

课程回顾

案例需求：

保存图书相关的信息

```
1  1.书名  String
2  2.出版日期 String
3  3.库存  int
4  4.是否清算?  boolean
5  5.单价  float
6  6.图书类型: char
7  '国'  '外'
8
9  按照以下格式输出:
10 书名      日期      库存....
11  《西游记》
12  《红楼梦》
13
```

参考代码

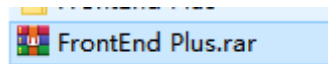
```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 描述图书相关的信息
5   * @Version: 1.0
6   */
7  public class BookInfo {
8      //JVM进入程序执行的入口
9      public static void main(String[] args) {
10         // 1.数据 容器
11         // 书名
12         String bookName="游西记1";
13         String bookName2="西游记2";
14         // 库存
15         int count1=200;
16         int count2=300;
17         // ctrl+/ 出版日期
18         String date="2012-12-31";
19         //4.是否清算?  boolean
20         boolean is=true;
21         //5.单价  float
22         float price=23.0F;
23         //6.图书类型: char
24         char type='N';
25
26         //2.处理 输出
27         System.out.println("书名\t\t\t出版日期\t\t\t是否清算");
28         System.out.println(bookName+"\t\t\t"+date+"\t\t\t"+is);
29
30         //准备第二本图书信息
31         bookName="红楼梦";
```

```
32         System.out.println(bookName+"\t\t\t"+date+"\t\t\t"+is);
33
34     }
35 }
```

反编译工具的使用

常见的反编译工具：

IDEA本身也可以执行反编译，另外也可以使用一些其他的反编译工具，比如教学素材提供：frontEnd



- 1 编译：java源代码编译成class文件（字节码文件），jvm可以执行class文件
- 2 反编译：将class文件逆向生成源文件过程。
- 3
- 4 反编译作用：阅读jvm实际执行的代码
- 5 使用场景：OOP编程。构造方法
- 6
- 7 主要用于程序调试。

课程目标

1 Scanner使用 ===== 掌握

2 运算符使用 ===== 掌握

3 数据类型转换 ===== 理解

课程实施

1 Scanner

回顾经历：变量和赋值，变量赋值，值从哪儿来的？？

int num=数据？？ 数据：自己随便写，灵活性差。

优化：用户输入的方式，接收数据

1-1 作用

Scanner主要的作用：完成接收用户输入值功能

使用场景：希望用户输入数据的时候。

1-2 使用

```
import java.util.Scanner;
```

import必须出现每个java文件的第一行

```
/**
 * @Author: lc
 * @Date: 2022/3/7
 * @Description: PACKAGE_NAME
 * @Version: 1.0
 */
public class QQDemo {
    public static void main(String[] args) {
        //1. 需要一个qq号码和qq密码
        //思考: 数据从哪儿来? 用户输入===》Scanner
        //idea支持自动导入, import不用程序员自己写, 也不用记
        Scanner input=new Scanner(System.in); //定义监听控制台用户输入
    }
}
```

TODO Problems Terminal Profiler Auto-build Build

- 1 1. 自己的程序中引入jdk提供的Scanner工具
- 2 import java.util.Scanner;
- 3
- 4 2. 创建一个Scanner的变量
- 5 Scanner 监听器名称=new Scanner(System.in); //System.out控制台 System.in 控制台
- 6
- 7 3. 开始监听: 接收用户输入的数据
- 8 建议: sout()给出输入值的一些提示信息
- 9 3-1 定义变量
- 10 int num; //提供给监听器使用
- 11 3-2 变量保存监听器监听的数据
- 12 num=监听器的名称.nextInt(); //监听输入整数int类型
- 13 监听器的名称.next(); //监听String
- 14 监听器的名称.nextDouble(); //监听输入整数Double类型
- 15 监听器的名称.nextFloat(); //监听输入整数Float类型
- 16 监听器的名称.nextByte(); //监听输入整数byte类型
- 17 监听器的名称.nextShort(); //监听输入整数short类型
- 18 监听器的名称.nextLong(); //监听输入整数long类型
- 19 监听器的名称.nextBoolean(); //监听输入整数boolean类型
- 20 注意事项: Scanner没有提供监听char类型数据的功能: 因此不存在: 监听器的名称.nextChar(); //监听输入整数int类型

课堂案例:



需求：使用Scanner接收用户输入用户名和密码，分别保存在两个变量

分析：

```
1 QQ: long
2 用户名: String
3 密码: String
```

参考代码

```
1 import java.util.Scanner;
2
3 /**
4  * @Author: lc
5  * @Date: 2022/3/7
6  * @Description: PACKAGE_NAME
7  * @Version: 1.0
8  */
9 public class QQDemo {
10     public static void main(String[] args) {
11         //1.需要一个qq号码和qq密码
12         //思考：数据从哪儿来？用户输入===》Scanner
13         //idea支持自动导入，import不用程序员自己写，也不用记
14         Scanner input=new Scanner(System.in); //定义监听控制台用户输入数据的监听器
15
16         //2.监听用户输入的数据
17         //2-1 先问
18         System.out.print("请输入QQ号码: ");
19         //2-2 再答
20         long qqNum=input.nextLong();
21
22
23         //3.输出用户输入的qq号码
24         System.out.println("您输入的qq号码是: "+qqNum);
25     }
26 }
```

学生练习:转账

需求：

1.定义两个变量：user1=500 user2=250保存账户现有余额

模拟ATM取款，提示用户1输入转账金额。使用Scanner接收用户输入的取款金额

计算user1和user2的卡内余额

扩展：if

1.转账金额必须是整百

2.转账金额必须小于等于卡内余额

参考代码

```
1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/7
6   * @Description: 两个 账户之间转账
7   * @Version: 1.0
8   */
9  public class ATM {
10     public static void main(String[] args) {
11         //1.数据
12         int user1=500;//银行卡余额
13         int user2=250;//银行卡余额
14
15         //2.模拟ATM输入转账金额
16         Scanner input=new Scanner(System.in);
17         System.out.print("请输入转账金额: ");
18         int money = input.nextInt();
19         //long l = input.nextLong();//long ctrl+alt+v自动根据数据值生成变量定义
20
21         //3.处理转账的结果
22         //3-1 判断转账金额是否是整百 转账金额是否充足: 小于等于卡内余额
23
24         //user1=user1-money;//转出之后的余额
25         //user1-=money;
26         //+= += -= *= /= %=
27         //user2=user2+money;//转入后的余额
28         //user2+=money;
29
30         //基于整百, 余额是否足够
31         user1=user1>=money&&money%100==0?user1-money:user1;
32         user2=user2>=money&&money%100==0?user2+money:user2;
33         //4.输出程序处理结果
34         System.out.println("user1卡内余额是: "+user1);
35         System.out.println("user2卡内余额是: "+user2);
36     }
37 }
```

2 运算符

2-1 概述

1+1=2用数学公式来看, 两个1参与数学运算的元素。

2-2 分类

分类的依据: 根据一个算式参与运算的元素个数, 划分成三类

一元运算符 二元运算符 三元运算符

一元运算符

- 1 +: 正数书写格式+12 正号, 可以省略
- 2 -: 负号
- 3
- 4 ++: 自增运算符
- 5 --: 自减运算符

课堂案例

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 演示一元运算符
5   * @Version: 1.0
6   */
7  public class Demo1 {
8      public static void main(String[] args) {
9          //int num=+12;//正数
10         //int num2=-90;//负数
11
12         //常用的方式
13         //++:在当前参与运算的数据基础上+1
14         //--:在当前参与运算的数据基础上-1
15         int a=10;
16         //++独立使用时, ++在前还是后面, 效果是一样的
17         //a++;//等价于a=a+1
18         //++a;//等价于a=a+1
19         //++ --放入sout时, ++ -- 在变量前或变量后, 意义不一样
20         //sout(++ --):++ --在变量后, 先输出, 再运算
21         //sout(++ --):++ -- 在变量前, 先运算, 再输出
22         //System.out.println(++a);//10 //1.运算 11 2.11
23         //System.out.println(a);//11
24
25         /*System.out.println(a++);//10
26         System.out.println(++a);//12
27         System.out.println(a--);//12
28         System.out.println(a);//11
29         System.out.println(--a);//10
30         System.out.println(a);//10*/
31
32         System.out.println(a++);
33         System.out.println(a++);
34         System.out.println(a++);
35         System.out.println(a++);
36         System.out.println(a++);
37         System.out.println(a++);
38         System.out.println(a++);
39     }
40 }
```

二元运算符 重点

```
1  算术运算符：同数学
2  + - * / %（读作模）
3
4  注意：+ - * /最后的计算结果遵循的原则是：参与运算的元素有小数，计算结果就是小数，都是整
   数，运算结果就是整数。
5  %只针对整除求余。小数求余没有意义！！！！
6
7  关系运算符：同数学
8  > >= < <= == !=
9
10 赋值运算符：
11  =（赋值符号）
12  +=
13  -=
14  *=
15  /=
16  %=
17  += -= *= /= %=的作用就是在变量的基础上做+ - * / %并将计算结果保存到原变量中
18
19 逻辑运算符：
20  &-与  |-或  !=非，取反  &&-短路与  ||--短路或
21  重要区分：短路与不短路区别？？
22  & &&: &&在复杂的逻辑运算表达式中，前面的表达式如果能够得出false的结果，那么&&后面的表达式
   就会忽略不去计算
23  | ||: ||在复杂的逻辑运算表达式中，前面的表达式如果能够得出true的结果，那么||后面的表达式
   就会忽略不去计算。所以，短路的&& ||可以优化表达式计算的效率
24  日常开发使用时，优先使用短路方式if while()
```

课堂演示案例

算术运算符

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 算术运算符
5   * @Version: 1.0
6   */
7  public class Demo2 {
8      public static void main(String[] args) {
9          //所有的整数类型小于浮点数，整数和小数一起运算，结果一定是小数
10         double num1=10;
11         int num2=-3;
12         //两数进行算术运算，计算结果与参与运算的两个数据中，类型最大的类型为准
13         System.out.println(num1+num2);//13
14         System.out.println(num1-num2);//7
15         System.out.println(num1*num2);//30
16         // /除法也是整除吗？？？
17         System.out.println(num1/num2);//3
18         //%:模 整除求余数 14/3=4....2 带不带负号，看被除数
19         System.out.println(num1%num2);//1
20     }
21 }
```

关系运算符

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 关系运算符
5   * @Version: 1.0
6   */
7  public class Demo3 {
8      public static void main(String[] args) {
9          //关系运算符计算结果一定是boolean、true false
10         //关系运算符连接元素形成一个运算公式，称为boolean表达式
11         int num1=12;
12         int num2=-100;
13         System.out.println(num1>num2);
14         System.out.println(num1>=num2);
15         System.out.println(num1<num2);
16         System.out.println(num1<=num2);
17         System.out.println(num1==num2);
18         System.out.println(num1!=num2);
19     }
20 }
```

逻辑运算符

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 逻辑运算符 & | ! && ||
5   * 提示：逻辑运算符前后只能boolean值或boolean表达式
6   * 逻辑运算符计算结果也是：boolean值
7   * @Version: 1.0
8   */
9  public class Demo4 {
10     public static void main(String[] args) {
11         boolean b1=true;
12         boolean b2=false;
13         //& &&前后两个boolean值，同真则真，有假即假
14         //false
15         System.out.println(b1&b2);//长得美 且 有工作 &表示且关系，同时成立，满足
16         System.out.println(b1&& b2);//false
17
18         // | || 前后两个boolean值，有真则真，同假即假
19         System.out.println(b1|b2);//长得帅 或 有钱 true
20         System.out.println(b1||b2);// true
21
22         // ! 非，取反 真则假，假则真
23         System.out.println(!b1); // false
24         System.out.println(!b2); // true
25
26     }
27 }
```


短路与不短路区别

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 短路与不短路区别
5   * 运算顺序: 自增自减>算术运算符>关系运算符>逻辑运算符>赋值运算符
6   * 如果改变运算顺序, 使用(), 永远优先计算() 中
7   * @Version: 1.0
8   */
9  public class Demo5 {
10     public static void main(String[] args) {
11         int a=1;
12         int b=2;
13         //& |:& | 第一个表达式已经决定了整个算式结果, 没有必要再让jvm帮你执行后面的复杂运
            算, 非常耗时
14         //&&: 第一个false, 第二个公式不去计算
15         //||: 第一个true, 第二个公式不用计算
16         boolean result=a>b&&++a==2;//false died code
17
18         System.out.println(result);
19
20         System.out.println(a);//2
21     }
22 }
```

赋值运算符

```
1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/7
6   * @Description: 两个 账户之间转账
7   * @Version: 1.0
8   */
9  public class ATM {
10     public static void main(String[] args) {
11         //1.数据
12         int user1=500;//银行卡余额
13         int user2=250;//银行卡余额
14
15         //2.模拟ATM输入转账金额
16         Scanner input=new Scanner(System.in);
17         System.out.print("请输入转账金额: ");
18         int money = input.nextInt();
19         //long l = input.nextLong();//long ctrl+alt+v自动根据数据值生成变量定义
20
21         //3.处理转账的结果
22         //3-1 判断转账金额是否是整百 转账金额是否充足: 小于等于卡内余额
23
24         //user1=user1-money;//转出之后的余额
25         user1-=money;
26         //+= += -= *= /= %=
27         //user2=user2+money;//转入后的余额
28         user2+=money;
29     }
```

```

30         //4.输出程序处理结果
31         System.out.println("user1卡内余额是: "+user1);
32         System.out.println("user2卡内余额是: "+user2);
33     }
34 }
35

```

三元运算符

- 1 ? : 也读作条件运算符
- 2 条件boolean表达式? 前面条件成立, 给的值: 前面条件不成立, 给的值

课堂案例

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/7
4   * @Description: 条件运算符（即三元运算符）
5   * @Version: 1.0
6   */
7  public class Demo6 {
8      public static void main(String[] args) {
9          int money=550;
10         //是整百, 得出true不是整百, false
11         //12/4=3...0  12/5=2...2
12         boolean b=money%100==0;
13         System.out.println(b);//false
14
15         String result=b?"转账成功":"转账失败";
16         System.out.println(result);
17     }
18 }

```

常见的坑点:

C:\Program Files\Java\jdk-11.0.10\bin\java.exe ...
 请输入QQ号码: *fsdfsfs*
 Exception in thread "main" java.util.**InputMismatchException** Create breakpoint
 at java.util.Scanner.throwFor(Scanner.java:864)
 at java.util.Scanner.next(Scanner.java:1485) 类型不匹配的异常: 用户输入的数据与存入的变量类型不兼容
 at java.util.Scanner.nextLong(Scanner.java:2222)
 at java.util.Scanner.nextLong(Scanner.java:2182)
 at QQDemo.main(**QQDemo.java:20**) ← 代码发生异常位置
 Process finished with exit code 1

课程总结

1 运算符的顺序

- 1 运算顺序: 自增自减>算术运算符>关系运算符>逻辑运算符>条件运算符>赋值运算符
- 2 * 如果改变运算顺序, 使用(), 永远优先计算()中

2 Scanner小结: ctrl+alt+v

3 运算符的小结

4 ?:尝试使用

预习安排

流程控制语句: 条件语法if

语法、执行流程

循环语句: 难点**