



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
UNIVERSITY OF PIRAEUS

# Αντικειμενοστρεφής Προγραμματισμός

## Ασκήσεις

### 14/5/2024

Εαρινό Εξάμηνο 2024  
Κούτσικας Χρήστος

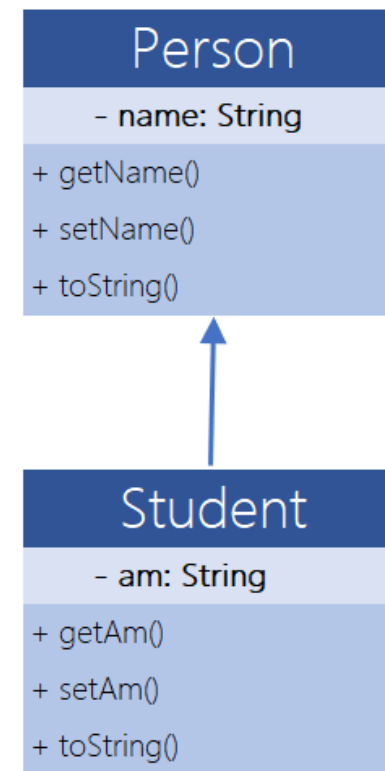
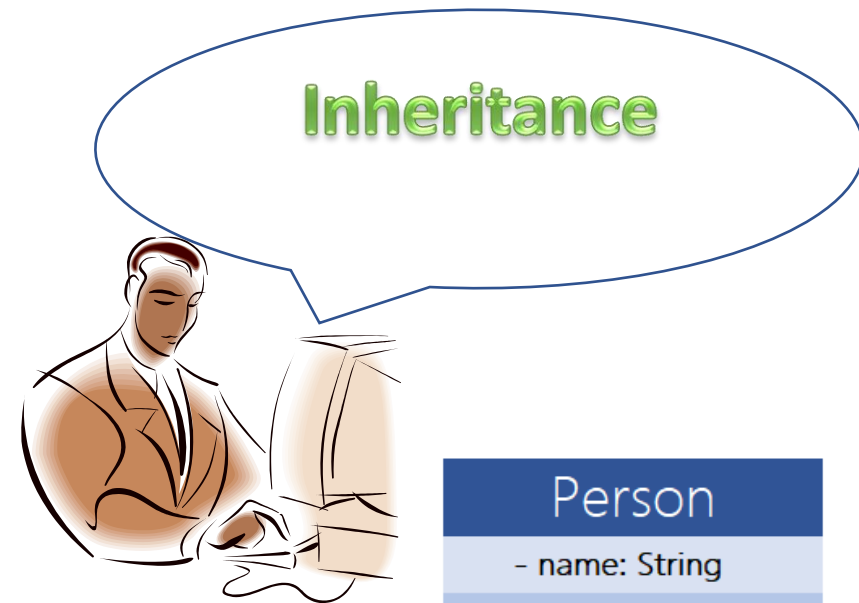


Δίνεται η παρακάτω κλάση:

```
public class mainApp {  
    public static void main(String[] args) {  
        Person p1 = new Person ("Iwannou Maria");  
        System.out.println(p1);  
        Person p2 = new Person ();  
        p2.setName("Iwannou Giannis");  
        System.out.println(p2);  
        Student s1 = new Student();  
        s1.setName("Karas Petros");  
        s1.setAm("3180000");  
        System.out.println(s1);  
        System.out.println("Main: Name " + s1.getName() + " AM " + s1.getAm());  
        Student s2 = new Student("Kara Iwanna", "3180046");  
        System.out.println(s2);  
        System.out.println("Main: Name " + s2.getName() + " AM " + s2.getAm());  
    }  
}
```

Γράψτε την κλάση `Person` που παριστά ένα άτομο με μοναδική ιδιότητα το όνομα (`name`) του ατόμου. Ορίστε τον ή τους κατασκευαστή/ες που απαιτούνται με βάση τον τρόπο δημιουργίας αντικειμένων τύπου `Person` στη `mainApp` κλάση που σας δίνεται. Επίσης ορίστε μεθόδους `get` και `set` για την ιδιότητα `name` και μέθοδο `toString`.

# 1<sup>η</sup> άσκηση





# 1<sup>η</sup> άσκηση

```
class Person {
    String name;

    public void setName(String name) {
        this.name = name;
    }
    public Person(String name) {
        this.name = name;
    }

    public Person() {
    }

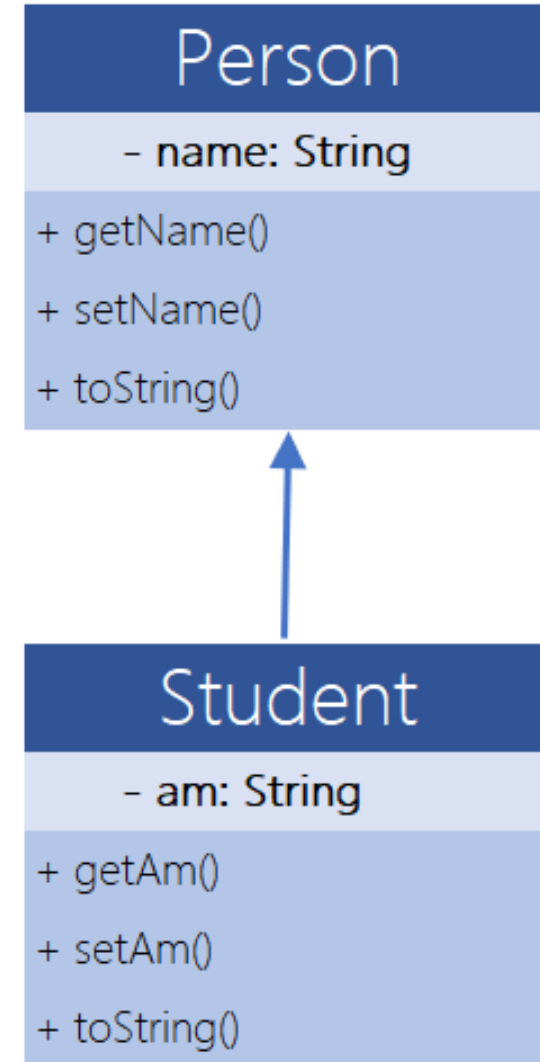
    public String getName() {
        return name;
    }
}

@Override
public String toString() {
    return String.format("Name: %s%n", name);
}
}
```



# 1<sup>η</sup> άσκηση

Γράψτε την κλάση Student που κληρονομεί την κλάση Person με επιπλέον - του ονόματος - ιδιότητα τον αριθμό μητρώου (am) του φοιτητή. Ορίστε τον ή τους κατασκευαστή/ες που απαιτούνται με βάση τον τρόπο δημιουργίας αντικειμένων τύπου Student στη mainApp κλάση που σας δίνεται. Επίσης ορίστε μεθόδους get και set για τον αριθμό μητρώου και μέθοδο toString.





# 1<sup>η</sup> άσκηση

```
class Student extends Person{
```

```
    String am;
```

```
    public void setAm(String am) {
```

```
        this.am = am;
```

```
    }
```

```
    public String getAm() {
```

```
        return am;
```

```
    }
```

```
public Student(String name, String am) {
```

```
    super(name);
```

```
    this.am=am;
```

```
}
```

```
public Student() {
```

```
}
```

```
@Override
```

```
public String toString() {
```

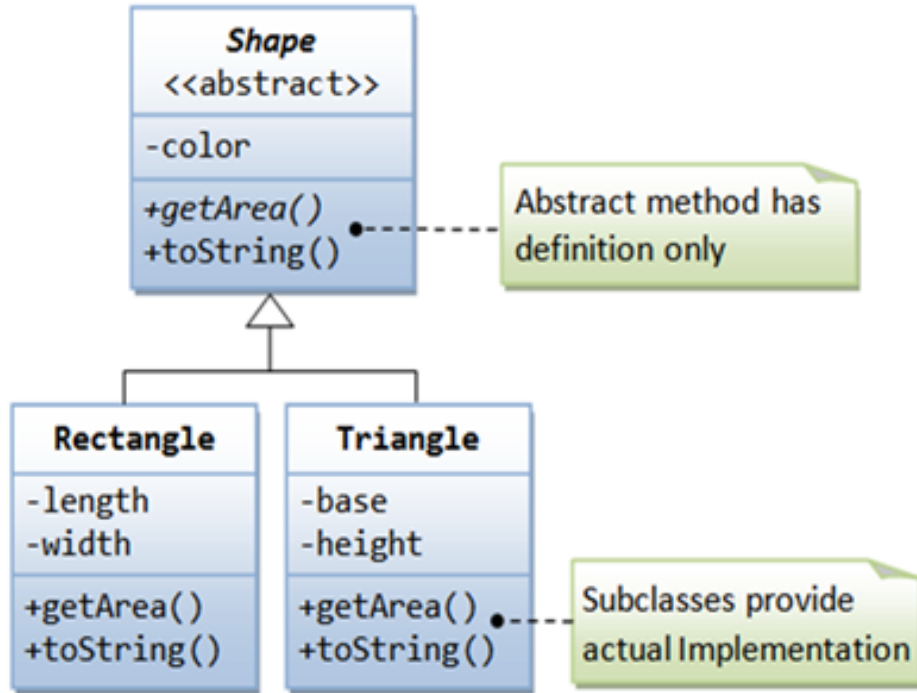
```
    return super.toString()+String.format("Am: %s%n", am);
```

```
}
```

```
}
```



# 2<sup>η</sup> άσκηση



Ορίστε την αφηρημένη κλάση Shape έτσι ώστε να έχει τη μεταβλητή color, μια μέθοδο-κατασκευαστή που να δίνει αρχική τιμή στη μεταβλητή αυτή και την αφηρημένη μέθοδο getArea() που όταν υλοποιηθεί θα πρέπει να υπολογίζει και να επιστρέφει το εμβαδόν του σχήματος. Επίσης θα πρέπει να υλοποιήσετε τη μέθοδο toString που εμφανίζει το χρώμα του σχήματος.



## 2<sup>η</sup> άσκηση

```
abstract class Shape {
    private String color;
    abstract double getArea();

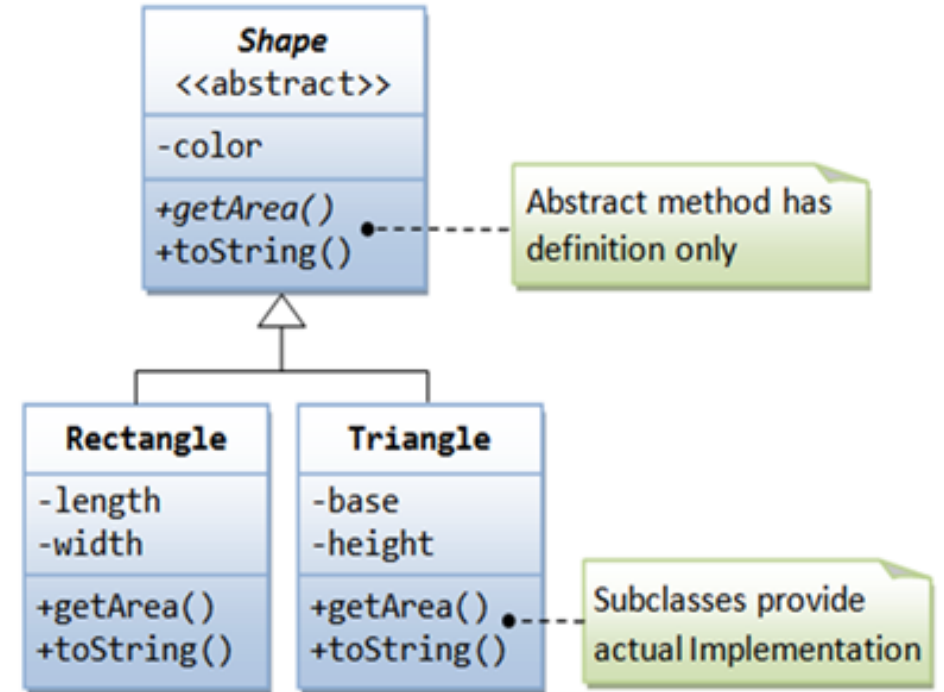
    public Shape(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return String.format("Color=%s%n", color);
    }
}
```



# 2<sup>η</sup> άσκηση

Γράψτε την κλάση Rectangle. Ορίστε τις μεταβλητές length και width, μια μέθοδο-κατασκευαστή που δίνει τιμές στις ιδιότητες του ορθογωνίου παραλληλόγραμου (χρώμα, μήκος, πλάτος). Επίσης υλοποιείστε τη μέθοδο getArea() και μια μέθοδο toString που εμφανίζει τις ιδιότητες του παραλληλόγραμμου.







## 2<sup>η</sup> άσκηση

```
class Rectangle extends Shape {
    double length, width;

    public Rectangle(String color, double length, double width) {
        super(color);
        this.length = length;
        this.width = width;
    }

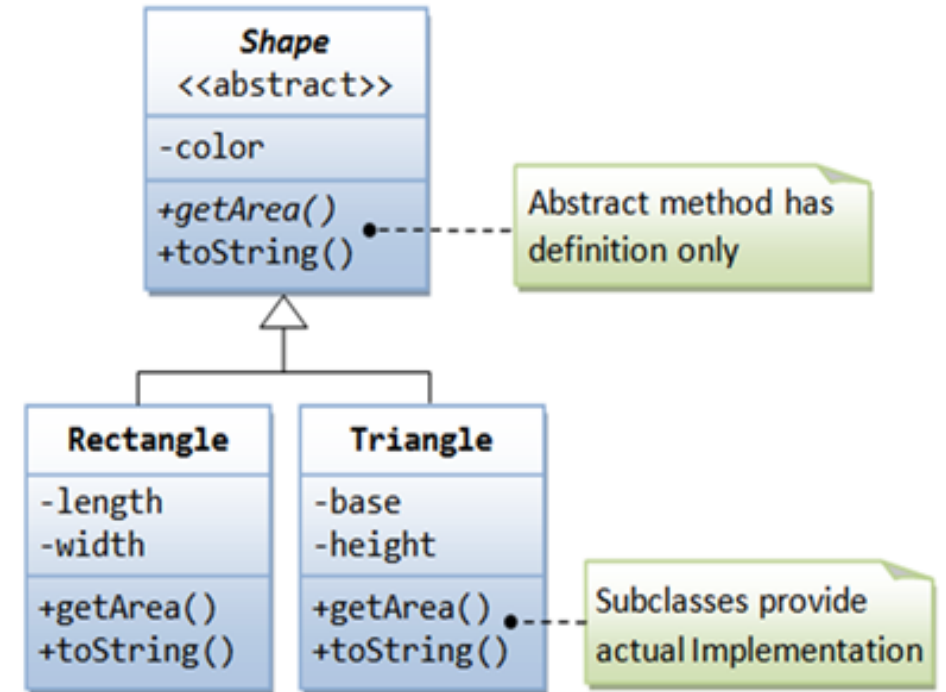
    public double getArea() {
        return length*width;
    }

    @Override
    public String toString() {
        return String.format("Length=%5.2f\nWidth=%5.2f\n", length, width);
    }
}
```



# 2<sup>η</sup> άσκηση

Γράψτε την κλάση Triangle. Ορίστε τις μεταβλητές base και height, μια μέθοδο-κατασκευαστή που δίνει τιμές στις ιδιότητες του τριγώνου (χρώμα, βάση, ύψος). Επίσης υλοποιείστε τη μέθοδο getArea() και μια μέθοδο toString που εμφανίζει τις ιδιότητες του τριγώνου.





# 2<sup>η</sup> άσκηση

```
class Triangle extends Shape {
    double base, height;

    public Triangle(double base, double height, String color) {
        super(color);
        this.base = base;
        this.height = height;
    }

    public double getArea() {
        return 0.5*base*height;
    }

    @Override
    public String toString() {
        return String.format("Base=%5.2f%Height=%5.2f%n",base,height);
    }
}
```



# 2<sup>η</sup> άσκηση

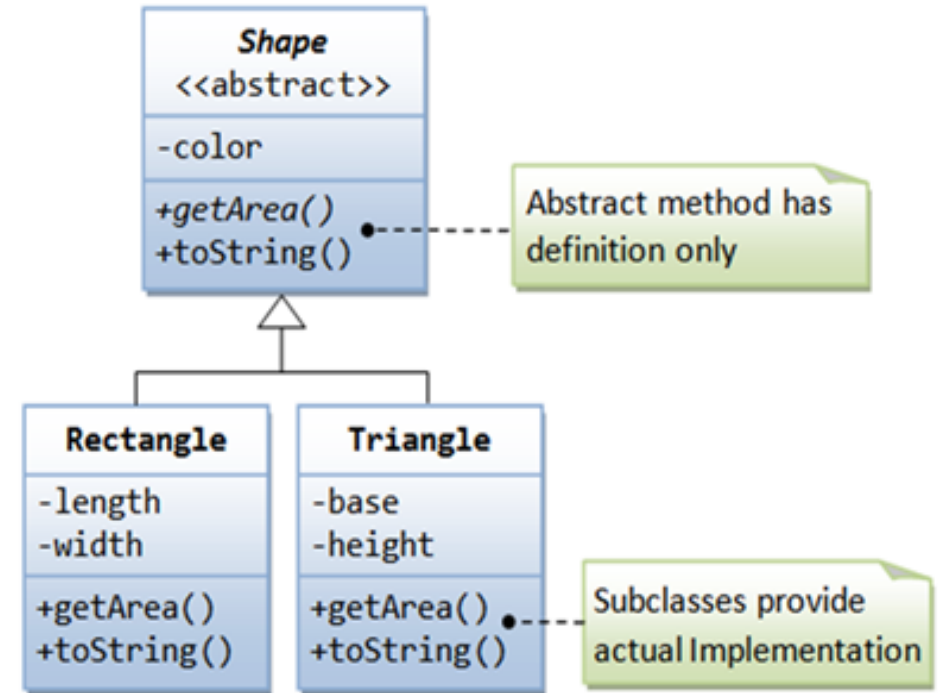
Πρέπει η κλάση Shape να δηλωθεί abstract;

Πρέπει η μέθοδος getArea να δηλωθεί abstract;

Πρέπει η μέθοδος getArea της Rectangle να δηλωθεί abstract;

Μπορούμε να δημιουργήσουμε στιγμιότυπα της κλάσης Shape;

Μπορούμε να δημιουργήσουμε στιγμιότυπα των κλάσεων Rectangle και Triangle;





# 2<sup>η</sup> άσκηση

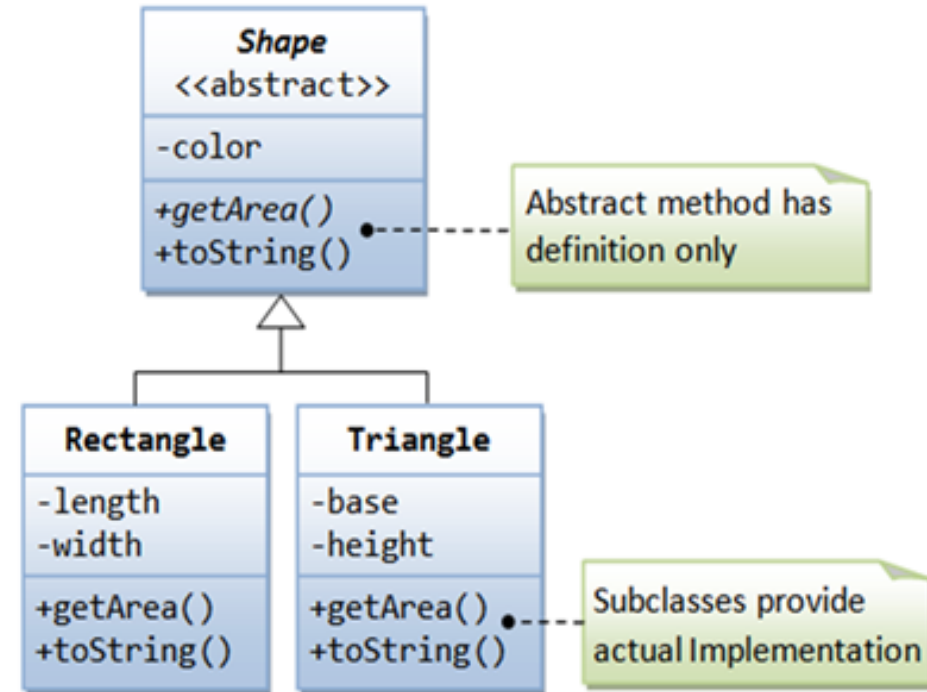
Πρέπει η κλάση Shape να δηλωθεί abstract; ΝΑΙ

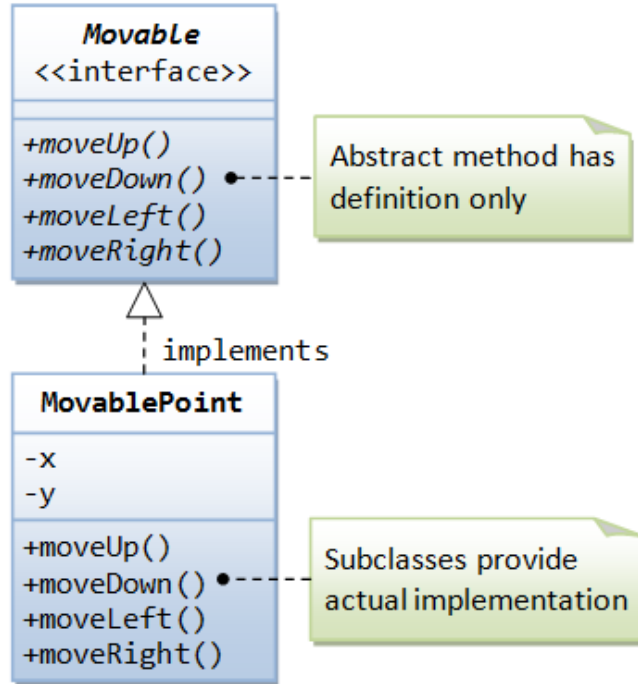
Πρέπει η μέθοδος getArea να δηλωθεί abstract; ΝΑΙ

Πρέπει η μέθοδος getArea της Rectangle να δηλωθεί abstract; ΟΧΙ

Μπορούμε να δημιουργήσουμε στιγμιότυπα της κλάσης Shape; ΟΧΙ

Μπορούμε να δημιουργήσουμε στιγμιότυπα των κλάσεων Rectangle και Triangle; ΝΑΙ





# 3<sup>η</sup> άσκηση

Interface



```
public class TestMovable {  
  
    public static void main(String[] args) {  
  
        MovablePoint m1 = new MovablePoint(8, 9);  
        System.out.println(m1);  
  
        m1.moveDown();  
        System.out.println(m1);  
  
        m1.moveRight();  
        System.out.println(m1);  
  
        m1.moveUp();  
        System.out.println(m1);  
  
        m1.moveLeft();  
        System.out.println(m1);  
  
    }  
}
```



# 3<sup>η</sup> άσκηση

```
interface Movable {  
    void moveUp();  
    void moveDown();  
    void moveLeft();  
    void moveRight();  
}
```

```
class MovablePoint implements Movable{  
    private double x,y;  
  
    public void moveUp() {y++; System.out.printf("Moving up. New position (%f, %f)", x,y);};  
    public void moveDown() {y--; System.out.printf("Moving down. New position (%f, %f)", x,y);};  
    public void moveLeft() {x++; System.out.printf("Moving left. New position (%f, %f)", x,y);};  
    public void moveRight() {x--; System.out.printf("Moving right. New position (%f, %f)",  
x,y);};  
}
```