

# Java方法作业题

---

## 选择题

---

**1、(单选题)关于方法的定义,以下说法错误的是: B**

- A、方法必须先定义后调用
- B、方法可以嵌套定义
- C、void表示无返回值,可以省略return
- D、方法可以提高代码复用性

**2、(单选题)以下关于形参和实参的描述正确的是: C**

- A、形参是方法调用时传递的参数
- B、实参是方法定义时声明的参数
- C、形参和实参的类型必须一致
- D、方法调用时实参的数量可以与方法定义的形参数量不同

**3、(多选题)关于带返回值的方法,以下说法正确的是:AB**

- A、返回值类型必须与方法定义时声明的类型一致
- B、返回值通常应该用变量接收
- C、void类型的方法也可以有返回值,例如: return 0;
- D、return语句可以无视方法声明上的数据类型,返回任意类型的值

**4、(单选题)分析以下方法参数的定义,哪个是正确的:B**

- A、public static void method(int a; int b) {...}
- B、public static int method(int a, int b) {...}
- C、public static void method(int a, b) {...}
- D、public static method(int a, int b) {...}

## 问答题

---

**5、请简述方法的"两个明确"原则是什么?在定义方法时如何应用这个原则?**

1、明确方法的功能,明确方法的输入和输出。2、方法功能名字清晰无歧义、定义所需参数和返回类型

**6、请解释什么是形参?什么是实参?它们之间有什么区别和联系?**

形参是定义方法时定义的需求参数,用于接收外部传入的数据,仅在方法内部有效;实参时传入方法内的具体数据需是确定的常量或变量。

区别:形参在方法定义和执行期间有效、实参在方法调用的瞬间有效。方法被调用时形参存到栈中,实参独立于形参。

联系:调用方法时形参和实参的数量需一致、类型必须可以兼容

# 编程题

## 7、编写一个综合程序,要求包含以下功能:

1. 定义一个方法,生成指定范围内的随机整数数组(方法接收数组长度、最小值、最大值三个参数)
2. 定义一个方法,计算数组中所有偶数的和(方法接收数组参数,返回计算结果)
3. 定义一个方法,找出数组中的最大值(方法接收数组参数,返回最大值)
4. 在main方法中:
  - 提示用户输入数组长度、最小值和最大值
  - 调用生成随机数组的方法
  - 调用计算偶数和方法并输出结果
  - 调用查找最大值方法并输出结果

要求:每个功能必须用单独的方法实现,方法参数和返回值要合理设计。

```
package zuoye;

import java.util.Random;
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);
        System.out.println("请设置生成数组的长度: ");

        int len = sc.nextInt();
        int[] arr1 = new int[len];

        System.out.println("请指定数组数值范围为: ");
        int be = sc.nextInt();
        int en = sc.nextInt();

        out_Array(be,en,arr1);
        System.out.print("数组内容为: ");
        for (int i = 0; i < arr1.length; i++) {
            System.out.print(arr1[i]+" ");
        }
        System.out.println();

        int sum = sumEvenNumber(arr1);
        System.out.println("偶数和为: "+sum);

        int maxx = findMax(arr1);

        System.out.println("最大值为: "+maxx);
    }

    public static void out_Array(int q,int p,int[] arr){//起始 和数组
        Random random = new Random();
        for (int i = 0; i < arr.length; i++) {
            arr[i]=random.nextInt(q,p+1);
        }
    }
}
```

```
}

public static int sumEvenNumber(int[] arr){
    int sum = 0;
    for (int i = 0; i < arr.length; i++) {
        if(arr[i]%2==0){
            sum+=arr[i];
        }
    }
    return sum;
}

public static int findMax(int[] arr){
    int maxx=arr[0];
    for (int i = 0; i < arr.length; i++) {
        if(arr[i]>maxx){
            maxx=arr[i];
        }
    }
    return maxx;
}

}
```