

# 第五章【面向对象基础】- 作业

---

## 1.谈谈你对面向对象和面向过程的理解

## 2.设计一款五子棋游戏

请分别用面向过程和面向对象思想实现，请大致描述你的思路。

### 五子棋游戏过程描述：

游戏启动，用户点击棋盘指定位置（第几行、第几列），该位置会被放置一颗棋子（黑子先，白子后，轮流落），然后系统自动判断输赢，如果判定某方赢，则游戏结束，如果没有赢，则继续游戏。

## 3.谈谈你对类、对象、引用的理解

## 4.根据下面案例，绘制对象内存图

```
1  class Dog {  
2      String name;  
3  
4      public void run() {  
5          System.out.println(name + " 在随风奔跑");  
6      }  
7  }  
8  
9  public static void main(String[] args) {  
10      Dog d1 = new Dog("小黑");
```

```
11      Dog d2 = new Dog("虎子");
12
13      d1.run();
14      d2.run();
15  }
```

## 5.面向对象基础编程

### 需求描述：

定义一个汽车类 Car

- 属性有：颜色、品牌、价格
- 行为有：前进、后退、转弯和输出对象属性

测试类实现功能：

- 实例化一个Car类对象，并给元素赋值
- 调用这个类中方法，输出对象属性信息
- 调用对象前进、后退、转弯等方法

### 命名规则注意事项：

属性、方法：若只有一个单词，全小写

若多个单词构成，第一个单词全小写，后面单词首字母大写

### 代码提示：

```
1  class Car {
```

```
2      //属性定义：颜色  品牌  价格
3
4      //方法定义：前进、后退
5
6  }
7
8  public class Test05_Car {
9      public static void main(String[] args) {
10         //1.实例化Car对象
11
12         //2.给对象属性赋值
13
14         //3.输出对象属性信息
15
16         //4.对象调用方法，让车跑起来
17
18     }
19 }
```

答案：

```
1
```

## 6.谈谈你对封装的理解

## 7.列举构造方法注意事项

## 8.构造方法编程

定义一个人类Person

描述：属性： 姓名（name）、年龄（age）；

要求：

1. 定义无参构造器
2. 定义两参构造器，参数为name和age
3. 定义一参构造器，参数为name，要求该构造器借助借助两参构造器实现功能
4. 定义show方法，输出类对象基本信息

提示：补上代码中缺的部分

```
1  class Person {
2      private String name;
3      private int age;
4
5      //补全代码
6  }
7
8  //测试类如下：
9  public class Test08_Person {
10     public static void main(String[] args) {
11         Person p1 = new Person("jack",21);
12         p1.show();
13
14         System.out.println("-----");
15
16         Person p2 = new Person("lucy");
17         p2.show();
18     }
```

```
19 }
```

答案:

```
1
```

## 9.简述this关键字的用法

## 10.程序阅读题

下面代码无法通过编译，要求：

- 修改代码使得程序能够正常运行
- 分析其输出结果

提示：该题主要用于考核this关键字的作用

```
1  package com.briup.chap05;
2
3  public class Test10_This {
4      public static void main(String[] args) {
5          Teacher t = new Teacher();
6          t.show();
7
8          System.out.println("-----");
9
10         t.disp();
11     }
12 }
13
14 class Teacher {
```

```

15      //显式初始化
16      private String name;
17      private int age;
18      private String gender = "男";
19
20      private void show() {
21          String name = "tom";
22          int age = 20;
23
24          System.out.println("name: " + name);
25          System.out.println("age: " + age);
26          System.out.println("gender: " + gender);
27      }
28
29      public void disp() {
30          String name = "jack";
31          int age = 18;
32          System.out.println("this.age: " + this.age);
33          System.out.println("this.name: " + this.name);
34          System.out.println("this.gender: " + this.gender);
35      }
36  }

```

答案:

1

## 11.根据要求编写时间类Duration

时间类 `Duration` 具体要求如下:

1. 包含属性hour、min、sec, 都是int类型
2. 提供无参构造器
3. 提供构造器 (三参) , 对hour、min、sec进行初始化

4. 提供构造器（一参），参数含义：总秒数int seconds，对hour、min、sec进行初始化，例如3661秒，转为a小时b分钟c秒，就是1小时1分钟1秒
5. 提供每个属性的get方法
6. 提供getTotalSeconds()方法，用于返回hour小时min分钟sec秒钟对应的总秒数seconds
7. 提供disp()方法，输出对象的属性信息和总秒数

测试类：

```
1  package com.briup.chap05;
2
3  public class Test11_Duration {
4      public static void main(String[] args) {
5          //借助三参构造器 实例化对象
6          Duration d1 = new Duration(1,1,1);
7          //获取总秒数
8          int totalSeconds = d1.getTotalSeconds();
9          System.out.println("1小时1分钟1秒: " + totalSeconds);
10
11         System.out.println("-----");
12
13         //借助单参构造器 实例化对象
14         Duration d2 = new Duration(3660);
15         //获取小时
16         int hour = d2.getHour();
17         System.out.println("hour = " + hour);
18
19         System.out.println("-----");
20
21         //输出对象属性信息及总描述信息
22         d2.disp();
23     }
24 }
```

答案：

```

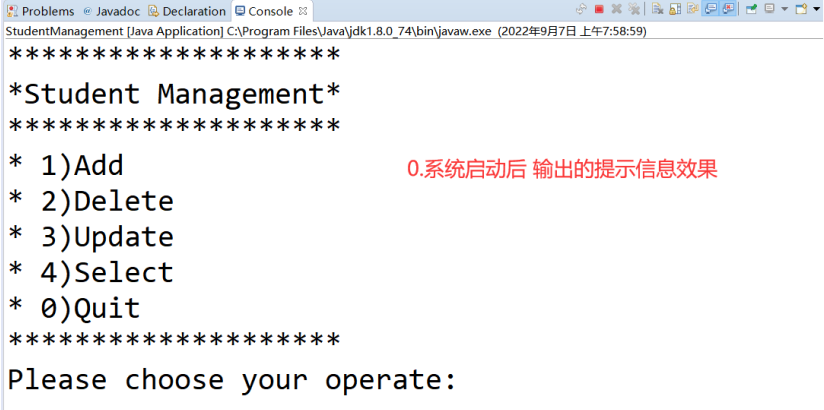
1 //基础类定义
2 class Duration {
3     private int hour;
4     private int min;
5     private int sec;
6
7     //补全下面代码
8 }
9

```

## 12.扩展题-学生管理系统

现有一个简单的学生管理系统，系统启动后运行效果如下图：

系统启动后提示界面如下图：



```

Problems Javadoc Declaration Console
StudentManagement [Java Application] C:\Program Files\Java\jdk1.8.0_74\bin\javaw.exe (2022年9月7日 上午7:58:59)
*****
*Student Management*
*****
* 1)Add
* 2)Delete
* 3)Update
* 4)Select
* 0)Quit
*****
Please choose your operate:

```

添加学生操作步骤如下：



```

* 1)Add
* 2)Delete
* 3)Update
* 4)Select
* 0)Quit
*****
Please choose your operate:1
Please input name:zs
Please input age:20
Please input gender:m
Add student success!
.....

```

1.添加学生操作

查看学生操作步骤如下:

```

* 3)Update
* 4)Select
* 0)Quit
*****
Please choose your operate:4
Please input name:zs
name:zs,age:20,gender:m

```

4.查看学生信息操作

更新学生操作步骤如下:

```

* 3)Update
* 4)Select
* 0)Quit
*****
Please choose your operate:3
Please input name:zs
Please input new name:ww
Please input new age:23
Please input new gender:m
Update student success!
*****

```

3.更新学生信息

删除学生操作步骤如下:

```

*****
* 1)Add
* 2)Delete
* 3)Update
* 4)Select
* 0)Quit
*****
Please choose your operate:2
Please input name:ww
Delete student success!

```

其中学生类Student，学生管理类 StudentManagement代码已经给出，但代码不完整！

请补全代码，使得程序能够实现完整上述完整功能：

```

1  package com.briup.chap05;
2
3  import java.util.Scanner;
4
5  //学生管理系统类
6  public class StudentManagement{
7      //学生对象数组
8      private Student[] stus;
9      //实际学生个数
10     private int counter;
11     //键盘录入对象
12     private Scanner in;
13
14     //无参构造器：默认容量100
15     public StudentManagement(){
16         stus = new Student[100];
17         in = new Scanner(System.in);
18     }
19     //有参构造器：size表示初始容量
20     public StudentManagement(int size){
21         stus = new Student[size];

```

```

22         in = new Scanner(System.in);
23     }
24
25     //输出 操作提示
26     public void prompt(){
27         System.out.println("*****");
28         System.out.println("*Student Management*");
29         System.out.println("*****");
30         System.out.println("* 1)Add");
31         System.out.println("* 2)Delete");
32         System.out.println("* 3)Update");
33         System.out.println("* 4)Select");
34         System.out.println("* 0)Quit");
35         System.out.println("*****");
36         System.out.print("Please choose your operate:");
37     }
38
39     //根据用户录入的学生名称，找到学生在数组中的索引返回
40     private int findStudent(){
41         int index = -1;
42         String name;
43         System.out.print("Please input name:");
44         //从键盘获取一个字符串
45         name = in.next();
46
47         //请补全下面核心代码
48         //...
49
50         return index;
51     }
52
53     //下面就是增删改查 四个方法
54     public void addStudent(){
55         String name;
56         int age;
57         String gender;

```

```

58         Student s;
59         //判断 管理系统容量是否 已满
60         if(counter >= stus.length){
61             System.out.println("Add failure,too many
student!");
62             //扩容 数组拷贝, System.方法 Arrays.方法
63             return;
64         }
65         System.out.print("Please input name:");
66         name = in.next();
67         System.out.print("Please input age:");
68         age = in.nextInt();
69         System.out.print("Please input gender:");
70         gender = in.next();
71         s = new Student(name,age,gender);
72
73         //关键代码
74         stus[counter] = s;
75         counter++;
76
77         System.out.println("Add student success!");
78     }
79
80     //请补全下面代码
81     public void deleteStudent(){
82
83     }
84
85     public void updateStudent(){
86
87     }
88
89     public void selectStudent(){
90
91     }
92

```

```

93     public static void main(String[] args){
94         //实例化对象
95         StudentManagement sm = new StudentManagement();
96         //用户操作标志
97         int option = -1;
98         Loop: while(true){
99             // 打印提示信息
100            sm.prompt();
101            //获取用户操作
102            option = sm.in.nextInt();
103            if(option < 0 || option > 4)
104                continue;
105
106            switch(option){//byte char short int
107                case 1:
108                    sm.addStudent();
109                    break;
110                case 2:
111                    sm.deleteStudent();
112                    break;
113                case 3:
114                    sm.updateStudent();
115                    break;
116                case 4:
117                    sm.selectStudent();
118                    break;
119                case 0:
120                    break Loop;
121            }//end switch
122        }// end while
123
124        System.out.println("游戏结束, byebye");
125    }//end main
126 }//end class
127
128 class Student {

```

```
129      // 属性
130      private String name;
131      private int age;
132      private String gender;
133
134      //默认|无参构造器
135
136      //有参构造器
137
138      //get|set方法
139
140  }
```

答案:

1