

2021.5.15 dmd.java基础

笔记本： 我的第一个笔记本

创建时间： 2021/5/15 16:45

更新时间： 2021/5/15 20:24

作者： dmd1810411032@163.com

2021.5.15 dmd.java基础

1.规则：

字符串>整数>字符 字符不可以单独输出，输出时是它的ASCC码

```
public static void main(String[] args) {
    char c='a';
    int num=10;
    String s="hello";
    System.out.println(c+num+s);
    System.out.println(c+s+num);
    System.out.println(c+(num+s));
    System.out.println((c+num)+s);
    System.out.println(s+num+c);
    System.out.println("** **");
    System.out.println('*'+'\t'+'*');
    System.out.println('*'+""'\t'+'*');
    System.out.println('*'+'\t'+"*");
    System.out.println('*'+(''\t'+"*"));
}
```

```
107hello
ahello10
a10hello
107hello
hello10a
* *
93
* *
51*
* *
```

2.由于字符串大于整数，所以当两个类型相加时，整数将变为字符串。当由大变小，由字符串变为整型，要进行封装,即用Integer.parseInt(n)

```
public static void main(String[] args) {
    String s="123+";
    System.out.println(s);
    int n= Integer.parseInt(s);
    System.out.println(n);
}
```

3. 整型: byte(1字节=8bit) \ short(2字节) \ int(4字节) \ long(8字节)

① byte范围: -128 ~ 127

浮点型: float(4字节) \ double(8字节)

① 浮点型，表示带小数点的数值

② float表示数值的范围比long还大

通常，定义浮点型变量时，使用double型。

字符型: char (1字符=2字节)

① 定义char型变量，通常使用一对",内部只能写一个字符

布尔型: boolean

① 只能取两个值之一: true 、 false

② 常常在条件判断、循环结构中使用
4.double,float是可以转化成int,这三者是可以相互转化的。
& && | || ! ^

说明:

1.逻辑运算符操作的都是boolean类型的变量

//区分& 与 &&

//相同点1: & 与 && 的运算结果相同

//相同点2: 当符号左边是true时,二者都会执行符号右边的运算

//不同点: 当符号左边是false时, &继续执行符号右边的运算。&&不再执行符号右边的运算。

//开发中,推荐使用&&

//开发中,如果希望变量实现+2的操作,有几种方法?(前提: int num = 10;)

//方式一: num = num + 2;

//方式二: num += 2; (推荐)

//开发中,如果希望变量实现+1的操作,有几种方法?(前提: int num = 10;)

//方式一: num = num + 1;

//方式二: num += 1;

//方式三: num++; (推荐)

```
int c=0;
Label: for (int i = 2; i <=1000; i++) {
    for (int j = 2; j <=Math.sqrt(i) ; j++)
        if(i%j==0){
            continue Label;
        }
    c++;
}

System.out.println(c);
long end=System.currentTimeMillis();
System.out.println(end-start);

long start=System.currentTimeMillis();
boolean isflag=true;
int c=0;
for (int i = 2; i <=1000; i++) {
    for (int j = 2; j <=Math.sqrt(i) ; j++) {
        if(i%j==0){
            isflag=false;
            break;
        }
    }
    if(isflag){
        c++;
    }
    isflag=true;
}

System.out.println(c);
long end=System.currentTimeMillis();
System.out.println(end-start);
```

