

java基础语言

2018年5月19日 16:38

代码只是思想的体现形式

java基础语言：关键字 标识符 注释 常量与变量 运算符 语句 函数 数组
了解其表现形式与用途
editplus (超级记事本)

关键字：紫色 **关键字都是小写 将类名的首字母大写 (规范)**

```
class Demo
{
    public static void main(String[] args)
    {
        System.out.println("hello java");
    }
}
```

标识符：程序中自定义的一些名称

26各字母大小写；数字；_ \$

不能以数字开头 不能用关键字 不能用空格

Demo_1Test

mingcheng\$zimingcheng

常量：

1.整数常量

2.小数常量

3.布尔型boolean常量：true false

4.字符常量：单个数字字符符号'

5.字符串常量：一个或多个字符" 空字符串("")和null不同

6.null常量

进制：

二进制：0,1 1关0开

八进制：0-7 以0开头

十进制

十六进制：0-9 A-F 以0x开头

byte 字节 (最小单位) =8个二进制位 (bit位)

1k=1024字节；1000

01101011

752=7*100+5*10+2*1

1011=1*2(0)+1*2(1)+0*2(2)+1*2(3)
=1 +2 +0 +8
=11

0 1 0 1 0 1 1
64 32 16 8 4 2 1

216 0 余数
213 1
1 1
h=110

注释：注释不进入class文件

// 单行注释

/*

王若潇 ———> 多行注释 (多行注释中不能有多行注释)
同济大学

*/

java特有的注释方式

/**

文档注释

注释中的文字可以由java提取

*/

—————> javadoc.exe
生成程序说明书



/*

需求：练习hello world

思路：

- 1.定义一个类，因为java程序都定义在类中，以字节码最终体现
- 2.定义一个主函数，可以让该类独立运行
- 3.因为要演示hello world，需要用输出语句

步骤

*/

010101110
↓
010-101-110
↓
2 5 6
八进制:0256

1010-1110
10 14
十六进制: 0xAE

结论：

八进制数，其实就是二进制位 3 个二进制位为一个八进制位

十六进制数，其实就是二进制位 4 个二进制位为一个十六进制位

二进制和十六进制最常见

IP地址最大255也就是11111111 (1个字节)

IPV4

IPV6

负数的二进制

1 11
6 = 110

负数的二进制

6

0000-0000 0000-0000 0000-0000 0000-0110 (内存中的存储方法)

-6

其实就是这个数的正数的二进制取反，加1

0000-0110

取反 1111-1001

加1 ~~0000-0001~~

1111-1010

负数二进制最高位是1

变量

2018年5月23日 9:45

变量：内存中的一个储存区域，用于存储不确定变量

java是强类型数据语言

名字+类型+指定类型的数据

变量用来不断的存放同一类型的常量并可以重复使用

当数据不确定时使用变量

基本数据类型：整数类型：byte short int long

字节（8位） $-2^7 \sim 2^7 - 1$
-128~127

现在通常用int（整数默认）

浮点类型：float double 8个字节
单精度 双精度

4个字节

16个bit位 -32768~32767

4个字节 $-2^{31} \sim 2^{31} - 1$ （上亿）

8个字节

'12'不行

字符型：char（2个字节）一个中文两个字节

布尔型：boolean

```
class VarDemo
```

```
{
    public static void main(String[] args)
    {
        //数据类型 变量名 = 初始化值;
        byte b=3;
        b=8;
        short s = 4000;
        int x = 12;
        long l = 12345678l; //后面加一个l表示长整形

        float f = 2.3f; //后面加f因为默认是double
        double d = 3.4;
        char ch = 'a';

        boolean bl = true;
        //必须有初始化值
        {
            int z = 9;
            System.out.println(z); //z只在{}域有用
        }

        System.out.println(ch);
    }
}
```

强制转换

class VarDemo2

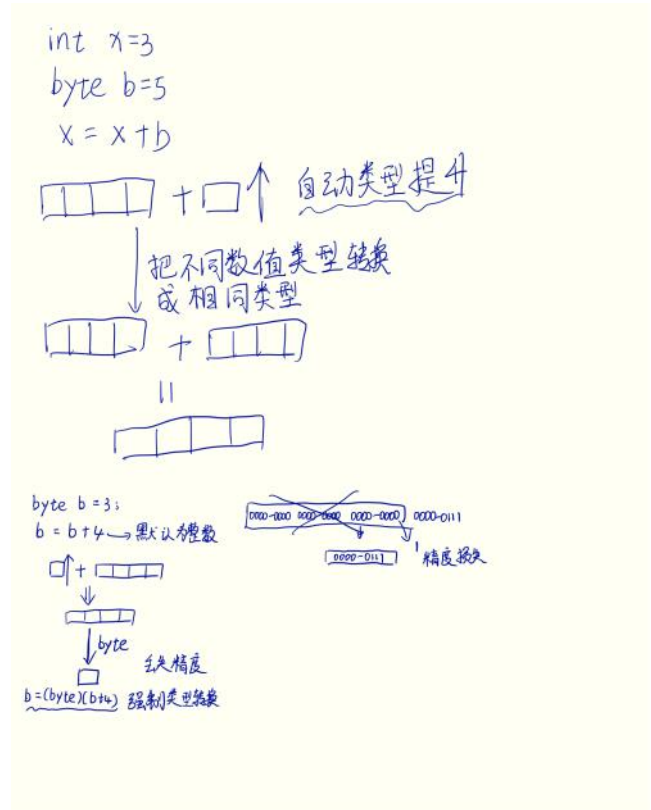
```

{
    public static void main(String[] args)
    {
        int x = 3;
        x = x + 5; //运算区运算 (同类型)
        //运算两端不同类型不能运算 (数值型除外)
        byte b = 5;
        x = x + b;

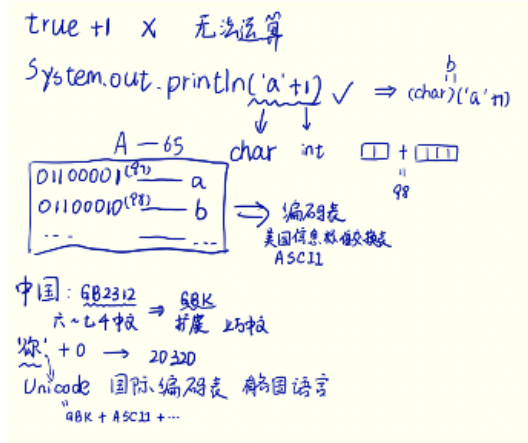
        byte c = 3;
        c = (byte)(c + 4); //强制类型转换

        System.out.println(c);
    }
}

```



System.out.println('你'+0);



关键点

byte b = 4; int 先判断是否存在 byte 范围内, 再自动转换

byte b1 = 3;

byte b2 = 7; 变量 (不确定, 无法检查)

b = b1 + b2; 可能损失精度 b = 3 + 7 ✓ 常量能检查

System.out.println(b)

int x;

x = b1 + b2; ✓

int x;

int x1 = 10;

int x2 = 98;

x = x1 + x2 ✓

int x1 = Integer.MAX_VALUE

int x2 = 98

x = x1 + x2

x = -2147483647 (超出自动变为负)

因为默认类型

```

int x3;
int x1=Integer.MAX_VALUE;
int x2=98;
x3 = x1 + x2;
System.out.println(x3);

```

算术运算符

2018年5月31日 21:46

数据 → 运算 2018. 6. 1

算术运算符: + - * / % (取余、模运算) ⇒ 开关算法的定义

int x = 6370; $5 \% 2 = 1$

$x = x / 1000 * 1000 \Rightarrow 6000$ $2 \% 5 = 2$

\downarrow
 $637 \rightarrow 6$ $-5 \% 2 = -1$ 取模只考虑绝对数

$5 / 2 = 2$ $5 \% -2 = 1$

+(连接符)

3 + "2" 字符串 = 字符串 "32"

System.out.println("5+5=" + 5+5) ⇒ 5+5=55

("5+5=" + (5+5)) ⇒ 5+5=10

int a = 4, b = 5

System.out.println(a + ", " + b); ⇒ 4, 5

("a=" + a + ", b=" + b); ⇒ a=4, b=5

自增: 在原有数据基础上加1

+ + - -

int a = 3, b;

// a++ → a=4 a=a+1

// ++a // a=a+1

// b=a++ → a=4 b=3

// b=++a → a=4 b=4

int i = 3;

i = i + 1; → i = 3 ⇒ temp = i;

temp = i + 1;

i = temp;

赋值运算符 += a += 2 ⇒ a = a + 2

short s = 3;

s = s + 4;

s = s + 4;

有自动转换过程

区别 ✓

有精度损失 → (short)(s+4) = s+4

```
class OperateDemo
{
    public static void main(String[] args)
    {
        //算术运算符
        int x = 6370;
        x = x / 1000 * 1000;
        System.out.println(x);
        int a = 4, b = 5;
        System.out.println("a=" + a + ", b=" + b); //a=4, b=5

        a = 3;
        b = a++;
        System.out.println(a + ", " + b);

        int i = 3;
        i = i++;
        System.out.println("i=" + i);

        //赋值运算符
        int a1, a2, a3;
        a1 = a2 = a3 = 4;
        a1 += 2; //a1 = a1 + 2
    }
}
```

键盘录入

2018年6月15日 13:53

/*

实现键盘数据的录入

A.导包

import java.util.Scanner;//(在class上面)

B.创建键盘录入对象

Scanner sc = new Scanner(System.in);

C.通过对象获取数据

int x = sc.nextInt();

*/

import java.util.Scanner;

class ScannerDemo{

 public static void main(String[] args){

 Scanner sc = new Scanner(System.in);

 System.out.println("请输入一个数据: ");

 int x = sc.nextInt();

 System.out.println("你输入的是: "+x);

 }

}

键盘录入练习

2018年6月15日 14:02

```
import java.util.Scanner;
class ScannerTest{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入第一个数据: ");
        int x = sc.nextInt();
        System.out.println("请输入第二个数据: ");
        int y = sc.nextInt();

        int sum = x+y;
        System.out.println("和为"+sum);
    }
}
```