第五章【面向对象基础】- 答案

1.谈谈你对面向对象和面向过程的理解

面向对象是一种以对象为中心的编程方法，它将程序的各个部分封装成对象并定义对象之间的交互关系。一个对象可以拥有数据、属性和方法，其它对象通过与该对象的交互来完成任务。

面向对象的编程强调的是"行为"与"状态"的抽象，便于代码的重用和扩展，同时也更易于组织和管理大型的软件系统。Java语言就是一种典型的面向对象的编程语言。

面向对象优缺点:

- 优点：易于维护、易于复用、易于扩展，由于面向对象有封装、继承、多态的特性，可以设计出低耦合的系统，使系统更加灵活、更加易于维护
- 缺点：比较抽象，性能比面向过程低

面向过程则是一种以任务为中心的编程方法，它将程序分解成若干个步骤或函数，每个函数都单独处理一部分任务。给定一个输入，程序按照一定的顺序依次执行这些函数，最后输出结果。面向过程的编程强调的是各个子任务之间的组合和流程控制，通常需要深入了解问题领域的许多具体细节。C语言就是一种典型的面向过程的编程语言。

面向过程优缺点:

- 优点：性能比面向对象高，适用于简单系统，容易来理解
- 缺点：不利于维护，复用，扩展

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。

面向过程实现步骤：

1. 游戏过程分析

```
1、开始游戏
2、黑方落子
3、绘制棋盘画面
4、输赢判断
    如果赢则结束游戏
    如果没有赢，则往下继续
5、白方落子
6、绘制棋盘画面
7、判断输赢
    如果赢则结束游戏
    没有赢则回到步骤2，继续往下运行
```

2. 把上面步骤中的功能设计成不同的方法

   比如：`startGame() luoZi() drawImage() isWin()` ...

面向对象实现步骤：

分析游戏中所需要的对象：棋子（包含黑方、白方）、游戏规则系统（负责判定是否犯规、输赢判断）、棋盘系统（负责绘制画面）

2. 定义棋子类、棋盘类、规则类，并添加合适的属性与方法
3. main方法中实例化所需的1个棋盘对象、1个规则对象，多个棋子对象，借助其方法实现最终的游戏功能

3.谈谈你对类、对象、引用的理解

类是一个模板，它描述一类对象的行为和状态

拿一条狗来举例，它的状态有：名字、品种、颜色，行为有：叫、摇尾巴和跑。

说白了，类就是我们自然界的一些统称，比如人、狗、车等。我们已经在实际生活中，将一些事物主动划分为某一类，将这个概念延伸至软件开发中，就是我们自己所写的 class 文件。

具体将哪些事物划分为某一类是由我们自己去调节的。比如，你可以将狗划分为一类，但是狗中的杜宾犬也可以成为一类。举个例子，一家专门出售杜宾犬的商店可以对所有的杜宾犬进行编号，记录每个杜宾犬的姓名、饮食时间。

对象是类的一个实例，具有状态和行为

如果说类是指的某一类，那么对象就是这一类当中具体的一个事物。比如狗，具体起来又分为拉布拉多、哈士奇，巴哥犬、吉娃娃等。它们都有自己独特的状态：名字、品种、颜色，以及独特的行为：叫、摇尾巴和跑。

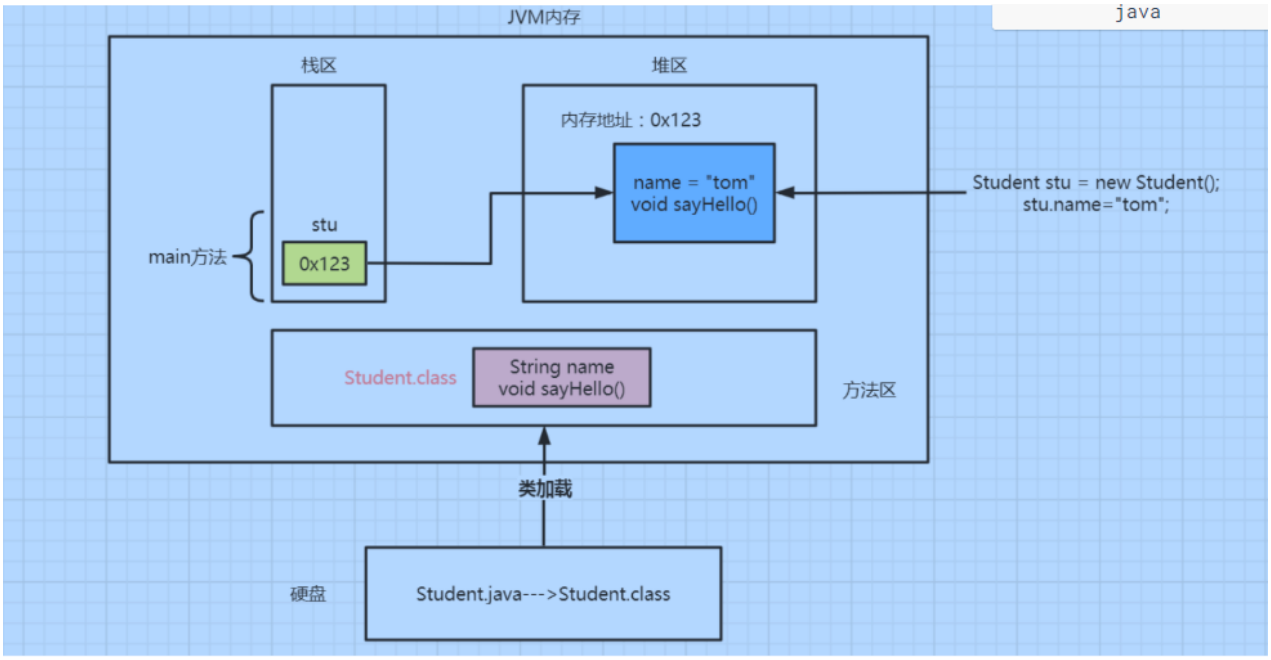
引用是一种数据类型，包含类类型、数组类型、接口类型（后续补充），用引用类型可以定义变量，引用变量对应的内存中存放的是某一对象在堆空间的内存地址

```
//创建一个Dog对象，名字叫小黑
Dog d = new Dog("小黑");

//通过对象名(引用类型变量)调用方法：让小狗跑起来
d.run();
```

4.根据下面案例，绘制对象内存图

```
class Dog {
    String name;

    public void run() {
        System.out.println(name + " 在随风奔跑");
    }
}

public static void main(String[] args) {
    Dog d1 = new Dog("小黑");
    Dog d2 = new Dog("虎子");

    d1.run();
    d2.run();
}
```

参考下图：



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：

```
class Car {
    //属性定义：颜色   品牌    价格

    //方法定义：前进、后退

}

public class Test05_Car {
    public static void main(String[] args) {
        //1.实例化Car对象

        //2.给对象属性赋值

        //3.输出对象属性信息

        //4.对象调用方法，让车跑起来

    }
}
```

答案：

```
package com.briup.chap05;

public class Test05_Car {
    public static void main(String[] args) {
        //1.实例化Car对象
```

```
        Car car = new Car();

        //2.给对象属性赋值
        car.color = "red";
        car.brand = "BYD";
        car.price = 215800;

        //3.输出对象属性信息
        System.out.println("颜色: " + car.color);
        System.out.println("品牌: " + car.brand);
        System.out.println("价格: " + car.price);

        System.out.println("----------------");

        //4.对象调用方法，让车跑起来
        car.forward();
        car.backOff();
    }
}

class Car {
    //定义属性
    //颜色   品牌    价格
    String color;
    String brand;
    double price;

    //前进   后退
    public void forward(){
        System.out.println("前进...");
    }

    public void backOff(){
        System.out.println("后退...");
    }
}
```

6.谈谈你对封装的理解

答案：封装是一种将数据和行为包装在类中的机制，以保护数据的安全性和可靠性。它是一种面向对象编程的基本概念，通过隐藏类的实现细节和访问限制，使得代码更易于维护和理解。

7.列举构造方法注意事项

答案：

- 构造方法一般使用`public`修饰
- 构造方法没有返回值类型，连`void`都没有
- 构造方法名和类名相同（区分大小写）
- 构造方法可以重载
- 用户不定义构造方法，系统会提供一个无参构造方法
- 用户定义构造方法，系统则不再提供无参构造方法
- 创建对象的时候调用，每创建一次对象，就会执行一次构造方法
- 用户不需要也不可以主动调用构造方法，系统会自动调用

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

```
class Person {
    private String name;
    private int age;

    //补全下面代码
}

//测试类如下：
public class Test08_Person {
    public static void main(String[] args) {
        Person p1 = new Person("jack",21);
        p1.show();

        System.out.println("--------------");

        Person p2 = new Person("lucy");
        p2.show();
    }
}
```

答案：

```
package com.briup.chap05;

//测试代码
public class Test08_Person {
    public static void main(String[] args) {
```

```
6           Person p1 = new Person("jack",21);
7           p1.show();
8
9           System.out.println("--------------");
10
11          Person p2 = new Person("lucy");
12          p2.show();
13      }
14  }
15
16  /*
17   *  1. 定义无参构造器
18      2. 定义两参构造器，参数为name和age
19      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
20      4. 定义show方法，输出类对象基本信息
21   *
22   */
23  class Person {
24      private String name;
25      private int age;
26
27      //1.定义无参构造器
28      public Person(){
29          System.out.println("无参构造器");
30      };
31
32      //2.定义2参构造器
33      public Person(String name, int age){
34          System.out.println("两参构造器");
35          this.name = name;
36          this.age = age;
37      }
38
39      //3.定义一参构造器
40      public Person(String name){
```

```
41          //输出语句不能放在第一行
42          //System.out.println("一参构造器");
43
44          //this(实参s) 必须为第一行有效代码
45          this(name,0);
46      }
47
48      //4.定义show方法，输出类对象基本信息
49      public void show() {
50          System.out.println("姓名: " + this.name + ", 年龄: " +
    age);
51      }
52  }
```

9.简述this关键字的用法

在类中的普通成员方法中，可以使用this关键，来表示当前方法的调用对象。

this关键字主要有三个应用：

1. 在成员方法中`this.数据成员`，表示该类对象的成员变量；
2. 在成员方法中`this.成员方法(实参列表);`表示该类对象调用本类中的成员方法；
3. 在构造方法中使用`this(实参列表)`，表示this调用本类中的其他构造方法，注意该代码必须为构造方法中的第一行有效代码。

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   第16行有误，private修饰符 应该修改为 public
2
3   程序输出结果为:
4   name: tom
5   age: 20
6   gender: 男
7   -----------
8   this.age: 0
9   this.name: null
10  this.gender: 男
```

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：

```
package com.briup.chap05;

public class Test11_Duration {
    public static void main(String[] args) {
        //借助三参构造器 实例化对象
        Duration d1 = new Duration(1,1,1);
        //获取总秒数
        int totalSeconds = d1.getTotalSeconds();
        System.out.println("1小时1分钟1秒： " + totalSeconds);

        System.out.println("--------------------");

        //借助单参构造器 实例化对象
        Duration d2 = new Duration(3660);
        //获取小时
        int hour = d2.getHour();
        System.out.println("hour = " + hour);

        System.out.println("--------------------");

        //输出对象属性信息及总描述信息
        d2.disp();
    }
}
```

答案：

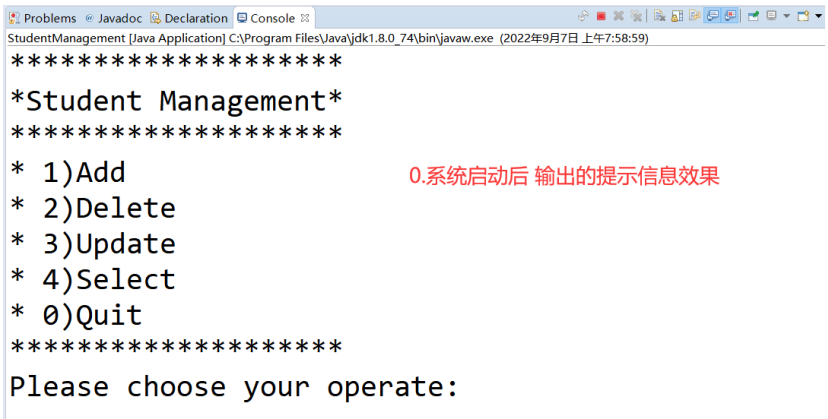
```java
//基础类定义
class Duration {
    private int hour;
    private int min;
    private int sec;

    //无参构造器
    public Duration() {}

    //三参构造器
    public Duration(int hour,int min,int sec) {
        this.hour = hour;
        this.min = min;
        this.sec = sec;
    }

    public Duration(int seconds) {
        int hour = seconds / 3600;
        int min = (seconds % 3600) / 60;
        int sec = seconds % 60;

        this.hour = hour;
        this.min = min;
        this.sec = sec;
    }

    //提供对应的get方法
    public int getHour() {
        return hour;
    }

    public int getMin() {
        return min;
    }

    public int getSec() {
```

```
37          return sec;
38      }
39
40      //获取总秒数
41      public int getTotalSeconds() {
42          return sec + min*60 + hour*60*60;
43      }
44
45      //输出对象基本信息
46      public void disp() {
47          System.out.println("hour: " + this.hour);
48          System.out.println("min: " + hour);
49          System.out.println("sec: " + sec);
50
51          //输出总秒数
52          System.out.println("totalSeconds: " +
    this.getTotalSeconds());
53      }
54  }
55
```

12.扩展题

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：



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!
```

2.删除学生操作

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：

```java
package com.briup.chap05;

import java.util.Scanner;

//学生管理系统类
public class StudentManagement{
    //学生对象数组
    private Student[] stus;
    //实际学生个数
    private int counter;
    //键盘录入对象
    private Scanner in;

    //无参构造器：默认容量100
    public StudentManagement(){
        stus = new Student[100];
        in = new Scanner(System.in);
    }
    //有参构造器：size表示初始容量
    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
```

```
class Student {
    // 属性
    private String name;
    private int age;
    private String gender;

    //默认|无参构造器

    //有参构造器

    //get|set方法

}
```

答案:

```
package com.briup.chap05;

import java.util.Scanner;

//学生管理系统类
public class StudentManagement {
    // 学生对象数组
    private Student[] stus;
    // 实际学生个数
    private int counter;
    // 键盘录入对象
    private Scanner in;

    // 无参构造器：默认容量100
    public StudentManagement() {
        stus = new Student[100];
        in = new Scanner(System.in);
    }

    // 有参构造器：size表示初始容量
```

```
21      public StudentManagement(int size) {
22          stus = new Student[size];
23          in = new Scanner(System.in);
24      }
25
26      // 输出 操作提示
27      public void prompt() {
28          System.out.println("********************");
29          System.out.println("*Student Management*");
30          System.out.println("********************");
31          System.out.println("* 1)Add");
32          System.out.println("* 2)Delete");
33          System.out.println("* 3)Update");
34          System.out.println("* 4)Select");
35          System.out.println("* 0)Quit");
36          System.out.println("********************");
37          System.out.print("Please choose your operate:");
38      }
39
40      // 下面就是增删改查 四个方法
41      private int findStudent() {
42          int index = -1;
43          String name;
44          System.out.print("Please input name:");
45          // 从键盘获取一个字符串
46          name = in.next();
47          for (int i = 0; i < counter; i++) {
48              if (name.equals(stus[i].getName())) {
49                  index = i;
50                  break;
51              }
52          }
53          return index;
54      }
55
56      public void addStudent() {
```

```
57          String name;
58          int age;
59          String gender;
60          Student s;
61          // 判断 管理系统容量是否 已满
62          if (counter >= stus.length) {
63              System.out.println("Add failure,too many
    student!");
64              // 扩容 数组拷贝，System.方法 Arrays.方法
65              return;
66          }
67          System.out.print("Please input name:");
68          name = in.next();
69          System.out.print("Please input age:");
70          age = in.nextInt();
71          System.out.print("Please input gender:");
72          gender = in.next();
73          s = new Student(name, age, gender);
74
75          // 关键代码
76          stus[counter] = s;
77          counter++;
78
79          System.out.println("Add student success!");
80      }
81
82      public void deleteStudent() {
83          int index = -1;
84          index = findStudent();
85          if (index == -1) {
86              System.out.println("Student not found!");
87              return;
88          }
89          for (int i = index; i < counter - 1; i++) {
90              stus[i] = stus[i + 1];
91          }
```

```
92          counter--;
93          System.out.println("Delete student success!");
94      }
95
96      public void updateStudent() {
97          int index = -1;
98          String name;
99          int age;
100         String gender;
101         index = findStudent();
102         if (index == -1) {
103             System.out.println("Student not found!");
104             return;
105         }
106         System.out.print("Please input new name:");
107         name = in.next();
108         System.out.print("Please input new age:");
109         age = in.nextInt();
110         System.out.print("Please input new gender:");
111         gender = in.next();
112         stus[index].setName(name);
113         stus[index].setAge(age);
114         stus[index].setGender(gender);
115         System.out.println("Update student success!");
116     }
117
118     public void selectStudent() {
119         int index = -1;
120         index = findStudent();
121         if (index == -1) {
122             System.out.println("Student not found!");
123             return;
124         }
125         System.out.print("name:" + stus[index].getName());
126         System.out.print(",age:" + stus[index].getAge());
```

```
127             System.out.println(",gender:" +
    stus[index].getGender());
128         }
129
130         public static void main(String[] args) {
131             // 实例化对象
132             StudentManagement sm = new StudentManagement();
133             // 用户操作标志
134             int option = -1;
135             Loop: while (true) {
136                 // 打印提示信息
137                 sm.prompt();
138                 // 获取用户操作
139                 option = sm.in.nextInt();
140                 if (option < 0 || option > 4)
141                     continue;
142
143                 switch (option) {// byte char short int
144                     case 1:
145                         sm.addStudent();
146                         break;
147                     case 2:
148                         sm.deleteStudent();
149                         break;
150                     case 3:
151                         sm.updateStudent();
152                         break;
153                     case 4:
154                         sm.selectStudent();
155                         break;
156                     case 0:
157                         break Loop;
158                 }// end switch
159             }// end while
160
161             System.out.println("游戏结束, byebye");
```

```
    }// end main
}// end class

class Student {
    // 属性
    private String name;
    private int age;
    private String gender;

    // 默认|无参构造器
    public Student() {}

    // 有参构造器
    public Student(String name, int age, String gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }

    // get|set方法
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

```

```
198         public String getGender() {
199             return gender;
200         }
201 
202         public void setGender(String gender) {
203             this.gender = gender;
204         }
205     }
```