答案：输出两个大整数相加的结果。

```
import java.math.BigInteger;

public class BigIntegerExample {
    public static void main(String[] args) {
        BigInteger num1 = new BigInteger("12345678901234567890");
        BigInteger num2 = new BigInteger("98765432109876543210");
        BigInteger sum = num1.add(num2);
        System.out.println(sum);
    }
}
```

编写一个程序，使用 `BigDecimal` 计算 `1 / 3` 的精确值，并保留 10 位小数输出。

```
import java.math.BigDecimal;
import java.math.RoundingMode;

public class BigDecimalDivision {
    public static void main(String[] args) {
        BigDecimal dividend = new BigDecimal("1");
        BigDecimal divisor = new BigDecimal("3");
        BigDecimal result = dividend.divide(divisor, 10, RoundingMode.HALF_UP);
        System.out.println(result);
    }
}
```

```
interface Animal {
    void makeSound();
}

class Dog implements Animal {
    @Override
    public void makeSound() {
        System.out.println("汪汪汪");
    }
}

class Cat implements Animal {
    @Override
    public void makeSound() {
        System.out.println("喵喵喵");
    }
}

public class InterfaceExample {
    public static void main(String[] args) {
```

```
        Dog dog = new Dog();
        Cat cat = new Cat();
        dog.makeSound();
        cat.makeSound();
    }
}
```

定义一个 `Flyable` 接口，有 `fly` 方法。创建 `Bird` 和 `Airplane` 类实现该接口，在 `main` 方法中分别创建对象并调用 `fly` 方法，输出相应的飞行信息。

```
interface Flyable {
    void fly();
}

class Bird implements Flyable {
    @Override
    public void fly() {
        System.out.println("鸟儿在天空自由飞翔");
    }
}

class Airplane implements Flyable {
    @Override
    public void fly() {
        System.out.println("飞机在高空飞行");
    }
}

public class FlyableExample {
    public static void main(String[] args) {
        Bird bird = new Bird();
        Airplane airplane = new Airplane();
        bird.fly();
        airplane.fly();
    }
}
```

异常

```java
public class ExceptionExample {
    public static void main(String[] args) {
        try {
            int[] arr = {1, 2, 3};
            System.out.println(arr[3]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("数组下标越界异常");
        } finally {
            System.out.println("无论是否发生异常，都会执行这里的代码");
        }
    }
}
```

编写一个方法 `divideNumbers`，接受两个整数作为参数，在方法中进行除法运算。如果除数为 0，抛出 `IllegalArgumentException` 异常，并在 `main` 方法中调用该方法并处理异常。

```java
public class ExceptionHandling {
    public static void divideNumbers(int dividend, int divisor) {
        if (divisor == 0) {
            throw new IllegalArgumentException("除数不能为 0");
        }
        System.out.println(dividend / divisor);
    }

    public static void main(String[] args) {
        try {
            divideNumbers(10, 0);
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        }
    }
}
```