1、环境变量

2、第一个程序；

3、系统输出语句；

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》

人脑：计算区域、存储区域

电脑：CPU、内存

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；

整数类型（byte、short、int、long）

整数型

浮点类型(float、double)

基本数据类型

字符型

布尔型

|  |  |  |  |
| --- | --- | --- | --- |
| 数据类型 | 内存空间(8位等于1字节) | 取值范围 | 举例 |
| byte | 8位 | -128~127 | byte a=1 |
| short | 16位 | -32768~32767 | short a=1 |
| int | 32位 | -2^31~2^31-1 | int a=1 |
| long | 64位 | -2^63~2^63-1 | long a=1L |
| float | 32位 |  | float a=3.01F |
| double | 64位 |  | double a=3.1D |
| char | 16位 |  | char st=’a’ |
| boolean | 1位 |  | boolean b=ture |

**变量命名规范**

变量命名只能以字母、下划线、美元符号、数字组成，第一个字符不能为数字，不能以关键字命名变量

变量命名一般以驼峰式命名

类名：每个单词的首字母都大写

变量名：首单词字母小写后面单词首字母大写

函数名：首单词字母小写后面单词首字母大写

常量：所有字母都大写

类型转换：

自动转换：低的优先级到高的优先级是自动转换；

强制转换：高的优先级到低的优先级就需要强制转换；

1. 运算符

算术运算符： + - \* / %（取余|取模） ；

比较运算符|关系运算符： > 、<、 ==、>=、 <=、 !=；

逻辑运算符： &&(and) 、||(or)、 !(not)；

**优先级：**

|  |  |  |
| --- | --- | --- |
| 优先级 | 描述 | 运算符 |
| 1 | 括号 | () |
| 2 | 正负号 | +、- |
| 3 | 一元运算符 | ++、--、！ |
| 4 | 乘除 | \*、/、% |
| 5 | 加减 | +、- |
| 6 | 移位运算 | >>、>>>、<< |
| 7 | 比较大小 | <、>、>=、<= |
| 8 | 比较是否相等 | ==、!= |
| 9 | 按位于运算 | & |
| 10 | 按位异或运算 | ^ |
| 11 | 按位或运算 | | |
| 12 | 逻辑与运算 | && |
| 13 | 逻辑或运算 | || |
| 14 | 三元运算符 | ?: |
| 15 | 赋值运算符 | = |

**++a/--a:表示在使用变量a之前，先使a的值加（减）1**

**a++/a--:表示在使用变量a之后，使a的值加（减）1**

1. 条件控制语句

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

}

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){

}

}

* Switch语句

Switch(待比较的表达式):括号里面的值的类型只能是byte、short、int、char、string

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；

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

}

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

一维数组实质上是一组相同类型数据的线性集合

一维数组和二维数组一样，在声明时没有分配内存空间，只有用new关键字来分配内存，然后才可以访问每个元素

**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***

条件

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