SOLID-PRINCIPLES ASSIGNMENT

INSTRUCTOR: MUSKAN GUPTA

Submitted By: Gaurav Shukla

Question:

Identify how the following code violates the Liskov Substitution Principle? Also, correct the code in order it to be correct as per the Liskov Substitution Principle.

```
class Rectangle {
  int m_width;
  int m_height;

  public void setWidth(int width){
    m_width = width;
  }
```

```
public void setHeight(int h) {
    m_height = h;
  }

  public int getWidth() {
    return m_width;
  }
public int getHeight() {
    return m_height;
  }

  public int getArea() {
    return m_width * m_height;
  }
}
```

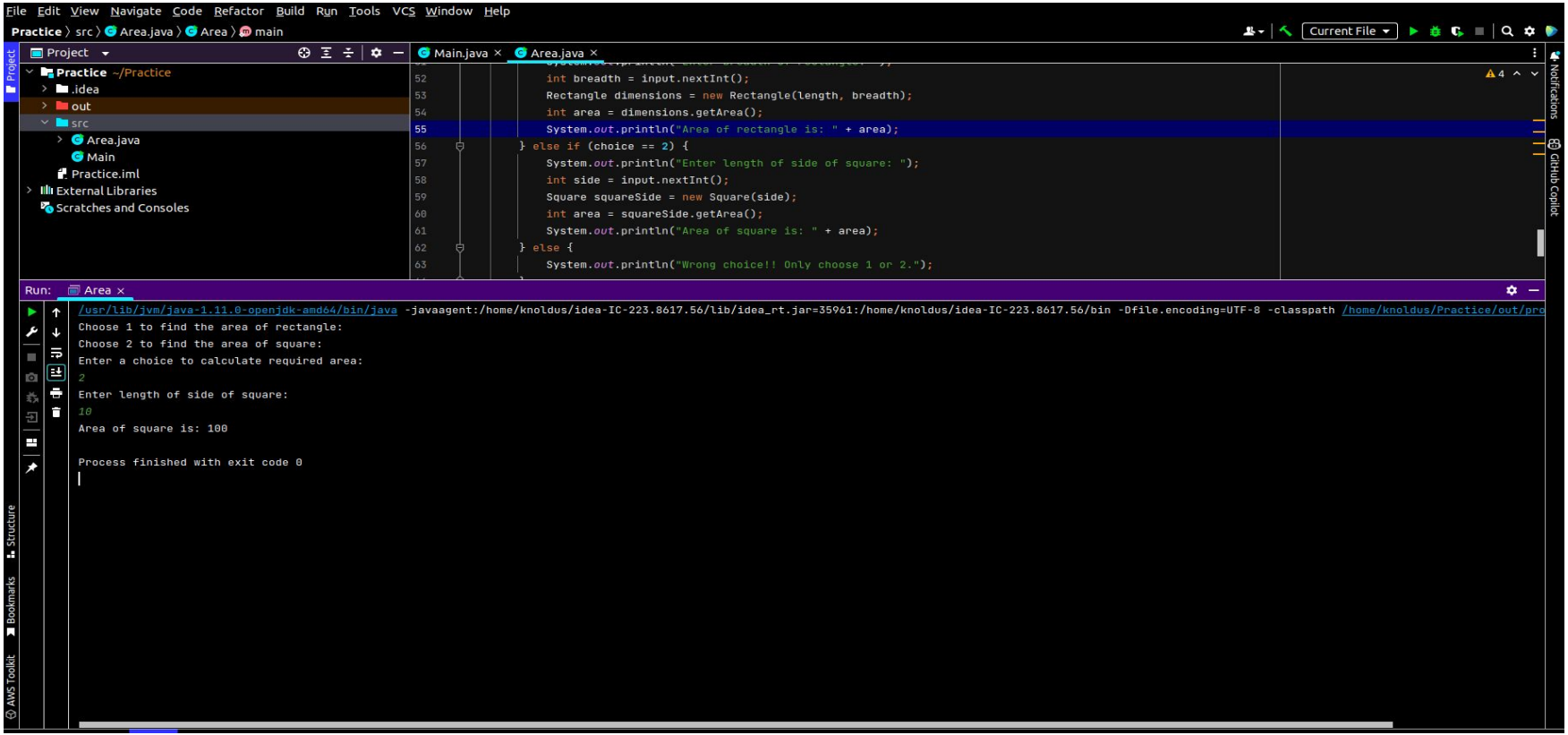
```
class Square extends Rectangle {
  public void setWidth (int width) {
    m_width = width;
    m_height = width;
  }

  public void setHeight (int height) {
    m_width = height;
    m_height = height;
  }
}
```

Problem in the Code->

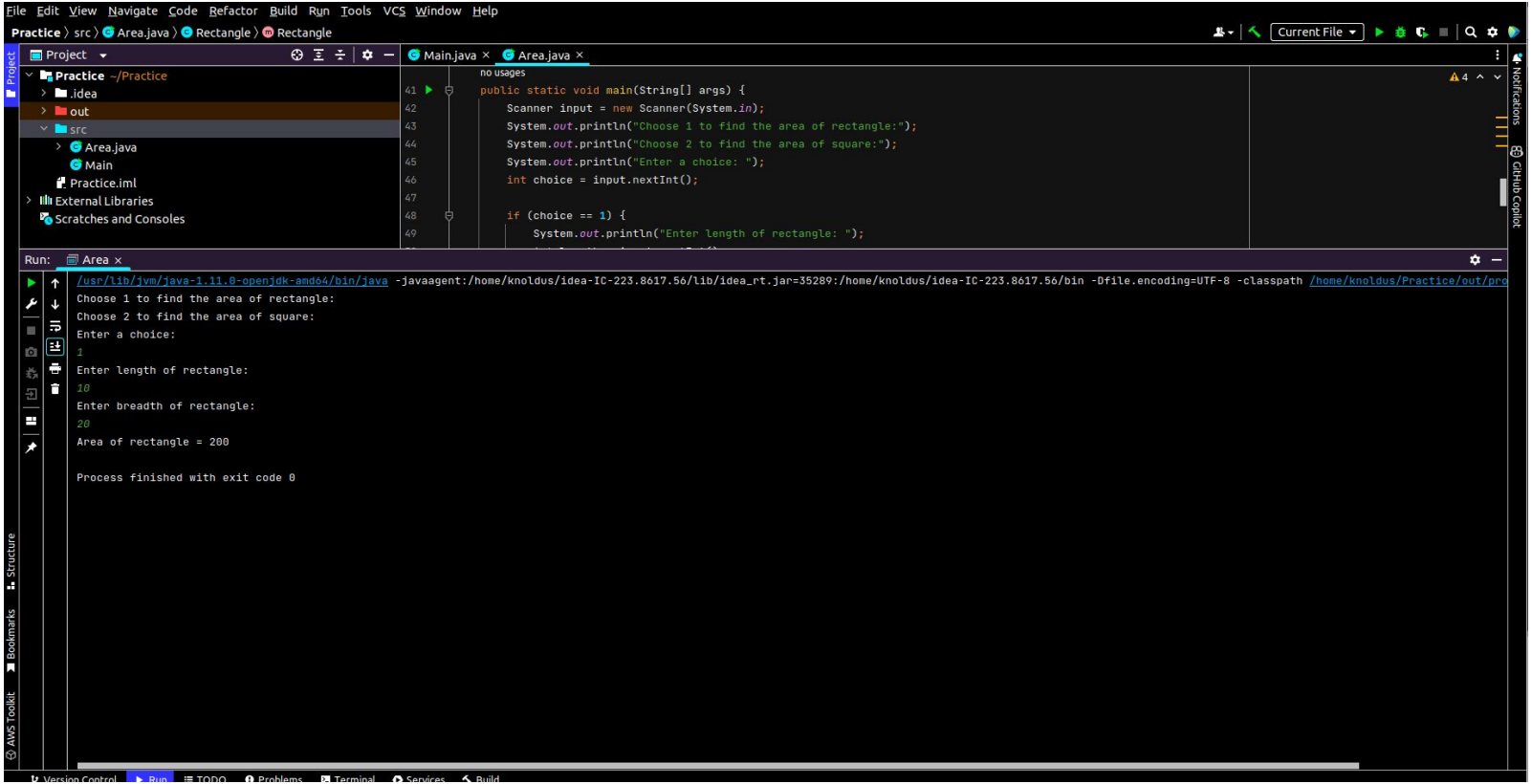
The above code violates the Liskov Substitution Principle because a square is a special case of a rectangle, but the Square class does not behave like a Rectangle in terms of its setters. Specifically, the Square class violates the Liskov Substitution Principle because it alters the behavior of the base class (Rectangle) in a way that is incompatible with the expectations of a Rectangle.

The setWidth and setHeight methods in the Square class override the corresponding methods in the base class, but they change the behavior of a Square object in a way that is incompatible with the behavior of a Rectangle object. Specifically, if we set the width and height of a Rectangle object separately, we can get different values for getWidth() and getHeight(). But if we set the width or height of a Square object, both properties must always have the same value.

Output After Fixing Code: 1



Output After fixing code: 2



THANK YOU!!