# 第一章初识JAVA

## JAVA环境搭建

下载JDK：JDK是用于开发java应用程序的开发包，它提供了编译、运行java程序所需的各种工具和资源。

配置环境变量：我的电脑—属性—高级系统设置—环境变量—（win10）找到path点击编辑再新建一个位置，将安装的JDK下的bin目录粘贴此处—在环境变量处新建java\_home，将JDK安装目录粘贴。

检查环境变量是否配置成功。Win+r键 打开命令行，输入cmd——再输入java -version 查看是否提示JDK版本号。也可以再输入java命令验证。

## JAVA执行过程

编写一个java文件 → 编译器 → 生成.class后缀的字节码文件 → JVM

# 第二章变量

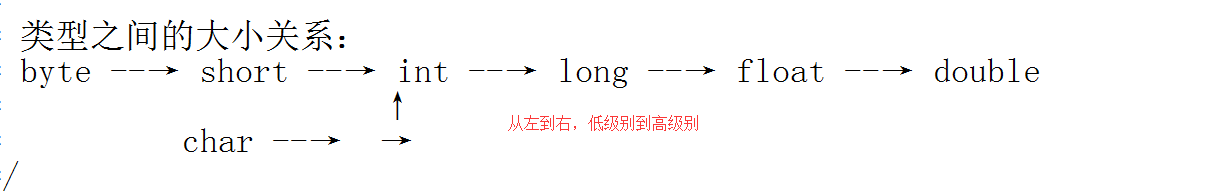
## 变量的命名法则和变量类型

1. 由字母、数字、下划线、$组成，其中首字符不能为数字
2. 区分大小写，没有长度限制
3. 不能与关键字、true、false、null重复
4. 建议用驼峰命名法命名，见名知意
5. 可以用中文命名，但不建议

## 2:java中的基本数据类型取值范围

|  |  |  |  |
| --- | --- | --- | --- |
| 基本类型 | 大小 | 实例 | 取值范围 |
| Boolean | 1字节8位 | true | true,false |
| byte | 1字节8位有符号整数 | -12 | -128~+127 |
| short | 2字节16位有符号整数 | 100 | -32768~+32767 |
| int | 4字节32位有符号整数 | 12 | -2^31~+2^31-1 |
| long | 8字节64位有符号整数 | 10000 | -2^63~+2^63-1 |
| char | 2字节16位Unicode字符 | ‘a’ | 0~65535 |
| float | 4字节32位浮点数 | 3.4F | -3.4E38~3.4E38 |
| double | 8字节64位浮点数 | -2.4e3D | -1.7E308~1.7E308 |

## 3：基本类型转换



1、自动类型转换:将低级别的类型赋值给高级别类型时，将进行自动类型转换

2、强制类型转换：高级别到低级别，需要增加强制转换符

## 4.注释

//：单行注释

/\*

\*

\*/:多行注释

/\*\*

\*

\*/：文档注释

## 5.算术运算符

+、-、\*、/、%(取模)、++、--

## 6.关系运算符

==、！=、<、>、<=、>=、，返回布尔值

## 7.逻辑运算符

|| & ！

## **8.条件运算符**

又称三目运算符/三元运算符

格式: 条件?表达式1:表达式2

条件为true则取表达式1的值，条件为false则取表达式2的值。

# 第三章：选择结构

## 单分支if语句

if(条件){

代码块;

}

## 双分支if-else语句

If(条件){

代码块1;

}else{

代码块2;

}

## 多分支语句

If(条件1){

代码块1;

}else if(条件2){

代码块2;

}else if(条件3){

代码块3;

}

...

else{

代码块n;

}

## switch语句

switch(表达式){

case 常量1:

代码块

break;(非必须)

case 常量2:

代码块

break;(非必须)

...

default:(非必须)

代码块

break;

}

default不一定在最后,表达式的值可为byte、short、int、char、String、枚举型.

# 第四章：循环结构

## while循环语句

while(循环条件){

循环体

}

## do-while循环语句

do{

循环体

}while(循环条件)

## for循环语句

for(表达式1;表达式2;表达式3){

循环体

}

循环次数确定，通常用for

不确定，通常用while、do-while

**break**

break的使用

break语句只能出现在switch和循环语句中。 表示结束switch，或循环

**continue**

continue的作用：跳过本次循环，而进行下一次循环 。

continue只能出现在循环语句中。

### return

结束当前的方法

# 第五章：数组

## 申明一维数组：

1. Int [ ] a=new int [ 5 ];
2. Int [ ] a ;

a=new int[5];

1. Int [ ] a={1,2,3};
2. Int [ ]a=new int[]{1,2,3};

## Arrays类的应用：

Arrays.equals(数组1，数组2）将两个数组进行比较，返回一个boolean类型；

Arrays.sort(数组） 将数组的元素进行升序排列；

Arrays.toString(数组) 将数组转化为String字符串；

Arrays.fill(数组，5） 将数组的每个元素赋值为5；

Arrays.copyOf(被复制的数组，新数组的长度) 复制一个数组

Arrays.binarySearch(数组，元素)查找元素的下标返回一个int类型

## 声明二维数组：

Int [ ][ ] a=new int[3][2];

Int [] [] a=new int[3][ ];

Int a[0][]=new int[0][2];

Int a[1][]=new int[0][3];

Int a[2][]=new int[0][4];

Int a[][]=new int{{1,2},{3,4,5},{6,7}};

Int a[][]={{1,2},{3,4,5},{6,7}};