**日期：2020/4/24**

**方法的使用**

1. **实验目的**
2. 掌握什么是方法，怎么定义和使用方法
3. 掌握如何定义有参数有返回值的方法和怎么调用
4. 掌握判断方法重载的依据
5. **实验内容**

1. 根据提示，在右侧编辑器Begin-End处补充代码，定义一个方法名为hello的方法用来和老师打招呼； 调用此方法输出如下结果：hello teacher!

2. 请根据任务描述 ，在编辑器Begin-End中定义一个名为 calcAvg 的方法，用来计算两门课程成绩的平均值，并返回结果。

3. 根据提示，在右侧编辑器Begin-End处补充代码，编写并调用求平均值的方法，传入两门成绩，求出平均值。

4. 在Begin-End区域创建一个 名为getMax的方法，以两个整数作为参数， 返回两个整数变量较大的值，最后在main方法中调用该方法，传入输入的数据。

5. 根据提示，在右侧编辑器Begin-End处补充代码，程序给出了部分代码，您需要参照实现思路达到任务描述的需求。

6. 在编辑器Begin-End处对print方法进行重载，并通过传入不同的参数调用匹配的重载方法。

1. 选择题
2. 根据任务描述，在右侧编辑器Begin-End处补充代码，定义五个方法，并编写代码实现方法对应的功能，达到输出要求。

测试说明

我会对你编写的代码进行测试： 第一题：求和

测试输入：1,2

预期输出： 3

第二题：求平均值

测试输入：1,2,3 预期输出： 2.0

第三题：输出矩形

测试输入：2,3

预期输出： \*\*\* \*\*\*

第四题：数组升序排序

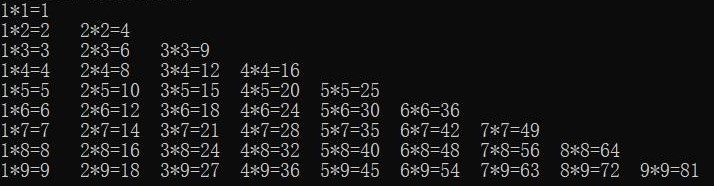
测试输入：5,7,12,9,32

预期输出： 5 7 9 12 32

第五题：打印99乘法表

测试输入：无

预期输出：



1. **实验结果**

1.

public class HelloWorld {

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    //定义一个方法，用来和老师打招呼

    public static void hello(){

        System.out.println("hello teacher!");

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    public static void main(String[] args) {

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        hello();

        //调用方法

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    }

}

2.

public class HelloWorld {

    public static void main(String[] args) {

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        // 调用calcAvg()方法，并将返回值保存在变量avg中

        double avg = calcAvg();

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

        System.out.println("平均成绩为：" + avg);

    }

    // 定义一个返回值为double类型的方法

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    public static double calcAvg() {

        double java = 92.5;

        double php = 83.0;

        double avg = (java + php) / 2; // 计算平均值

        return avg;

        // 使用return返回值

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    }

}

3.

import java.util.Scanner;

public class HelloWorld {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int score1 = sc.nextInt(); //第一门成绩

        int score2 = sc.nextInt(); //第二门成绩

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        // 调用方法，传入两门课程的成绩

        calcAvg(score1,score2);

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    }

    /\*

     \* 功能：计算两门课程考试成绩的平均分并输出平均分

     \* 定义一个包含两个参数的方法，用来传入两门课程的成绩

     \*/

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    public static void calcAvg(int score1,int score2 ){

        int avg = (score1+score2)/2;

        System.out.println("平均分："+avg);

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

4.

import java.util.Scanner;

public class HelloWorld {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int i = sc.nextInt();   //获取第一个输入的整数

        int j = sc.nextInt();   //获取第二个输入的整数

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        //在这里调用方法获取返回值

        System.out.println( i+"和"+j+"比较，最大值是："  +  getMax(i,j));

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    }

    /\*在这里创建getMax方法，以两个整数作为参数，返回两个整数中较大的值\*/

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    public static int getMax(int i,int j){

        if(i>j){

            return i;

        }

        return j;

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

5.

import java.util.Arrays;

public class HelloWorld {

    public static void main(String[] args)

        int[] scores={79,52,98,81};

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        //调用方法，传入成绩数组，并获取成绩的个数

        int count= sort( scores );

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

        System.out.println("共有"+count+"个成绩信息！");

    }

    /\*

     \* 功能：将考试成绩排序并输出，返回成绩的个数

     \*

     \*/

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    public static int sort(   int[] scores   ){

       // 数组从小到大排序处理

         for(int i=0;i< scores.length;i++){

      for(int j=i+1;j<scores.length;j++){

        if(scores[i]>scores[j]){

          int m = scores[i];

          scores[i]=scores[j];

          scores[j]=m;

        }

      }

    }

       // 打印出数组

          System.out.println(Arrays.toString(scores));

       //返回数组中元素的个数

           return scores.length;

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

6.

public class HelloWorld {

    public static void main(String[] args) {

        /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

        // 调用无参的方法

        print();

        // 调用带有一个字符串参数的方法

        print("educoder");

        // 调用带有一个整型参数的方法

        print(666);

        /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

    }

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    // 无参print方法的定义

    public static void print(){

        System.out.println("无参的print方法");

    }

    // 字符串print方法的定义

    public static void print(String name){

        System.out.println("带有一个字符串参数的print方法，参数值为："+name);

    }

    // 整型print方法的定义

    public static void print(int age){

        System.out.println("带有一个整型参数的print方法，参数值为："+age);

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

7.

请选择下列选项中对方法描述正确的所有选项：（ BDE ）

A、方法重载时方法名不能相同

B、静态方法可以直接调用，不需要通过类实例

C、方法名与方法参数相同的两个方法可以同时在一个类中

D、方法重载必须在同一个类中

E、调用有参方法时传入的值可以为object

8.

import java.util.Arrays;

import java.util.Scanner;

public class HelloWorld {

    /\*\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*\*/

    /\*\*

     \* 第一题：定义一个方法 接收两个整数类型的参数 a和b，返回两个数的和  返回值类型为int 方法名为：getSum

     \*/

        public static int getSum(int a,int b){

        int c=a+b;

        return c;

    }

    /\*\*

     \* 第二题： 定义一个方法 接收三个double类型参数a，b，c， 返回这三个数的平均值 返回值类型为double 方法名为：getAvg

     \*/

    public static double getAvg(double a,double b,double c){

    double d = (a+b+c)/3;

    return d;

  }

    /\*\*

     \* 第三题： 定义一个方法 接收两个参数 a 和b 打印a行 b列的一个矩形  不需要返回值   方法名为：printRect

     \*/

    public static void printRect(int a,int b){

    for(int i=0;i<a;i++){

      for(int j=0;j<b;j++){

        System.out.print("\*");

      }

      System.out.println();

    }

  }

    /\*\*

     \* 第四题：定以一个方法，接收整形数组 为参数  对这个数组进行升序排序  最后输出该数组 不需要返回值 方法名为 sortArr

     \*/

    public static void sortArr(int[] arr){

    // Arrays.sort(arr);

   // System.out.println(Arrays.toString(Arrays.copyOf(arr, arr.length)));

    for (int i = 0; i < arr.length; i++) {

      for (int j = 0; j < arr.length-i-1; j++) {

        if(arr[j] > arr[j+1]) {

          int temp = arr[j];

          arr[j] = arr[j+1];

        arr[j+1] = temp;

        }

      }

    }

     for (int m = 0; m < arr.length; m++) {

    System.out.println(arr[m]);

  }

  }

    /\*\*

     \*  第五题

     \* 你只需要实现for 循环括号的内容就可 ，打印语句系统已经给你完成

     \*/

    public static void Print99() {

        for ( int i = 1; i < 10; i++  ) {

            for( int j = 1; j <= i; j++  ) {

                System.out.print(j + " \* " + i + " = " + i \* j + "\t");

            }

            System.out.println();

        }

    }

    /\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

public static void main(String[] args){

  Scanner sc = new Scanner(System.in);

  System.out.println(getSum(sc.nextInt(),sc.nextInt()));

  System.out.println(getAvg(sc.nextDouble(),sc.nextDouble(),sc.nextDouble()));

  printRect(sc.nextInt(),sc.nextInt());

 int[] arr = new int[5];

 for (int i=0;i<arr.length;i++){

    arr[i] = sc.nextInt();

  }

  sortArr(arr);

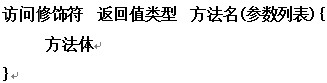
  Print99();

}

}

**四、实验总结**

1. 一般情况下，定义一个方法的语法是：



2. 如果方法的返回类型为 void ，则方法中不能使用 return 返回值。方法的返回值最多只能有一个，不能返回多个值。方法返回值的类型必须兼容，例如，如果返回值类型为 int，则不能返回 String型值。

3. 调用带参方法时，必须保证实参的数量、类型、顺序与形参一一对应。调用方法时，实参不需要指定数据类型。方法的参数可以是基本数据类型，如 int、double 等，也可以是引用数据类型，如 String、数组等。当方法参数有多个时，多个参数间以逗号分隔。

4. 当调用被重载的方法时，Java会根据参数的个数和类型来判断应该调用哪个重载方法，参数完全匹配的方法将被执行