

语言基础第二天：

回顾：

1. java开发环境：

- 编译运行过程：
 - 编译期：.java源文件，经过编译，生成.class字节码文件
 - 运行期：JVM加载并运行.class(0和1)
 - 跨平台、一次编译到处使用
- 名词：
 - JVM：java虚拟机
 - 加载.class并运行.class
 - JRE：java运行环境
 - JRE = JVM+系统类库(小零件)
 - JDK：java开发工具包
 - JDK = JRE+编译、运行等命令工具

2. idea:

- JetBrains公司的，分社区版(免费)和终级版(收费)
- 开发步骤：
 - 新建Java项目-----小区
 - 新建Java包-----楼+单元
 - 新建Java类-----房子
 - main：System.out.println("hello world");
- 注释：解释性文本(计算机不执行)
 - 单行注释：//
 - 多行注释：/* */
 - 文档注释：/** */-----后期才讲

精华笔记：

1. 变量：存数的

- 声明：-----在银行开了个帐户
- 初始化：-----给帐户存钱
- 使用：-----使用的是帐户里面的钱
 - 对变量的使用就是对它所存的那个数的使用
 - 变量在使用之前必须声明并初始化
- 命名：
 - 只能包含字母、数字、_和\$符，并且不能以数字开头
 - 严格区分大小写

- 不能使用关键字
 - 允许中文命名，但不建议，建议"英文的见名知意"、"小驼峰命名法"
2. 八种基本数据类型：byte、short、int、long、float、double、boolean、char
- int：整型，4个字节，-21个多亿到21个多亿
 - 整数直接量默认为int类型，但不能超出范围，若超范围则发生编译错误
 - 两个整数相除，结果还是整数，小数位无条件舍弃(不会四舍五入)
 - 运算时若超范围会发生溢出，溢出不是错误，但需要避免
 - long：长整型，8个字节，很大很大很大
 - 长整型直接量需要数字后加L或l
 - 运算时若有可能溢出，建议在第1个数字后加L
 - double：浮点型，8个字节，很大很大很大
 - 浮点数直接量默认为double型，若想表示float则需在数字后加F或f
 - double或float型数据运算时，有可能会发生舍入误差，精确场合不能使用
 - boolean：布尔型，1个字节
 - 只能取值为true或false
 - char：字符型，2个字节
 - 采用Unicode编码格式，一个字符对应一个码
 - 表现的形式是字符char，但本质上是码int(0到65535之间)
 - ASCII码：'a'---97 'A'---65 '0'---48
 - 字符直接量必须放在单引号中，只能装1个
 - 特殊符号需要通过\来转义

3. 类型间的转换：

数据类型由小到大依次为：byte, short, int, long, float, double
char

- 两种方式：
 - 自动/隐式类型转换：小类型到大类型
 - 强制类型转换：大类型到小类型
- 语法：(要转换成为的数据类型)变量
- 注意：强转有可能溢出或丢失精度
- 两点规则：
 - 整数直接量可以直接给byte,short,char赋值，但不能超范围
 - byte,short,char型数据参与运算时，系统会自动将其统一转换为int再运算

笔记：

1. 变量：存数的

- 声明：-----在银行开了个帐户

```
int a; //声明一个整型的变量，名为a
int b,c,d; //声明三个整型的变量，名为b,c,d
//int a; //编译错误，变量不能同名
```

- 初始化: -----给帐户存钱

```
int a = 250; //声明整型变量a并赋值为250-----开户的同时存了个250
int b;      //声明整型变量b
b = 250;    //给变量b赋值为250-----先开户再存了个250
b = 360;    //修改变量b的值为360
```

- 使用: -----使用的是帐户里面的钱
 - 对变量的使用就是对它所存的那个数的使用

```
int a = 5;
int b = a+10; //取出a的值5, 加10后, 再赋值给整型变量b
System.out.println(b); //输出变量b的值15
System.out.println("b"); //输出b, 双引号中的原样输出
a = a+10; //在a本身基础之上增10
System.out.println(a); //15
```

- 变量在使用之前必须声明并初始化

```
System.out.println(i); //编译错误, 变量i未声明
int i;
System.out.println(i); //编译错误, 变量i未初始化
```

- 命名:
 - 只能包含字母、数字、_和\$符, 并且不能以数字开头
 - 严格区分大小写
 - 不能使用关键字
 - 允许中文命名, 但不建议, 建议"英文的见名知意"、"小驼峰命名法"

```
int a1,a_5$, _3c,$_5;
//int a*b; //编译错误, 不能包含*号等特殊符号
//int 1a; //编译错误, 不能以数字开头
int aa = 5;
//System.out.println(aA); //编译错误, 严格区分大小写
//int class; //编译错误, 不能使用关键字

//int j; //不直观, 不建议
//int 年龄; //允许中文, 但不建议
//int nianLing; //必须杜绝, 既不直观也不专业, 不建议
int age; //建议"英文的见名知意"
int score,myScore,myJavaScore; //建议"小驼峰命名法"
```

2. 八种基本数据类型: byte、short、int、long、float、double、boolean、char

- int: 整型, 4个字节, -21个多亿到21个多亿
 - 整数直接量默认为int类型, 但不能超出范围, 若超范围则发生编译错误

```
int a = 25; //25为整数直接量, 默认为int类型
//int b = 10000000000; //编译错误, 100亿默认为int类型, 但超出范围了
//int c = 25.678; //编译错误, 数据类型不匹配
```

- 两个整数相除, 结果还是整数, 小数位无条件舍弃(不会四舍五入)

```
System.out.println(5/2); //2
System.out.println(2/5); //0
System.out.println(5/2.0); //2.5
```

- 运算时若超范围会发生溢出，溢出不是错误，但需要避免

```
int c = 2147483647; //int的最大值
c = c+1;
System.out.println(c); //-2147483648(int的最小值)，发生溢出了
```

- long: 长整型，8个字节，很大很大很大

- 长整型直接量需要数字后加L或l

```
long a = 25L; //25L为长整型直接量
//long b = 10000000000; //编译错误，100亿默认为int型，但超出int范围了
long c = 10000000000L; //100亿L为长整型
```

- 运算时若有可能溢出，建议在第1个数字后加L

```
long d = 1000000000*2*10L;
System.out.println(d); //200亿
long e = 1000000000*3*10L;
System.out.println(e); //不是300亿
long f = 1000000000L*3*10;
System.out.println(f); //300亿
```

- double: 浮点型，8个字节，很大很大很大

- 浮点数直接量默认为double型，若想表示float则需在数字后加F或f

```
double a = 25.678; //25.678为浮点数直接量，默认double型
float b = 25.678F; //25.678F为float型
```

- double或float型数据运算时，有可能会发生舍入误差，精确场合不能使用

```
double c=3.0,d=2.9;
System.out.println(c-d); //0.10000000000000009，有可能发生舍入误差
```

- boolean: 布尔型，1个字节

- 只能取值为true或false

```
boolean a = true; //true为布尔型直接量
boolean b = false; //false为布尔型直接量
//boolean c = 250; //编译错误，数据类型不匹配
```

- char: 字符型，2个字节

- 采用Unicode编码格式，一个字符对应一个码

表现的形式是字符char，但本质上是码int(0到65535之间)

ASCII码: 'a'---97 'A'---65 '0'---48

- 字符直接量必须放在单引号中，只能装1个

```

char c1 = '女'; //字符女
char c2 = 'f'; //字符f
char c3 = '6'; //字符6
char c4 = ' '; //空格符
//char c5 = 女; //编译错误, 字符直接量必须放在单引号中
//char c6 = ''; //编译错误, 必须有字符
//char c7 = '女性'; //编译错误, 只能有1个字节

char c8 = 97; //0到65535之间
//println()时会依据c8的数据类型来输出数据
System.out.println(c8); //A, c8为char类型, 所以会以字符的格式来输出

```

- 特殊符号需要通过\来转义

```

char c9 = '\\'; //\为转义符
System.out.println(c9); //\

```

3. 类型间的转换:

数据类型由小到大依次为: byte, short, int, long, float, double

char

○ 两种方式:

- 自动/隐式类型转换: 小类型到大类型
- 强制类型转换: 大类型到小类型

语法: (要转换成为的数据类型)变量

注意: 强转有可能溢出或丢失精度

```

int a = 5;
long b = a; //自动类型转换
int c = (int)b; //强制类型转换

long d = 5; //自动类型转换
double e = 5; //自动类型转换 //练习-----4:41继续

long f = 10000000000L;
int g = (int)f;
System.out.println(g); //1410065408, 强转有可能发生溢出
double h = 25.987;
int i = (int)h;
System.out.println(i); //25, 强转有可能丢失精度

```

○ 两点规则:

- 整数直接量可以直接给byte,short,char赋值, 但不能超范围
- byte,short,char型数据参与运算时, 系统会自动将其统一转换为int再运算

```
byte b1 = 5;
byte b2 = 6;
byte b3 = (byte)(b1+b2);

System.out.println(2+2);    //4
System.out.println(2+'2');  //52, 2加上'2'的码50
System.out.println('2'+2);  //100, '2'的码50, 加上, '2'的码50

System.out.println('2');    //2, 只有运算时才会转换为int类型
```

补充:

1. 标识符命名法:

- 小驼峰命名法: 第1个单词首字母小写, 其余单词首字母大写

score, myScore, myJavaScore-----变量

- 大驼峰命名法/帕斯卡命名法: 所有单词首字母大写

Score, MyScore, MyJavaScore-----类

2. 数据类型分类: 基本数据类型和引用数据类型

3. 内存单位换算:

```
1G=1024M(兆)
1M=1024KB(千字节)
1KB=1024B(字节)
1B=8bit(位)
```

4. Unicode: 万国码、统一码、通用码, 世界级通用的定长(固定16位)字符集

5. 明日单词:

```
1)name: 姓名
2)number/num: 数字
3)flag: 标记
4)max: 最大值
5)if: 如果
6)price: 价格
7)else: 否则、其它的
8)operator/oper: 运算符
```