java笔记

# day0

program files 默认为64位安装目录

program files（x86）默认为32位安装目录

dos命令 黑窗口

1.mkdir abc 创建目录

cls 清屏

dir 列出当前路径的目录

exit 退出dos命令窗口



2.切换盘符 直接输入盘符回车 例如： D:回车

3.切换目录 （只能改变当前盘符下的目录）使用cd命令来完成目录的切换 cd change directory 更换目录

cd ..回到上级目录

命令格式 cd 路径

路径 相对路径、绝对路径

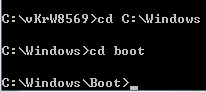
在Windows操作系统的起点是盘符的 为绝对路径

相对路径一定是相对于当前所在“位置”而言的

C:\Users\LENOVO\.uuid

D:\AutoCAD作业

切换相对路径 相对路径切换是相对于当前路径为起点往后找



cd ..回到当前路径上级目录

cd \直接回到根目录

# Day1

dos 常见命令

del 删除文件

del \*.后缀名模糊匹配删除多个文件

ipconfig /all可以查看更详细的网络信息

如何查看两台电脑怎样正常通信

使用ping ip地址

http://

当光标在行尾选中一行shift+home选中一行

当光标在行首选中一行shift+end选中一行

Ctrl+home回到文件头

Ctrl+end回到文件尾

JVM（c++语言写的一个虚拟机）、GC（垃圾回收机制）

Jdk、jre、jvm三者之间的关系

Jdk：java开发工具

Jre：java运行环境

Jvm：java虚拟机

Jdk包括jre，jre包括jvm

只运行软件，安装jre即可

要开发就需要安装jdk

.java文件就是源文件，这个文件中编写源代码

.class 文件就是字节码文件，这个文件是编译代码而得到的

注意：1个java源文件是可以编译生成多个class文件的

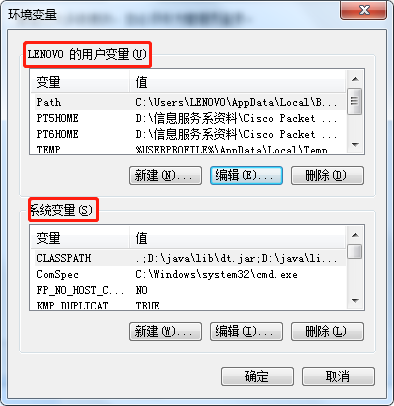
最终运行的是class文件

xxx．Java源文件经过编译之后生成的A.class、B.class、C.class等文件

称A是一个类、B是一个类、C是一个类，其中A、B、C是类的名字。

配置环境变量

环境变量包括“系统变量”“用户变量”



如何查看编译器版本及java虚拟机版本

C:\Users\LENOVO>javac -version

javac 1.8.0\_202

C:\Users\LENOVO>java -version

java version "1.8.0\_202"

Java(TM) SE Runtime Environment (build 1.8.0\_202-b08)

Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

Dos命令窗口，该软件 命令找不到错误了， 配置一下环境变量path

Javac命令怎么用？

语法格式：javac java源文件路径

神操作：javac 把源文件直接拖进来即可编译

在dos命令窗口中快速定位到某个路径 找到想要的路径，在地址栏直接输入cmd即可定位

运行阶段

class文件 使用java命令运行

语法格式 java类名 例如：helloword.class java hellworld

Java运行步骤：

1. 先使用cd命令切换到class文件所在的路径
2. 执行java 类名

Java命令下只能跟类名

Java的jdk所属环境变量，有一个叫做JAVA\_HOME

需要JAVAWEB的时候安装Tomcat服务器，这时才需配置JAVA\_HOME环境变量

# Day2

类加载器（classloader）

Java运行后会启动jvm虚拟机，调用类加载器默认从当前路径寻找class文件

类加载器可以更改默认寻找路径，但必须先设置一个环境变量：classpath

Classpath环境不属于windows操作系统，是隶属于java

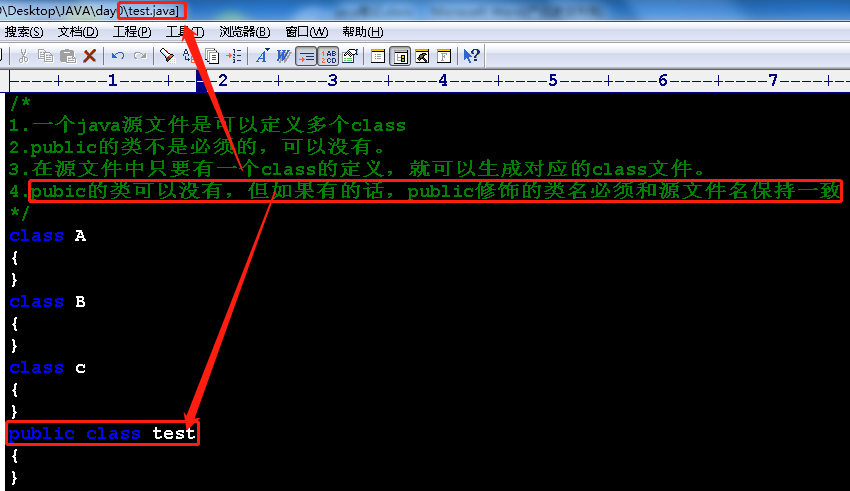
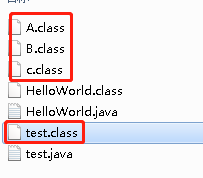
Classpath环境变量是java特有的

Classpath=A路径;B路径;C路径

Classpath配置之后只会在配置后的路径去指路，不再从当前路径去找类了

Java程序解释:

1. 在java中任何有效的代码必须写到”类体当中”，最外层必须是一个类的定义
2. Public表示公开的，class表示定义一个类，HelloWorld，为一个类名，类名后面必须是一个大括号，一对大括号被称为”类体”



Public的类只能有一个

标识符

1.在java源代码中白色颜色为标识符

2.标识符可以标识类名、变量名、接口名、常量名、方法名

3.凡是程序员自己有权利命令的单词都是标识符

4.main是一个方法的名称，属于标识符，但不能修改。

5.标识符的命名规则

规则1；标识符只能由数字、字母（包括中文）、下划线、美元符号$组成

规则2：标识符不能以数字开头

规则3：关键字不能做标识符。例如：public class static void 这些都是关键字，不能做标识符

规则4：标识符严格区分大小写例如：class a{ }

Class A{ } 编译只会编译排在第一个的标识符生成a.class

关键字

提前定义好的单词，且全部小写，具有特殊含义，不能用作标识符

字符变量

注意：所有的字符型只能使用单引号括起来‘A’

所有的字符串只能使用双引号括起来“ABC”、

字符型一定是单个字符才能成为【字符型】

System.out.println(‘你’); 属于字符型

System.out.println(“你”); 属于字符串型区别就是看是单引号还是双引号

什么是变量

变量就是内存当中存储数据的最基本的单元

整数型：int

带小数的（浮点型）：double

字符串类型：String

字符型类型：char

布尔型：Boolean （true，false）

：

对于一个变量来说包括三要素：1.变量的数据类型2.变量的名字3.变量中保存的值

数据类型决定空间的大小

变量声明

语法格式数据类型变量名; 例如：intnianling;

在同一个域当中，变量名不能重复声明。

Int a=5;

Int a=6; 这就不行不能重复定义

可以一行定义多个变量

Inta,b,c=1; 但是只会给c赋值，a、b不会被赋值；仅仅只是定义了a，b变量，并没有赋值，必须自己赋值

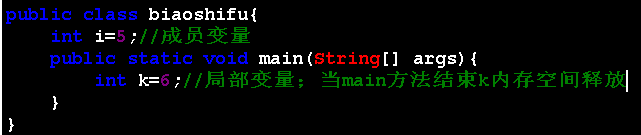
关于变量分类

1.变量根据出现的位置进行划分

在方法体中声明的变量：局部变量

在方法体外声明的变量：成员变量

2．局部变量只在方法体当中有效，方法体执行结束该变量的内存就释放了。



变量的作用域

1. 变量的有效范围就是作用域
2. 关于变量的作用域，即出了大括号就不认识了出了大括号就无法访问
3. Java中有一个很重要的原则，就近原则

int a=10;

int b=a;

System.out.println(a);

System.out.println(b);

10

10 可以这样编写，意思：将a变量的值”复制”一份给b

数据类型

1. 数据类型用来声明变量，程序在运行过程中根据不同的数据类型分配不同大小的空间。
2. 数据类型包括：1.基本数据类型

基本数据分为四大类，8小种

第一类：整数型

第二类：浮点型

第三类：布尔型

第四类：字符型

8小种：byte，short，int，long

Float,double

Boolean

Char

2.引用数据类型

字符串型string属于引用数据类型

String字符串不属于基本数据类型

8种基本数据类型中整数型：byte、short、int、long区别

区别：占用的空间大小不同

类型 占用字节数量

---------------------------------------------------

Byte 1

Short 2

Int 4

Long 8

Float 4

Double 8

Boolean 1

Char 2

# Day3

编码和解码应采用同一个编码方式，不然会出现乱码。

八种基本数据类型详解

字符型：char

1. char占用两个字节
2. char取值范围【0~65535】
3. 采用unicode编码方式
4. Char类型的字面量使用单引号括起来‘b’‘ab’错这是字符串
5. Char可以储存一个汉字一个汉字用两个字节保存

Java中的转义字符

char a=’\t’; \t 实际上是一个字符，不属于字符串转义字符：制表符tab

java语言中 \ 表示转义 \t 表示tab

\n 表示换行

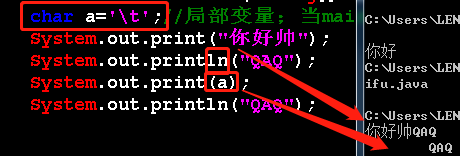
\’表示一个单引号字符

\\ 表示一个一个反斜杠

\”表示转义双引号

\u 表示后面的是一个字符的Unicode编码（16进制）

例如：char z=’\u4e2d’; 输出z，就是中



print打印不换行；println打印换行

6.char x=97; 输出x，会变成a 当一个整数赋值给char类型变量的时候，会自动转换成插入字符型，最终的结果是一个字符

整数型：byte、short、int、long

long a= 100L; 这就被用long类型处理字面量后加L即可

小容量可以直接赋值给大容量，被称之为自动类型转换

大容量不能直接赋值给小容量，需要使用强制类型转换符进行强转。

例如：long a=100L;

Int b=(int)a; //这个(int)就是强制转换符。

多种数据类型混合运算：

多种数据类型做混合运算的时候，最终结果类型是”最大容量”对应的类型，byte、char、short除外。

Byte、char、short做混合运算的时候，各自先转换成int再做处理。

浮点型：float、double

Long类型占用8个字节；float占用4个字节。

Float容量大于long 任意浮点型都比整数型空间大

Java中规定，任何一个浮点型数据默认当做double来处理，如果想让浮点型自变量被当做float处理，只需在字面量后添加F/f

例如：1.0会被默认当成double型处理

1.0F才会被当成float型处理

double pi=3.1415926; 正常编译不会报错

float pi=3.14; 编译报错，因为3.14被默认为double处理，大容量转小容量，报错

改正：float pi=3.14f;

或者强制类型转换，但可能会损失精度

float pi=(float)3.14; 正常编译

int a=10.0/5; 编译报错，5会先转为double型，处理结果为double型，需要强转

更改：int a=(int)10.0/5; 或者int a=(int)(10.0/5);

布尔型：Boolean

Boolean类型只有两个值 true，false

在开发中，主要用来充当条件

Boolean flag=20<30;先运算等号后面的；

Syetem.out.println(f;ag); //true

在类型转换需遵循的规则：

1. 八种数据类型中，除boolean类型不能转换，剩下的七种类型可以转换
2. 若整数型字面量没有超出byte、short、char的取值范围，可以直接赋值
3. 小容量向大容量转换为自动类型转换；容量从小到大转换排序：

Byte<short(char)<int<long<<float<double 其中short和char都占用两个字节，但char可以表示更大的正整数

1. 大容量转换为小容量，称为强制类型转换，编写是必须添加“强制类型转换符”，但运行时可能会损失精度，谨慎使用
2. Byte，short，char类型混合运时，先各自转换成int类型再做运算
3. 多种数据类型混合运算，各自先转换成最大容量的哪一种进行运算

运算符

算术运算符 ++ 自身加1 ++可以出现在变量前，也可以出现在变量后

i++; ++i;

语法：当++出现在变量后，会先做赋值运算，再自加1

Int m=10;

Int n=m++;

输出m // 11

输出n // 10

当++出现在变量前，会先自身加1，在做赋值运算

Int m=10;

Int n=++m;

输出m // 11

输出n // 11

题目：int c=90;

System.out.println(c++); //90 拆解int temp=c++; System.out.println(temp);

System.out.println(c); //91

int d=80;

System.out.println(++d); //81 拆解int temp=++d; System.out.println(temp);

System.out.println(d); //81

关系运算符

所有的关系运算符的运算结果都是布尔类型，不是true就是false。不可能是其他值。

逻辑运算符

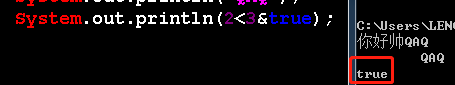
& 与

| 或

! 非

&& 短路与

|| 短路或



短路与和逻辑与的区别

1. 两则运算结果一致，只不过短路与会发生短路现象

Int x=10;

Int y=11;

System.out.println(x>y&x>y++); //输出结果为false，但是x>y++;这个语句执行了 y=12

System.out.println(x>y&&x>y++); //输出结果为false， x>y++;这个语句没有执行 y=11

使用短路与时，当左边的表达式为false时，右边的表达式不执行。这中现象为短路现象。

从效率方面来说，短路与比逻辑与更高效。

赋值运算符

赋值运算符：包括基本赋值运算符合扩展运算符

注意：扩展运算符编写时，中间不能有空格

使用扩展运算符的时候，永远不会改变运算结果类型

Byte x=100;

X+=1; //编译不会报错 x=x+1; //编译报错，x为byte类型；1为int类型；运算结果为int类型，，最左边x为byte类型；所以报错

+=运算符为代表

Int k=10;

K+=20; //k变量追加20 与k=k+20相似

System.out.println(k); //30

条件运算符

语法格式：布尔表达式?表达式1:表达式2

执行原理：布尔表示为true时，表达式1的执行结果，为整个表达式的结果

布尔表示为false时，表达式2的执行结果，为整个表达式的结果

例如：boolean sex=false;

char a=sex?’男’:’女’; //前面变量a的类型不能随意编写，最终计算结果是字符型，所以变量也要使用char类型

运算符

+运算符求和、字符串连接

当+运算符任意一边是字符串类型，那么+会进行字符串拼接操作

注意：字符串拼接完任是一个字符串

Int x=10;

Int x=x+x++; // x++ 为10，先赋值再自身加1 该行代码只要结束完，x就为11了

System.out.println(“a=”+a); // 20 (x到这里就变成11了)

Int b=x+++x; // ++x先自身加1，再赋值。此时就变为 11+12

Inti=5;

I++; //结果为6

System.out.println(i++); // 执行结果为6，但是执行之后，i变成了7

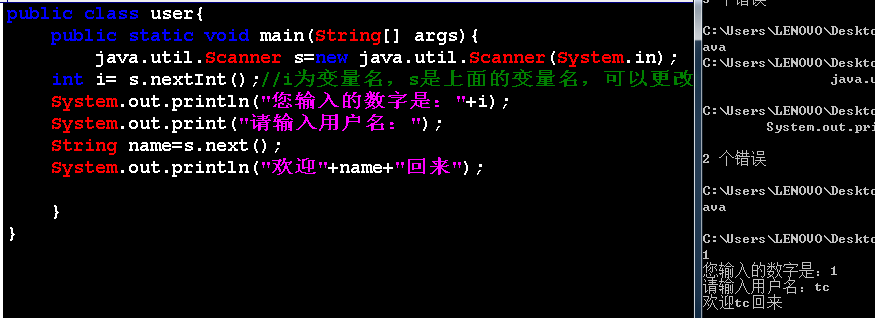
怎么接收用户键盘输入

Java.util.Scanner s=new java.util.Scanner(System.in); //s和下面i、str为自变量可以更改

inti=s.nextInt(); //接收整数

String str=s.next();// 接收字符串

double fenshu=s.nextDouble(); //接收小数点的数字



# Day4

控制语句：

选择语句：if、switch

循环语句：for、while、do…while

转向语句：break、continue、return

If语句的语法结构以及运行原理：

If（布尔表达式）{

Java语句;

}else{

Java语句;

}

或者 if（布尔表达式1）{

Java语句;

}else if（布尔表达式2）{

Java语句;

}else if（布尔表达式3）{

Java语句;

}else if（布尔表达式4）{

Java语句;

}

或者if（布尔表达式1）{

Java语句;

}else if（布尔表达式2）{

Java语句;

}else if（布尔表达式3）{

Java语句;

}else if（布尔表达式4）{

Java语句;

}else{

Java语句;} //以上条件没有一个成立，这个else就执行了

如果布尔表达式1为true，则执行分支1，然后if语句就结束了。

Boolean sex=true;

If (sex){

System.out.println(“男”);

}else{

System.out.println(“女”);

}

# Day5

Switch语句

1. switch语句也是选择语句
2. 语法格式： switch(值){

case 值1:

java语句;

java语句;…..

break;

case 值2:

java语句;

java语句;…..

break;

default:

java语句;

}

以上是一个完整的switch语句

其中break语句不是必须的；default分支也不是必须的

Switc的值：int类型、string类型

注意jdk版本，jdk8之前，switch语句只支持int类型，jdk8以后才支持string类型

Switch语句中“值”与“值1”、“值2”比较，会使用“==”进行比较。

1. switch语句执行原理

注意：如果分支执行了，但是分支最后没有break，此时会发生case穿透现象。即继续执行下面语句不再与后面的case值比较，直到遇到break结束switch语句。

所有case都没有匹配成功，default最后才会执行。

Case合并 case 1：case2：case3:

Java语句1;//switch值与case值“1“2“3任意匹配都会执行java语句1

# Day6

For循环

语法：for(初始化表达式;布尔表达式;更新表达式){

循环体; //循环体由java语句构成

}

注意：初始化表达式最先执行，并且只执行一次

例如：for(i=0;i<10;i++){

Java语句;}

For死循环

For(;;){

Java语句;} //一直执行java语句

For循环嵌套

For(){

//在分析外层for循环的时候，把里面的for就当做一段普通的java语句

For(){

}

}

While 循环：一上来先判断条件在循环。

语法：while(布尔表达式){

循环体;

更新表达式;}

Do while循环：一上来先执行循环体，再判断条件。（绝对会至少执行1次）

语法：do{

循环体;

}while(布尔表达式);

例如：inti=0;

Do{

System.out,println(i);

i++;

}while(i<10);

转向语句：

Break、continue、return（return，学方法的时候学习）

Break

1. 用于switch语句和循环语句（for、while、do…while）当中终止循环的执行。
2. break语句的执行，主要终止最近的循环，不会结束后面的语句，
3. break终止指定的循环,,,break a; 来终止指定循环。

a:for(){

b:for(){

}

}

Continue

1. 与break区别，continue结束本次循环，开始下一次循环。

方法：是一段可以被重复利用的代码片段，并且完成特定的某个功能。

定义：[修饰符列表]返回值类型方法名(形式参数列表){

方法体;

}

注意：[]中括号里面的内容表示不是必须的，是可选的。

方法体为java语句构成。

1. 方法定义在类体中
2. 定义方法的位置不限
3. Main方法自动执行，其他定义的方法需调用才会执行

关于修饰符列表

修饰列表不是必选项，目前统一写出public static

关于返回值类型

1.返回值类型可以是任何数据类型。

2.返回值一般指一个方法执行结束后的结果，通常是一个数据，被称为“值”或“返回值”

3.当一个方法执行结束，不返回任何值的时候，返回值类型也不能为空白，必须写上void main方法结束后不需要给jvm返回任何执行结果。

4.方法执行结束后，返回的结果给调用者了。（谁调用返回给谁）

5.如果返回值类型不是void，那么方法体执行结束后，必须使用”return 值;”这样的语句来完成“值”的返回，如果没有”return 值;”这样的语句编译器报错。

注意：只要有”return 值;”语句执行，当前方法必然结束。

Return 结束当前方法，不是结束程序。

关于方法名

在标识符命名规范中，要求首字母小写，后面每一个单词首字母大写。

关于形式参数列表

简称：形参

注意：形式参数列表中的每一个参数都是“局部变量”，方法结束之后内存释放

形参的个数：0~n个。

例如：public static void sumInt(){}

public static void sumInt(int x){}

public static void sumInt(intx,doubey,String s){}

形参有多个时，用逗号隔开

方法调用：

语法：类名.方法名(实际参数列表);

例如：计算两个int类型数据的商

public static int chufa(intx,int y){

int z=x/y;

return z;

} 类名.chuFa(2,4);

使用一个变量来接收方法的返回值；int a=类名.chuFa(2,4);

注意：变量的数据类型与返回结果类型一致

当一个方法有返回值的时候，可以选择不接收。即不使用变量去接收。（只不过无意义）

调用方法：类名.可以省略，调用的就是本类定义的方法；当跨类调用的时候不能省略类名.

调用程序不一定写到main方法当中，main也是普通方法，可以在一个方法里调用宁一个方法

Break和return区别

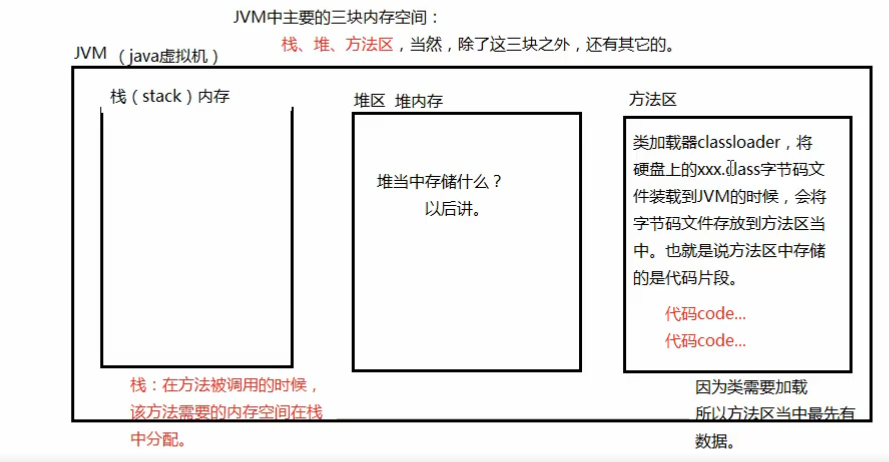
Break退出循环

Return退出方法

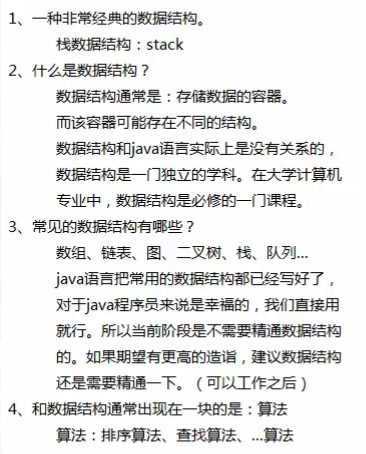
在同一个域中，return语句后不能再编写其他代码。

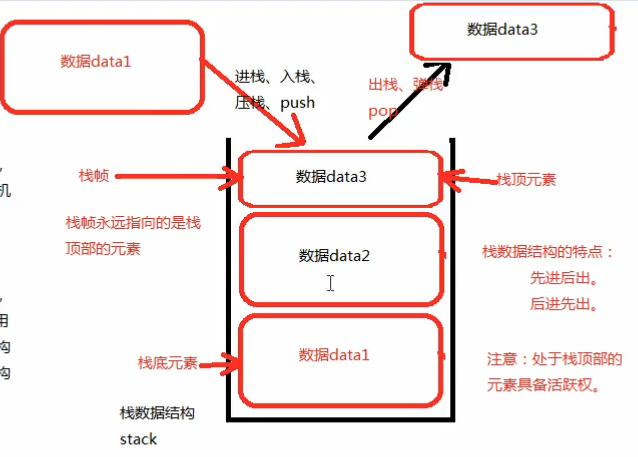
# Day7

Jvm中主要的三块内存空间：



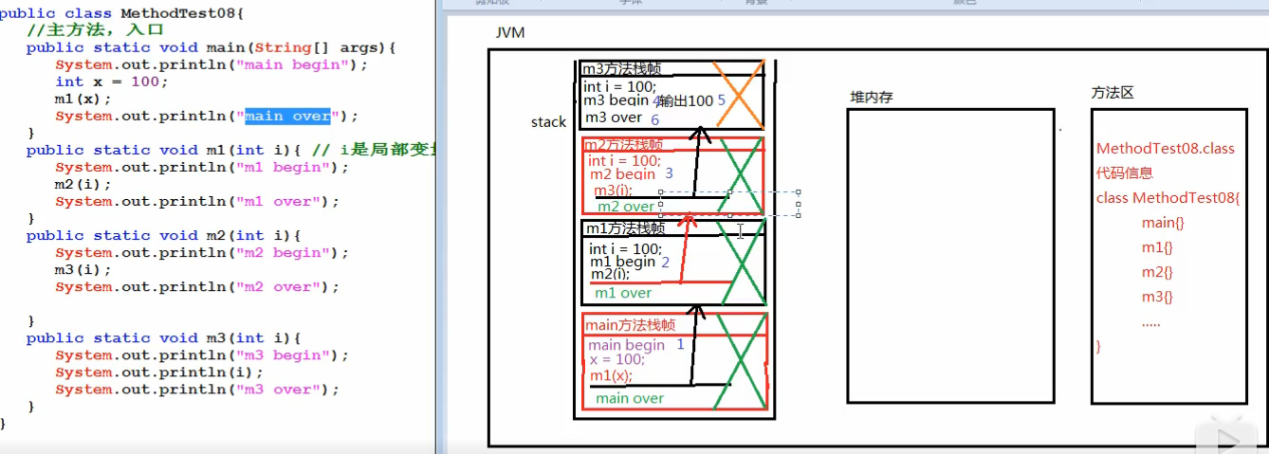
栈数据结构：

数据结构与算法：程序效率；数据类型和算法

方法只有在调用的时候才会在栈中分配空间，并且调用时就是压栈

方法执行结束之后，该方法所需要的空间就会释放，此时发生弹栈动作

栈中存储方法运行过程中需要的变量，以及栈中会存储方法的局部变量。

栈特点：先进后出；后进先出。