

Java03-数据类型与数据结构

Task1. 变量和数据类型

Java 的八种基本数据类型如下：

1. 2.

byte —— 1字节，表示范围： -128 ~ 127

short —— 2字节，表示范围： -32,768 ~ 32,767

int —— 4字节，表示范围： -2,147,483,648 ~ 2,147,483,647

long —— 8字节，表示范围： -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

float —— 4字节，单精度浮点数

double —— 8字节，双精度浮点数

char —— 2字节，表示单个字符（Unicode编码）

boolean —— 1字节，表示 true 或 false

3. 隐式类型转换, b为52, 因为'0'会转换为整数48(ASCII)

4. (1)false, 因为x和y记录的是不同的地址值 (2)true,z和k指向的都是18(缓存机制) (3)false, 超出了缓存的范围, 会返回新的对象

Task2. 运算符

5. 答案为13,6 8

int a = 5 ; a 初始值为 5。

int b = 7 ; b 初始值为 7。

int c = (++a) + (b ++);

++a 是前置自增，a 先加 1 变为 6，然后参与运算。b++ 是后置自增，先用 b 的当前值 7 参与运算，运算后 b 再加 1 变为 8。所以 c = 6 + 7 = 13。System.out.println(c); 输出 13。

System.out.println(a+" "+b); 输出 a 和 b 的当前值，即 6 8。

6. (网上查阅)

int (整数) int 类型在计算机中以二进制补码的形式存储。以 32 位 int 为例，最高位是符号位（0 表示正数，1 表示负数），其余位表示数值。补码的设计使加减法运算统一，便于硬件实现。

float (浮点数) float 类型采用 IEEE 754 标准存储，包括三部分：符号位、指数位和尾数（有效数字）位。这样可以表示很大或很小的小数，但存在精度误差。

为什么两个正数相加会变成负数？

这是因为溢出。int类型有最大值（如32位int最大为2,147,483,647），当两个较大的正数相加超过这个范围时，结果会“绕回”到最小的负数区间，表现为负数。这是补码运算的自然结果。

Task3.数据结构

代码请见 ThirdQuestion

自个鼓捣了一个多小时也没能用一个程序把四个功能全部运行完成,最后借助了ai的力量,但运行结果令我十分费解,我觉得代码方面应该没有问题,运行结果却与我想的大相径庭(如下)

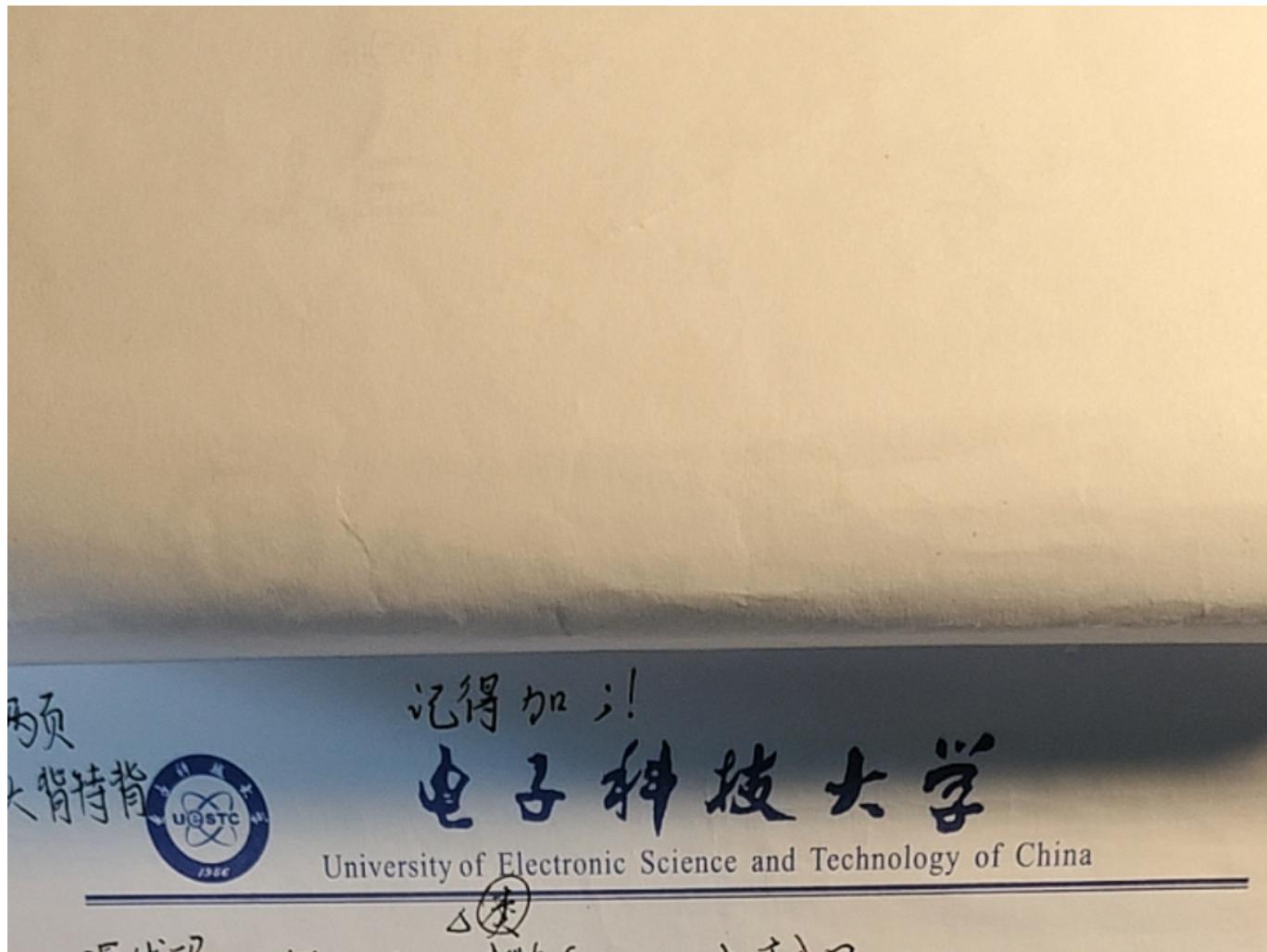
```
1 package ThirdQuestion;
2 import java.util.*;
3
4 public class one {
5     Run | Debug
6     public static void main(String[] args) {
7         // 用于存储每个人的错题统计
8         Map<String, Map<String, Integer>> allData = new HashMap<>();
9         Scanner scanner = new Scanner(System.in);
10
11         while (true) { ...
12             scanner.close();
13         }
14
15         // 定义一个方法用于输出错题最多的科目及其错题数
16         public static void sout(Map<String, Integer> map){...
17     }
18 }
```

问题 37 输出 调试控制台 终端 端口

```
PS C:\Users\29867\Desktop\daima> & 'D:\java\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:52214' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\29867\AppData\Roaming\Code\User\workspaceStorage\9a8851d6034f7cd4fca62250d46cd95c\redhat.java\jdt_ws\daima_d46075f1\bin' 'ThirdQuestion.one'
请输入姓名（如小明），输入end结束：
小明
请输入科目和错题数，（格式为科目：错题数）输入end结束
end
C?}的错题统计: {}
请输入姓名（如小明），输入end结束：
end
PS C:\Users\29867\Desktop\daima> []
```

实在是燃尽了

一些学习笔记如下(还有一些塞不下):



```

public class 入口类 {
    public static void main(String[] args) {
        System.out.println("输出");
        // 基本数据类型 提升为 int
        整数 byte short int long +L 单作后缀
        浮点 double float +F 后缀 取值不太相同
        字符 char
        布尔 boolean: true or false
        String + 字符串
    }
}

```

标识符 数母 \$ 且数不开头 ! 不能带 . 知否!

命名法 1. 一词不大 2. 多: 首不大后大 [方法变量]

1. 首词大 2. 全首大 [类名]

△ 键盘录入: 1. 导包 `Scanner import java.util.Scanner`

2. 创建对象用 scanner 类 `Scanner sc = new Scanner(sys`

3. 接收数据 `int i = sc.nextInt(); nextDouble(); ne`

最后将 i 打印 例: `if i=123`

运算符/表达式

6种: 1. 算术: + - * / % (取模、取余) double

强制转换: 类型变1 =⁺ (类型) 变2 long char

"+" 出现字符串时会进行拼接. (从左到右逐个进行)



电子科技大学

University of Electronic Science and Technology of China

2. 自增减 $++$, $--$

3. 赋值: $=$ $+=$ $-=$ ~~$\star =$~~ $\star =$ $/=$ $\% =$ (PS: 感觉不好用)
[隐藏了一个强转] [不行, 特定情况可用]

4. 关系/比较 [$true/false$] 布尔类型

$==$ 判相等 $!=$ 判不相等 $\dots\dots$ $\% ==$

5. 逻辑 ~~\neq &~~ $\&$ (与) $|$ (或) \wedge (异或) $!$ (非)
 \downarrow 两方都是 \downarrow 一方是 同否反 true

6. 短路逻辑 $\&&$ $||$ 效率高

若左侧能确定, 右侧便不会再运行

7. 三元运算符

格式: 关系表达式? 表达式1: 表达式2

e.g. $a > b ? a : b$. 真则左, 假则右

8. 运算符优先级

() 为顶级优先

地址：成都市建设北路二段四号
高新区西区西源大道2006号

查询电话：83201114

邮政编码：610054（沙河校区）
611731（清水河校



电子科技大学

University of Electronic Science and Technology of China

二、造型

① 定义类、接口、方法时 同时声明多个类型变量 (int String 等)

② 约束编辑时所能操作的数据类型

引用=可以！
int String 等)
基本不行！

避免强转和一些异常

③ eg: `ArrayList< >` [泛型类]

`public static <T> void printArray(T[] names) { ... }`

(可用于打印不同数组) [泛型方法]

④ 通配符

即“?” 使用泛型时代表一切类型。

`E T K V` 是定义泛型时使用。

⑤ 上下限

(I) 上限: `? extends Car` \uparrow Car 或其子类

(II) 下限: ~~? extends Car~~ \downarrow super Car \uparrow Car 或其父类

⑥ 改进方法 (基本也能用)

使用包装类 (见前, 使其变引用)

(-127 ~ 128 不新建, 但之后会新创建)

eg: `Integer it = Integer.valueOf(100);`

事实上, 自动装箱:

`Integer it = 100;`

自动拆箱

`int i = it;`

地址: 成都市建设北路二段四号
高新区西源大道2006号

查询电话: 83201114

邮政编码: 610054 (沙河校区)

611731 (清水河校区)



电子科技大学

University of Electronic Science and Technology of China

- (包) 其他功能：
1. 把基本数据转成字符串 (没啥用)
 2. 把字符串转为基本 (很有用)

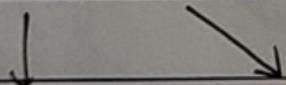
eg: int i1 = Integer.parseInt(str) / valueOf()

三. 集合进阶

整体：Collection : 单列集合

Map : 双列集合

Collection



List

ArrayList
LinkedList

Set

HashSet
LinkedHashSet
TreeSet

常用功能：(1) ~~add~~ add [boolean]

(2) clear() [void]

(3) size() [int]

(4) `remove() [boolean]`

(5) `isEmpty() [boolean]`

(6) `contains() [boolean]` 包含相

(7) `Object[] arr = list.toArray()`

接着用 `Arrays.toString()`

地址：成都市建设北路二段四号
高新区西源大道2006号

查询电话：83201114

邮政编码：610054（沙河校区）
611731（清水河校区）



电子科技大学

University of Electronic Science and Technology of China

合编历

① 迭代器

Iterator

方法：`hasNext(i)` 判断当前位置是否有元素

`next()` 获取当前位置元素并移到下一位

`list.iterator().next() - name.iterator()`

~~for = Iterator <--> int - numbers - iteration , ,~~

④ for 增强

格式: for(类型 变量名: 数组或集合) { ... }

[本质 迭代器简化]

⑤ Lambda

-> collection 中的方法 = ~~Lambda~~ -> ~~foreach~~

a. forEach (new Consumer<? super T> action)

{ 重写 action 方法即可 }

删数据 / 修改

① 有索引: (1) 倒着来 (2) 删完 i--.

② 无索引: 用迭代器 remove

ArrayList 与 LinkedList

ArrayList.

(1) 基于数组 细节请回温几遍

① 查询快

② 增删慢: 前移数据

地址: 成都市建设北路二段四号 查询电话: 83201114
高新区西源大道2006号

邮政编码: 610054 (沙河
611731 (清水)