java简易上手笔记

夏瑞鸿

2024 年 10 月 6 日

0.1 JAVA历史

- 种类有java se ee 以及 me
- se是标准版 ee为企业版，提供更多的api和库 me则是为了嵌入式的阉割版，这里我应该学习ee，为了做后端加大数据
- jre是java字节码的虚拟机 jdk 包含jre且提供编译器调试工具
- 规范有 jsr规范和jcp规范

0.2 从第一个代码开始吧

```
public class hello{
    public static void main(String[] args){
        System.out.println("hello world");
    }
}
```

开始解释

- hello为类名大小写敏感
- public表示类是公开的
- main方法是可执行代码块 main方法里有参数参数类型是字符串
- static表示公开青苔 void为返回的类型
- 类名必须和文件名一样才行
- java规定 public static void main(String[] args)是Java程序的固定入口

0.3 java程序基础

0.3.1 java程序的基本结构

基本单位是class 类名要求

- 类名必须是英文字母开头，后面接字母数字下划线
- 应该用大写字母开头，当然很明显我上面的实例就没有这样做，下次注意

对于我这种熟练c的来说，方法也就是函数，方法内的代码会依次顺序执行，当然了那么构造器呢，这个以后写thinking in java的博客的时候再说注释方法，和c一样

0.3.2 变量和数据类型

先来说变量

- 首先必须先定义再使用，定义的时候可以给一个初始值（注意c++不可以）
- 类型就是 byte short int long float double char boolean 特点csapp的笔记里写过了，不多赘述

然后是引用类型比如说string ，就比较像一个指针，指向这么个地址 **常量** 加个final 类似于const **var** 可以用来自动推断创造类型

```
var a=new StringBuilder()
```

接下来是变量的应用范围就是完全和c语言差不多就是那个块，不懂就去学c

0.3.3 整数运算与浮点数运算

会溢出 ¿¿是算数右移 ¿¿¿逻辑右移 **剩下的csapp笔记里面都有awa**

0.3.4 字符串

字符串连接类似于py Java字符串由于是引用类型，所以字符串不可变，他是一个指向（很明显这权限不如指针稍微说一下原理吧，当创建一个s，就是创建一个指针当你对s赋值的时候，jvm会创建一个字符串，这个字符串是你的值，然后s指向这个字符串所以当你改变赋值的时候，也不过是另外创建一个指向，而不是更改原有的字符串（多亏了有垃圾回收

0.3.5 数组类型

初始化为0 0.0 或者 false ，剩下的懂得都懂特点就是和字符串一样

0.4 流程控制

0.4.1 io

println 是 print line 输出并且换行很明显我们c代码手更喜欢print 首先需要知道**java.util,Scanner**这个包然后上代码

```
    import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input your name: ");
        String name = scanner.nextLine();
        System.out.print("Input your age: ");
        int age = scanner.nextInt();
        System.out.printf("Hi, %s, you are %d\n", name, age);
    }
}
```

我只能说不如cin cout 甚至不如scanf

0.4.2 各种语句判断

if()else 可嵌套 java这里判断还可以是判断引用类型是否相等 switch()case 1: break; case 2: break...; while(); do while(); for 和c一样 break continue 和c一样

0.4.3 多维数组

并没什么好说的，说实话，有些知识点在什么情况下都不会发生，谁写那种代码我就把他狗头打掉

0.4.4 命令行参数

java程序入口是main方法，main方法可以接收一个命令行参数，他是一个string【】数组这个命令行参数jvm从客户端接收并且传给main方法

0.5 oop

0.5.1 定义class

class 类名 xxxx 然后需要创建实例，必须用**new方法** 类名变量名= new 类名（）还记得上面的var吗，懂了吧这样一来，我们就创建了一个这个类的实例，并且用变量名指向如此我们可以变量名.字段进行操作了

0.5.2 方法

为什么在类里面又说一次因为爱情bushi 因为当类的成员field（以后统一写为field）为private的时候，需要用方法来赋值这是使用间接方法来修改的那么来写一下格式吧 **修饰符方法返回类型方法名（参数）方法语句；return 返回值**其实就是函数的样子

0.5.3 private方法

public方法就是公开那么private就是私人，不允许外部调用的类型当然，在内部是可以调用的，这里就不举例了

0.5.4 this的用法

- 用于区分成员变量和参数

```
public class a{
    private int ww;
    public aaa(int ww){
        this.ww=ww;
    }
}
```

  使用this指向当前对象

- 使用this调用其他构造函数注意如果参数的顺序类型相同的话，就会导致冲突
- 用this把当前对象当作参数传给函数（方法，后面就叫函数就叫函数就叫函数）
- 可以返回当前对象

0.5.5 杂碎知识点

可变参数使用类型。。。定义

```
class a{
    private String[] names;

    public void setname(string... names){
        this.names=names
    }
    }
```

喜欢给什么参数我就要什么构造方法可以直接使用构造器，用方法构造太过于麻烦，有更好用的构造器就不多说了，thinking java第六章会详细说明

0.5.6 方法重载

在一个类中，我们可以定义多个方法。如果有一系列方法，它们的功能都是类似的，只有参数有所不同，那么，可以把这一组方法名做成同名方法。简单的来说就是，通过相同的名字不同的参数来建立不同的函数。奇怪的Java特性，在方法名后面加上下划线加上对应的功能难道不可以更好的增加可读性吗

0.5.7 继承

java中使用关键字**extends**实现继承关键字

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

    public String getName() {...}
    public void setName(String name) {...}
    public int getAge() {...}
    public void setAge(int age) {...}
}
class Student extends Person {
    private int score;

    public int getScore() {}
    public void setScore(int score) { }
}
```

值得一提的是，**一定不要定义重复了** 稍微说一下继承树，在你定义没有继承的时候，那么编辑器就会自动加上一个叫做object的父类，除了object以外的任何类都有父类

继承有一个特点，子类无法访问父类的private方法所以请拜托使用protected吧这样也很隐私而且可以给子类用 **super** 当子类引用父类的filed的时候可以用super.加上名字，当然也可以用this或者正常使用那么什么时候是必须用的呢在java中任何class的构造方法，第一行必须调用父类的构造方法，编辑器会自己补上如果你不写的话。当你的父类并没有明确的参数，记住一定要super（相应的参数），否则无法正常去初始化父类，自然也就无法调用父类方法杂碎知识点：当你的引用变量类为子类的时候，这个变量同时也可以指向父类，当然了，测试和运维会打爆你的狗头持有的事情懒得说了

0.5.8 多态

也就是子类对父类的函数重写行业潜规则是在重写之前要先@Override那么什么是真正的多态呢多态的意思是，针对于某个方法的调用，真正执行的方法取决于运行其实际的函数

如果想要调用被复写的方法，可以通过super调用，调用方法你懂的如果不想自己写的父类被复写，那么也可以使用final不让复写如果一个类不想被继承也可以使用final修饰——真是自私啊

0.5.9 抽象类

简单来说就是只定义函数不写函数，这样可以用于继承也防止调用父类方法了

0.5.10 api

接口（interface）是一个定义了对象或类应具备的功能和行为的结构，但不包含具体的实现。在编程中，接口用于规定一组方法和属性，允许不同的类实现这些功能，从而实现多态性。

```
interface preson{
    void run();
    String lug();
}
```

如此就声明了一个接口，当一个具体的class实现api的时候需要**implements**
例如

```
class student implements person{
    xxxx
}
```

同时抽象也可以继承其他的抽象类也可以用default方法，不细讲

0.5.11 静态字段和静态方法

static 注意api里面的静态字段必须为final类型简单来说就是全局变量
懂吧

0.5.12 包

```
package hahah
public calss hahahahahah{

}
```

下面做出解释，先声明包名为hahah 然后再创建这个包里的类hahahahahahahah
调用的时候就hahah.hahahahahah

0.5.13 import

简单的来书哦就是import hahah*可以直接调了所有的class import static
hahah.hahahahahah;只搞这个类懂吧

0.5.14 作用域修饰符

public 可以被任何东西访问，最开放的一级 **private** 这玩意就是在类
内能用 **protected** 能给子类访问 class里面的class 这很无聊，只要你对块
的理解到位，相信一定能自己领悟的

0.6 java的核心类

0.6.1 字符串和编码

在Java中，String是一个引用类型，它本身也是一个class。但是，Java编译器对String有特殊处理，即可以直接用"..."来表示一个字符串：两个字符串的比较必须用equals() 为什么不用==因为指针地址之类的至于其他的呵呵，我从来不喜欢记住一些什么函数自然也不会写

0.6.2 StringBuilder

这是一个可变的string类当在StringBuilder中新增字符的时候，不会创建新的对象

```
StringBuilder sb = new StringBuilder(1024);
for (int i = 0; i < 1000; i++) {
    sb.append(',');
    sb.append(i);
}
String s = sb.toString();
```

喵的，第三方又或者是第一方的包有什么好讲的，等用得到再说吧、

0.7 异常处理

捕获异常使用try...catch语句，把可能发生异常的代码放到try ...中，然后使用catch捕获对应的Exception及其子类：可以使用多个catch语句，每个catch分别捕获对应的Exception及其子类。JVM在捕获到异常后，会从上到下匹配catch语句，匹配到某个catch后，执行catch代码块，然后不再继续匹配。简单地说就是：多个catch语句只有一个能被执行。无论发生或者不发生，都可以用finally在最后执行

0.7.1 抛出异常

throw new xxxxx（）