



链滴

Java 学习之第 2 章 java 编程基础

作者: [xiaoqian](#)

原文链接: <https://ld246.com/article/1642560123290>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

第2章 java编程基础

2.1 Java的基本语法

2.1.1 Java代码的基本格式

修饰符 class 类名{

程序代码

}

注意点:

1. Java是严格区分大小写的
2. 每条功能执行语句后都要用分号结束 (与C语言类似)
3. Java程序中一个连续的字符串不能分开在两行中书写, 但可以分开后用+连接

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

2.1.2 Java中的注释

1. 单行注释

```
int x = 1;    //定义一个整型变量x
```

2. 多行注释

```
/* int x = 1;  
   int y = 2; */
```

3. 文档注释

```
/**  
 * Person实体类  
 */
```

2.1.3 Java中的标识符

可以由字母、数字、美元符号和下划线组成, 但不能以数字开头, 也不可以使用关键字

命名规范:

1. 类名和接口名首字母一律大写
2. 包名一律小写

3. 常量名一律大写

4. 变量名和方法名第一个单词首字母小写，从第二个开始每个单词首字母大写

2.2.3 变量的类型转换

强制类型转换

```
package com.itheima.example;

public class Example01 {
    public static void main(String[] args) {
        int i = 5;
        byte b = (byte) i;
        System.out.println(b);
    }
}
```

2.5 选择语句结构

2.5.1 if条件语句

```
package com.itheima.example;

public class Example02 {
    public static void main(String[] args) {
        int month=1;
        if (month>=3 && month <= 5){
            System.out.println("春季");
        }else if (month>=6 && month <= 8){
            System.out.println("夏季");
        }else if (month>=9 && month <= 11){
            System.out.println("秋季");
        }else{
            System.out.println("冬季");
        }
    }
}
```

2.5.2 switch语句

```
package com.itheima.example;

public class Example03 {
    public static void main(String[] args) {
        String country = "中国";
        switch (country) {
            case "中国":
                System.out.println("你好: "+country);
                break;
            case "美国":
                System.out.println("hello: "+country);
        }
    }
}
```

```
        break;
    default:
        System.out.println("输入错误");
        break;
    }
}
}
```

2.6 循环语句

2.6.1 while循环语句

```
package com.itheima.example;

public class Example04 {
    public static void main(String[] args) {
        int i=1;
        int sum=0;
        while (i <= 10) {
            sum += i;
            i++;
        }
        System.out.println("sum="+sum);
    }
}
```

2.6.2 do-while语句

```
package com.itheima.example;

public class Example05 {
    public static void main(String[] args) {
        int i=1,sum=0;
        do {
            sum+=i;
            i++;
        } while (i<=10);
        System.out.println("sum="+sum);
    }
}
```

注意区别:

while和do-while的区别:

while: 先判断再执行

do-while: 先执行一次再判断

2.6.3 for循环语句

```

package com.itheima.example;

public class Example06 {
    public static void main(String[] args) {
        int sum=0;
        for (int i = 0; i <=10;i++){
            sum += i;
        }
        System.out.println("sum="+sum);
    }
}

```

2.6.4 循环嵌套

使用循环嵌套用#输出直角三角形

```

package com.itheima.example;

public class Example07 {
    public static void main(String[] args) {
        for (int i = 1; i <=8;i++){
            for (int j = 1; j <= i; j++) {
                System.out.print("#");
            }
            System.out.print("\n");
        }
    }
}

```

此处有一个问题，在java中

System.out.print

System.out.println

这两个是不一样的，下面一个输出后进行一次换行

2.6.5 跳转语句

1.break语句

出现在switch中时是终止某个case并向下执行

出现在循环中时是跳出当前循环，执行后面代码

```

package com.itheima.example;

public class Example08 {
    public static void main(String[] args) {
        int x=1;
        while (x <=5){
            System.out.println("x="+x);
            if (x==4){

```

```
        break;
    }
    x++;
}
}
```

2.continue语句

终止本次循环，执行下一次循环

```
package com.itheima.example;
/**
 * 对1-100之内的偶数进行求和
 */
public class Example09 {
    public static void main(String[] args) {
        int sum=0;
        for (int i = 1; i <=100;i++){
            if (i%2!=0){
                continue;
            }
            sum+=i;
        }
        System.out.println("sum="+sum);
    }
}
```

2.7 方法(其实就是函数)

2.7.1 方法的概念

```
package com.itheima.example;

public class Example10 {
    public static void main(String[] args) {
        int area = getArea(4,5);
        System.out.println("矩形的面积为:"+area);
    }

    private static int getArea(int i, int j) {
        return i*j;
    }
}
```

2.7.2 方法的重载

```
package com.itheima.example;

public class Example11 {
    public static void main(String[] args) {
```

```

int sum1=add(2,3);
int sum2=add(2,3,4);
double sum3 = add(2.5,3.5);
System.out.println("sum1="+sum1);
System.out.println("sum2="+sum2);
System.out.println("sum3="+sum3);
}

private static double add(double d, double e) {
    return d+e;
}

private static int add(int i, int j, int k) {
    return i+j+k;
}

private static int add(int i, int j) {
    return i+j;
}
}

```

注意：方法的重载与返回值类型无关，只需要满足两个条件：1.方法名相同。2、参数个数或参数类不相同

2.8 数组

2.8.1数组的定义

```

int[] x= new int[10];

package com.itheima.example;

public class Example12 {
    public static void main(String[] args) {
        int []arr=new int[4];
        arr[0]=2;
        arr[1]=3;
        System.out.println("arr[0]="+arr[0]);
        System.out.println("arr[1]="+arr[1]);
        System.out.println("arr[2]="+arr[2]);
        System.out.println("arr[3]="+arr[3]);
        System.out.println("数组的长度是:"+arr.length);
    }
}

```

2.8.2 数组的常见操作

1.数组遍历

```

package com.itheima.example;

public class Example13 {

```

```
public static void main(String[] args) {
    int[] arr = {3,5,1,2,6};
    for (int i = 0; i < arr.length; i++){
        System.out.println(arr[i]);
    }
}
```

2.数组最值

```
package com.itheima.example;

public class Example14 {
    public static void main(String[] args) {
        int[] arr = {3,5,1,2,6};
        int max=arr[0];
        for (int i = 0; i < arr.length; i++){
            if(arr[i]>max){
                max = arr[i];
            }
        }
        System.out.println("max="+max);
    }
}
```

3.数组排序

```
package com.itheima.example;

public class Example15 {
    public static void main(String[] args) {
        int[] arr = {4,8,3,9,7,5};
        System.out.print("排序前:");
        printArray(arr);
        bubbleSort(arr);
        System.out.print("排序后:");
        printArray(arr);
    }

    private static void bubbleSort(int[] arr) {
        for (int i = 0; i < arr.length-1; i++){
            for (int j = 0; j < arr.length-1-i; j++){
                if(arr[j]>arr[j+1]){
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j+1]=temp;
                }
            }
            System.out.print("第" + (i+1) + "轮排序后:");
            printArray(arr);
        }
    }
}
```



```

private static void printArray(int[] arr) {
    for (int i = 0; i < arr.length; i++){
        System.out.print(arr[i]+" ");
    }
    System.out.print("\n");
}
}

```

2.8.3 Arrays工具类

Arrays是一个专门操作数组的工具类，位于java.util包中

方法声明

static void sort(int[] a)
升序进行排序

static int binarySearch(Object[] a, Object key)
用二分搜索法搜索指定数组，以获得指定对象

static int[] copyOfRange(int[] original,int from,int to)
指定数组的指定范围复制到一个新数组

static void fill(Object[] a,Object val)
Object引用分配给指定Object数组的每个元素

static String toString(int[] arr)
的字符串表示形式

功能描述

对指定的int兴数组按数

将指定的O

返回指定数组内

```
package com.itheima.example;
```

```
import java.util.Arrays;
```

```

public class Example16 {
    public static void main(String[] args) {
        int[] arr={9,8,3,5,2};
        int[] copied=Arrays.copyOfRange(arr, 1, 7);
        System.out.println("复制数组:"+Arrays.toString(copied));
        Arrays.sort(arr);
        System.out.println("排序后:"+Arrays.toString(arr));
        Arrays.fill(arr, 8);
        System.out.println("填充数组:"+Arrays.toString(arr));
    }
}

```

2.9 String类和StringBuffer类

2.9.1 String类

直接初始化:

```
String str1="abc";
```

使用构造方法:

```
String str1=new String();
String str2=new String)("abc");
```

方法声明

功能描述

<code>int indexOf(int ch)</code> 一次出现处的索引	返回指定字符在此字符串中
<code>int indexOf(String str)</code> 字符串中第一次出现处的索引	返回指定子字符串在此
<code>char charAt(int index)</code> 的字符,其中 index的取值范围是0 ~ 字符串长度-1	返回字符串中index位置
<code>boolean ends With(String suffix)</code> 是否以指定的字符串结尾	判断此字符
<code>int length</code>	返回此字符串的长度
<code>boolean equals(Object anObject)</code> 指定的字符串比较,如果相等则返回 true,否则返回false	将此字符串
<code>boolean isEmpty()</code> true	当且仅当字符串长度为0时返回
<code>boolean startsWith(String prefix)</code> 是否以指定的字符串开始	判断此字符
<code>boolean contains(CharSequence cs)</code> 串中是否包含指定的字符序列	判断此字
<code>String toLowerCase()</code> tring中的所有字符转换为小写	使用默认语言环境的规则将
<code>String toUpperCase()</code> String中的所有字符转换为大写	使用默认语言环境的规则将
<code>char[] toCharArray()</code> 组	将此字符串转换为一个字符
<code>String replace (CharSequence oldstr,CharSequence newstr)</code> 回一个新的字符串,它是通过利用newstr替换此字符串中出现的所有oldstr得到的	
<code>String[] split(String regex)</code> 的字符串分割为若干个子字符串	根据参数regex将原
<code>String substring(int beginIndex)</code> 串,它包含从指定的beginIndex处开始,直到此字符串末尾的所有字符	返回一个新字
<code>String substring (int beginIndex, int endIndex)</code> 回一个新字符串,它包含从指定的beginIndex处开始,直到索引endIndex-1处的所有字符	
<code>String trim</code> 首尾的空格	返回一个新字符串,它去除了原字符

```
package com.itheima.example;
```

```
public class Example17 {
    public static void main(String[] args) {
        String s="ababcAbc,dedcba";
        System.out.println("e第一次出现在:"+s.indexOf("e"));
        System.out.println("bc第一次出现在:"+s.indexOf("bc"));
        System.out.println("第8个位置上是:"+s.charAt(8));
    }
}
```

```

System.out.println("字符串是否以a结尾:"+s.endsWith("a"));
System.out.println("字符串是否以z结尾:"+s.endsWith("z"));
System.out.println("字符串长度为:"+s.length());
System.out.println("字符串长度为0?:"+s.isEmpty());
System.out.println("字符串是否以a开始:"+s.startsWith("a"));
System.out.println("字符串是否以z开始:"+s.startsWith("z"));
System.out.println("字符串是否包含了ded:"+s.contains("ded"));
System.out.println("转换为小写:"+s.toLowerCase());
System.out.println("转换为大写:"+s.toUpperCase());
System.out.println("将a全部替换为z:"+s.replace("a", "z"));
System.out.println("从第6个字符到结束:"+s.substring(6));
System.out.println("从第5个字符到第10个字符:"+s.substring(5,10));
System.out.println("返回一个去除空格的字符串:"+s.trim());
}
}

```

2.9.2 StringBuffer类

方法声明

功能描述

StringBuffer append(char c)
uffer对象中

添加参数到 String

StringBuffer insert(int offset,String str)
中的offset位置插入字符串str

在字符

StringBuffer delete(intstart,int end)
Buffer对象中指定范围的字符或字符串序列

删除 Strin

StringBuffer deleteCharAt(int index)
指定位置的字符

移除此序

StringBuffer replace(int start, int end,String s)
StringBuffer对象中替换指定的字符或字符串序列

void setCharAt(int index, char ch)
置index处的字符序列

修改指定

StringBuffer reverseO
取代

将此字符序列用其反转形

String toString(
符串

返回 StringBuffer 缓冲区中的

```
package com.itheima.example;
```

```

public class Example18 {
    public static void main(String[] args) {
        System.out.println("*****1.添加*****");
        add();
        System.out.println("*****2.删除*****");
        remove();
        System.out.println("*****3.修改*****");
        alert();
    }

    public static void alert() {

```

```

StringBuffer sb = new StringBuffer("abcdefg");
sb.setCharAt(2, 'h');
System.out.println("修改指定位置字符结果:" + sb);
sb.replace(1, 3, "yy");
System.out.println("修改指定位置字符(串)结果:" + sb);
System.out.println("字符串翻转结果:" + sb.reverse());
}

public static void remove() {
    StringBuffer sb = new StringBuffer("abcdefg");
    sb.delete(2, 4);
    System.out.println("删除指定位置结果:" + sb);
    sb.deleteCharAt(3);
    System.out.println("删除指定位置结果:" + sb);
    sb.delete(0, sb.length());
    System.out.println("清空缓冲区结果:" + sb);
}

public static void add() {
    StringBuffer sb = new StringBuffer();
    sb.append("abcdefg");
    System.out.println("append添加结果:" + sb);
    sb.insert(3, "321");
    System.out.println("insert结果:" + sb);
}
}

```

2.10 包装类

基本数据类型 应的包装类	对应的包装类	基本数据类型	基本数据类型
char	Character	long	Lo
g			
byte	Byte	float	Float
int	Integer	double	Doub
e			
short	Short	boolean	Bo
lean			

装箱：指将基本数据类型的值转为引用数据类型

拆箱：将引用数据类型的对象转为基本数据类型

END~~~~