# Day01

JDK官网 http://www.oracle.com（javac把.java文件编译为.class文件，java运行文件）

配置环境变量（bin）

class（定义一个类）

字面量类型（整数，小数，字符串，字符，布尔型，空类型）

制表符'\t'把字符串补齐8位

数据类型 变量名=数据值；(变量请直接赋值)

二进制（0b）八进制（0）十进制（null）十六进制（0x）

任意进制转十进制（公式：系数\*基数的权次幂 相加）

byte的取值范围（-128~127）

标识符：类，方法，变量的名称；不能以数字开头；由字母，数字，-,$组成；

键盘录入：Scanner (ScannerDomeOne)

(

import java.until.Scanner;

Scanner sc=new Scanner(System.in);

int name=sc.nextIn();

)

IDEA(下载官网：https://www.jetbrains.com/idea/)

IDEA四层结构（project 项目；module 模块；package 包；class 类；）

小数参与运算可能丢失精度

算术运算符：=，-，乘，/，%；

小自动升大，大强制缩小。

byte，short，char升为int；

强制转换类型：目标数据类型 变量名=（目标数据类型）被强制转换变量名；

例：int a=（int）b；（可能精度丢失）

字符串只有拼接操作；

自增自减：先用后加（b=a++），先加后用(b=++a);

加等减等：加后赋值（a+=b），赋值后加（a=+b）；

&与|或！非，^异或；

&&短路与（都真才真），||短路或（有真则真）；

# **Day02**

**三元运算符：**

格式：关系表达式？ 表达式1：表达式2

解释：关系表达式为true运行表达式1，false运行表达式2。

# Day 03

原码，反码，补码（单位：比特byte，8比特=1字节）；

原码:-128（补码：1000 0000）~127

反码：正数的反码不变；

负数进行取反，符号位不变；

补码：负数的反码+1；解决+0和-0的问题，正数和负数的运算，计算机的存储和计算；

Byte：1

Short：2

Int：4

Long：8；

&逻辑与：都为1才为true

|逻辑或：有1就为1；

<<左移：一位是\*2

>>右移：一位/2.

# Day04 流程控制语句

### 顺序结构

先后顺序，依次执行。

### 分支结构

If语句：多条件判断（范围判断）

if（关系表达式1）{

··· 语句体1；

}

Else if（关系表达式2）{

··· 语句体2；

}…

else{

语句体2

}

Switch语句

格式：有限个数据列举出，任选其一

Switch（表达式）{

Case 值1：

语句体1；

Break；

Case 值2：

语句体2；

Break；

。。。

Default：（default可以省略，可以写在任何位置）

语句体n+1；

Break；

}

Case穿透：break不能省略（本次循环结束）

使用场景：如果多个case重复了，可以用case简化。

Eg：switch（表达式）{

Case 值1->语句体；

Case 值1->语句体；

Case 值1->语句体；

Default->语句体；

}

循环结构·

变量的作用范围，只在所属的·大括号里面有效

如果我们把变量定义在循环里面，那么当前的变量只能在本次循环有效

第二次循环开始的时候就会重新定义了变量

##### For循环：

格式：

for（初始化语句；条件判断语句；条件控制语句）{

循环语句

}

执行流程：

1. 执行初始化语句。
2. 执行条件判断语句，看其结果是true还是false

如果是false，循环结束。

如果是true，执行循环语句题。

3．执行条件控制语句

4.回到2继续执行判断语句

##### While循环

格式：

初始化语句；

While（条件判断语句）{

循环体语句；

条件控制语句；

}

For和while对比：

语法上无区别，知道范围用for，不知道范围用while

##### Do。。。while循环

格式

Do{

循环体语句；

条件控制语句；

}while（条件判断语句）；

先执行再判断。