转义字符

2017年5月13日 10:00

代码所在位置：
C:\others\code\JAVA\ABCjichu
\Escapedcharacters.java

常见的转义字符
\b backspace(退格键)
\n Linefeed newline（换行）
\r Carriage Return（回车）将当前位置移到本行开头
\t 一个tab键一般等于4个空格；
\\ Backslash （反斜杠)

在windows系统中，如果是操作文件的时候需要换行，是需要\r\n一起使用。

九九乘法表

```
public void simples2(){
        for(int i = 1;i<=9;i++){
            for(int j = 1;j<=i;j++){
                System.out.print(i+"*"+j+"="+i*j+"\t");
            }
            System.out.println();
        }
    }
```

转义字符	意义	ASCII码值（十进制）
\a	响铃(BEL)	007
\b	退格(BS)，将当前位置移到前一列	008
\f	换页(FF)，将当前位置移到下页开头	012
\n	换行(LF)，将当前位置移到下一行开头	010
\r	回车(CR)，将当前位置移到本行开头	013
\t	水平制表(HT)（跳到下一个TAB位置）	009
\v	垂直制表(VT)	011
\\	代表一个反斜线字符"\'	092
\'	代表一个单引号（撇号）字符	039
\"	代表一个双引号字符	034
\?	代表一个问号	063
\0	空字符(NULL)	000
\ooo	1到3位八进制数所代表的任意字符	三位八进制
\xhh	1到2位十六进制所代表的任意字符	二位十六进制

函数基本知识

2017年5月13日 10:02

代码位置：C:\others\code\JAVA\ABCjichu\OverLoaded.java

1、函数的基本格式

修饰符 返回值类型 变量名（形式参数）{

函数体

}

2、特点：

1）提高代码复用性

2）将功能代码进行封装

3、return 关键字的作用

1、返回数据给函数调用者

2、在返回值类型为void的函数中，可以使用return;不能返回具体的值 return a;这种写法在返回值类型为void的函数中就会报错。

3、可以使用return;结束一个函数的运行。

注意：如果一个函数的返回值类型是具体的数据类型，那么该函数就必须保证在任意情况下，该函数都有返回值。

```
public String getGrade(int score){
     if(score>=90&&score<=100){
          return "A";
     }else if (score<90&&score>=80) {
          return "B";
     }else if(score<80&&score>60){
          return "C";
     }else if(score <=60){
          return "D";
     }
}
```

上面这段带有就有错误，不能运行。

-------函数重载---------

```
public void add(int a,int b){
          System.out.println(a+b);
     }
     public void add(int a,int b,int c){
          System.out.println(a+b+c);
     }
```

在一个类中出现两个或者两个以上的同名函数，这个称作为函数的重载

作用：同一个函数名可以出现了不同的函数，以应对不同个数或者不同类型的参数

要求：

1、函数名一致

2、形式参数列表不一致（参数个数，参数对应的数据类型）

3、与函数返回值类型无关

数组

2017年5月13日 13:08

数组：数组是同一种数据类型的集合容器

定义格式：

数据类型[] 变量名 = new 数据类型[长度];

int[] scores = new int[50];

解析数组：

左边：int[] scores 声明了一个int类型的数组变量，

变量名为scores

int 表示该数组只能存储int类型的数据。

[] 表示这是一个数组类型

右边：new int[50]: 创建了一个长度为50的int类型数组对象。

new ：创建数组对象的关键字

Int：表示该数组对象只能储存int类型数据

int[] scores = new int[4];

上面这个程序段是定义在main方法内部在。因此会在栈内存中定义一个变量scores。

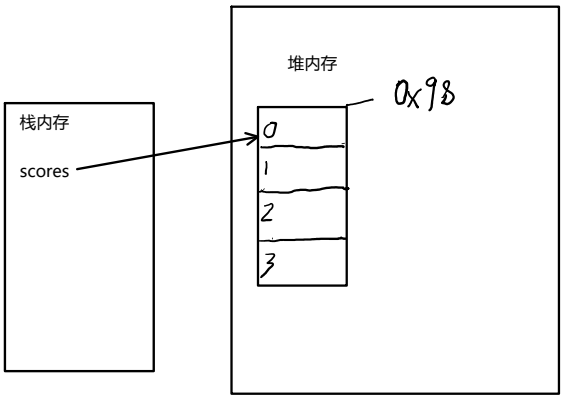
同时注意：

凡是以new关键字创建的对象，jvm(java虚拟机）都会在堆内存中开辟一个新的

右上角的0x98表示数组在堆内存中的地址

=赋值运算符

就是把数组对象的内存地址赋值给变量scores



局部变量：

如果一个变量是在一个方法（函数）的内部声明的，那么改变量就是一个局部变量

成员变量:

成员变量就是定义在方法外的，类之内的。

```
public class Array {
    private int b = 5;//成员变量。
    public static void main(String[] args){
        int a = 0;//局部变量
        int[] scores = new int[4];
    }
```

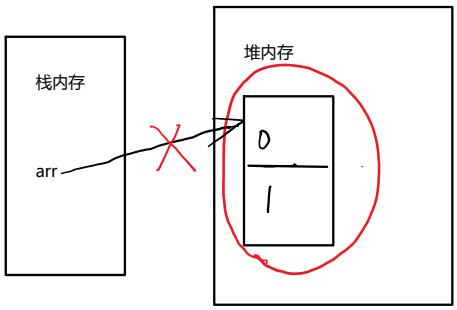
栈内存的特点：栈内存存储的存储的是局部变量变量一旦出了自己的作用域，那么就马上会从内存中消失，释放内存空间。

堆内存的特点：堆内存存储的都是对象数据，对象一旦被使用完，并不会马上从内存中消失，而是等待垃圾回收器不定时的把垃圾对象回收，这时候对象才会消失，释放内存。

对象使用完：对象如果没有变量引用它，它就是一个垃圾对象了。

对象数据：

除了8种基本数据类型，也称为引用类型数据。




```
int[] arr = new int[2];
arr = null;
arr[1] = 10;
System.out.println(arr[1]);
```

执行arr = null，之后图中红圈包围起来的对象就是没有变量引用，此时，她就是一个垃圾对象。

数组遍历

查看数组所有元素

for语句。

数组初始化

动态初始化：

```
int[] arr = new int[5];
```

静态初始化：

int[] arr = {1,2,3,4,5};

静态初始化的内存图和动态的一样，虽然省略的new，但是jvm会自动帮你加上。

数组中常见问题：

NullPointException 空指针异常

原因：引用类型变量没有指向任何变量，而访问了对象的属性或者调用了对象的方法。

ArrayIndexOutOfBoundsException 索引值越界（下标越界）

如果程序一开始就确定了数据，建议使用静态初始化。

动态数据，从控制台输入的情况

```
int[] arr = new int[4];
Scanner scanner = new Scanner(System.in);
for(int i = 0;i<arr.length;i++){
    arr[i] = scanner.nextInt();
}
System.out.println(arr[3]);
```

编程实践

1、需求：定义一个函数，接收一个int类型的数组对象，找出数组对象中的最大元素，然后返回。 Integer.MIN_VALUE; **int max = arr[0].**

```
public int getMaxNum(int[] arr){
    int max = Integer.MIN_VALUE;
    for(int i = 0;i<arr.length;i++){
        if(max<arr[i]){
            max = arr[i];
        }
    }
    return max;
}
```

2、选择排序

选择排序之降序排序：就是从未排序的数据中选择一个最大的元素，放入到已排序的后面

```
public void sortByMax(int[] arr){
    for(int i = 0;i<arr.length;i++){
        for(int j = i;j<arr.length;j++){
            if(arr[i]<arr[j]){
                //交换位置
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

3、线性查找

查找指定数在数组中的位置，找到返回下标，找不到返回-1

求数组中的最大值，最小值

4、需求：目前存在的数组：int arr = {0,0,12,1,0,4,6,0},编写一个函数接收该数组然后把数组的0清空，返回一个不存在 0元素的数组。

首先：应该计算数组中0的个数，然后才能知道新定义的数组的长度是多少。

```
package jichu;
import java.util.Arrays;
public class Array {

    public static int[] clearZero(int[] arr){
        //计算数组中值不为0的个数
        int count = 0;
        for(int i =0;i<arr.length;i++){
            if(arr[i] == 0){
                count++;
            }
        }
        int[] newArr = new int[arr.length-count];
        int index = 0;//用于新数组的下标
        for(int i = 0;i<arr.length;i++){
            if(arr[i]!=0){
                newArr[index] = arr[i];
                index++;
            }
        }
        return newArr;
    }
```

```
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

升序的话只需要将图中的第二个方框中的“<”改成”>“即可。

```
    return newArr;
}


public static void main(String[] args) {
        int[] arr = {0,0,12,1,0,4,6,0};
        int[] newArr = clearZero(arr);
        //需要导入包文件  import java.util.Arrays;
        System.out.println(Arrays.toString(newArr));
        //[12,1,4,6]
    }
}
```

面向对象（1）

2017年5月13日 21:30

代码位置：C:\others\code\JAVA\ABCjichu\Object

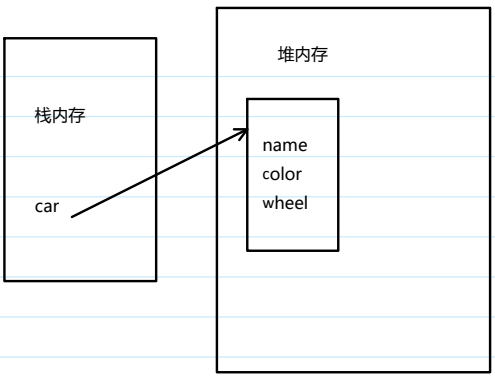
面向对象内存分析

```
Car car = new Car();
car.name = "BMW";
car.color = "red";
car.wheel = 5;
//默认值输出是null,null,0
System.out.println(car.name+","+car.color+","+car.wheel);
```

注意：成员属性有默认的初始值

数据类型	int	double	float	char	String	引用数据类型	boolean
默认值	0	0.0	0.0	''	null	null	false

对象一旦创建，对象的成员变量也会发生分配默认初始值



一个小例子

有两个事物，车和修车厂。

车具有公共属性：轮子数、名字、颜色，还具有跑的功能行为。跑之前要检测轮子个数是否少于4个如果少于四个，需要送到修车厂修理，修理之后，轮子数要补充回来4个。然后车就可以继续跑起来

修车厂具有公共属性名字、地址和电话。公共的行为：修车

注意事项：

1、变量在同一个作用域（大括号）上是可以直接访问的。

2、如果一个类要访问另一个类的变量时，这个时候只能通过对象进行访问

```
class Car{
    public String name;
    public String color;
    public int wheel;

    public void run(){
        //这里可以直接使用wheel，因为在相同的作用域中
        if(wheel >= 4){
            System.out.println("The car can run!");
        }else{
            System.out.println(name+"轮子数是"+wheel);
        }
    }
}
class Garage{
    private String name;
    private String address;
    private String tel;

    public void repair(Car c){
        if(c.wheel>=4){
            System.out.println("not repair");
        }else if(c.wheel<4){
            System.out.println("need repair");
            c.wheel = 4;
        }
    }
}
```

匿名对象

没有引用类型变量指向的对象称作为匿名对象。

new Student()就是一个匿名对象

```
new Car().wheel = 4;
System.out.println(new Car().wheel);//0
```

1、我们一般不会给匿名对象赋予属性值，因为永远无法获取

2、两个匿名对象永远不可能是同一个对象。

```
//==号用于引用类型变量的时候，比较的是两个对象的内存地址。
System.out.println(new Car() == new Car());//false
```

应用场景：

1、如果一个对象需要调用一个方法，而调用完这个方法之后，该对象就不再使用，这时候就可以使用匿名对象。

2、可以作为实参调用一个函数。

```
public void repair(Car c){
    if(c.wheel>=4){
        System.out.println("not repair");
    }else if(c.wheel<4){
        System.out.println("need repair");
        c.wheel = 4;
    }
}
}

public class Object {
    public static void main(String[] args){
//      调用repair方法
        Garage g = new Garage();
//      Car c = new Car();
//      g.repair(c);
//      这种写法需要专门创建一个Car对象用于传入参数，因此在只需要使用一次Car对象的时候，可以使用下面
```

成员变量与局部变量的区别

1、局部变量是定义在方法内部的。

```
public void test(){
    for(int i = 0;i<4;i++){
        System.out.println(i);
    }
    System.out.println(i);//这里会出错
}
```

这里的 i是局部变量，只能在for循环中访问。

2、成员变量是定义在方法之外，类之内的。

作用上的区别：

1、成员变量用于描述一类事物的公共属性

2、局部变量就是提供一个变量给方法内部使用。

生命周期的区别：

1、成员变量：随着对象的创建而创建，随着对象的消失而消失

```
class Car{
    public String name;
}
```

Car c = new Car();//new 之后name才出现在内存中，也就是在创建Car对象的时候才出现在内存中。

2、局部变量：在调用对应的方法时，执行到了创建变量语句时存在，局部变量一旦出了自己的作用域就马上从内存中消失。比如上面的test()方法。

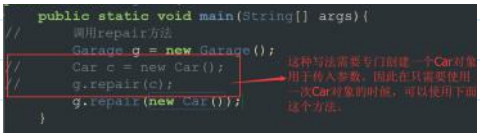
初始值的区别：

1、成员变量是有默认的初始值的

2、局部变量没有默认的初始值，必须要初始化才能使用。

```
public void test(){
    int i;
    System.out.println(i);//这里会出错
}
```

这里会报错，因为没有初始化。


public static void main(String[] args){
//   调用repair方法
        Garage g = new Garage();
//   Car c = new Car();
//   g.repair(c);
        g.repair(new Car());
    }
这种写法需要专门创建一个Car对象
用于传入参数，因此在只需要使用
一次Car对象的时候，可以使用下面
这个方法。

面向对象（2）

2017年5月14日 20:51

封装

1、权限修饰符

public：public修饰的成员变量或者方法，任何人都可以访问。

private：私有的，private修饰的成员变量或者方法只能在本类中进行直接访问。

2、封装步骤

1) 提供private修饰要封装的属性

2）提供公共的方法设置私有的属性和获取该私有的成员属性

命名规范：

set属性名 setName()

get属性名 getName()

疑问：封装一定要使用get和set方法吗？？？

不一定，是根据需求而定的。

规范：在现实开发中，一般实体类的所有成员属性（成员变量）都要封装起来。

实体类：实体类就是用于描述一类事物的。

工具类： Arrays（数组）就是工具类。

好处：1、提高了安全性 2、操作简单 3、隐藏了实现

构造函数

作用：给对应的对象进行初始化

定义格式：

修饰符 函数名（形式参数）{

函数体...

}

注意：

1、构造函数没有返回值

2、构造函数名必须与类名一致

3、构造函数并不是我们自己手动调用的，而是在创建对象的时候，JVM就会主动调用到对应的构造函数

4、如果一个类没有显式的写上一个构造方法，java编译器在编译的时候会自动加一个无参数的构造函数。

5、如果一个类已经显式的写上一个构造方法时，那么java编译器则不会在为该类添加一个无参的构造方法。

6、构造函数可以在一个类中以函数重载的形式存在多个。

疑问：java编译器添加的无参数构造方法的权限修饰符是什么？

```
public class Person{
    int id;
    String name;

    public Person(int id,String name){
        this.id = id;
        this.name = name;
    }
    public void cry(){
        System.out.println("cry");
    }
}
```

与类的权限操作符是一致的。如果类没有权限符（一个.java文件中只能有一个public修饰的类），则java编译器添加的构造函数是Person()..就没有public修饰。

构造函数与普通函数的区别：

1、构造函数没有返回值类型，而普通函数有返回值类型。

2、函数名的区别

3、调用方式的区别

4、作用上的区别

构造函数用于初始化对象。

该看07 3 构造代码块。

一个小例子：

需求：描述一个计算器类，具备操作数1，操作数2，运算符这三个属性，还具备计算功能

要求：不能直接对操作数1，操作数2，运算符进行直接的赋值。

定义运算符可以直接使用char类型，因为它是一个单独的+ - * /

实现

```
class Calculator{
    private int num1;
    private int num2;
    private char option; //运算符是单个字符，可以直接使用char类型定义

    public void initCalculator(int n1,int n2,char c){
        //不需要判断n1，n2，因为如果输入的是abc这样的字符串，编译就会出现问题
        //这就是java强类型的好处
        this.num1 = n1;
        num2 = n2;
        if(c == '+' || c == '-'|| c =='*' || c== '/'){
            //注意这里是char类型，因此可以直接使用 ==  ，而不是使用equals
            //注意是单引号
            option = c;
        }else{
            option = '+';//默认是做加法运算。
            System.out.println("操作符输入不在预定范围内");
        }
    }

    public int compute(){
        int result  = 0;
        switch(this.option){
        case '+':
            result = this.num1+this.num2;
            break;
        case '-':
            result = this.num1 - this.num2;
            break;
        case '/':
            if(num2 != 0){
                result = this.num1 / this.num2;
            }else{
                System.out.println("除数不能为0");
            }
            break;
        default:
            System.out.println("不能进行此类运算");
            break;
        }
        return result;
    }
}
```

面向对象（3）--小结

2017年5月15日 19:01

对象：真是存在的唯一的事物

类：同一种类型的事物公共属性与公共行为的抽取

找对象方式：

1、sun已经定义好了很多的类，只需要认识这些类就可以创建对象使用

2、通过自定义类创建对象

自定义类：

1、声明一个类

```
class 类{
        事物的公共属性；
        事物的公共行为；
}
```

2、通过类创建对象

3、通过对象访问或者设置对象的属性和方法

成员变量和局部变量的区别：成员变量与局部变量的区别

1、定义位置的区别

1）成员变量是定义在方法之外，类之内的。

2）局部变量是声明在方法内部的变量

2、作用上的区别：

1）成员变量的作用是描述一类事物的属性

2）局部变量作用是提供一个变量给方法内部使用

3、声明周期区别：

1）成员变量随着对象的创建而存在，随着对象的消失而消失

2）局部变量是当该方法执行到了创建改变量的语句时存在，

一旦出了自己的作用域，马上从内存中消失。

4、初始值的区别：

1、成员变量是有默认初始值的。

2、局部变量没有默认初始值，必须初始化之后才能使用

匿名对象：匿名对象

封装：封装