# 第一章 初始java

1. 程序：为了让计算机执行某些操作或解决某个问题而编写的一系列有序指令的集合。

Java SE

Java技术平台

Java EE

2. 开发java程序的步骤（使用记事本编写java源程序）

1 ）编写源程序

Java源程序文件使用.java作为扩展名

2 ）编译

3 ）运行

1. 安装JDK
2. 配置环境变量

Java 程序的结构

1. 编写程序框架

public class Hello world { }

1. 编写main方法的框架

Public static void main(String[] args){ }

（一个程序只能有一个main方法。）

1. 编写代码

System.out.println(“Hello World!!!”);

Java 程序的注释

1 单行注释 // 注释代码

2 多行注释 /\* 注释代码 \*/

Java 编码规范

* 类名必须使用public修饰
* 一行只写一条语句
* 用｛｝括起来的部分通常表示程序的某一层次结构。“｛”一般放在这一结构开始行的最末，“｝”与该结构的第一个字母对齐，并单独占一行。
* 低一层次的语句或注释应该比高一层次的语句或注释缩进若干个空格后在书写，是程序更加清晰，增加程序的可读性。

Java程序开发软件——MyEclipse平台

1. 安装MyEclipse
2. 配置环境变量
3. 在MyEclipse中，择“windows” “prefernces” 选项，打开“prefernces” 窗口，展开prefernces窗口左侧“java”节点，选择其下的“installed JREs”子节点。
4. 在左侧窗口中单击“Add”按钮，进入“Add JRE”窗口。在“浏览文件夹”对话框中，选择JDK1.7安装目录，单击“确定”按钮，成功导入JDK1.7。
5. 选中JDK1.7前的复选框，设置为默认版本，单击“Add JRE”窗口下方的”finish”按钮，返回“prefernces” 窗口。
6. 展开“prefernces” 窗口左侧“java”节点的“compiler”子节点，在窗口右侧“compiler compliance level”下拉列表框中选择JDK 1.7 的版本，完成环境变量。

使用myeclipse开发java程序的步骤

1. 创建一个java项目
2. 创建并编写java源程序
3. 编译java源程序
4. 运行java程序

Java项目组织结构

1. 包资源管理器
2. 导航器

常见错误

1. 类名不可以随便命名

Public修饰的类的名称必须与JAVA文件同名

1. Void不可少

main（）方法中的void不可少

3.Java对英文字母的大小写敏感

4.“；”是必须的

5.“””是必须的

第二章 变量、数据类型和运算符

变量：（可以发生变化的量）变量是存储数据的一个基本单元，不同的变量相互独立。

数据类型

Java常用的数据类型

|  |  |  |
| --- | --- | --- |
| 数据类型 | 说明 | 举例 |
| int | 整型 | 用于存储整数， |
| double | 双精度浮点型 | 用于存储带有小数的数字 |
| char | 字符 | 用于存储单个字符 |
| String | 字符串 | 用于存储一串字符 |

变量的申明及使用

申明变量

1.根据数据类型在内存中申请的空间，即给变量命名

语法：数据类型 变量名；

1. 给变量赋值；即“将数据存储至对应的内存空间”。

语法 ：变量名=值；

1. 调用变量，使用存储的变量，称为调用变量。

变量的命名规则

1. 变量必须以字母、下划线“\_”或“$”符号开头。
2. 变量可以包括数字，但不能一数字开头。
3. 除了“\_”或“$”符号以外，变量名不能包含任何特殊字母
4. 不能使用java语言的关键字

注意：1.变量未赋值先使用

1. 使用非法的变量名
2. 变量不能重名

赋值运算符“=”

算数运算符：“+”“-”“\*”“/”“%”

从控制台输入

控制语句：Scanner input=new Scanner(System.in);

Int a=input.nextInt();

Double a=input.nextDouble();

String a=input.next();

自动类型转换

将一种类型的变量赋给另一种类型的变量时，发生的转换

数据类型转换

强制类型转换：满足条件1.两种类型要兼容

2.目标类型大于源类型

语法 double d=3.14;

Int b=(int)d;

Boolean类型的值：

|  |  |
| --- | --- |
| 值 | 说明 |
| true | 真 |
| false | 假 |

关系运算符 ：

|  |  |  |
| --- | --- | --- |
| 关系运算符 | 说明 | 举例 |
| > | 大于 | 99>100,结果为false |
| < | 小于 | 大象的寿命<乌龟的寿命，结果为true |
| >= | 大于等于 | 你的考试成绩>=200分，结果为false |
| <= | 小于等于 | 每次的考试成绩<=60分，结果为false |
| == | 等于 | 地球的大小==篮球的大小，结果为false |
| != | 不等于 | 水的密度！=铁的密度，结果为true |

注意：“=”’和“==”的区别

在计算机里面，第一个表示赋值，第二个表示相等

第三章 选择结构（一）

If选择结构：根据条件判断之后再做处理的一种语法结构。

1.if结构语法

语法：if（条件）｛

代码块

｝

2.if-else结构

语法：if（条件）｛

代码块1

｝else｛

代码块2

｝

3.if-else if-else结构

语法：if（条件1）｛

代码块1

｝else if（条件2）｛

代码块2

｝else｛

代码块3

｝

嵌套if选择结构

语法：if（条件1）｛

If（条件2）｛

代码块1

｝else｛

代码块2

｝

｝else｛

代码块3

｝

逻辑运算符：

|  |  |  |
| --- | --- | --- |
| 说明符 | 表达式 | 说明 |
| && | 条件1&&条件2 | 两个条件同时为真，则结果为真，两个条件有一个为假，则结果为假 |
| ｜｜ | 条件1｜｜条件2 | 两个条件有一个为真，则结果为真，两个条件同时为假，则结果为假 |
| ！ | ！条件 | 条件为真时，结果为假。条件为假时，结果为真 |

随机数的产生：

Int random=(int)(math.random()\*10);

第四章 选择结构（二）

switch循环

switch的关键字可以是：

Int

short

byte

char

String

枚举类型

case后可以是

int

short

byte

char

String

枚举

switch语法结构

switch（表达式）｛

case常量1：

//代码块1：

break；

case常量2：

//代码块2:

case常量3:

//代码块3：

.......

default;

//代码块n：

break；

｝

第五章 循环结构（一）

循环：循环就是重复的做。

循环的三种方式：

1. while循环
2. do——while循环
3. for循环

while循环步骤：

1. 声明并初始化变量
2. 判断循环条件是否满足，如果满足则执行循环操作；否则退出。
3. 执行完循环操作后，再次判断循环条件，决定继续执行循环或退出循环。

程序调试：

1. 分析错误，设置断点。
2. 启动调试，单步执行。

do--while循环

语法

do｛

//循环操作

｝while（循环条件）

do--while循环

do--while循环步骤：

1. 分析循环条件和循环操作；
2. 套用do--while语法写出代码
3. 检查循环能否退出

第六章 循环结构(二)

for循环的语法：  
for（表达式1；表达式2；表达式3）｛

//循环体

｝

* 初始部分：设置循环的初始状态
* 循环体：重复执行的代码
* 迭代部分：下一次循环开始要执行的部分，在while循环结构中它作为循环体的一部分
* 循环条件：判断是否继续循环的条件

for循环结构执行的顺序：

1. 执行初始部分（int i=0；）
2. 进行循环条件判断（i<100;）
3. 根据循环条件判断结果
4. 执行迭代部分，改变循环变量值（i++）
5. 依次重复步骤（2）~步骤（4），直到退出for循环结构

在循环中可以使用break和continue语句控制程序的流程

1. break语句用于终止某个循环，程序跳转到循环体外的下一条语句。
2. continue语句用于跳出本次循环，进入下一次循环。

第八章 数 组

数组：数组就是一个变量，用于将相同数据类型的数据存储在内存中，数组中的每一个数据元素都属于同一类型。

数组的基本要素：

* 标识符：数组名称
* 数组元素：即数组中所存放的数据
* 数组下标：每个数据所对应的位置
* 元素类型：存储在数组中的数组元素应该是同一数据类型

使用数组的步骤：

1. 声明数组：

语法：

数据类型[] 数组名；

数据类型 数组名[]

1. 分配空间：

语法：

数组名=new 数据类型[数组长度]；

数据类型[] 数组名=new 数据类型[数组长度]；

1. 赋值：

语法：

数组名[下标值]；

数据类型[] 数组名=｛值1，值2，。。。。。。,值n｝；

1. 对数据进行处理：

数组应用： 数组升序：Arrays.sort(数组名)；

数组求最大值：

If(scores[i]>max){

Max=scores[i];

}

向数组中插入元素：p161

第九章 循环结构进阶

二重循环：一个循环体内包含另一个完整的循环结构

语法：

while与while嵌套

while（循环条件1）｛

循环操作1

while（循环条件2）｛

循环操作2；

｝

｝

do-while与do-while嵌套：

do｛

循环操作1

do｛

循环操作2

｝while（循环条件2）；

｝while（循环条件1）；

for与for循环嵌套：

for（循环条件1）｛

循环操作1

for（循环条件2）｛

循环操作2

｝

｝

while与for循环嵌套：

while(循环条件1)｛

循环操作1

for（循环条件2）｛

循环操作2

｝

｝

注意：在二重循环中，外层循环变量变化一次，内层循环变量要从初始值到结束值变化一遍。

continue和break语句的对比：

当continue和break语句用在内层循环时，只会影响内层循环的执行，对外层循环没有影响。

continue语句是跳出本次循环，进行下一次循环。二break是跳出本层循环，即提前结束本层循环，执行循环下面的语句。

第十一章 类和对象

对象：客观存在的事物就称为对象，在java的世界中“万物皆对象”。

对象是用来描述客观事物的一个实体。

属性

对象

方法

封装：就是把一个事物包装起开，并尽可能隐藏内部细节。

类定义了对象将会拥有的特征（属性）和行为（方法）。

类和对象的关系：类是多个对象进行综合抽象的结果，是实体对象的概念模型，而一个对象是一个类的实例。

在面向对象设计中，类是程序的基本单元。

java的类模板

语法：

public class <类名>｛

属性1的类型 属性1；

属性2的类型 属性2；

......

属性n的类型 属性n；

方法1；

方法2；

......

方法n；

｝

类的命名规则：

* 不能使用java中的关键字，
* 不能包含任何嵌如的空格或点号“.”以及除下划线“\_”字符“**$**”外的特殊字符。
* 不以数字开头。
* 类名通常由多个单词组成，每个单词的首字母大写。

1. 定义类名

public class 类名{

}

1. 编写类的属性
2. 编写类的方法

访问修饰符 返回值类型 方法名（）{

方法体

}

创建和使用对象：

类名 对象名=new 类名（）；

对象名.属性 //引用对象的属性

对象名.方法名（） //引用对象的方法

Java数据类型的默认值

|  |  |
| --- | --- |
| int | 0 |
| double | 0.0 |
| boolean | false |
| string | null |
| char | ‘\u0000’ |

面向对象的优点：

1. 与人类的思维习惯一致
2. 信息隐藏，提高了程序的可维护性和安全性
3. 提高了程序的可重用性

第十二章 类的无参方法

类是由一组具有相同属性和共同行为的实体抽象而来的。

类的方法必须包括三个部分：

1. 方法的名称
2. 方法的返回值类型
3. 方法的主体

编写方法时分两步完成

1. 定义方法名和返回值类型
2. 在｛｝中编写方法的主体部分

编写方法时注意一下三点：

1. 方法体放在一对｛｝中，方法体就是一段程序代码，完成一定的工作
2. 方法名主要在调用这个方法时使用，在java中一般采用骆驼时命名法
3. 方法执行后可能会返回一个结果，该结果的类型称为返回值类型，使用return语句返回值

return语句是跳转语句的一种，主要做两件事：

1. 跳出方法
2. 跳出结果

方法调用：

语法：对象名.方法名（）；

常见错误：

1. 方法的返回值类型为void时，方法中不能有return语句
2. 方法不能返回多个值
3. 多个方法不能嵌套定义
4. 不能在程序外直接写逻辑代码

成员变量和局部变量的区别：

1. 作用域不同：成员时大家的，都能用，局部时自己的，只能自己用
2. 初始值不同：初始值给成员，局部是值，不是初始值

注意作用域：

1. 在同一个方法中，不允许有同名的局部变量，在不同的方法中可以有同名的变量
2. 在同一类中，成员和局部变量可以重名，使用时优先使用局部变量

JavaDoc注释以“/\*\*”开头，以“\*/”结尾并且提供了JavaDoc注释标签，使用JavaDoc

技术可以生成开发文档

第十四章 带参数的方法

定义带参方法：

语法：<访问修饰符>返回值类型<方法名>(<带参列表>)｛

//方法的主体

｝

调用带参方法：

语法：对象名.方法名（参数1，参数2，......参数n）；

调用方法时，需要注意两点：

1. 先实例化对象，在调用方法
2. 实参的类型、数量、顺序都要与形参一一对应

包名：package net.javagroup.research.powerproject;

research:部门名 powerpoject:项目名

第十五章 字符串

使用字符串：

1. 定义并初始化字符串
2. 使用字符串，对字符串进行一些处理

字符串的长度：字符串1.length（）；

字符串的比较：

字符串1.equals（字符串2）；

说明：“==”判断是否是同一字符串，“equals”判断两个字符串的值是否相等

字符串的连接：字符串1.concat（字符串2）；

字符串的拆分：字符串1.split（String separator，int limit）；

StringBuffer类

1. 声明StringBuffer对象并初始化
2. 使用StringBuffer对象

String类操作字符串的方法

* 获得字符串的长度：length（）。
* 比较字符串：equals（）。
* 连接字符串：concat（）。
* 提取字符串：indexOf（）。
* 拆分字符串：split（）。

常用的StringBuffer类的方法：

* 拆分成String类型：toString（）。
* 连接字符串：append（）。
* 插入字符串：insert（）。