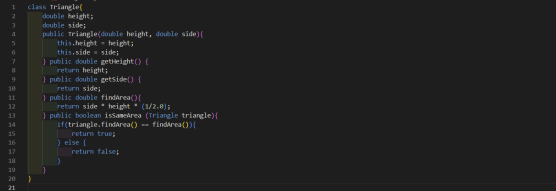
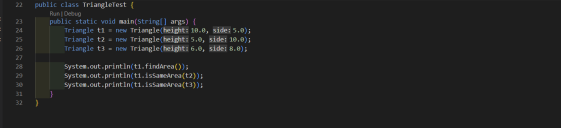
**Assignment 9**

**- 이름 : 정유진**

1. 삼각형을 나타내는 Triangle 클래스를 작성하시오. 삼각형의 속성으로는 실숫값의 밑변과 높이를, 동작으로는 넓이 구하기(findArea), 넓이 비교(isSameArea), 접근자, 생성자를 포함한다. 또한, 작성한 클래스를 다음 코드를 사용하여 테스트하시오.

|  |
| --- |
| public class TriangleTest {  public static void main(String[] args) {  Triangle t1 = new Triangle(10.0, 5.0);  Triangle t2 = new Triangle(5.0, 10.0);  Triangle t3 = new Triangle(6.0, 8.0);    System.out.println(t1.findArea());  System.out.println(t1.isSameArea(t2));  System.out.println(t1.isSameArea(t3));  }  } |

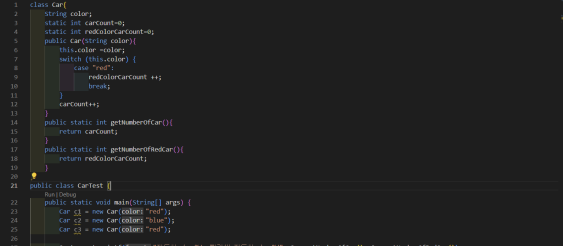






2. 생산된 모든 자동차와 빨간색 자동차의 개수를 출력하는 Car 클래스를 작성하시오. 그리고 다음 코드를 사용하여 테스트하시오.

|  |
| --- |
| public class CarTest {  public static void main(String[] args) {  Car c1 = new Car(“red”);  Car c2 = new Car(“blue”);  Car c3 = new Car(“red”);    System.out.printf(“자동차 수: %d, 빨간색 자동차 수: %d,  Car.getNumberOfCar(), Car.getNumberOfRedCar());  }  } |







3. 다음을 만족하는 클래스 Account를 작성하시오.

· 다음의 2 개의 필드를 선언

private String owner;

private long balance;

· 위 모든 필드에 대한 getter와 setter의 구현

· 위 모든 필드를 사용하는 가능한 모든 생성자의 구현

· 메소드 deposit()의 헤드는 다음과 같으며 인자인 금액을 저축하는 메소드

public long deposit(long amount)

· 메소드 withdraw()의 헤드는 다음과 같으며 인자인 금액을 인출하는 메소드

public long withdraw(long amount)

· Account 클래스의 main() 메소드에서 Account 객체를 생성하여 적당한 저축과 인출을 수행한 후 잔금을 출력

4. 위에서 구현된 메소드 withdraw()를 다음 조건에 맞게 다시 작성하시오.

· 인출 상한 금액은 잔액까지로 하며, 이 경우 이러한 상황을 출력

· 클래스 AccountTest의 main() 메소드에서 인출 상한 이상의 금액을 인출하려는 메소드를 호출하여 출력

