

| 学号             | 姓名  | 提交时间                | 总分  | 成绩 |
|----------------|-----|---------------------|-----|----|
| 20225102010209 | 张博林 | 2024-03-28 19:24:10 | 100 | 95 |

1. 【简答题】

得分：95/100

作业：

- 1、修饰方法和属性的常用修饰符有哪些？作用是啥？
- 2、构造方法的特点？
- 3、要想调用一个类下的方法和属性需要做什么？
- 4、静态修饰的属性和方法的特点
- 5、静态语句块的结构与特点
- 6、封装的特点
- 7、继承的关键字
- 8、局部变量和成员变量的区别
- 9、堆和栈结构特点
- 10、面向对象的特点

编程题：编写一个计算器的类，

计算器类根据用户类输入的 $+ - * /$ 符号和运算数，

返回相应的运算结果

学生答案：8. 局部变量和成员变量的区别在于：

- 局部变量：定义在方法内部，只在方法内部有效，方法执行完毕后销毁。
- 成员变量：定义在类中，整个类中的方法都可以访问，存在于对象的生命周期中。

9. 堆和栈是内存中存储数据的两种区域：

- 堆：存储对象和数组，由 Java 虚拟机自动管理，对象存在于堆中，通过引用访问。
- 栈：存储基本数据类型和对象的引用，方法调用时会在栈中创建方法栈帧，方法执行完毕后栈帧被销毁。

10. 面向对象的特点包括封装、继承、多态和抽象，通过对象的方式来组织和处理数据，提高了代码的可维护性和可扩展性。

1. 修饰方法和属性的常用修饰符包括：

- 方法修饰符：`public`、`protected`、`private`、`static`、`final`、`abstract`、`synchronized`、`native`、`strictfp`等。
- 属性修饰符：`public`、`protected`、`private`、`static`、`final`等。

这些修饰符的作用包括控制访问权限、实现静态方法或属性、限制方法的重写、实现抽象方法、实现线程同步等。

2. 构造方法的特点包括：

- 构造方法与类同名。
- 构造方法没有返回类型。
- 在创建类的实例时被调用，用于初始化对象。
- 可以有多个构造方法，通过参数列表的不同来区分。

3. 要想调用一个类下的方法和属性，需要创建该类的对象实例，然后通过对象实例来访问方法和属性。

4. 静态修饰的属性和方法的特点包括：

- 静态属性属于类，所有对象共享同一份静态属性。
- 静态方法可以直接通过类名调用，不需要实例化对象。
- 静态方法不能访问非静态方法和属性。

5. 静态语句块是用`static`关键字修饰的代码块，在类加载时执行，用于初始化静态变量或执行静态代码。

6. 封装的特点包括隐藏对象的属性，只能通过特定的方法来访问和修改属性，提高了安全性和灵活性。

7. 继承的关键字是`extends`，用于子类继承父类的属性和方法。

```
package java_0328_zuoye;

public class jisuanlei {
    1 usage
    public static double jisuanlei(double shu, char fuhao, double shu1) {
        2 double j = 0;
        3 switch (fuhao) {
        4     case '+':
        5         j = shu + shu1;
        6         break;
        7     case '-':
        8         j = shu - shu1;
        9         break;
        10    case '*':
        11        j = shu * shu1;
        12        break;
        13    case '/':
        14        if (shu1 != 0) {
        15            j = shu / shu1;
        16        } else {
        17            System.out.println("除数不能为0");
        18            return 0;
        19        }
        20        break;
        21    default:
        22        System.out.println("无效的运算符");
        23        return 0;
        24    }
        25    return j;
    }
}

package java_0328_zuoye;

import java.util.Scanner;

public class yonghulei {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("请输入一个数。");
        double shu = scanner.nextDouble();
        System.out.println("请输入一个运算符。");
        char fuhao = scanner.next().charAt(0);
        System.out.println("请输入一个数。");
        double shu1 = scanner.nextDouble();
        double result = jisuanlei.jisuanlei(shu, fuhao, shu1);
        System.out.println("运算结果" + shu + fuhao + shu1 + "=" + result);
    }
}
```

参考答案：参考笔记和课堂练习