

简介day01

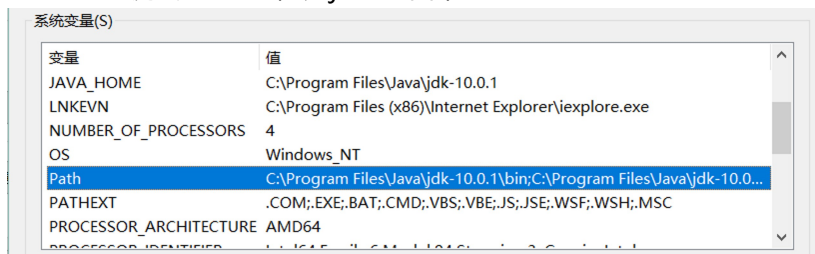
2018年10月27日 9:05

1. 发展史 sun公司

- a. 1991年 oak(橡树)
- ★b. 1995年 java 爪哇岛的咖啡
- c. 1996年 jdk1.0发布
- ★d. 1998年 jdk1.2发布
 - ★ J2SE:(JAVASE) java标准版
 - ★ J2EE: (JAVAE) java企业版
 - ★ J2ME:java微型版
- e. 2002年 jdk 1.4
- f. 2004年 jdk 1.5
- g. 2006年 jdk 1.6
- ★h. 2009年 被Oracle (甲骨文) 收购
- i. 2011年 jdk 1.7
- j. 2014年 jdk 1.8
- k. 2017年 jdk 1.9

2. Java的开发环境

- a. java先编译, 后运行
- b. 开发java必须有jdk (Java开发包)
- c. 编译好的Java程序, 只需要jre (Java运行环境)
- d. JVM:Java虚拟机
- ★e. 关系jdk包含jre, jre包含jvm
- ★f. jdk的安装:
 - i. 配置路径: 1) JAVA_HOME: c:\Program Files\java\jdk-10.0.1;
path:%JAVA_HOME%\bin;
2) path:C:\Program Files\Java\jdk-10.0.1\bin;
 - ii. 测试: cmd 命令 javac+回车



3. java的开发过程

- a. 使用编辑器编写源文件 (.java)
 - //类的声明: 文件名必须和public修饰的类名相同
 - public class HelloWorld{//类的声明
 - //方法的声明
 - public static void main(String[] args){
 - //控制台输出
 - System.out.println("HelloWord!");

```
    }//方法体
} //类体
```

b. 把 *.java 编译成 *.class (javac *.java) ,也叫字节码文件

! `javac Helloworld.java`

c. 运行Java程序 (java *)

! `java Helloworld`

4. Helloworld(Eclipse开发)

a. 创建Java工程：工程的名称通常小写

b. 在src下创建Java类

定义包名：域名反过来写，功能名称：例如：cn.tedu.demo

类名：1) 驼峰起名法：HelloWord、Student等

2) 见名知其意

3) 改名：在class文件上右键选择：Refactor—Rename（快捷键：Alt+Shift+R）

```
package cn.tedu.demo;

public class HelloWorld {
    //入口方法的特点
    //1.方法名：main
    //2.方法的返回类型void类型，并且由public和static
    //3.参数列表中的数据类型必须是String[]
    public static void main(String[] args) { //变量名可以换
        System.out.println("Hello World!");
        /*打印图形
        *
        * *
        * * *
        * * * *
        * * */
        for(int a = 1;a<5;a++) {
            for(int b = 1;b<=a;b++) {
                System.out.print("*"); //不换行
            }
            System.out.println();//换行
        }
    }
}
```

★ c. 快捷键：Ctrl+D 删除光标所在的一行

Ctrl Shift+/ 多行注释

Ctrl Shift+\ 取消多行注释

Ctrl +/ 单行注释

Ctrl+Shift+o 导包

5. 变量

a. 两部分 变量名和变量值

b. 定义变量：数据类型 变量名称 int age;

c. 使用变量（给变量赋值或获取变量值）

i. age = 20;

System.out.println(age);获取age的值

ii. int score = 100;

```
package cn.tedu.demo;
/**
```

```

* 1.类体中只能定义变量和方法
* 2.语句必须在方法内完成 age = 20 是赋值语句;
* 3.当定义变量为同一数据类型, 那么变量之间用“,”隔开
* 4.变量的分类: 成员变量和局部变量
*         4.1:成员变量:定义在类体中, 不被方法包含;
*             访问范围:整个类内, 当前类的所有方法
*         4.2:局部变量:在方法中定义的变量
*             访问范围:只能局限于定义的方法内访问
★ * 5.使用变量的规则: 使用变量时, 保证变量已经被初始化
*/

public class Demo1 {

    int age = 20; //声明变量同时给变量赋值 (属于变量定义语句)
    int age1; //声明变量
    int x = 1, y = 2;

    void test1() {
        age1 = 20; //给变量赋值 (赋值语句)
        System.out.println(age);
        System.out.println(age1);
        int sum = x+y;
        System.out.println(sum);
    }
    // int a; //没有默认值, 不可以使用, 可以改为以下语句
    int a = 2;
    System.out.println(a);
}
void test2() {
    // sum = x+y;
}

public static void main(String[] args) {
    Demo1 d = new Demo1();
    d.test1();
}

}

```

d. 标识符

- i. 定义: 用来起名字的符号 (类名, 变量名, 方法名) (驼峰规则, 第一个字母小写)
- ii. 标识符的组成: 0到9, 大小写字母, _, \$, Unicode字符集 (各个国家的文字)
- iii. 合法性: 第一位不能是0-9之间的数字


```

int age1;
int _456;
int $$;
int 3$; //error
int aB;
int a_b;
int a,b; //error
int 成绩;
int studentAge; 驼峰规则, 第一个字母小写

```

e. 关键字

- i. 定义: 被赋予特定意义的单词
- ii. 关键字有哪些: 59个+2个保留字 (goto、const)
- iii. 注意: 都不能作为合法的标识符用

注释: 单行注释和多注释

作用: 解释说明

6. 数据类型

a. 分为：基本数据类型和引用类型

b. 基本数据类型：8个

i. 整形数据：byte, short, int, long (4、-9)

1) byte: 一个字节 (8位) $[-128, 127] = [-2^7, 2^7-1]$

2) short: 两个字节 (16位) $[-2^{15}, 2^{15}-1]$

3) int: 4个字节 (32位) $[-2^{31}, 2^{31}-1]$

4) long: 8个字节 (64位) $[-2^{63}, 2^{63}-1]$

```
System.out.println(Byte.MIN_VALUE+", "+Byte.MAX_VALUE);  byte类型的最小值和最大值
System.out.println(Integer.MIN_VALUE+", "+Integer.MAX_VALUE);  int类型的最小值和最大值
System.out.println(Short.MIN_VALUE+", "+Short.MAX_VALUE);  short类型的最小值和最大值
System.out.println(Long.MIN_VALUE+", "+Long.MAX_VALUE);  long类型的最小值和最大值
```

计算机只识别二进制：010

▪ Java语言识别 十进制、八进制、十六进制

十进制：int i = 10;

八进制：int i2 = 012; //0(零)表示是八进制的数，12表示具体的八进制的数

int i3 = 018; //error

十六进制：int i4 = 0x33ff; //0-9abcdef

二进制和十六进制的关系(十六进制是二进制的简单表示法)

1000 0100 1010 0001

8 4 a 1 = 0x84a1

ii. 浮点型：float, double (精度更高) (3.2、-8.9)

1) float: 单精度(4个字节--32位; 1个字节存符号位, 8个字节指数, 23个字节尾数)

float f = 4.5; //默认为double类型

float f = 4.5f; //正确, 大小写都可以

2) double: 双精度 (8个字节--64位; 1个字节存符号位, 11个字节指数, 52个字节尾数)

double d = 4.6; //double也可以带d/D

! 适合科学运算和工程运算, 不适合商业运算

```
package cn.tedu.demo;

★ import java.math.BigDecimal;
import java.text.DecimalFormat;

public class Demo1 {
    void test1() {
        float f = 2.5f;
        double d = 3.5D;

        double d1 = 2.3;
        double d2 = 1.5;
        System.out.println(d1-d2);

        //double类型适合做科学运算和工程运算
        //不适合做商业运算, 解决方案: BigDecimal类
        //Ctrl+Shift+o 导包快捷键
        BigDecimal bc1 = new BigDecimal("2.3");
        BigDecimal bc2 = new BigDecimal("1.5");
        System.out.println(bc1.add(bc2)); //加法
```

```

        System.out.println(bc1.subtract(bc2));//减法
        System.out.println(bc1.multiply(bc2));//乘法
        System.out.println(bc1.divide(bc2, 1, BigDecimal.ROUND_HALF_UP));//除法

        //格式化输出: DecimalFormat
        BigDecimal bc3 = new BigDecimal("2.312");
        BigDecimal bc4 = new BigDecimal("1.534");
        System.out.println(bc3.multiply(bc4));//乘法
        //格式化 (商业运算四舍五入)
        DecimalFormat df = new DecimalFormat("###.##");//###.##点后面几个#就保留几位
        System.out.println(df.format(bc3.multiply(bc4)));
    }

    public static void main(String[] args) {
        Demo1 demo = new Demo1();
        demo.test1();
    }
}

```

iii. 字符型: char 可以存一个字符(A、a、中)

- 1) 用来存字符 char c = 'A'; char c1 = '中'; char c2 = 65;
- 2) 占2个字节, 16位, 没有负数0-65535
- 3) 字符型为什么能存汉字: Java使用的是Unicode字符集 (2个字节可以表示一个汉字)

char c3 = '\u4e2d';*//十六进制的写法, 代表 '中'*

```

void test2() {
    char c = 'A';
    char c1 = '中';
    char c2 = 65;
    char c3 = '\u4e2d';//十六进制的写法
    System.out.println(c);
    System.out.println(c1);
    System.out.println(c2);
    System.out.println(c3);
}

```

4) 转义字符

```

char c4 = '\t';//表示TAB键
char c5 = '\n';//表示换行符
//c:\java\Demo.java
char c6 = '\\';//表示转义为\
System.out.print("aaa");
System.out.print(c6);
System.out.print("aaa");

```

iv. 布尔类型: boolean 存两个值 (true、false)

- 1) 表示Java里的真和假

```

boolean b1 = true;//真
boolean b2 = false;//假

```

c. 引用类型: 除了基本数据类型, 其他的数据类型都为引用类型

```

String name;
int[] score;
Demo1 demo;

```