

# 实验5 异常处理

班 级： 软工1805  
学 号： 201806061219  
姓 名： 王程飞  
完成日期： 2019.11.11

1.输入一个加法表达式，计算表达式的结果。表达式形式为12+24+5，如果输入格式错误将产生异常。提示：用String类的split方法分离出加数。

## 代码

```
private static void test01() {
    Scanner scanner = new Scanner(System.in);
    while (scanner.hasNextLine()) {
        String exp = scanner.nextLine();
        String[] items = exp.split("\\+");
        int sum = 0;
        try {
            for (String item : items) {
                int i = Integer.parseInt(item);
                sum += i;
            }
            System.out.println(sum);
        } catch (NumberFormatException e) {
            System.out.println("格式错误");
        }
    }
}
```

## 结果

```
input: 1+2+3+4
10
input: 1-+2
格式错误
```

2.编写一个三角形类Triangle的方法isTriangle()判断3个参数能否构造三角形，如果三个参数能构成三角形则返回true，如果三个参数不满足三角形的条件，则抛出异常IllegalArgumentException，并打印输入的三个参数不能构成三角形，返回false。最后用main方法进行测试。

## 代码

```
private static void test02() {
    class Triangle {
        Triangle(double a, double b, double c) {
            if (check(a, b, c)) {
                System.out.println("可以构成三角形");
            } else {
                throw new IllegalArgumentException("不能构成三角形");
            }
        }
    }
}
```

```

    }
    private boolean check(double a, double b, double c) {
        return !(a + b <= c) && !(a + c <= b) && !(b + c <= a);
    }
}
try {
    new Triangle(10, 2.0, 3.0);
} catch (IllegalArgumentException e) {
    System.out.println("不能构成三角形");
}
}

```

## 结果

可以构成三角形  
不能构成三角形

3.编写一个抽象类Shape，并提供一个抽象的方法getArea()，再派生出三个子类圆Circle、矩形类Rectangle和三角形类Triangle重写getArea()方法。然后编写一个测试程序TestShape，根据用户输入的命令行参数数量来计算圆、矩形和三角形的面积。如果输入的参数为1个、2个、3个分别对应圆、矩形和三角形，如果没有参数或则参数错误，则通过异常处理方法显示错误信息。

## 代码

```

private static void test03(String[] args) {
    try {
        switch (args.length) {
            case 1:
                System.out.println("圆形");
                System.out.println(new Circle(args).getArea());
                break;
            case 2:
                System.out.println("矩形");
                System.out.println(new Rectangle(args).getArea());
                break;
            case 3:
                System.out.println("三角形");
                System.out.println(new Triangle(args).getArea());
                break;
            default:
                throw new IllegalArgumentException();
        }
    } catch (Exception e) {
        System.out.println("参数异常");
    }
}

abstract class Shape {
    abstract double getArea();
}

class Circle extends Shape {
    private double r;
    Circle(String[] args) {
        try {
            r = Double.parseDouble(args[0]);
        } catch (NumberFormatException | ArrayIndexOutOfBoundsException e) {
            System.out.println("args error");
        }
    }
}

```

```

    }
}

@Override
double getArea() {
    return Math.PI*r;
}
}
class Rectangle extends Shape {
    Rectangle(String[] args) {
        try {
            a = Double.parseDouble(args[0]);
            b = Double.parseDouble(args[1]);
        } catch (NumberFormatException | ArrayIndexOutOfBoundsException e) {
            System.out.println("args error");
        }
    }
    private double a,b;
    @Override
    double getArea() {
        return a*b;
    }
}
class Triangle extends Shape {
    private double a, b, c;
    Triangle(String[] args) {
        try {
            a = Double.parseDouble(args[0]);
            b = Double.parseDouble(args[1]);
            c = Double.parseDouble(args[2]);
        } catch (NumberFormatException | ArrayIndexOutOfBoundsException e) {
            System.out.println("args error");
        }
    }
    @Override
    double getArea() throws IllegalArgumentException {
        if (check(a, b, c)) {
            double p = (a+b+c)/2;
            return Math.sqrt(p*(p-a)*(p-b)*(p-c));
        } else {
            throw new IllegalArgumentException();
        }
    }
    private boolean check(double a, double b, double c) {
        return !(a + b <= c) && !(a + c <= b) && !(b + c <= a);
    }
}
}

```

## 结果

```

-args: 1 2.5 3
三角形
1.1709371246996996

```

```

-args: 1 2.5
矩形
2.5

```

```

-args: 1
圆形

```

3.141592653589793

-args: 1 2 3 4

参数异常

-args: 1 2 3

三角形

参数异常