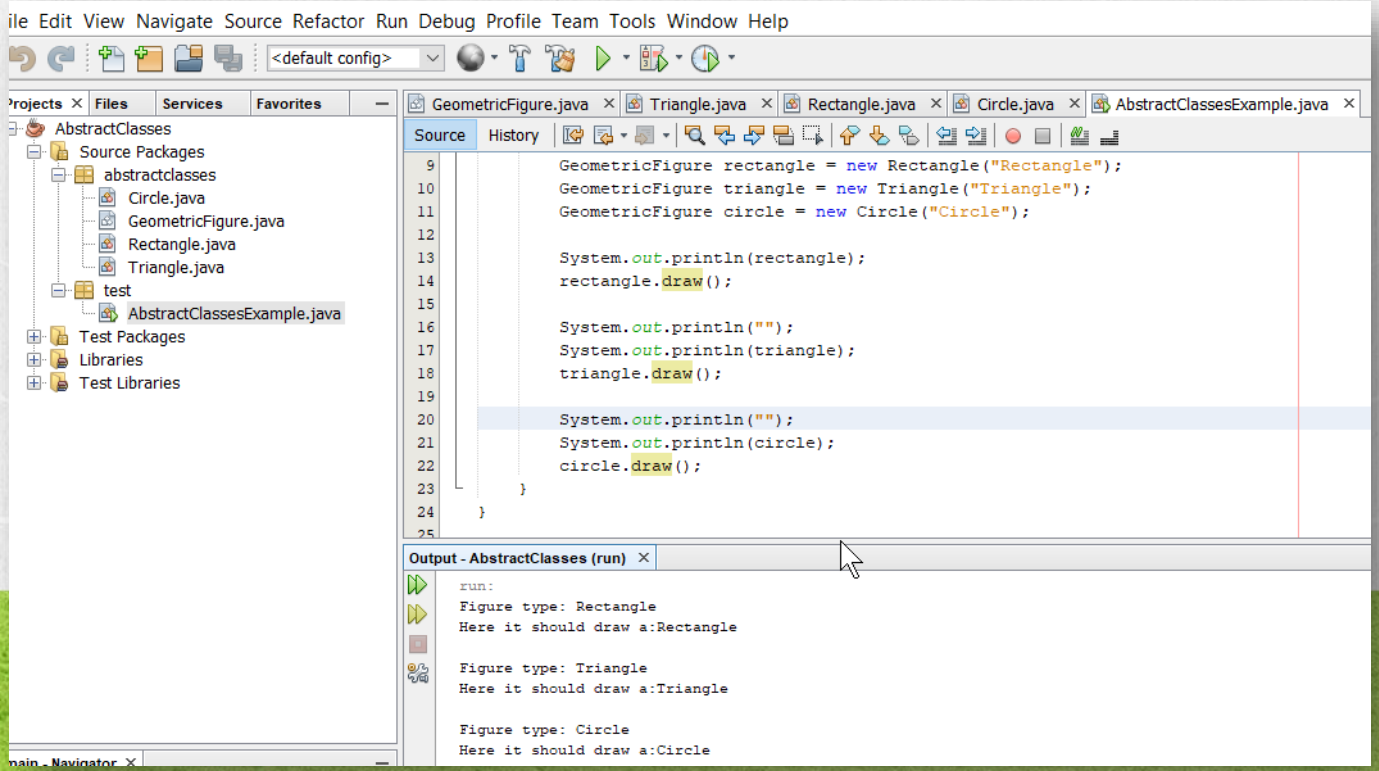
JAVA PROGRAMMING COURSE

EXERCISE

ABSTRACT CLASSES IN JAVA

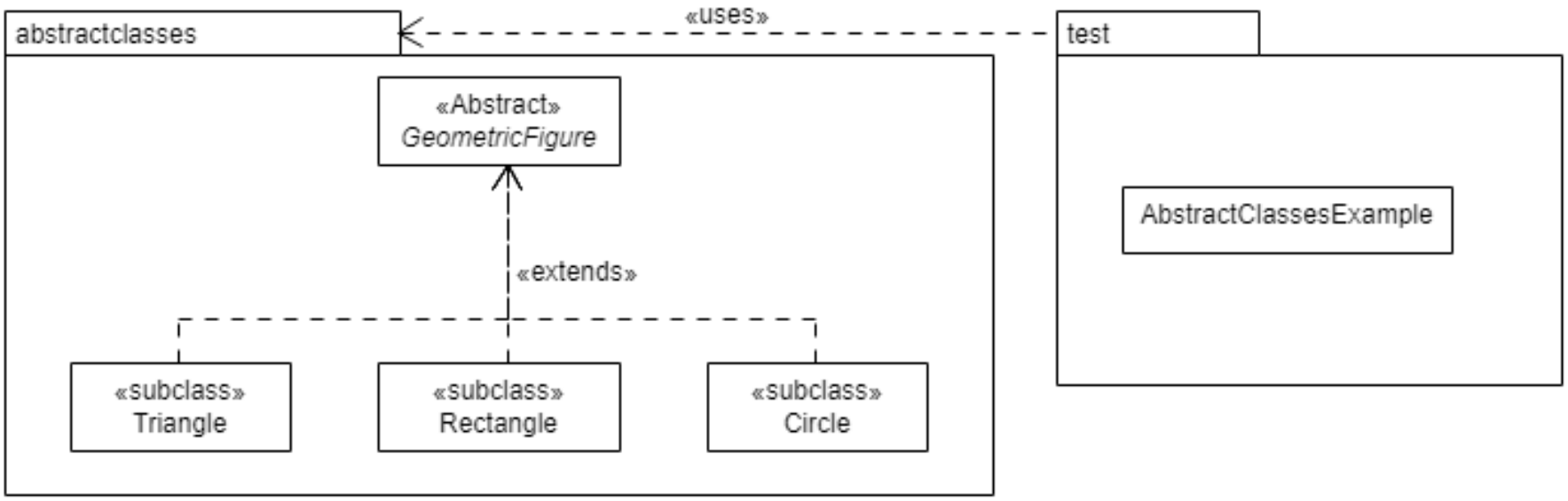
EXERCISE OBJECTIVE

Create a program to practice the concept of Abstract Classes in Java. At the end we should observe the following:



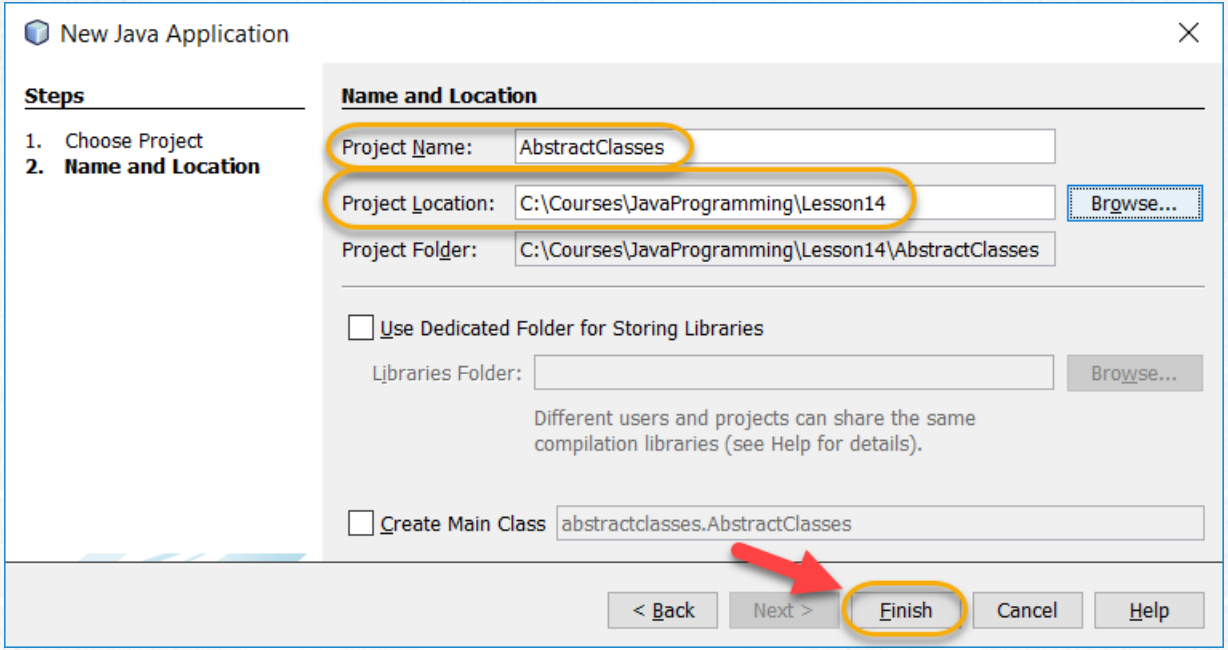
UML DIAGRAM

The project will be based on the following class diagram:



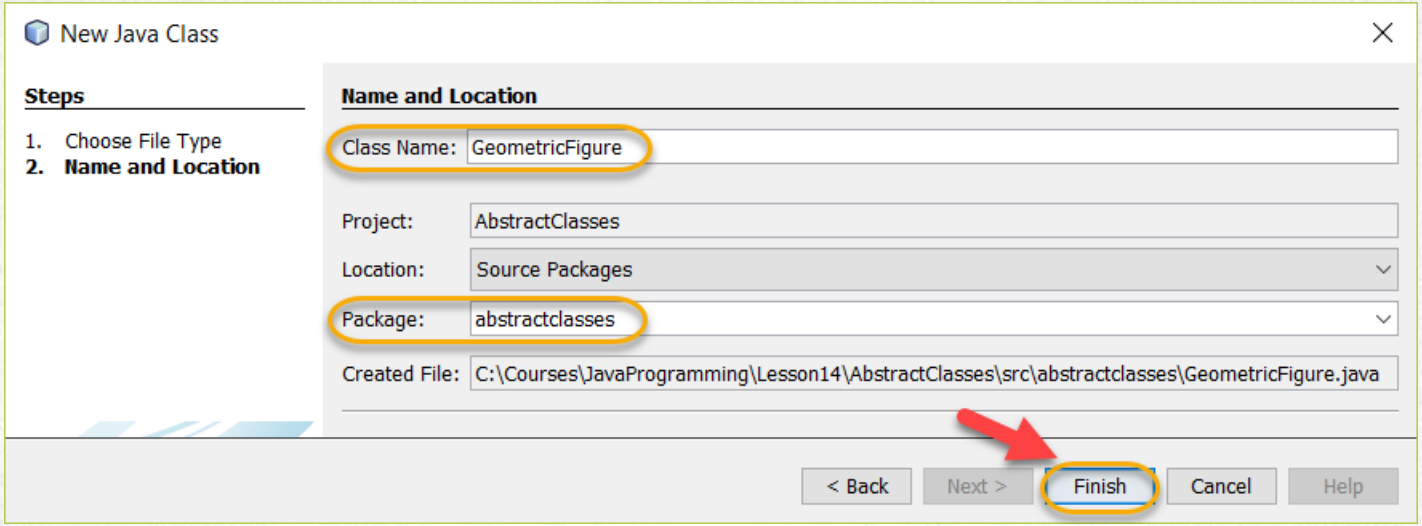
1. CREATE A NEW PROJECT

Create a new Project:



2. CREATE A NEW CLASS

Create a new class:



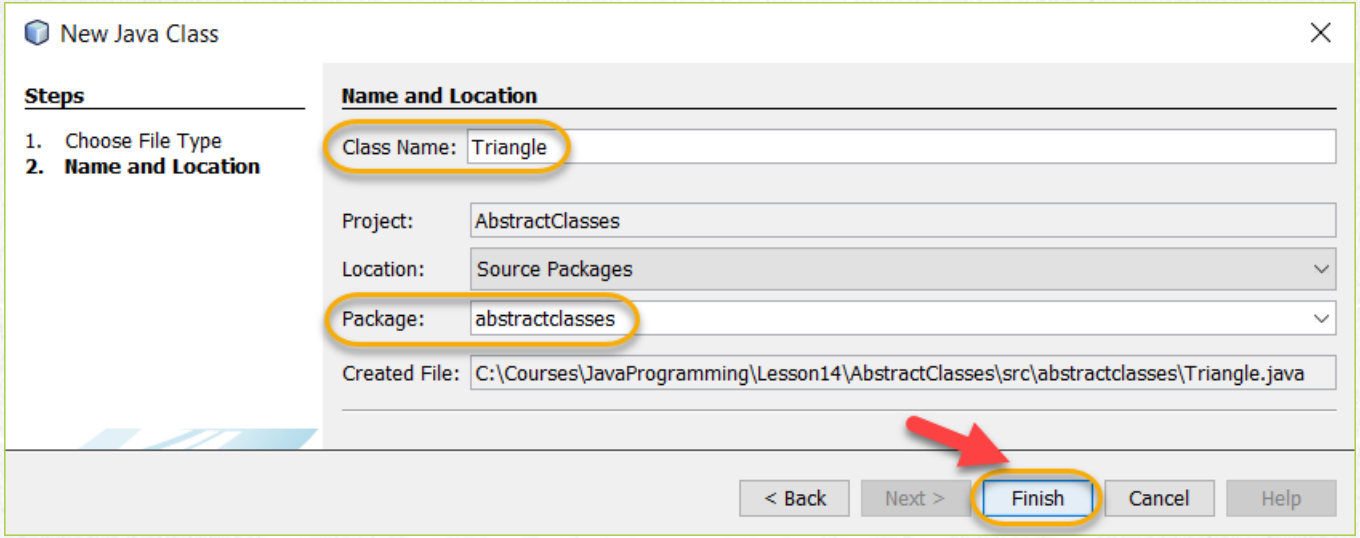
3. MODIFY THE CODE

GeometricFigure.java:

```java
package abstractclasses;

public abstract class GeometricFigure {

    protected String figureType;

    protected GeometricFigure(String figureType){
        this.figureType = figureType;
    }

    //The parent class does not define behavior
    public abstract void draw();

    public String getFigureType() {
        return figureType;
    }

    public void setFigureType(String figureType) {
        this.figureType = figureType;
    }

    @Override
    public String toString() {
        return "Figure type: " + this.figureType;
    }
}
```

4. CREATE A NEW CLASS

Create a new class:



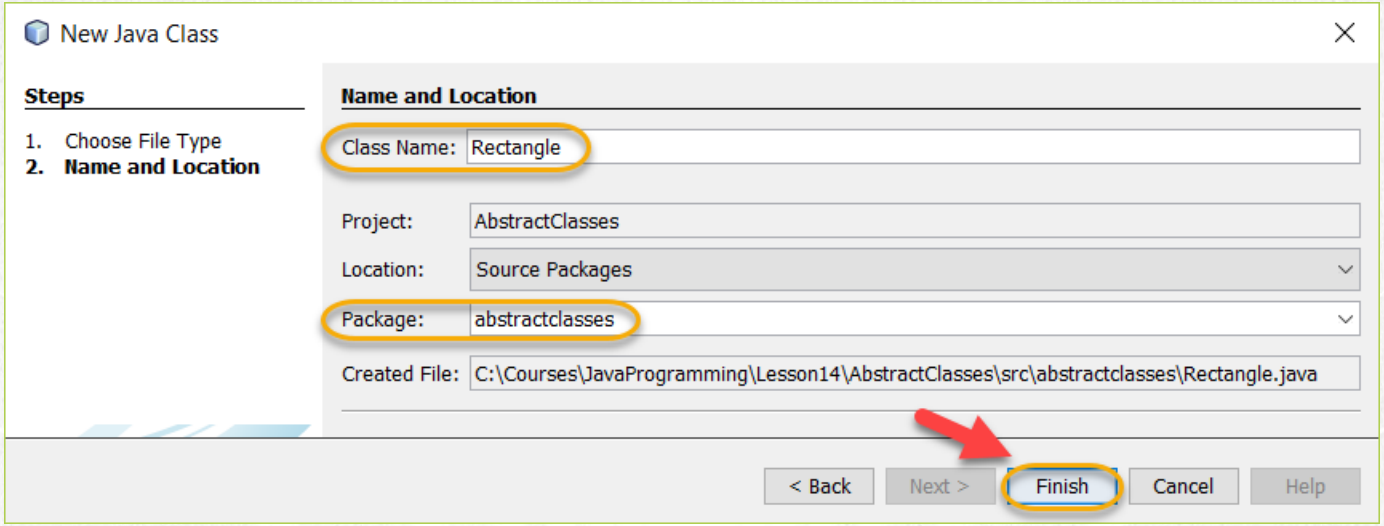
5. MODIFY THE CODE

Triangle.java:

```
package abstractclasses;

public class Triangle extends GeometricFigure{

    public Triangle(String figureType) {
        super(figureType);
    }

    @Override
    public void draw() {
        //Implementation of the inherit drawing method of the GeometricFigure class
        System.out.println("Here it should draw a:" + this.getClass().getSimpleName());
    }
}
```

6. CREATE A NEW CLASS

Create a new class:



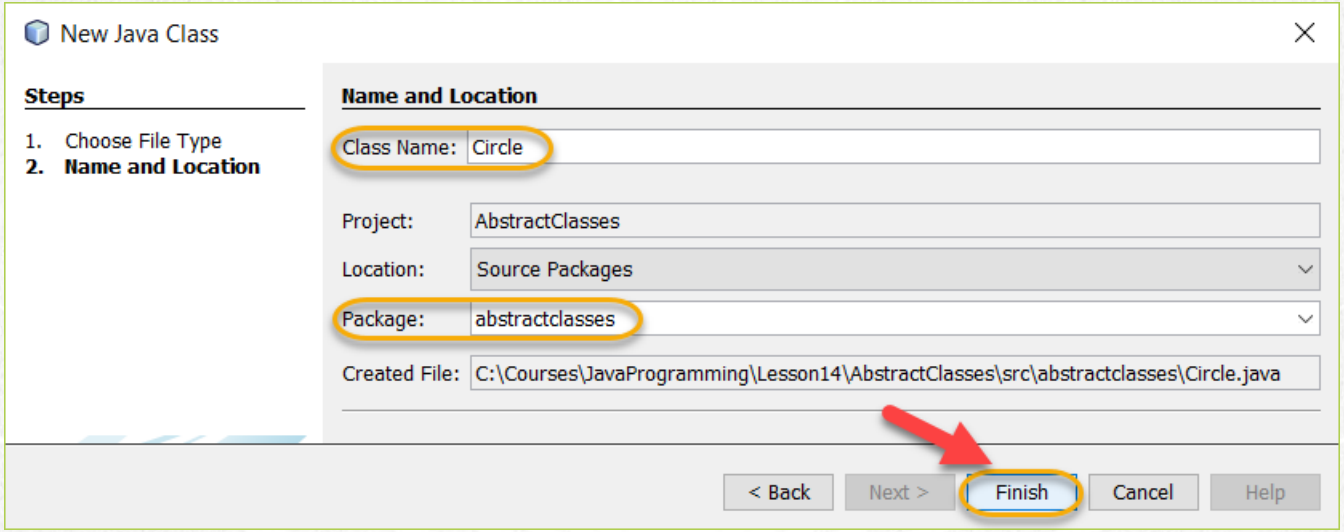
7. MODIFY THE CODE

Rectangle.java:

```
package abstractclasses;

public class Rectangle extends GeometricFigure{

    public Rectangle(String figureType) {
        super(figureType);
    }

    @Override
    public void draw() {
        //Implementation of the inherit drawing method of the GeometricFigure class
        System.out.println("Here it should draw a:" + this.getClass().getSimpleName());
    }
}
```

8. CREATE A NEW CLASS

Create a new class:



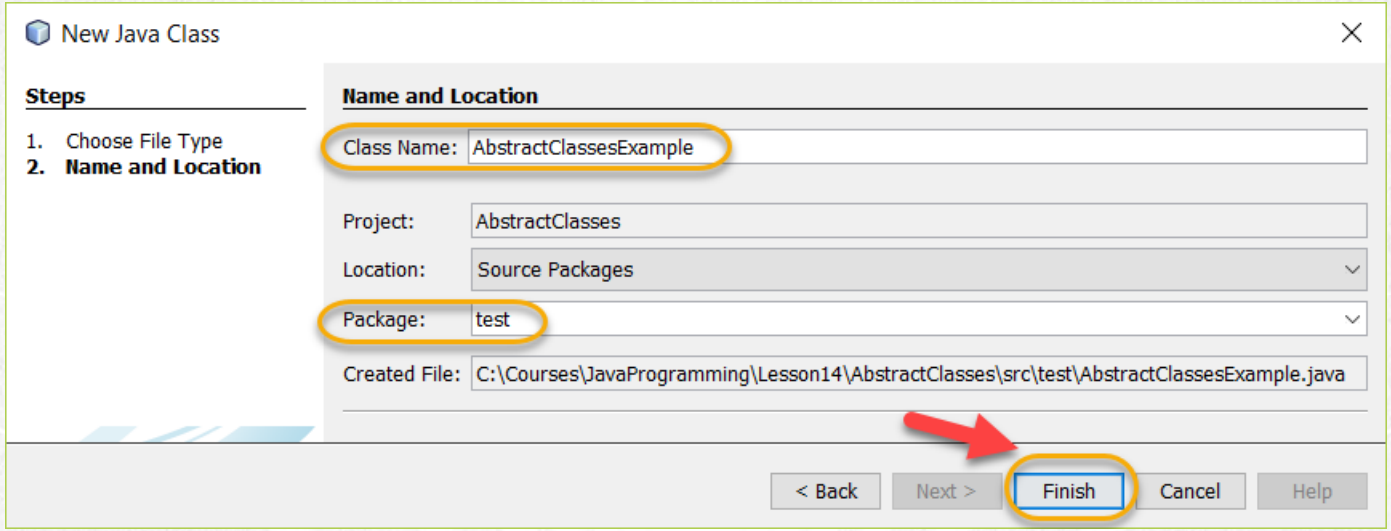
9. MODIFY THE CODE

Circle.java:

```
package abstractclasses;

public class Circle extends GeometricFigure{

    public Circle(String figureType) {
        super(figureType);
    }

    @Override
    public void draw() {
        //Implementation of the inherit drawing method of the GeometricFigure class
        System.out.println("Here it should draw a:" + this.getClass().getSimpleName());
    }
}
```

10. CREATE A NEW CLASS

Create a new class:



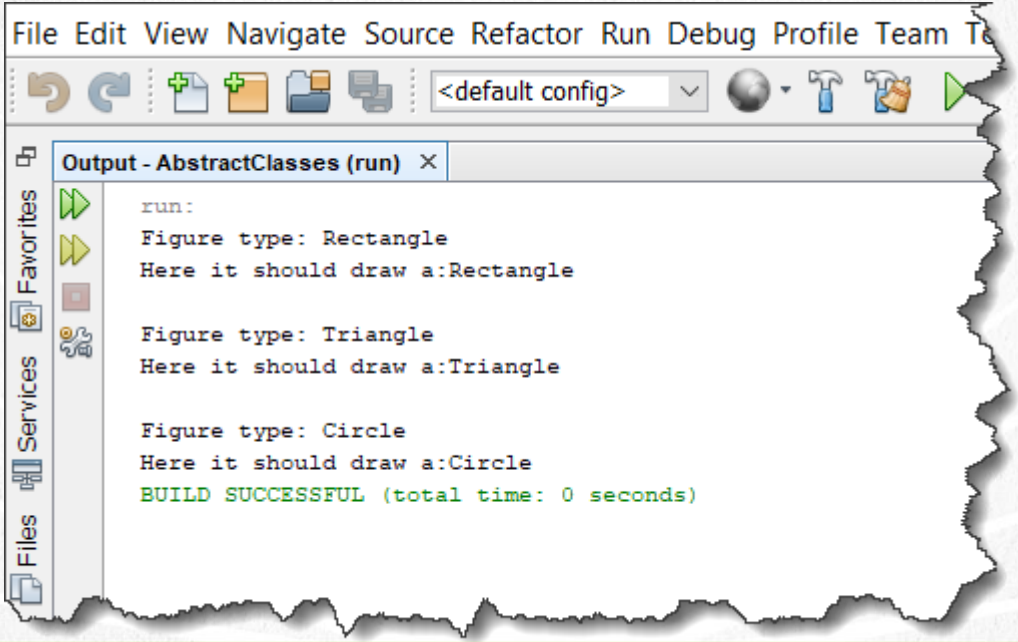
11. MODIFY THE CODE

AbstractClassExample.java:

```
package test;

import abstractclasses.*;

public class AbstractClassesExample {

    public static void main(String args[]) {
        //Creation of objects
        GeometricFigure rectangle = new Rectangle("Rectangle");
        GeometricFigure triangle = new Triangle("Triangle");
        GeometricFigure circle = new Circle("Circle");

        System.out.println(rectangle);
        rectangle.draw();

        System.out.println("");
        System.out.println(triangle);
        triangle.draw();

        System.out.println("");
        System.out.println(circle);
        circle.draw();
    }
}
```

12. EXECUTE THE PROJECT

The result is as follows:



EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of abstract classes in Java.

- We have seen how the same polymorphism rules apply, however, an abstract class can not be instantiated, and therefore a subclass is required to implement the abstract method defined in the parent class.

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta