

# 课程回顾

## 1 选择结构

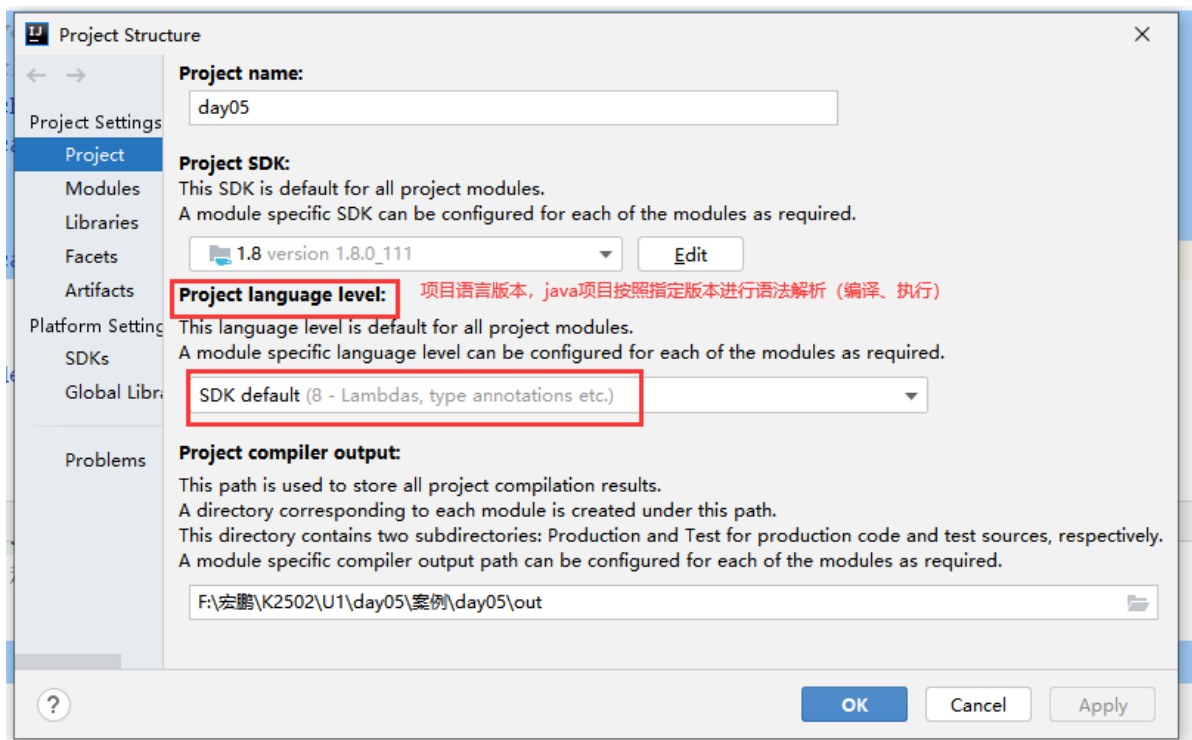
```
1  if语句
2  if-else语句
3  if-else if-else if-...-else 多重if
4  switch-case语句 ***
5
6  嵌套if, if结构整合使用: 保证if语句结构完整的。
7  最多嵌套层级不要超过3层。
8  提出一些流程控制关键字: return!!! break continue
9  if(){
10     if(){
11         if(){
12             switch(){
13
14             }
15         }
16     }else{
17
18     }
19 }
```

```
if(second<10) {
    //1-2 是否需要继续获取性别?
    //System.out.println("请选择性别: 输入1表示男 输入2表示女");
    System.out.println("请输入性别: "); //可能会输入除了男和女之外的词, 也要判
    String sex = input.next();
    //sex是男 还是女 if(sex=="男") {}

    switch (sex){
        case '男':
            System.out.println("男子决赛组");
            break;
        case '女':
            System.out.println("女子决赛组");
            break;
        default:
            System.out.println("请输入男或女的性别!");
            break;
    }
} else {
    System.out.println("继续加油!!!");
}
```

if(sex == 常量值)

switch后面的变量类型和case后面的常量值类型兼容的



```
// 性别 男 女 (sex == 男 / 女)
```

```
if (sex == "男") {  
    System.out.println("男子决赛组");  
} else {  
    System.out.println("女子决赛组");  
}
```

关系运算符主要用于基本类型之间值比较

```
/* switch (sex) { // String 类型比较 JDK 1.7 开始才 switch 支持 String 类型  
    ~~~~~  
}
```

## 课后作业第八题

### 分析

- 1 伪代码:
- 2 If() { // 判断选手是否进入决赛
- 3 If() { // 判断性别, 根据性别进行分组
- 4 // 1. Scanner 接收性别: 男 女 char 类型保存, 没有提供 nextChar()
- 5 // 2. String 保存性别, String 比较字符串是否相等, 怎么比? == !=
- 6 // 3. 扩展: 优化嵌套级别 return 单词使用

### 方案一

```
1 import java.util.Scanner;  
2  
3 /**  
4  * @Author: lc  
5  * @Date: 2022/3/9  
6  * @Description: 课后作业第八题: 嵌套 if 的使用  
7  *  
8  * @Version: 1.0  
9  */  
10 public class Exercise1 {
```

```

11     public static void main(String[] args) {
12         //1.准备数据 自动导包的快捷键 alt+enter
13         Scanner input = new Scanner(System.in);
14         //1-1 获取用户比赛成绩: int
15         System.out.print("请输入比赛的成绩: (秒)");
16         int second = input.nextInt();
17
18         if(second<10) {
19             //1-2 是否需要继续获取性别? ?
20             //System.out.println("请选择性别: 输入1表示男 输入2表示女");
21             System.out.println("请输入性别: "); //可能会输入除了男和女之外的词, 也要
判断BOY    boy Boy
22             String sex = input.next();
23             //java中, 哪些是常量? final定义 或 所有的字面量
24             //sex是男 还是女 if(sex=="男"){
25             if(sex.equals("boy")){//sex字符串的内容是"男",如果是true, 否则就是
false
26                 System.out.println("男子决赛组");
27             }else if(sex.equals("girl")){
28                 System.out.println("女子决赛组");
29             }else{
30                 System.out.println();
31             }
32             }else{
33                 System.out.println("继续加油!!!");
34             }
35
36         }
37     }

```

## 方案二:

```

1     import java.util.Scanner;
2
3     /**
4      * @Author: lc
5      * @Date: 2022/3/9
6      * @Description: 课后作业第八题: 嵌套if的使用
7      *
8      * @Version: 1.0
9      */
10    public class Exercise1 {
11        public static void main(String[] args) {
12            //1.准备数据 自动导包的快捷键 alt+enter
13            Scanner input = new Scanner(System.in);
14            //1-1 获取用户比赛成绩: int
15            System.out.print("请输入比赛的成绩: (秒)");
16            int second = input.nextInt();
17
18            if(second<10) {
19                //1-2 是否需要继续获取性别? ?
20                //System.out.println("请选择性别: 输入1表示男 输入2表示女");
21                System.out.println("请输入性别: "); //可能会输入除了男和女之外的词, 也要
判断BOY    boy Boy
22                String sex = input.next();
23                //java中, 哪些是常量? final定义 或 所有的字面量
24                //sex是男 还是女 if(sex=="男"){

```

```

25         switch (sex){//String类型比较 JDK1.7开始才switch支持String类型
26             case "男":
27                 System.out.println("男子决赛组");
28                 break;
29             case "女":
30                 System.out.println("女子决赛组");
31                 break;
32             default:
33                 System.out.println("请输入男或女的性别！");
34                 break;
35         }
36     }else{
37         System.out.println("继续加油！！");
38     }
39 }
40 }
41 }
42

```

## 方案三：使用return改良嵌套if

```

1  public class ReviewDemo {
2      public static void main(String[] args) {
3          //1.准备数据 自动导包的快捷键 alt+enter
4          Scanner input = new Scanner(System.in);
5          //1-1 获取用户比赛成绩: int
6          System.out.print("请输入比赛的成绩：（秒）");
7          int second = input.nextInt();
8          //2 给出不满足继续往后执行代码的条件，终止
9          /*if(second<10){
10             //进入决赛
11             */
12             if(second>=10){
13                 //没有进入决赛
14                 System.out.println("加油！！");
15                 return;//一旦执行return，程序停止了
16             }
17             //继续写代码，什么情况可以执行到这个位置else
18             //1-2 是否需要继续获取性别？
19             //System.out.println("请选择性别：输入1表示男 输入2表示女");
20             System.out.println("请输入性别：");//可能会输入除了男和女之外的词，也要判断
21             BOY boy Boy
22             String sex = input.next();
23             if(sex.equalsIgnoreCase("BOY"))//sex字符串的内容是"男",如果是true，否则
24             就是false
25                 System.out.println("男子决赛组");
26             }else if(sex.equals("girl")){
27                 System.out.println("女子决赛组");
28             }else{
29                 System.out.println(".....");
30             }
31         }
32     }
33 }
34

```

## 课程目标

## 1 String比较字符串相等

## 2 return关键字的作用

## 3 while循环 === 掌握

## 4 do-while循环 === 掌握

# 课程实施

## 1 String类型如何比较两个字符串是否一样（即是否相等）

验证：用户名 密码一般区分大小写

### 1-1 区分大小写比较字符串是否相等

- 1 字符串2.equals(字符串1)
- 2 : 字符串1和字符串2是否相等，equals，哪个字符串在前哪个字符串在后，没有影响

验证：验证码不区分大小写

### 1-2 不区分大小写比较两个字符串是否相等

- 1 字符串2.equalsIgnoreCase(字符串1)
- 2 : 字符串1和字符串2是否相等，equalsIgnoreCase，哪个字符串在前哪个字符串在后，没有影响

## 2 return关键字

return:返回

### 作用

终止程序，不再继续往后执行。简单粗暴理解：停止程序执行。jvm不再运行了

### 课堂案例

需求：登录功能：登录成功/用户名或密码不对

### 分析

```
1  if(用户名正确){
2      if(密码输入正确){
3          ok
4      }else{
5          密码输入有误
6      }
7  }else{
8      用户名输入有误
9  }
10
```

## 参考代码

```
1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/9
6   * @Description: 登录
7   * @Version: 1.0
8   */
9  public class ReturnDemo {
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12         System.out.println("用户名: ");
13         //next()和nextLine()都可以接受String。next()遇到空格停止接受  nextLine()
遇到回车不再继续接收
14         String username = input.nextLine();
15         System.out.println("密码");
16         String pwd = input.next();
17         System.out.println("username="+username);
18         System.out.println("pwd="+pwd);
19         // return挑出不合适、不合法
20         //if(username.equals("admin")==false){
21         if(!username.equals("admin")){//if(!true)
22             System.out.println("用户名输入有误");
23             return;
24         }
25         if(!pwd.equals("admin")){
26             System.out.println("密码输入有误!");
27             return;
28         }
29         System.out.println("登录成功!");
30     }
31 }
```

## 练习案例

- 需求：乘坐公交车
1. 首先提示用户输入是否有绿码：true-有 false-没有  
接收boolean String，不区分大小写true
2. 如果有绿码，请支付车费：2元，多了不行，少了也不行
3. 如果有绿码且能够支付2元车费，sout("一路顺风")。
- 没有绿码：提示没有绿码，不能乘坐公共交通工具
- 有绿码，但是不能支付2元车费：提示公共交通工具，不设找零，请支付2元车费

## 分析

```

1  if(有绿码){
2      if(得有2元钱){
3          sout();
4      }else{
5          sout();
6      }
7  }else{
8      sout()
9  }
10
11 //不写嵌套代码结构，也能实现需求
12

```

## 参考代码

```
1 |
```

## 3 循环结构

### 3-1 概述

循环：重复做同一件事情

循环案例体现：写代码 做梦

重复说一万次：我爱你

### 3-2 循环使用

实现：sout打印一万次“我爱你”

```

1  while(输出次数不足一万次){
2      打印 “我爱你”
3  }

```

如何实现？

方案一：顺序结构

### 3-3 循环结构

```

//变量：一块内存，存取数据
int count=0; //count保存实际执行sout的次数
while(count<10000){
    System.out.println("第"+count+"次输出：我爱你");
}
/*System.out.println("I Love You");
System.out.println("I Love You");
*/

```

无限循环：也称为死循环  
发生：内存溢出问题

## while循环

```
1 while(boolean表达式条件){//当....发生时
2     //代码段：循环操作
3 }
4 解析：
5     boolean条件：循环条件
6     循环操作：重复做的那件事情
```

### 课堂案例：输出一万次我爱你

#### 方案一：顺序结构实现

```
1 /**
2  * @Author: lc
3  * @Date: 2022/3/9
4  * @Description: while循环入门案例
5  * @Version: 1.0
6  */
7 public class Demo1 {
8     public static void main(String[] args) {
9         //重复输出一句话：传统顺序结构解决需求，存在问题??
10        //存在代码冗余 重复? 中国人 I Love You 维护困难
11        System.out.println("I Love You");
12        System.out.println("I Love You");
13        System.out.println("I Love You");
14        System.out.println("我爱你");
15        System.out.println("我爱你");
16        System.out.println("我爱你");
17        System.out.println("我爱你");
18        System.out.println("我爱你");
19        System.out.println("我爱你");
20        System.out.println("我爱你");
21        System.out.println("我爱你");
22        System.out.println("我爱你");
23        System.out.println("我爱你");
24        System.out.println("我爱你");
25        System.out.println("我爱你");
26        System.out.println("我爱你");
27        //此处省略N次输出
28    }
29 }
30
```

方案一存在的代码问题有：

1.代码冗余，造成的代码后期修改困难的问题

#### 方案二：使用循环结构实现

```
1 /**
2  * @Author: lc
3  * @Date: 2022/3/9
4  * @Description: while循环入门案例
5  * @Version: 1.0
6  */
```



```

7 public class Demo1 {
8     public static void main(String[] args) {
9         //重复输出一句话：传统顺序结构解决需求，存在问题？？
10        //存在代码冗余 重复？中国人 I Love You 维护困难
11        //循环结构：主要应用于解决重复做同一件事情
12        //循环变量
13        //变量：一块内存，存取数据
14        int count=0;//count保存实际执行sout的次数
15        while(count<10000){
16            //修改循环变量
17            count++;//count=count+1;
18            System.out.println("第"+count+"次输出：我爱你");
19        }
20    }
21 }
22

```

## 学生练习：

需求：输出1-100之间所有的偶数

```

1  循环操作：
2  sout(偶数)
3
4  循环条件：1-100之间
5
6  循环变量：int num=1
7  while(num<=100){
8      if(num%2==0){
9          sout(num);
10     }
11     num++;
12 }

```

## 参考代码

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/9
4   * @Description: while循环应用案例一：求1-100之间偶数和
5   * @Version: 1.0
6   */
7  public class Demo2 {
8      public static void main(String[] args) {
9          //1.输出1-100之间所有的数
10         //1-1 保存输出次数
11         int count=1;
12         //1-2 保存1-100之间偶数和
13         int sum=0;
14         while(count<=100){
15             //2.挑出偶数
16             if (count%2==0) { //基于条件才能做输出 弹出自动嵌套的代码结构：
17                 ctrl+alt+t
18                 System.out.println(count);
19             }
20             count++;
21         }
22     }
23 }

```

```

20     }
21 }
22 }
23

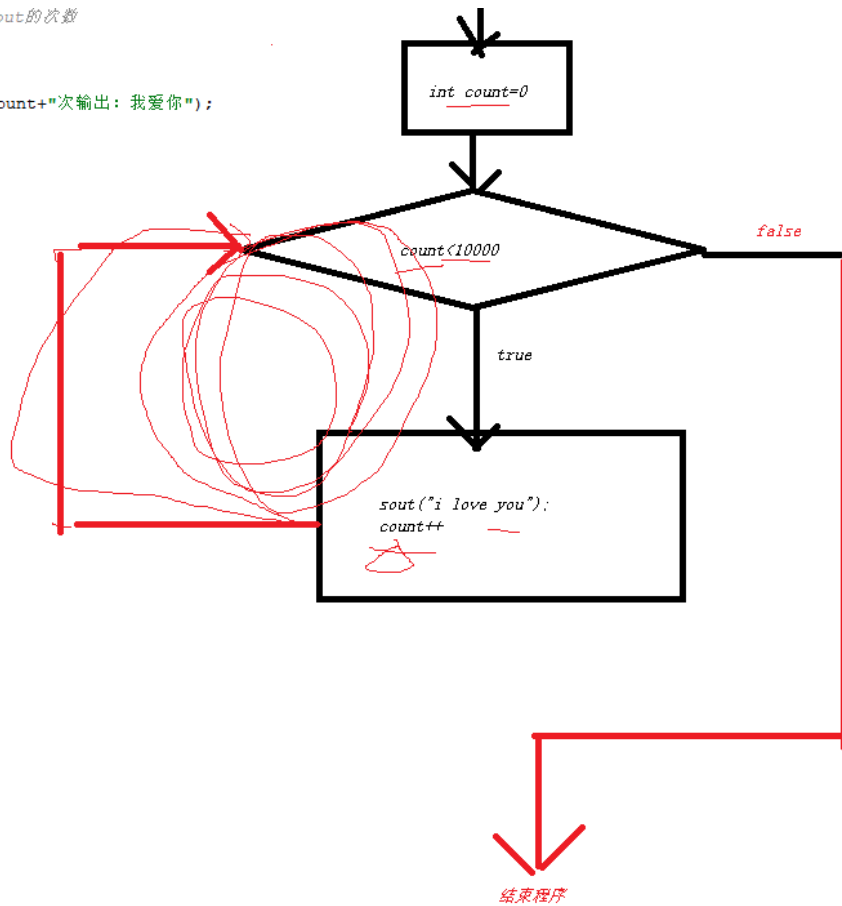
```

## while循环执行的流程

```

int count=0; //count保存实际执行sout的次数
while(count<10000){
    //修改循环变量
    count++; //count=count+1;
    System.out.println("第"+count+"次输出: 我爱你");
}

```



## 课堂案例：求1-100之间偶数和

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/9
4   * @Description: while循环应用案例一：求1-100之间偶数和
5   * @Version: 1.0
6   */
7  public class Demo2 {
8      public static void main(String[] args) {
9          //1. 输出1-100之间所有的数
10         //1-1 保存输出次数
11         int count=1;
12         //1-2 保存1-100之间偶数和
13         int sum=0;
14         while(count<=100){
15             //2. 挑出偶数
16             if (count%2==0) { //基于条件才能做输出 弹出自动嵌套的代码结构:
17                 //偶数加在一起，存入sum
18                 //
19                 * sum=sum+count 简化sum+=count

```

```

20         * 当count=2 第一个偶数 sum=sum+2 sum=0+2 2
21         * 当count=4 第二个偶数 sum=sum+4 sum=2+4 sum存6
22         * 当count=6 第三个偶数 sum=sum+6 sum=6+6 sum存12
23         * 当count=8 第四个偶数 sum=sum+8
24         */
25         //sum=第一个数+第二个偶数+第三个偶数+....+偶数;
26         sum=sum+count;
27         System.out.println(count);
28     }
29     count++;
30 }
31 System.out.println("1-100之间偶数和: "+sum);
32 }
33 }
34

```

## 执行流程特点：

while循环基于先判断，再执行。假如循环条件一次也不满足，循环操作就一次也不会执行。

## 循环使用心得

循环三要素

- 1 定义控制循环条件的变量
- 2 循环条件boolean表达式
- 3 循环操作（重复做的事情、修改循环变量）

## 课堂案例

需求：吃饺子

```

1  假设餐厅吃饺子，收费方式一个饺子0.9元，老板统计一个人这顿饭该支付多少钱，统计一共吃了多少个
   饺子。
2
3  吃一个饺子，问自己：吃饱了吗？吃饱了，就结束。没有吃饱，接着吃一个饺子，问自己：吃饱了？
4
5  伪代码：
6  String daAn=???;
7  while(输入答案不是yes){
8      //循环操作
9      吃饺子
10     计数
11     sout("吃饱了没有？输入yes吃饱了，输入其他都是没吃饱");
12     daAn=input.next();
13 }
14 sout("一共吃了个饺子，应该支付的金额是: "+ (count*0.9))

```

## 参考代码

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/9
6   * @Description: while循环应用案例二：String作为循环的条件。吃饺子案例

```

```

7      * @version: 1.0
8      */
9      public class Demo3 {
10         public static void main(String[] args) {
11             Scanner input = new Scanner(System.in);
12             //System.out.println("饿不? 输入yes表示饱了, 输入其他表示饿了");
13             String daAn="yes";//默认值给什么合适???
14             //保存吃得饺子个数
15             int count=0;
16             while(true){
17                 System.out.println("吃一个饺子");
18                 //计数
19                 count++;
20                 System.out.println("饿不? 输入yes表示饱了, 输入其他表示饿了");
21                 daAn=input.next();
22                 if(daAn.equals("yes")) {
23                     break;//可以
24                 }
25             }
26             System.out.println("吃了"+count+"个饺子, 支付的钱: "+(count*0.9));
27         }
28     }
29

```

## 学生练习

- 1 需求: 重复写代码学习
- 2 张三判断今天要不要继续写代码, 判断依据: 作业写完了么?
- 3
- 4 张三写作业, 提交给老师批改, 问老师批改的结果, 老师说: 通过, 可以不再继续写作业量, 未通过, 张三继续写代码, 直到老师通知作业通过为止。
- 5
- 6 三要素:
- 7 循环操作: 写作业 改作业      修改循环变量的代码: scanner与老师互动得到的答案
- 8 循环条件: 老师评价: 未通过
- 9 循环变量: String 变量=""

## 程序调试

bug: 称为漏洞 错误。日常英语: 臭虫

debug: 日常英语: 捉臭虫 理解: 找漏洞

工具使用, 使用步骤:

- 1 1. 根据经验, 锁定臭虫出没大概范围
- 2 2. 范围这一篇代码的第一行加断点 (breakPoint, 程序遇到断点会停止)。使用debug模式运行程序
- 3 3. 程序启动后, 遇到断点, 就会停下来, 此时使用F8单步执行方式, 地毯式搜索臭虫
- 4 4. 监控代码执行过程中, 变量、表达式值变化, 根据经验拿实际值和预期值对比。找到程序出错的位置、分析代码原因

作用: 找问题位置!!!

## 预习安排

for循环

嵌套循环

数组：数组定义 循环 数据的操作：添加、删除数据、替换数据、修改数据

吃货联盟在线点餐系统

# 课程总结

---

## 1 String的比较字符串相等 理解

---

## 2 return流程控制的关键字

---

## 3 循环执行流程、三要素

---

## 4 循环语法

---

