvgGrade** υπολογίζεται ο μέσος όρος των επιτυχημένων ( $\geq 5$ ) εξετάσεων ενός φοιτητή. Συμπληρώστε τον κώδικα της μεθόδου.



# 1<sup>η</sup> άσκηση

```
public class Student{
    private String name;
    private double[] grades;
    private int numGrades=0;

    public Student (String name){
        this.name = name;
        grades = new double[100];
    }

    public void addGrade(double grade) {
        if (numGrades<100) {
            grades[numGrades]=grade;
            numGrades++;
        }
    }

    public double getAvgGrade() {
        double sum=0;
        int cnt=0;
        for (int i=0; i<numGrades; i++) {
            if (grades[i]>=5){
                sum+=grades[i];
                cnt++;
            }
        }
        if (cnt!=0)
            return (sum/(double)cnt);
        else
            return 0;
    }
}
```

γ) Με τη μέθοδο **getAvgGrade** υπολογίζεται ο μέσος όρος των επιτυχημένων ( $\geq 5$ ) εξετάσεων ενός φοιτητή. Συμπληρώστε τον κώδικα της μεθόδου.





# Πως μπορούμε να αλλάξουμε το μέγεθος;

```
public class Student{

    public void addGrade(double grade) {
        int oldLength=grades.length;
        if (numGrades<grades.length) {
            grades[numGrades]=grade;
            numGrades++;
        }
        else {
            double[] gradesNew = new double[oldLength+100];
            for (int i; i<oldLength; i++) {
                gradesNew[i]=grades[i];
            }
            gradesNew[numGrades]=grade;
            numGrades++;
            grades=gradesNew;
        }
    }
}
```



# Άλλη λύση χωρίς numGrades

```
public class Student{
    private String name;
    private double[] grades;

    public Student (String name){
        this.name = name;
        grades = new double[100];
        for (double d: grades)
            d=-1.0;
    }

    public void addGrade(double grade) {
        for (int i=0; 100; i++) {
            if (grades[i]==-1) {
                grades[i]=grade;
                break;
            }
        }
    }

    public double getAvgGrade() {
        double sum=0;
        int cnt=0;
        for (int i=0; 100; i++) {
            if ((grades[i]!=-1) && (grades[i]>=5)) {
                sum+=grades[i];
                cnt++;
            }
        }
        if (cnt!=0)
            return (sum/(double)cnt);
        else
            return 0;
    }
}
```



# Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() { }
    public Point(int x, int y) { }
    public Point(Point p) { }

    public void setXY(int x, int y) {
        this.x=x;
        this.y=y;
    }
}
```

α) Συμπληρώστε τον κώδικα των τριών constructors, χρησιμοποιώντας τη μέθοδο **setXY**.

Exceptions





# Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() {
        setXY(0,0);
    }
    public Point(int x, int y) {
        setXY(x,y);
    }
    public Point(Point p) {
        setXY(p.x,p.y);
    }

    public void setXY(int x, int y) {
        this.x=x;
        this.y=y;
    }

}
```

α) Συμπληρώστε τον κώδικα των τριών constructors, χρησιμοποιώντας τη μέθοδο **setXY**.



## Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() {
        setXY(0,0);
    }
    public Point(int x, int y) {
        setXY(x,y);
    }
    public Point(Point p) {
        setXY(p.x,p.y);
    }

    public void setXY(int x, int y) {
        this.x=x;
        this.y=y;
    }
}
```

β) Εάν, με την κλάση Point θέλουμε να δημιουργούμε αντικείμενα που να αντιστοιχούν σε σημεία που ανήκουν στην ευθεία  $y=5x+3$ , προσθέστε κώδικα που να εξασφαλίζει την ορθότητα των δεδομένων.



# Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() throws Exception {
        setXY(0,0);
    }
    public Point(int x, int y) throws Exception {
        setXY(x,y);
    }
    public Point(Point p) throws Exception {
        setXY(p.x,p.y);
    }

    public void setXY(int x, int y) throws Exception {
        if (y==(5*x+3)) {
            this.x=x;
            this.y=y;
        }
        else throws new Exception("Point is invalid");
    }
}
```

β) Εάν, με την κλάση Point θέλουμε να δημιουργούμε αντικείμενα που να αντιστοιχούν σε σημεία που ανήκουν στην ευθεία  $y=5x+3$ , προσθέστε κώδικα που να εξασφαλίζει την ορθότητα των δεδομένων.



# Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() throws Exception {
        setXY(0,0);
    }
    public Point(int x, int y) throws Exception {
        setXY(x,y);
    }
    public Point(Point p) throws Exception {
        setXY(p.x,p.y);
    }

    public void setXY(int x, int y) throws Exception {
        if (y==(5*x+3)) {
            this.x=x;
            this.y=y;
        }
        else throws new Exception("Point is invalid");
    }
}
```

γ) Εάν η κλάση Point είναι μη μεταλλάξιμη, τι αλλαγές προτείνετε να γίνουν;



# Άσκηση 2

```
class Point{
    private int x;
    private int y;

    public Point() throws Exception {
        setXY(0,0);
    }
    public Point(int x, int y) throws Exception {
        setXY(x,y);
    }
    public Point(Point p) throws Exception {
        setXY(p.x,p.y);
    }

    public void setXY(int x, int y) throws Exception {
        if (y==(5*x+3)) {
            this.x=x;
            this.y=y;
        }
        else throws new Exception("Point is invalid");
    }
}
```

γ) Εάν η κλάση Point είναι μη μεταλλάξιμη, τι αλλαγές προτείνετε να γίνουν;

**Δεν πρέπει να υπάρχει μέθοδος που να αλλάζει τις τιμές των x, y πέρα από την setXY και η setXY δεν πρέπει να είναι public.**

**Εναλλακτικά οι ιδιότητες x, y μπορούν να δηλωθούν final.**





# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) { }
```

α) Συμπληρώστε τον κώδικα στον constructor **public Polygon(Polygon pol)**.

**ArrayList, Objects**





# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) {
        points = new ArrayList<Point>();
        for (Point p: pol.points)
            points.add(p);
    }
}
```

α) Συμπληρώστε τον κώδικα στον constructor **public Polygon(Polygon pol)**.



# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) {
        points = new ArrayList<Point>();
        for (Point p: pol.points)
            points.add(p);
    }

    public Boolean hasPoint(Point p) {}
}
```

β) Με τη μέθοδο **hasPoint(Point p)** ελέγχουμε εάν το σημείο p αποτελεί κορυφή του πολυγώνου. Για να αποτελεί κορυφή του πολυγώνου, αρκεί να υπάρχει κορυφή στο πολύγωνο που να έχει ίδιες συντεταγμένες με τις συνταταγμένες του σημείου p. Συμπληρώστε τον κώδικα της μεθόδου.



# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }

    public int getX() {return x;}
    public int getY() {return y;}
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) {
        points = new ArrayList<Point>();
        for (Point p: pol.points)
            points.add(p);
    }

    public Boolean hasPoint(Point p) {
        Boolean ret=false;
        for (Point p2: points)
            if ((p2.getX()==p.getX()) && (p2.getY()==p.getY())) {
                ret=true;
                break;
            }
        return ret;
    }
}
```

β) Με τη μέθοδο **hasPoint(Point p)** ελέγχουμε εάν το σημείο p αποτελεί κορυφή του πολυγώνου. Για να αποτελεί κορυφή του πολυγώνου, αρκεί να υπάρχει κορυφή στο πολύγωνο που να έχει ίδιες συντεταγμένες με τις συνταταγμένες του σημείου p. Συμπληρώστε τον κώδικα της μεθόδου.



# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }

    public int getX() {return x;}
    public int getY() {return y;}
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) {
        points = new ArrayList<Point>();
        for (Point p: pol.points)
            points.add(p);
    }

    public Boolean hasPoint(Point p) {
        Boolean ret=false;
        for (Point p2: points)
            if ((p2.getX()==p.getX()) && (p2.getY()==p.getY())) {
                ret=true;
                break;
            }
        return ret;
    }

    public Polygon getPositivePolygon() {}
}
```

γ) Με τη μέθοδο **getPositivePolygon** υπολογίζεται ένα νέο πολύγωνο που αποτελείται από τα σημεία που ανήκουν στο 1<sup>ο</sup> τεταρτημόριο (έχουν μη αρνητικές συντεταγμένες). Συμπληρώστε τον κώδικα της μεθόδου.



# Άσκηση 3

```
import java.util.ArrayList;

class Point{
    private int x;
    private int y;

    public Point (int x, int y){
        this.x=x;
        this.y=y;
    }

    public int getX() {return x;}
    public int getY() {return y;}
}

class Polygon {
    private ArrayList<Point> points;

    public Polygon() {
        points = new ArrayList<Point>();
    }

    public Polygon(Polygon pol) {
        points = new ArrayList<Point>();
        for (Point p: pol.points)
            points.add(p);
    }

    public Boolean hasPoint(Point p) {
        Boolean ret=false;
        for (Point p2: points)
            if ((p2.getX()==p.getX()) && (p2.getY()==p.getY())) {
                ret=true;
                break;
            }
        return ret;
    }

    public Polygon getPositivePolygon() {
        Polygon newpol = new Polygon();
        for (Point p: points)
            if ((p.x>=0) && (p.y>=0))
                newpol.points.add(p);
        return newpol;
    }
}
```

γ) Με τη μέθοδο **getPositivePolygon** υπολογίζεται ένα νέο πολύγωνο που αποτελείται από τα σημεία που ανήκουν στο 1<sup>ο</sup> τεταρτημόριο (έχουν μη αρνητικές συντεταγμένες). Συμπληρώστε τον κώδικα της μεθόδου.