

# 操作题

## 1. 操作题，练习Debug模式！

针对以下代码，说明其中成员变量，静态成员变量的赋值顺序以及结果。**要求留下必要的痕迹文字说明。**

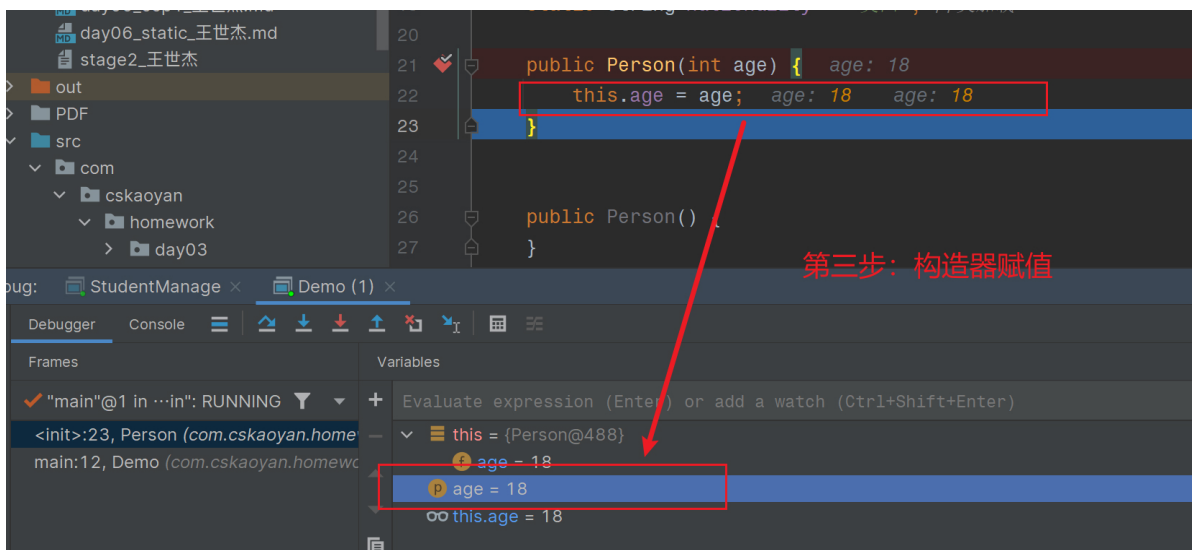
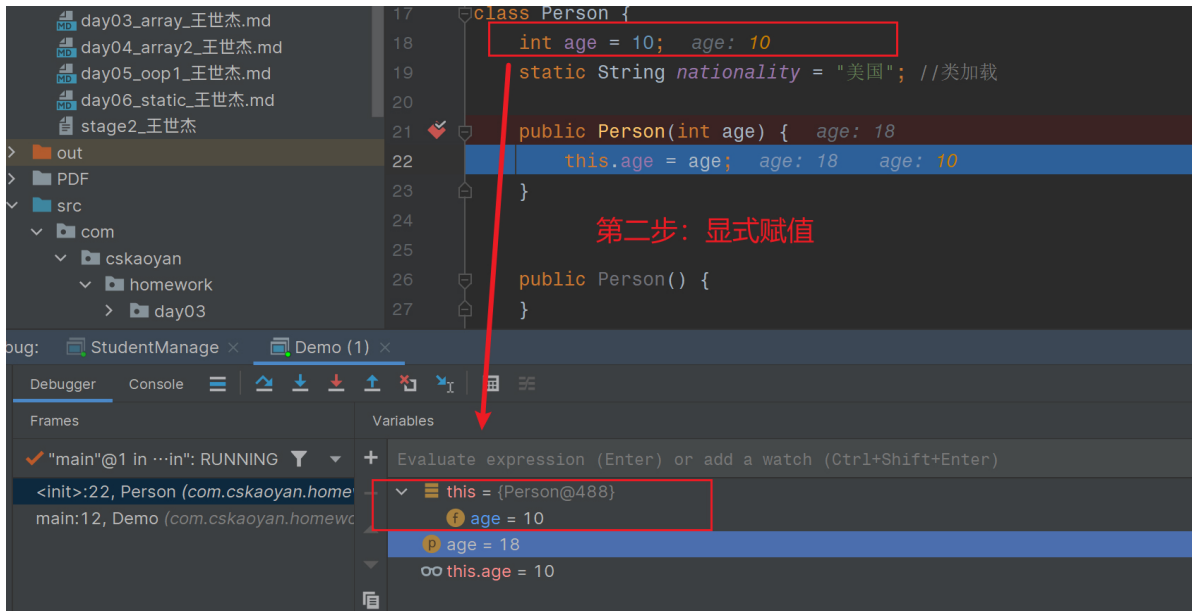
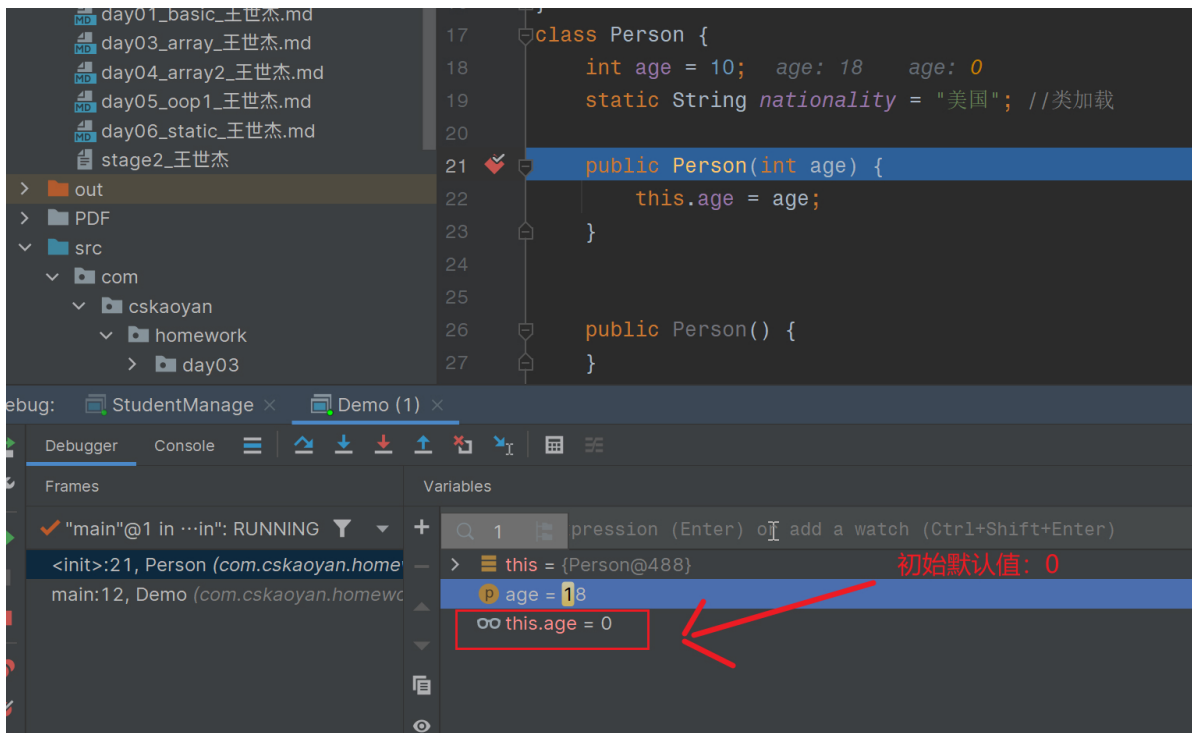
```
public class Demo {
    public static void main(String[] args) {
        // 注意不要同时放开两个new语句，避免互相影响
        // debug语句一，可以进入查看age的赋值过程
        Person p = new Person(18);
        // debug语句二，进入查看nationality赋值过程
        // Person p2 = new Person("中国");
    }
}

class Person {
    int age = 10;
    static String nationality = "美国"; //类加载

    public Person(int age) {
        this.age = age;
    }

    public Person() {
    }

    public Person(String nationality) {
        Person.nationality = nationality; //创建对象
    }
}
```



# 简答题

以下简答题，直接将答案写在题目下面即可。（都是一些概念，虽然我们学得是技术，但基本的概念还是需要记忆的）

1. static修饰的成员属于谁？如何访问？非static修饰的成员属于谁？如何访问？

static修饰的成员属于全体对象，直接调用  
非static修饰的成员属于对象本身，使用对象.成员变量调用

2. 静态成员变量在类全局唯一吗？成员变量在类全局唯一吗？

静态成员变量唯一  
成员变量不唯一

3. 总结目前为止，成员变量和静态成员变量的赋值以及顺序。（分别指的是创建对象时期，和类加载时期）

创建对象会先类加载，类加载会对静态成员变量赋值，类加载后会创建对象然后对对象的成员变量进行赋值

# 编程题

## 对象数组练习

定义一个Student类：

成员变量：String name, int stuld, String gender, int age, double score（表示学生Java考试成绩）

构造器：无参构造器，全参构造器等，自由发挥。

成员方法：print(), 打印对象所有属性的取值。

类定义完毕后，创建五个Student对象存入Student对象数组中，完成以下两个操作：

1. 遍历输出每个学生的属性信息（遍历对象调用print()方法）

- 2. 求成绩平均值。

注：

1. 数组既可以存储基本数据类型（的值），也可以存储引用数据类型（的引用）。它们的使用没有本质区别。
2. 可以考虑提取方法实现功能，不要把所有代码一股脑写在main方法中。

```
package com.cskaoyan.homework.day06;

/**
 * @author 王世杰
 * @created 2022/3/26 12:51
 */

public class Test {

    private static double getAverageScore(Student[] students) {
        double sumScore = 0;
        for (Student student : students) {
            sumScore += student.score;
        }
        return sumScore / students.length;
    }

    public static void main(String[] args) {
        Student s1 = new Student("王世杰", "男", 22, 4.4);
        Student s2 = new Student("屈正", "男", 25, 4.3);
        Student s3 = new Student("新冉冉", "男", 25, 4.2);
        Student s4 = new Student("刘露", "男", 23, 4.1);
        Student s5 = new Student("卢本伟", "男", 30, 4.0);

        Student[] students = {s1, s2, s3, s4, s5};
        for (Student student : students) {
            student.print();
        }
        double averageScore = getAverageScore(students);
        System.out.println("平均绩点为: "+averageScore);
    }
}

class Student {
    String name;
    int stuId;
    String gender;
    int age;
    double score;
    static int currentID = 10001;
}
```

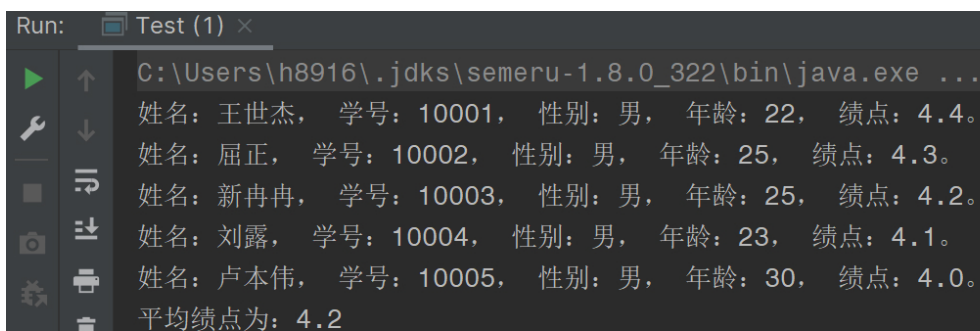
```

public Student() {
    currentID++;
}

public Student(String name, String gender, int age, double score)
{
    this.name = name;
    this.stuId = currentID++;
    this.gender = gender;
    this.age = age;
    this.score = score;
}

void print(){
    System.out.print("姓名: "+name+", ");
    System.out.print("学号: "+stuId+", ");
    System.out.print("性别: "+gender+", ");
    System.out.print("年龄: "+age+", ");
    System.out.println("绩点: "+score+"。");
}
}

```



```

Run: Test (1) x
C:\Users\h8916\.jdk\semeru-1.8.0_322\bin\java.exe ...
姓名: 王世杰, 学号: 10001, 性别: 男, 年龄: 22, 绩点: 4.4。
姓名: 屈正, 学号: 10002, 性别: 男, 年龄: 25, 绩点: 4.3。
姓名: 新冉冉, 学号: 10003, 性别: 男, 年龄: 25, 绩点: 4.2。
姓名: 刘露, 学号: 10004, 性别: 男, 年龄: 23, 绩点: 4.1。
姓名: 卢本伟, 学号: 10005, 性别: 男, 年龄: 30, 绩点: 4.0。
平均绩点为: 4.2

```

## static基础语法练习

当类中的成员变量被static修饰时，那么它就是一个静态成员变量。当类中的成员方法被static修饰时，那么它就是一个静态成员方法。

普通成员（变量和方法）必须创建对象才能够访问，它们是属于对象的。静态成员（变量和方法）属于类，通过"类名点"就可以访问，它们是属于类的。

静态成员变量还有一个非常重要特性：**由于类加载只有一次，所以静态成员变量在类全局都是唯一的！**

现在你需要完成以下基础语法练习：

编写一个类模拟银行账户（Account）的功能，包含的属性有“账户名（ID）”、“密码”、“账户余额”、“储蓄年利率”等。

其中：

1. **账号名**：要求每个账户的账户名都是唯一的，且按照固定格式自动生成：
  - 第一位账户的账户名是"10001"
  - 其后账户的账户名依次加1。比如第二位账户的账户名就是"10002"，第三位就是"10003".....
2. **密码**：新建账户时，可以选择自定义密码，也可以选择使用默认密码："000000"（6个0）
3. **账户余额**：可以选择在新建账户时预设一个值，也可以保持默认值0
4. **储蓄年利率**：由银行设置，和账户本身没有关系。你可以将它设置为"0.3%"（活期年利率一般都非常低，聊胜于无）

定义完属性后，再给出两个成员方法：

1. 打印账户**自身**所有属性。
2. 根据传入的年限和储蓄年利率，计算本息合计后的余额。

**注：计算利息时简单点，不考虑利滚利。即利息 = 利率 \* 时间 \* 本金**

提示：

1. 账户ID是独属于某个账户的，它肯定是一个成员变量。但是它的赋值却是在一个初始值的基础上，创建一个对象就+1，这需要一个不受创建对象影响的变量。
2. 储蓄年利率肯定不是属于账户对象的。

```
package com.cskaoyan.homework.day06;

import java.util.Arrays;
import java.util.Scanner;

/**
 * 编写一个类模拟银行账户（Account）的功能，包含的属性有
 * "账户名（ID）"、“密码”、“账户余额”、“储蓄年利率”等。
 * 其中：
 * 1. **账号名：**要求每个账户的账户名都是唯一的，且按照固定格式自动生成：
 * - 第一位账户的账户名是"10001"
 * - 其后账户的账户名依次加1。比如第二位账户的账户名就是"10002"，第三位就是"10003".....
 * 2. **密码：**新建账户时，可以选择自定义密码，也可以选择使用默认密码："000000"（6个0）
 * 3. **账户余额：**可以选择在新建账户时预设一个值，也可以保持默认值0
 * 4. **储蓄年利率：**由银行设置，和账户本身没有关系。你可以将它设置为**"0.3%"**（活期年利率一般都非常低，聊胜于无）
 * <p>
 * 定义完属性后，再给出两个成员方法：
 * 1. 打印账户自身有属性。
 * 2. 根据传入的年限和储蓄年利率，计算本息合计后的余额。
 * <p>
```

```
* 注：计算利息时简单点，不考虑利滚利。即利息 = 利率 * 时间 * 本金**
* <p>
* 提示：
* 1. 账户ID是独属于某个账户的，它肯定是一个成员变量。但是它的赋值
* 却是在一个初始值的基础上，创建一个对象就+1，这需要一个不受创建对象影响的变
量。
* 2. 储蓄年利率肯定不是属于账户对象的。
*
* @author 王世杰
* @created 2022/3/26 13:08
*/
```

```
public class StaticTest {

    public static void main(String[] args) {
        Account a1 = new Account();
        a1.save(1000);
        a1.setPassword("123456".toCharArray());
        a1.print();
        System.out.print("输入定期的年限: ");
        Scanner scanner = new Scanner(System.in);
        int year = scanner.nextInt();
        a1.setAccountBalance(year);
        a1.print();
    }
}

class Account {
    int accountID;
    char[] password = {'0', '0', '0', '0', '0', '0'};
    double accountBalance = 0;

    static double interestRate = 0.003;
    static int currentID = 10001;

    public Account() {
        this.accountID = currentID++;
    }

    public Account(double accountBalance) {
        this.accountID = currentID++;
        this.accountBalance = accountBalance;
    }

    public Account(char[] password) {
        this.accountID = currentID++;
        this.password = password;
    }
}
```

```

public Account( char[] password, double accountBalance) {
    this.accountID = currentID++;
    this.password = password;
    this.accountBalance = accountBalance;
}

public void save(double money) {
    this.accountBalance += money;
}

public void get(int money) {
    if ((double) money > this.accountBalance) {
        System.out.println("余额不足");
        return;
    } else
        this.accountBalance -= money;
}

public void setPassword(char[] password) {
    this.password = password;
}

void print(){
    System.out.println("账户ID: "+accountID+",    密码: "+
String.valueOf(password) +",    账户余额: "+accountBalance);
}

public void setAccountBalance(int years) {
    this.accountBalance *= (1+years*interestRate);
}
}

```



```

StaticTest x
C:\Users\h8916\.jdk\semeru-1.8.0_322\bin\java.exe ...
账户ID: 10001,    密码: 123456,    账户余额: 1000.0
输入定期的年限: 3
账户ID: 10001,    密码: 123456,    账户余额: 1008.9999999999999
Process finished with exit code 0

```

## 猜数小游戏

百度和各种搜索引擎就是生产力，尝试自己解决以下问题：



- 先生成一个随机数（1~100之间的整数），再键盘输入猜测的数
- 如果猜的数大了或者小了，给出提示，继续猜，直到猜中为止。

注：Java如何生成随机数，自己百度查一下非常简单。这点学习能力还是需要具备的。

```
package com.cskaoyan.homework.day06;

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

/**
 * ## 猜数小游戏
 * <p>
 * 百度和各种搜索引擎就是生产力，尝试自己解决以下问题：
 * <p>
 * 先生成一个随机数（1~100之间的整数），再键盘输入猜测的数
 * 如果猜的数大了或者小了，给出提示，继续猜，直到猜中为止。
 * <p>
 * 注：Java如何生成随机数，自己百度查一下非常简单。这点学习能力还是需要具备的。
 *
 * @author 王世杰
 * @created 2022/3/26 13:40
 */

public class GuessNumber {

    static int randomNumber = new Random().nextInt(100);

    public GuessNumber() {
    }

    public static boolean compare(int input) {
        if (input == randomNumber) {
            System.out.println("=====恭喜你猜对了! =====");
            return false;
        } else if (input > randomNumber) {
            System.out.println("您输入的值大了! 请重新输入: ");
            return true;
        } else {
            System.out.println("您输入的值小了! 请重新输入: ");
            return true;
        }
    }

    public static void game(){
        boolean flag = true;
        Scanner scanner = new Scanner(System.in);
```

```

    int input;
    System.out.println("请输入一个你猜的数字:(0~100)");
    while (flag) {
        input = scanner.nextInt();
        flag = compare(input);
    }
}

public static void main(String[] args) {
    game();
}
}

```

```

Run: GuessNumber x
C:\Users\h8916\.jdk\semeru-1.8.0_322\bin\java.exe ...
请输入一个你猜的数字:(0~100)
50
您输入的值大了! 请重新输入:
25
您输入的值小了! 请重新输入:
37
您输入的值大了! 请重新输入:
31
您输入的值大了! 请重新输入:
28
=====恭喜你猜对了! =====
Process finished with exit code 0

```

## 自己编写数组工具类

- 工具类：指的是类中方法全部是静态方法的类，工具类在使用时无需创建对象（静态方法无需对象调用）
- 以下方法的实现，都是我们上课写过的代码，如果觉得自己很熟练了，直接抄代码即可
- 如果觉得不熟悉或者想再练习一遍，再写一遍也可以~

写一个数组的工具类**ArrayTool**，要求提供如下方法：

遍历，求最大值，最小值，逆置数组元素

查询（在数组中查找指定元素，若不存在，待查找元素返回-1，若存在返回元素在数组中首次出现的位置）

查询（在数组中查找指定元素，若不存在，待查找元素返回-1，若存在返回元素在数组中最后一次出现的位置）

```
package com.cskaoyan.homework.day06;
```

```

import java.util.Arrays;
import java.util.Scanner;

/**
 * 自己编写数组工具类
 * 工具类：指的是类中方法全部是静态方法的类，工具类在使用时无需创建对象（静态方法无需对象调用）
 * 以下方法的实现，都是我们上课写过的代码，如果觉得自己很熟练了，直接抄代码即可
 * 如果觉得不熟悉或者想再练习一遍，再写一遍也可以~
 * <p>
 * 写一个数组的工具类ArrayTool，要求提供如下方法：
 * 遍历，求最大值，最小值，逆置数组元素
 * 查询（在数组中查找指定元素，若不存在，待查找元素返回-1，若存在返回元素在数组中首次出现的位置）
 * 查询（在数组中查找指定元素，若不存在，待查找元素返回-1，若存在返回元素在数组中最后一次出现的位置）
 *
 * @author 王世杰
 * @created 2022/3/26 13:57
 */

public class ToolTest {

    public static void main(String[] args) {
        double[] numbers = getNumbers();
        double[] reverseNumbers =
Arrays.copyOf(numbers, numbers.length);
        ArrayTool.printNum(numbers);
        System.out.println("数组的最大值
为: "+ArrayTool.getMax(numbers));
        System.out.println("数组的最小值
为: "+ArrayTool.getMin(numbers));
        System.out.println("翻转数组为: ");
        ArrayTool.reverse(reverseNumbers);
        ArrayTool.printNum(reverseNumbers);
        Scanner scanner = new Scanner(System.in);
        System.out.print("请输入要查询的元素: ");
        double searchNumber = scanner.nextDouble();
        System.out.println(searchNumber+"是数组
第"+ArrayTool.searchFirstSite(numbers, searchNumber)+"个元素");
        System.out.println(searchNumber+"是数组
第"+ArrayTool.searchLastSite(numbers, searchNumber)+"个元素");
    }

    private static double[] getNumbers() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("请输入数组长度: ");
        int len = scanner.nextInt();
        double[] numbers = new double[len];
    }

```

```

        System.out.print("请连续输入" + len + "个数: ");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = scanner.nextDouble();
        }
        return numbers;
    }
}

class ArrayTool {

    public static void printNum(double[] numbers) {
        System.out.println(Arrays.toString(numbers));
    }

    public static double getMax(double[] numbers) {
        double max = numbers[0];
        for (double number : numbers) {
            if (number > max)
                max = number;
        }
        return max;
    }

    public static double getMin(double[] numbers) {
        double min = numbers[0];
        for (double number : numbers) {
            if (number < min)
                min = number;
        }
        return min;
    }

    public static void reverse(double[] numbers) {
        double temp;
        for (int i = 0; i < numbers.length / 2; i++) {
            temp = numbers[i];
            numbers[i] = numbers[numbers.length - i - 1];
            numbers[numbers.length - i - 1] = temp;
        }
    }

    public static int searchFirstSite(double[] numbers, double
searchNumber) {
        for (int i = 0; i < numbers.length; i++) {
            if (searchNumber == numbers[i]) {
                return i + 1;
            }
        }
        return -1;
    }
}

```

```
}

    public static int searchLastSite(double[] numbers, double
searchNumber) {
        int lastSite = -1;
        for (int i = 0; i < numbers.length; i++) {
            if (searchNumber == numbers[i]) {
                lastSite = i + 1;
            }
        }
        return lastSite;
    }
}
```

Run: ToolTest x

C:\Users\h8916\.jdk\semeru-1.8.0\_322\bin\java.exe ...

请输入数组长度: 10

请连续输入10个数: 1 2 3 4 5 5 5 8 9 10

[1.0, 2.0, 3.0, 4.0, 5.0, 5.0, 5.0, 8.0, 9.0, 10.0]

数组的最大值为: 10.0

数组的最小值为: 1.0

翻转数组为: |

[10.0, 9.0, 8.0, 5.0, 5.0, 5.0, 4.0, 3.0, 2.0, 1.0]

请输入要查询的元素: 5

5.0是数组第5个元素

5.0是数组第7个元素

Process finished with exit code 0