# 零散知识点

只能由表达弱的向表达强的转变

Java数字与布尔类型不能相互转换

类：

源文件有且只有一个与之同名的public类。

计算机中[]表示选写，即可写可不写

| 表示或者

Instanceof表示测试对象的类型

Object是所有类别的父类

Number是int，float，double的父类

Public所有都可以用

Private:只能在本类中使用

Protect:在子类中可以使用

Default:在本包中可以用

Java的三个特点：继承、封装、多态

运行一个编译 好的 Java 字节码程序，需要调用 Java 的解释器软件, 解释运行的三个阶段为 载入 、代码校验和 解释执行 。

# Searcch

使用search可以一键退出所有的循环至最外层，但是search：要写在循环最前端，但是会跳到后方（也可以将search改成任意词）

文本

描述已自动生成

# 循环语句

循环语句写完后要注意一下四点：

1.初始化

2.循环条件

3.循环边界

4.步近

# Switch case：

文本

描述已自动生成

Case：后米可以直接空加:

# 数组

文本

描述已自动生成

对数组进行等号赋值时类似于指针，都公用一块空间存储数组中的元素

public static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)

src:源数组;

srcPos:源数组要复制的起始位置;

dest:目的数组;

destPos:目的数组放置的起始位置;

length:复制的长度

array.sort():使得数组排序

# 二维不规则数组

图形用户界面, 文本, 应用程序

描述已自动生成

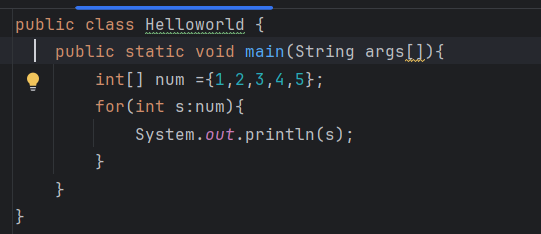
# Arraycopy

文本

描述已自动生成

Arraycopy中分别填写（源数组，开始复制位置，目的数组，目的数组开始存放位置，复制长度）

# For 语句增强



和python语法相似，s遍历数组总的每一个元素。

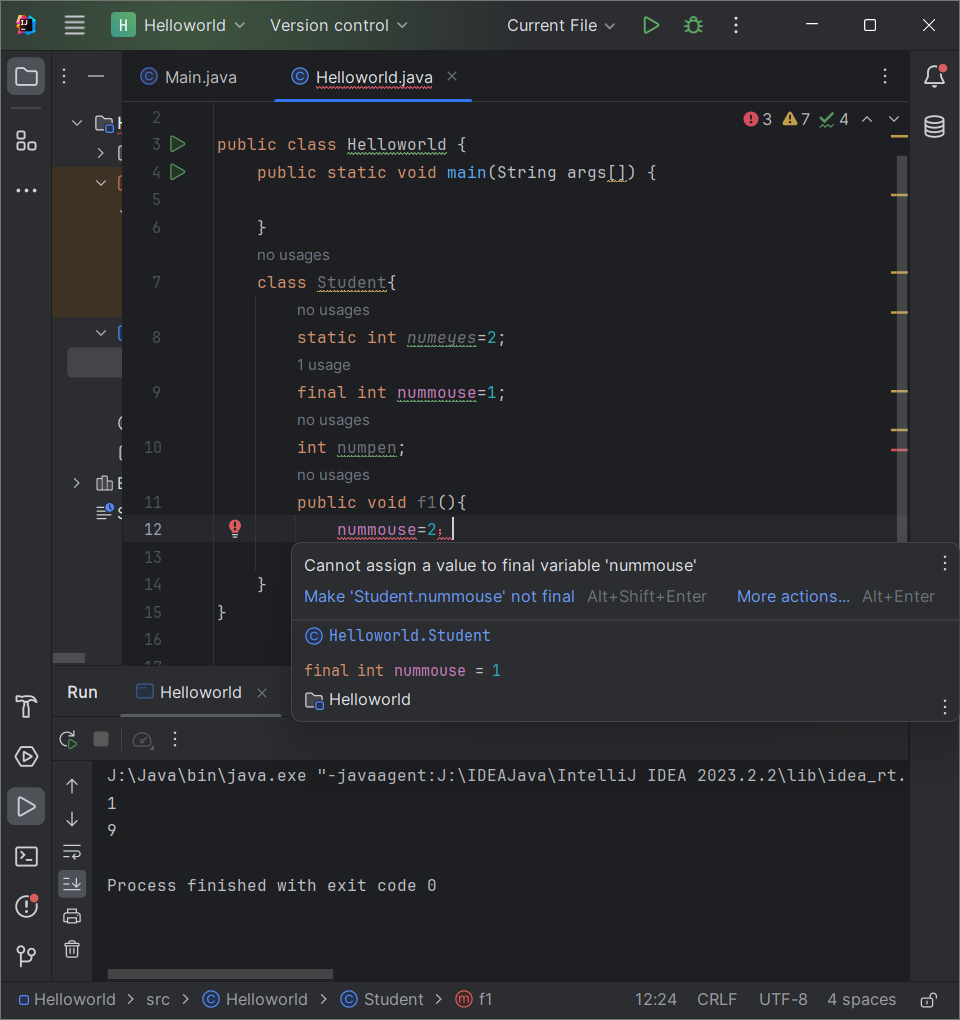
# 成员变量

Static：类属性

未加static：对象属性

Final：该变量不可变

|  |  |  |  |
| --- | --- | --- | --- |
| 变量名 | 解释 | 是否有默认 |  |
| 方法变量 | 写在方法里的变量 |  |  |
| 局部变量 | 写在类里的变量 | 有 |  |
| 全局变量 | Static修饰 |  |  |



Transient：定义暂时性变量（当存储整个文件信息时不会被存储）

Volatile：线程共享。

# 类的声明

Abstract：没定义方法（有方法名但是没有方法体）

Final：不可被继承

Extends：继承（单继承）

Implements：接口（可多个）

类定义是默认default

类的构造函数会自动调用父类的默认构造函数，如果父类定义了构造函数则会报错

需要使用super调用父类构造函数。

创建对象时会先定义一块空间，根据继承关系，自顶而下显示初始化，调用构造函数。

# 方法声明

Final：不可重写

Abstra：虚方法，无方法体。

# 作用域

文本

描述已自动生成

方法可以改变类的属性，但是不可以改变参数

# 方法参数可变

文本

描述已自动生成

可以是数组或者多个int

但不可以是int和数组都放进参数里。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 同一个类 | 同一个包 | 子类 | 全局 |
| Privat | √ |  |  |  |
| Default | √ | √ |  |  |
| Protected | √ | √ | √ |  |
| Public | √ | √ | √ | √ |

# 内部类

一般内部类可以使用外部类属性方法（public class）

局部内部类（class）

# 接口(interface）

接口中都是抽象方法，当类实现接口时要对接口中所有的函数全部重写

一个类可以定义多个接口但只可以继承一个父类但是接口不是类体系的一部分。

# 重载与重写

重载时参数列表不一样，返回值可以任意，因为再转为汇编语言时会将其参数也加入到函数名后，相当于再写了一个函数（又称为编译时多态）

重写是重写父类的函数，返回值是相同的（称为运行时多态）

# 静态变量与函数

**何时静态变量加载：**

如果一个类中定义了多个静态语句块，则这些语句块都按在类中出现次序 运行

# Final

类前加final，表示该类无法被继承

方法前加final，表示该方法无法重写

类的成员前+final，表示该成员值无法更改

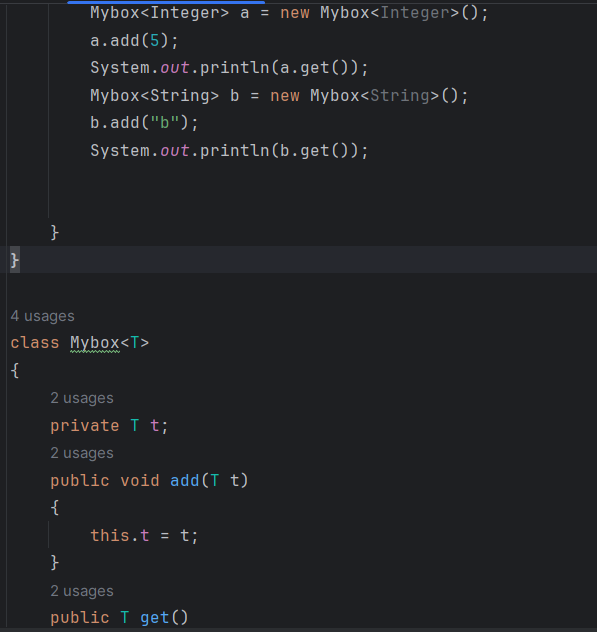
# 抽象类

在类前+abstract，表示方法不提供实现

# 包的定义与使用

文件开头的package pkg1是相对于src的路径，默认的路劲下的文件无法被引用，一般写文件的主程序在默认的文件夹

# 泛型



输出：图形用户界面, 应用程序

描述已自动生成

与c中teplete一样

<>中首字母必须大写。因为使用的不是举出数据类型而是类，

同时，可以使用一个父类作为类型参数调用泛型，而在后续对参数化类的访问中，使用该父类的子类对象。

# 输入

文本

描述已自动生成

# 输入输出IO流



代码如下：

public static void main(String[] args) {  
 OutputStream os = null;  
 InputStream is = null;  
 try {  
 is =new FileInputStream("J:\\桌面\\新建 文本文档.txt");  
 os = new FileOutputStream("J:\\桌面\\新建 文本文档3.txt");  
 byte[] buf = new byte[1024];  
 int length = -1;  
 while((length = is.read(buf) )!= -1)  
 {  
 os.write(buf,0,length);  
 }  
 System.*out*.println("代码拷贝成功！！！");  
 }catch (FileNotFoundException e)  
 {  
 System.*out*.println("没找到！！！");  
  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 is.close();  
 }catch (IOException a)  
 {  
 a.printStackTrace();  
 }  
 }