

Java基础语法

ZCX

一、功能单元-方法

- 方法：功能单元，封装了功能，便于代码复用，提高代码的复用性，提高代码的维护性，提高代码的扩展性。
- 方法可以相当于函数。
- 功能的最小单位是一个一个的方法。

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("hello world!");  
        System.out.println(getCode());    //在主函数内调用getCode()方法  
        print();                          //在主函数内调用print()方法  
        System.out.println(add(1,2));    //在主函数内调用add()方法  
    }  
  
    //定义一个方法，帮我生成一个验证码  
    public static String getCode(){        //这就是一个方法  
        String code = "";  
        for (int i = 0; i < 4; i++) {  
            int num = (int)(Math.random()*10);  
            code += num;  
        }  
        return code;  
    }  
  
    //定义一个方法输出三行你好世界  
    public static void print(){            //这也是一个方法  
        for (int i = 0; i < 3; i++) {  
            System.out.println("你好世界");  
        }  
    }  
  
    //定义一个方法表示两数相加  
    public static int add(int a,int b){    //这同样也是一个方法  
        return a+b;  
    }  
}
```

二、Java基本语法-注释

- 注释是程序员在程序中添加的文字，用来解释代码的含义，让程序员方便自己和他人查看，以便理解

1. java注释的分类

文档注释	单行注释	多行注释
<code>/** */</code>	<code>//</code>	<code>/* */</code>

```
package com.cxz.comment;

public class comment_demo {

    /**
     * 一般是在主函数上面或者整个程序上面写文档注释
     * 比如这是一个main方法，是程序的入口，有他才能执行程序
     * 这就是一个文档注释
     */
    public static void main(String[] args) {
        //目标:认识三种注释
        //1.单行注释
        //这就是单行注释

        /*
        2.多行注释
        这是多行注释
        */
    }
}
```

2. 注释的特点

- i. 注释不影响程序的运行,javac编译后的class文件会把所有注释全去掉
- ii. AI可以自动生成代码注释，但是需要手动修改
- iii. 比如我选中生成验证码的方法,选择AI生成注释得到以下文字,可以选择直接插入生成的注释

```
/**
 * 生成一个四位的随机数字验证码
 *
 * 该方法用于生成一个四位的数字验证码，用于用户验证、安全校验等场景
 * 验证码由0-9之间的随机数组成，每次调用该方法时生成的验证码都不相同
 *
 * @return 生成的四位数字验证码，以字符串形式返回
 */
public static String getCode(){
    // 初始化验证码字符串，此时为空
    String code = "";
    // 循环四次，每次生成一个随机数字并将其添加到验证码字符串中
    for (int i = 0; i < 4; i++) {
        // 生成一个0到9之间的随机数
        int num = (int)(Math.random()*10);
        // 将生成的随机数字添加到验证码字符串中
        code += num;
    }
    // 返回生成的四位数字验证码
    return code;
}
```

三、Java基本语法-字面量

- 字面量：在程序中直接出现的值，如：1,2,3,true,false,null,hello,world,etc.

1. 字面量的分类

常用数据	程序中怎么写	说明
整数	6,8,10	不需添加什么字符
小数	1.2,3.14	不需添加什么字符
字符	'a','我'	程序中必须使用单引号,且只能有一个字符
字符串	"helloworld","你好世界"	程序中必须使用双引号,且可以有多个字符
布尔值	true,false	只有两个值true:真和false:错
空值	null	一个特殊的值
特殊字符字面量	\t,\n,\r,\b,\f	\t:制表符,\n:换行,\r:回车,\b:退格,\f:换页

```
package com.cxz.literal;

public class literal_demo {
    /**
     * 目标:掌握常见字面量的书写格式
     */
    public static void main(String[] args) {
        printlnliteral();
    }
    public static void printlnliteral(){
        //请帮我直接输出常见的字面量
        System.out.println(true);    //输出布尔型字面量
        System.out.println(false);
        System.out.println(1);    //输出整型字面量
        System.out.println(1.0);    //输出浮点型字面量
        System.out.println('a');    //输出字符型字面量,且有且有一个字符
        System.out.println("hello world");    //输出字符串字面量
        System.out.println('a'+1);
        System.out.println("hello\nworld\n");    //\n换行
        System.out.println("hello\tworld");    //\t制表符,表示缩进
    }
}
```

四、Java基本语法-变量

- 变量：用来存储数据的容器，用来保存数据，用来接收数据，用来操作数据。
- 定义格式：数据类型 变量名 = 值;

```
package com.cxz.variable;

public class variable {
    public static void main(String[] args) {
        //目标：认识变量
        variable_demo();
    }
    //定义一个方法，学习变量的定义
    public static void variable_demo(){
        //定义一个变量
        int a = 10;
        //定义一个变量
        int b = 20;
        //定义一个变量
        int c = a + b;
        //定义一个变量
        String name = "cxz";
        //定义一个变量
        char gender = '男';
        //定义一个变量
        double score = 99.99;
        //定义一个变量
        boolean flag = true;
        //定义一个变量
        int age = 18;
        //输出上述变量
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(name);
        System.out.println(gender);
        System.out.println(score);
        System.out.println(flag);
        System.out.println(age);
    }
}
```

1. 变量的特点

- i. 减少重复，提高处理数据的灵活性和维护性
- ii. 变量里装的数据可被替换

```
public static void variable_replace(){
    int a = 10;
    a = a+2;
    System.out.println(a);
}
```

a的值可以被替换,输出从10变为12

2. 变量实际应用

//微信钱包目前有9.5元，收到6元的红包，又发出去5元设计一个方法，实时输出目前钱包的钱数

```
public static void wallet(){
    //定义一个变量，表示微信钱包的余额
    double money = 9.5;
    System.out.println(money);
    //定义一个变量，表示收到的红包
    double redpacket = 6;
    money=money+redpacket;
    System.out.println(money);
    //定义一个变量，表示发出去的红包
    double redpacket2 = 5;
    money=money-redpacket2;
    System.out.println(money);
}
```

3. 变量的存储原理

i. 常量数据进行二进制存储:逢二进一,整数部分除二取余,小数部分乘二取1

二进制转十进制,十进制转二进制

ii. 字节是数据在计算机中最小的存储单元(byte,简称B)

1字节=8位(8比特位)

即 **1B=8bit**

iii. 数据单位:KB,MB,GB,TB等

1B=8bit 1KB=1024B 1MB=1024KB 1GB=1024MB 1TB=1024GB

iv. 字符数据同样是用二进制存储的,但是存储的是字符的ASCII码值,通过ASCII码一一对应每个字符

```
char ch='A';
System.out.println(ch+1);
char ch1='a';
System.out.println(ch1+1);
```

输出为66和98

v. 图片数据同理,存储的是图片的像素值,通过像素值(由三原色数据大小共同决定)一一对应每个像素点的颜色

vi. 声音数据同理,声音波纹绘制波形图,每个值对应一个二进制,存储二进制

4. 各种进制的转换

i. 二转八:三位一组

ii. 二转十六:四位一组

iii. 八转十六:每位转成三位二进制,再四位一组转成十六进制

iv. 十转八:先转二,再转八

v. 十转十六:先转二,再转十六

5. 变量的数据类型

- 分类:基本数据类型,引用数据类型

- **基本数据类型:**

数据类型	数据书写	内存占用 (字节数)	数据范围
整型	byte	1	-128~127
	short	2	-32768~32767
	int(默认)	4	-2147483648~2147483647
	long	8	-9223372036854775808~9223372036854775807
浮点型	float	4	-3.4028235E+38~3.4028235E+38 (IEEE754标准 共32位,1位符号位,8位整数位,23位小数位)
	double(默认)	8	-1.7976931348623157E+308~1.7976931348623157E+308
字符型	char	2	0~65535
布尔型	boolean	1	true,false

• 以下为**实例**

```
//掌握八种基本数据类型的定义
public static void basic_data_type(){
int a = 10;
System.out.println(a);
char ch = 'A';
System.out.println(ch);
double d = 99.99;
System.out.println(d);
boolean flag = true;
System.out.println(flag);
byte b = 127;
System.out.println(b);
short s = 32767;
System.out.println(s);
long l = 999999999999999999L;
```

注意一般整型数据默认为int,如果不在数据后面加L,则会报错(虽然用的是long定义但是超了int范围),**long需要加L**才行(才能表示long数据类型)

```
System.out.println(1);
***float f = 99.99f;***
```

注意一般浮点型数据默认为double,如果不在数据后面加f,则会报错(虽然用的是float定义但是超了double范围),**float需要加f**才行(才能表示float数据类型)

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

- **引用数据类型(其一)**——string,用于存储字符串对象

```
//输出一个引用数据类型  
String str = "cxz";  
System.out.println(str);
```

五、Java基本语法-关键字、标识符

- **关键字**:指编程语言本身预留出来的,具有特殊意义的单词,不能用作变量名,方法名,类名等,用于定义程序的结构,控制流程,或数据类型等,如
if,else,for,while,do,break,continue,return,class,extends,implements,interface,new,this,super,static,final,abstract等
- **标识符**:用于标识变量,方法,类,接口,数组等,由字母,数字,下划线组成,不能以数字开头,区分大小写,不能使用关键字作为标识符.合法的标识符如:
dsdfdsee,_fsfgjoms,\$fgnsmde等