

Object-Oriented Programming

Lab session #2



To learn more about UML for class diagram: <https://www.youtube.com/watch?v=UI6lqHOVHic>

To learn how to use Lucidchart:

<https://www.youtube.com/playlist?list=PLUoebdZqEHTz0aKtk6dygh4dQIz6WDP99>

<https://www.lucidchart.com/pages/uml-class-diagram>

Question 1: Points in 2 Dimension

(25 points)

Write a class **Point2D** representing points in 2D space with integer coordinates. The **Point2D** class provides the following common methods on points:

- Initialization constructors (3 constructors overloading):
 - **Point2D()**: initialize the point at the origin (0,0).
 - **Point2D(int x, int y)**: initialize point with x coordinate, y coordinate.
 - **Point2D(Point2D p)**: initialize point at the position of point p.
- Enter the coordinates for the point from the keyboard: **public void input()**.
- Returns a string representing the coordinates of the point as "(x, y)": **public String ToString()**.
- Move point to new coordinate: **public void move(int x, int y)**.
- Check if the point is origin (0,0): **public boolean isOrigin()**.
- Function to calculate the distance from point to point p: **public double distance(Point2D p)**.
- Static function to calculate the distance between two points p1 and p2: **public static double distance(Point2D p1, Point2D p2)**.

To test your Point2D class, your main method could look like:

```
public class TestingPoint2D {
    public static void main(String[] args) {
        //Test my Point2D class
        Point2D p1 = new Point2D();
        System.out.printf("The initial value of p1: %s\n", p1);
        System.out.printf("Is p1 at the origin?: %s\n", p1.isOrigin());
        System.out.println("Asking user to change values for p1!");
        p1.input();
        System.out.printf("The new value of p1: %s\n", p1);
        Point2D p2 = new Point2D(4,7);
        System.out.printf("The value of p2: %s\n", p2);
        Point2D p3 = new Point2D(p2);
        System.out.printf("The value of p3: %s\n", p3);
        System.out.printf("First way to calculate distance between p1 and p2: %.2f\n",
            p1.distance(p2));
        System.out.printf("Second way to calculate distance between p1 and p2: %.2f\n",
            Point2D.distance(p1, p2));
        System.out.printf("First way to calculate distance between p2 and p3: %.2f\n",
            p2.distance(p3));
        System.out.printf("Second way to calculate distance between p2 and p3: %.2f\n",
            Point2D.distance(p2, p3));
    }
}
```

The sample output could be:

```
The initial value of p1: (0,0)
Is p1 at the origin?: true
Asking user to change values for p1!
Enter X:
3
Enter Y:
5
The new value of p1: (3,5)
The value of p2: (4,7)
The value of p3: (4,7)
First way to calculate distance between p1 and p2: 2.24
Second way to calculate distance between p1 and p2: 2.24
First way to calculate distance between p2 and p3: 0.00
Second way to calculate distance between p2 and p3: 0.00
```

After finishing question 1, please **create an UML class diagram** for class **Point2D**!

Question 2: Triangle Calculation

(25 points)

Write a **Triangle** class that represents the information of a triangle in 2D space with **3 vertices of type Point2D, which you created in Question 1.**

The Triangle class provides the following common operations on points:

- Initialization constructor:
 - **Triangle(Point2D p1, Point2D p2, Point2D p3)**
- Calculate the perimeter of the triangle: **public double perimeter()**. (perimeter: chu vi)
- Calculate the area of the triangle: **public double area()**. (area: diện tích)

Hint: the way to calculate the area of a triangle with 3 sides can be found here:

[cuemath.com/measurement/area-of-triangle-with-3-sides](https://www.cuemath.com/measurement/area-of-triangle-with-3-sides)

To test your Triangle class, your main method could look like:

```
public class TriangleTesting {  
    public static void main(String[] args) {  
        // write your code here  
        Point2D p1 = new Point2D();  
        Point2D p2 = new Point2D(2,7);  
        Point2D p3 = new Point2D(3,4);  
        System.out.println(p1);  
        System.out.println(p2);  
        System.out.println(p3);  
        Triangle myTriangle = new Triangle(p1, p2, p3);  
        System.out.printf("The perimeter is %.2f\n", myTriangle.perimeter());  
        System.out.printf("The area is %.2f\n", myTriangle.area());  
    }  
}
```

The sample output could be:

```
(0,0)  
(2,7)  
(3,4)  
The perimeter is 15.44  
The area is 6.50
```

This is the correct answer (yayyyy!!!) for the triangle with 3 points (0,0), (2,7) and (3,4) because you can check the result online here:

<https://www.triangle-calculator.com/?what=vc>

myTriangle looks like this with 3 points (0,0), (2,7) and (3,4) which matches exactly with my above calculations:

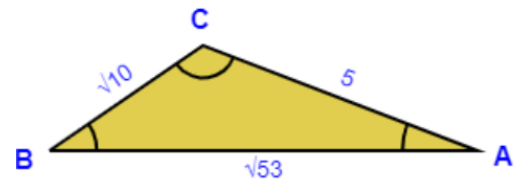
Obtuse scalene triangle.

Sides: $a = 3.162$ $b = 5$ $c = 7.28$

Area: $T = 6.5$

Perimeter: $p = 15.442$

Semiperimeter: $s = 7.721$

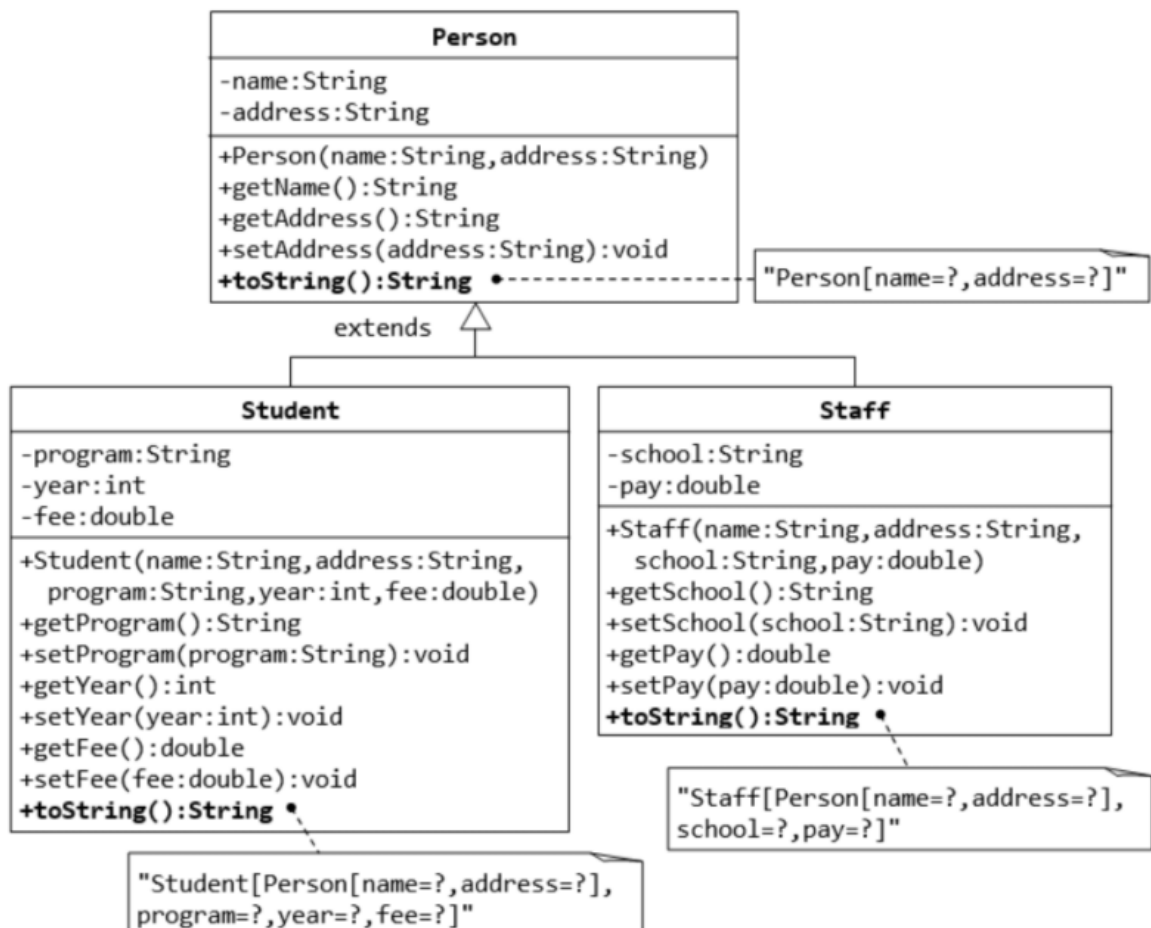


After finishing question 2, please **create an UML class diagram for class Triangle, class Point2D and their relationship**.

Question 3: Inheritance for Student and Staff

(50 points)

Build the **superclass class Person** and **2 subclass classes Student and Staff** inherits from Person class as follows:



Hints:

Step 1: Create the superclass (base class) Person

```
public class Person {  
    // Declare attributes  
  
    // Declare Methods  
  
}
```

Step 2: Create class Student which is inherited from class Person

- Declare attributes and methods for class Student
- Overriding method toString()

```
public class Student extends Person{  
    // Declare attributes  
  
    // Declare Methods  
  
    //Override toString() method  
    @Override  
    public String toString(){  
  
    }  
}
```

Step 3: Create class Staff which is also inherited from class Person just like in step 2

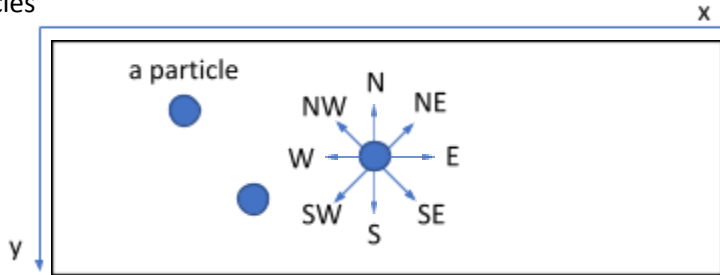
```
public class Staff extends Person{  
    // Declare attributes  
  
    // Declare Methods  
  
    //Override toString() method  
    @Override  
    public String toString(){  
  
    }  
}
```

After finishing these three classes, please do your own testing in your main method to create different objects for Students and Staff. You can try to print out information in each object, or change that information or even create a list of students and staff.

Question 4 (Bonus): Particle Behavior in the Box Simulation

(100 points)

Box of particles



Imagine that we have a box to store particles. Initially, we randomly place 3 particles in that box. After each step, these particles will move freely inside the box (move randomly). If two particles collide, a new particle will be placed randomly in the box.

We want to simulate the movement particles for n steps and count the number of particles in the box.

Implement a box of particles (write a **class of box**, a **class for particle**) in such a way that

1. A box has a fixed size: fixed width and height (10pts)
2. Each particle has a position (x, y) where $0 \leq x \leq \text{width of the box}$, and $0 \leq y \leq \text{height of the box}$ (10pts)
3. A particle can move randomly in one of the directions below but cannot move out of the box (20pts).
 - + North (decreasing its y by 1),
 - + North East (decreasing its y by 1 and increasing its x by 1),
 - + East (increasing its x by 1),
 - + South East (increasing its y by 1 and increasing its x by 1)
 - + South (increasing its y by 1),
 - + South West (increasing its y by 1 and decreasing its x by 1),
 - + West (decreasing its x by 1),
 - + North West (decreasing its y by 1 and decreasing its x by 1)
- Hint: you can try to organize directions as an arrayList or declare an enum type.
4. If two particles collide, a new particle will be placed randomly in the box (20pts)

Write a main class (or testing class) for simulation where for each step:

5. It makes all particles in the box move (5pts)
6. It shows the number of particles in the box (5pts)
7. It visualizes the box with particles inside (10pts)*

```

-----
| *  *      * |
|   *      ** |
|       *     |
|-----
  
```

8. Search about singleton pattern and make the box as a singleton (20pts)

Reference:

<https://www.baeldung.com/java-singleton>

<https://gpcoder.com/4190-huong-dan-java-design-pattern-singleton/>

