1. 环境变量

Path :（目的）在电脑任何地方可以执行javac和java指令

在path中加入如下配置：D:\Java\jdk1.8.0\_152\bin

因为path属于系统的变量，修改错误的话，会导致系统出bug。

所以引入JAVA\_HOME，来保护path。

JAVA\_HOME: D:\Java\jdk1.8.0\_152(Finally)

将path修改为 %JAVA\_HOME%\bin(Finally)

1. 第一个程序；

Public class HelloWorld{

Public static void main(String[] args){

System.out.println(“Hello World!”);

}

}

1. 系统输出语句；

System.out.println(“Hello World!”);

4、**变量**

为什么要有变量；-- 计算 数据（0~9）、字符|字符串、布尔值、

数据类型：《重点》

①自动类型转换：容量小的数据类型自动转换为容量大的数据类型。

short s = 12;

int i = s + 2;

注意：byte short char之间做运算，结果为int型！

②强制类型转换：是①的逆过程。使用“（）”实现强转。

人脑：计算区域、存储区域

电脑：CPU、内存

变量的定义：存储计算后的结果数据（中间数据）

有效范围、有效期：作用域；

输入的数据，也是映射到大脑皮层；

变量：在内存中开僻出的一个空间（区域），用来 存储、暂时的；该区域可以重复存放不同数据；

数据类型 变量名 ；//申明变量；

变量名 = 数据 ；//赋值语句；赋值符号；

**变量命名规范：**

* 命名的规则：（一定要遵守，不遵守就会报编译的错误）
* 由26个英文字母大小写，0-9 ，\_或 $ 组成
* 数字不可以开头。
* 不可以使用关键字和保留字，但能包含关键字和保留字。
* Java中严格区分大小写，长度无限制。
* 标识符不能包含空格。

1. 运算符

算术运算符： + - \* / %（取余|取模） ；

%：被除数 % 除数 = 商………余数

%求出来的就是这里的余数。

比较运算符|关系运算符： > 、<、 ==、>=、 <=、 !=；

比较运算符得到的结果只可能是boolean值。（true或者是false）.

逻辑运算符： &&(and) 、||(or)、 !(not)；

**优先级：**

|  |  |
| --- | --- |
| 高 低 | **() {}** |
| **++ -- ！** |
| **\* / %** |
| **+ -** |
| **< > <= >=** |
| **== !=** |
| **&&** |
| **||** |
| **+= -= /= %= += -=** |

1. 条件控制语句

格式：

①if(条件表达式){ }

②if(条件表达式){

//执行的语句1；

}else{

//执行的语句2；

}

如果 （条件满足） {

做什么事件//执行语句；代码块

}

否则{

做其它事件

}

Example1：

Int i =scanner.nextInt();

If(i < 5){

}else{

// i>=5

}

* 多重条件控制语句：

If (条件表达式1){

}else if(条件表达式2){

}else{

}

~~If (条件表达式1){~~

~~}~~

~~if(！条件表达式1 && 条件表达式2){~~

~~}~~

这种方式：代码复杂；计算运行效率低；

* 嵌套语句

条件表达式的代码块中（大括号中），再放一个条件表达式；

If(条件表达式1){

If(条件表达式2){

}

}

【注意】

* 一旦满足某个条件表达式，则进入其执行语句，执行完毕，跳出当前的条件判断，不会执行其以下的条件结构语句。
* 如果多个条件表达式之间为“互斥”关系，多个结构可以上下调换顺序。
* Switch语句

Switch(待比较的表达式)

只能做等值比较；

Case 关键字；

Case 比较值 **：{**

**}**

Break 关键字；

中断、跳出当前switch语句；

Default 关键字；

【注意】

* 待比较的表达式的数据类型只能为：byte short int char enum String
* Case 后只能写值，不能写范围。

特殊情况：当多个case语句处理的执行代码一致时，可以统一来书写；

* default 是可以省略的。并且其位置也是灵活的，但是一般将其放在case语句之后；
* 一般满足某个case语句，则进入该执行代码。直到遇到break或者程序结束；
* 若要判断的语句的条件满足switch变量的数据类型，且值不多的情况下，建议选择switch-case、除此之外，选择if-else。

1. 循环语句；

* While循环语句
* 初始化语句、比较语句、变量叠代；

While(比较语句){

//循环体；需要重复执行的代码；

//添加退出机制（步长）

//或者用break退出;

}

* Do While

//先执行一次，再判断是否继续循环

// while: 先判断是否继续循环，满足条件才执行

Dowhile最后的分号“；”不要漏掉；

**循环语句，也可以嵌套；**

* For循环

把数据初始化、比较语句、叠代语句，放在一起；

For(数据初始化 **;** 比较语句 **;** 叠代语句){

//循环的内容

}

【注意】

了解循环的执行流程。

Do-while

While

For

三个循环的区别，以及过程。

Break：中断或者退出循环体;

Continue：本次循环中，关键字后面的不执行，直接转到下次循环;

**Break 和continue，只在当前循环中有效；**

* **数组**

方便管理大量数据（变量）

数据类型 [] 数组名；//申明；

两种初始化

**基本初始化**

* New 数据类型 [数组长度]

数组中，没有任何数据值，只默认值 ；

带数据的初始化

* New 数据类型[]{数据1，数据2}；

此种情况，不用指定长度，使用后面数据个数作为长度；

给数组赋值

**数组名[索引|下标] = 数据**

动车001 ，第8号车厢；

使用数组：

**数组名[索引|下标]**

***二维数组***

人与人交流：自然的语言

人与机交流：程序语言；

计算机内部：计算机语言；01