

Phillip Ball

CST-239

Prof. Couch

4/26/2022

Activity 3: Interfaces and Polymorphism

Part 1: Person Interface

```
These persons are not identical using ==
These persons are not identical using equals()
This copied person is identical using equals()
My class is class app.Person Will Ball
My class is class app.Person Chelsea Ball
My class is class app.Person Will Ball
I am walking
I am running
Person 1 is running true
I am walking
Person 1 is running false
```

```
These persons are not identical using ==
These persons are not identical using equals()
This copied person is identical using equals()
My class is class app.Person Will Ball
My class is class app.Person Chelsea Ball
My class is class app.Person Will Ball
I am walking
I am running
Person 1 is running true
I am walking
Person 1 is running false
My class is class app.Person Chelsea Ball
My class is class app.Person Phillip Ball
My class is class app.Person Troy Ball
My class is class app.Person William Ball
```

```
My class is class app.Person Phillip Ball
My class is class app.Person Troy Ball
My class is class app.Person Chelsea Smith
My class is class app.Person William Smith
My class is class app.Person Chelsea Ball 18
My class is class app.Person Phillip Ball 20
My class is class app.Person William Ball 22
My class is class app.Person Troy Ball 23
My class is class app.Person William Ball 22
My class is class app.Person Phillip Ball 20
My class is class app.Person Chelsea Ball 18
```

The output was displayed in screenshot one due to the implementation of the PersonInterface Walk();, Run();, and isRunning(); methods being implemented inside of the actual Person class.

Screenshot 2 and 3 are both the same function but the second screenshot has all the same last name and the third screenshot displays 2 sets of last names, which shows that the function is sorting correctly. Screenshot 4 and 5 is the same thing but with age being sorted instead of first name. Screenshot 4 is low to high and screenshot 5 is high to low.

Part 2: Polymorphic Shapes

```
<terminated> test (2) [Java Application] C:\Program Files\Java\jdk-18\bin\
This is a shape named Rectangle with an area of 2000
This is a shape named Triangle with an area of 250

package app;

public class Test
{
    public static void main(String[] args)
    {
        ShapeBase[] shapes = new ShapeBase[4];
        shapes[0] = new Rectangle(10, 200, "Rectangle");
        shapes[1] = new Triangle(10, 50, "Triangle");
        shapes[2] = new Circle(12, "Circle");
        shapes[3] = new Hexagon(34, "Hexagon");

        for(int i = 0; i < shapes.length; i++)
        {
            DisplayArea(shapes[i]);
        }
    }

    private static void DisplayArea(ShapeBase shape)
    {
        System.out.println("This is a shape named "
            + shape.GetName()
            + " with an area of "
            + shape.CalculateArea());
    }
}
```

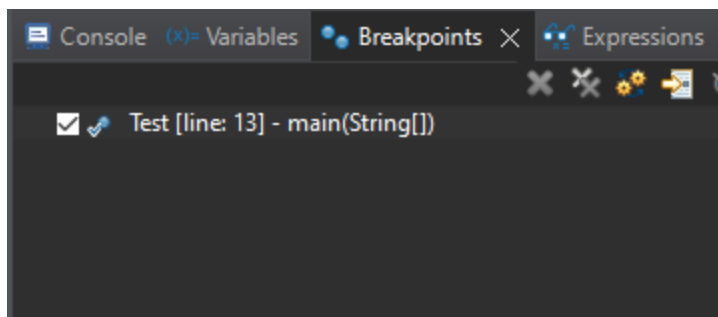
Polymorphism was demonstrated here by all of the class extensions from ShapeBase having the ability to be completely different from one another although they share the same base class. These classes are being morphed by their own terms and continuously share the base. Screenshot 2 displays the area of a circle with a radius of 12 and displays the area of a hexagon with a side length of 34.

Part 3: Polymorphic Weapons

```
-----> I am a Bomb
In the Bomb.Activate() with an enable of true
In Bomb.FireWeapon() with a power of 5
In the Gun.Activate() with an enable of true
In Gun.FireWeapon() with a power of 5
```

Polymorphism was demonstrated here from both the Bomb and Gun class extending from a WeaponInterface outputting separately from their own classes. Using @Override functions, these classes are able to use the same exact functions but differently.

Part 4: Practice Using the Debugger



Name	Value
no method return value	
args	String[0] (id=20)
shapes	ShapeBase[4] (id=21)

Name	Value
no method return value	
this	Rectangle (id=27)
width	10
height	200
name	"Rectangle" (id=30)

Test (2) [Java Application]

- app.Test at localhost:64495
 - Thread [main] (Suspended)
 - Rectangle.<init>(int, int, String) line: 13
 - Test.main(String[]) line: 13

C:\Program Files\Java\jdk-18\bin\javaw.exe (Apr 26, 2022, 4:48:49 PM) [pid: 8484]