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.class并运行.class

            > 特点：跨平台，一次编译到处使用

    - 名词：
        - JVM：java虚拟机 JVM加载.class并运行.class
        - JRE：java运行环境 JRE = JVM+java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，分为社区版和终级版
    - 开发步骤：
        - 新建Java项目/工程
        - 新建Java包
        - 新建Java类
    - 注释：解释性文本
        - 单行注释：//
        - 多行注释：/* */
        - 文档注释：/** */

精华笔记：

1. 变量：存数的
    - 声明：----相当于在银行开了个帐户
    - 初始化：----相当于给帐户存钱
    - 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
        - 变量的用之前必须声明并初始化
    - 命名：-----相当于给帐户起名
        - 只能包含字母、数字、_和$符，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)
        - 运算时若超出范围，则发生溢出，溢出不是错误，但是需要避免
    - long：长整型，8个字节，-900万万亿多到900万万亿多
        - 若想表示长整型直接量，需在数字后加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
        - 小数直接量默认为double型，若想表示float，需在数字后加F或f
        - 不能表示精确数据，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
        - 只能存储true或false
    - char：字符型，2个字节
        - 采用的是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：'a'----97　'A'----65　'0'----48)
        - 字符型直接量必须放在单引号中，有且仅有1个
        - 特殊符号需要通过\来转义
3. 类型间的转换：
    - 基本数据类型从小到大依次为：
        - 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：强转有可能会溢出或丢失精度
    - 两点规则：
        - 整数直接量可以直接赋值给byte,short,char，但不能超出范围
        - byte,short,char型数据参与运算时，系统会将其自动转换为int类型再运算

笔记：

1. 变量：存数的
    - 声明：----相当于在银行开了个帐户

```
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：----相当于给帐户存钱

```
int e = 250; //声明整型变量e并赋值为250----开户的同时存钱
int f;   //声明整型变量f----------先开户
f = 250; //给变量f赋值为250-------后存钱
f = 360; //修改变量f的值为360
int g=5,h=8,i=10; //声明三个整型变量g,h,i，并分别赋值为5,8,10
```

- 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 j = 5; //声明整型变量j并赋值为5
int k = j+10; //取出j的值5，加10后，再赋值给变量k
//输出时若不加双引号，则java认为它就是一个变量
System.out.println(k);   //输出变量k的值15
System.out.println("k"); //输出k，双引号中的原样输出
j = j+10; //在j本身基础之上增10
System.out.println(j); //15
int balance = 5000; //帐户余额
balance = balance-1000; //取款1000
System.out.println(balance); //4000
```

  - 变量的用之前必须声明并初始化

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

- 命名：-----相当于给帐户起名
  - 只能包含字母、数字、_和$符，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,$5b;
//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 年龄; //正确，但不建议
int nianLing; //必须杜绝
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，-2^31到2^31-1(正负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 = 3.14; //编译错误，整型变量中只能装整数
```

- 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
```

- 运算时若超出范围，则发生溢出，溢出不是错误，但是需要避免

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

- long：长整型，8个字节，-2^63到2^63-1(正负900万万亿多)
  - 若想表示长整型直接量，需在数字后加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int类型，但超出int范围了
long c = 10000000000L; //100亿L为长整型直接量
//long d = 3.14; //编译错误，长整型变量中只能装整数
```

  - 运算时若有可能溢出，建议在第1个数字后加L

```
long e = 1000000000*2*10L;
System.out.println(e); //200亿
long f = 1000000000*3*10L;
System.out.println(f); //不是300亿
long g = 10000000000L*3*10;
System.out.println(g); //300亿
```

- double：双精度浮点型，8个字节
  - 小数直接量默认为double型，若想表示float，需在数字后加F或f

```
double a = 3.14; //3.14为小数直接量，默认为double型
float b = 3.14F; //3.14F为float型直接量
```

  - 不能表示精确数据，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
  - 只能存储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，布尔型变量中只能存储true或false
```

- char：字符型，2个字节
  - 采用的是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：'a'----97　'A'----65　'0'----48)
  - 字符型直接量必须放在单引号中，有且仅有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，字符型直接量必须放在单引号中
//char c6 = ''; //编译错误，必须有字符
//char c7 = '10'; //编译错误，只能存储1个字符

char c8 = 65; //0到65535之间
System.out.println(c8); //println()会依据变量的类型做输出展示
//c8为char型，所以会以字符的形式输出

char c9 = '\\';
System.out.println(c9); //\
```

  - 特殊符号需要通过\来转义

```
char c9 = '\\';
System.out.println(c9); //\
```

3. 类型间的转换：
   - 基本数据类型从小到大依次为：
     - 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：强转有可能会溢出或丢失精度

```
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
System.out.println(e); //5.0，默认保留一位小数
```

```
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赋值给byte,short,char，但不能超出范围
  - byte,short,char型数据参与运算时，系统会将其自动转换为int类型再运算

```
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，因为没有运算，所以输出的是字符2

int m = 'a';
System.out.println(m); //97，查看字符对应的码
char n = 97;
System.out.println(n); //a，查看码对应的字符
```

补充：

1. 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----------变量/方法

     ```
     score,myScore,myJavaScore
     ```

   - 帕斯卡命名法/大驼峰命名法：所有单词首字母大写----------类

     ```
     Score,MyScore,MyJavaScore
     ```

2. Unicode：万国码、统一码、通用码，是世界级通用的定长(16位)字符集
3. 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
9)score:成绩
```