# 实验六 Java API

【实验目的】

1. 掌握String类和StringBufffer类。

2. 掌握其它系统类的应用。

【实验要求】

1．编写使用字符串类的程序。

2. 编写程序练习其它系统类的应用。

【实验内容】

1.编写一个java程序,完成以下功能:

* 1. 声明一个名为s的字符串,并使它的内容为”Call me Ishmael.”;
  2. 打印整个字符串
  3. 使用length()方法打印字符串的长度
  4. 使用charAt()方法打印字符串的第一个字符
  5. 使用charAt()和length()方法打印字符串的最后一个字符
  6. 使用indexOf()方法和substring()方法打印字符串中的第一个单词。

public class Main {  
 public static void main(String[] args) {  
 String s = "Call me Ishmael.";  
 System.*out*.println(s);  
 System.*out*.println(s.length());  
 System.*out*.println(s.charAt(0));  
 System.*out*.println(s.charAt(s.length() - 1));  
 System.*out*.println(s.substring(s.indexOf("Call"), 4));  
 }  
}

2. 设x,y是整型变量,d是双精度型变量,使用Math类编写一个完整的程序。

(1)求x的y次方。

(2)求x和y的最小值。

(3)求d取整后的结果。

(4)求d四舍五入后的结果。

(5)求atan(d)的数值。

public class Q2 {  
 public static void main(String[] args) {  
 int x = 2, y = 3;  
 double d = 114.514d;  
 System.*out*.println(Math.*pow*(x, y));  
 System.*out*.println(Math.*min*(x, y));  
 System.*out*.println(Math.*floor*(d));  
 System.*out*.println(Math.*round*(d));  
 System.*out*.println(Math.*atan*(d));  
 }  
}

1. 教材例6.1-6.20

6-1:

package Src.com.learn;

public class Demo\_03 {

public static void main(String[] args)

{

String str1 = new String();

String str2 = new String("abcd");

char[] charArray = new char[]{'D','E','F'};

String str3 = new String(charArray);

byte[] arr = new byte[]{97,98,99};

String str4 = new String(arr);

System.out.println("a" + str1 + "b");

System.out.println(str2);

System.out.println(str3);

System.out.println(str4);

}

}

6-2:

package Src.com.learn;

public class Demo\_04 {

public static void main(String[] args)

{

String s = "ababcdedcba";

// 获取字符串长度

System.out.println("字符串的长度为:" + s.length());

System.out.println("字符串中第一个字符：" + s.charAt(0));

System.out.println("字符c第一次出现的位置：" + s.indexOf("c"));

System.out.println("字符串c最后一次出现的位置：" + s.lastIndexOf("c"));

System.out.println("字符串ab第一次出现的位置：" + s.indexOf("ab"));

System.out.println("字符串ab最后一次出现的位置：" + s.lastIndexOf("ab"));

}

}

6-3:

package Src.com.learn;

public class Demo\_05 {

public static void main(String[] args)

{

String str = "abcd";

System.out.println("字符串转换为字符数组的结果：");

char[] charArray = str.toCharArray();

// 依次输出字符串的每个字符，中间采用逗号隔开

for(int i = 0;i < charArray.length; i++)

{

if(i == charArray.length - 1) System.out.print(charArray[i]);

else System.out.print(charArray[i] + ",");

}

System.out.println();

System.out.println("将int类型的值转换为String类型的字符串的结果是：" + String.valueOf(12));

System.out.println("将字符串转换为大写的结果为：" + str.toUpperCase());

}

}

6-4

package Src.com.learn;

public class Demo\_06 {

public static void main(String[] args) {

String s = "itcast";

// 字符串的替换操作

System.out.println("将it替换成cn.it的结果为：" + s.replace("it","it.cn"));

// 字符串的去除空格操作

String s1 = " itca st ";

// 去除字符串两端的空格

String trim = s1.trim();

System.out.println("去除字符串两端的空格的结果是：" + trim);

// 去除所有的空格

System.out.println("去除所有的空格之后的结果为：" + s.replace(" ",""));

}

}

6-5:

package Src.com.learn;

public class Demo\_07 {

public static void main(String[] args) {

String s1 = "String";

String s2 = "string";

System.out.println("判断s1字符串是否以Str开头：" + s1.startsWith("Str"));

System.out.println("判断s1字符串是否以ng结尾：" + s1.endsWith("ng"));

System.out.println("判断s1字符串是否包含字符串tri:" + s1.contains("tri"));

System.out.println("判断s1字符串是否为空：" + s1.isEmpty());

System.out.println("判断字符串s1和字符串s2的值是否相同：" + s1.equals(s2));

System.out.println("忽略大小写的情况下判断s1和s2的内容是否相同：" + s1.equalsIgnoreCase(s2));

System.out.println("按照对应字符的Unicode比较字符串s1和s2的大小(小于0则s2大，等于0则一样大，大于0则s1大):" + s1.compareTo(s2));

}

}

6-6:

package Src.com.learn.StringDemo;

public class Demo\_08 {

public static void main(String[] args) {

String str = "石家庄-武汉-哈尔滨";

// 字符串截取操作：

System.out.println("从第五个字符开始截取到末尾的结果为：" + str.substring(4));

System.out.println("从第五个字符开始截取到第六个字符结果为：" + str.substring(4,6));

// 字符串分割操作：

System.out.println("分割后字符串数组中的元素依次为：");

String[] split = str.split("-");

for(int i = 0;i < split.length; i++)

{

if(i==split.length - 1) System.out.print(split[i]);

else System.out.print(split[i] + ",");

}

}

}

6-7:

package Src.com.learn.StringDemo;

public class Demo\_09 {

public static void main(String[] args) {

try {

String s = "itcast";

System.out.println(s.charAt(8));

} catch (Exception e) {

System.out.println("数组越界异常！！！");

}

}

}

6-8:

package Src.com.learn.StringDemo;

public class Demo\_10 {

public static void main(String[] args) {

System.out.println("-----------------1.添加字符串--------------------");

add();

System.out.println("================================================");

System.out.println("-----------------2.修改字符串--------------------");

alter();

System.out.println("================================================");

System.out.println("-----------------3.删除字符串--------------------");

delete();

System.out.println("================================================");

System.out.println("-----------------4.截取.添加字符串--------------------");

sub();

}

//添加字符串

public static void add()

{

StringBuffer stringBuffer = new StringBuffer();

stringBuffer.append("Hello!!!");

stringBuffer.append("learn").append("yyds").append("8787");

System.out.println("append添加结果为：" + stringBuffer);

//在指定位置插入字符串

stringBuffer.insert(1,"yyds");

System.out.println("insert插入之后的结果为：" + stringBuffer);

}

//修改字符串

public static void alter()

{

StringBuffer stringBuffer = new StringBuffer("learnyyds");

stringBuffer.setCharAt(0,'Z');

System.out.println("修改指定位置的字符之后的结果为：" + stringBuffer);

stringBuffer.replace(7,10,"8787");

System.out.println("替换指定位置字符或者字符串的结果为：" + stringBuffer);

System.out.println("字符串反转之后的结果为：" + stringBuffer.reverse());

}

//删除字符串

public static void delete()

{

StringBuffer stringBuffer = new StringBuffer("learnyyds");

stringBuffer.delete(0,2);

System.out.println("删除指定范围的结果为：" + stringBuffer);

stringBuffer.deleteCharAt(3);

System.out.println("删除指定位置的元素的结果为：" + stringBuffer);

stringBuffer.delete(0,stringBuffer.length());

System.out.println("清空字符串缓冲区后：" + stringBuffer);

}

//截取字符串

public static void sub()

{

StringBuffer stringBuffer = new StringBuffer();

System.out.println("获取stringBuffer的初始容量:" + stringBuffer.capacity());

stringBuffer.append("learnyyds");

System.out.println("添加之后的结果为：" + stringBuffer);

System.out.println("截取第8-11个元素：" + stringBuffer.substring(7,11));

}

}

6-9:

package Src.com.learn.Array;

public class Demo\_01 {

public static void main(String[] args) {

int[] soueceArray = new int[]{10,11,12,13,14,15};

int[] targetArray = new int[]{20,21,22,23,24,25,26};

// 复制元素，从原数组索引为2的位置开始复制，复制长度为4，从目标数组的索引为3的位置开始存放

System.arraycopy(soueceArray,2,targetArray,3,4);

System.out.println("复制之后的数组为：");

for(int i = 0;i < targetArray.length; i++)

{

System.out.print(targetArray[i] + " ");

}

}

}

6-10:

package Src.com.learn.Array;

public class Demo\_10 {

private static long ans;

public static void main(String[] args) {

long begin = System.currentTimeMillis();

for(int i = 1;i <= 100000000; i++)

{

ans+=i;

}

long end = System.currentTimeMillis();

System.out.println("循环执行的时间为：" + 1.0\*(end - begin)/1000 + "s");

}

}

6-11:

package Src.com.learn.Array;

import java.util.Enumeration;

import java.util.Properties;

public class Demo\_11 {

public static void main(String[] args) {

// 获取当前系统的所有属性

Properties properties = System.getProperties();

// 获取当前系统属性所有的键

Enumeration propertyNames = properties.propertyNames();

// 只要这个键存在值

while(propertyNames.hasMoreElements())

{

// 获取这个键，同时指针指向下一个键

String key = (String) propertyNames.nextElement();

// 获取当前键对应的值

String value = System.getProperty(key);

System.out.println(key + "--->" + value);

}

}

}

6-12

: package Src.com.learn.Array;

public class Demo\_12 {

public static void main(String[] args)

{

// 创建Person对象

Person person = new Person("张三", 20);

// 将person置为null，使其成为垃圾

person = null;

// 调用gc方法进行垃圾回收

System.gc();

// 为了延长程序运行的时间，执行空循环

for(int i = 0;i < 100000; i++)

{

}

}

}

class Person{

private String name;

private int age;

public Person(String name,int age)

{

this.name = name;

this.age = age;

}

@Override

public String toString() {

return "name:" + this.name + "," + "age:" + this.age;

}

// 下面定义的finalize（）方法会在垃圾回收之前被调用

public void finalize() throws Throwable

{

System.out.println("对象被释放！！！");

}

}

6-13:

package Src.com.learn.Array;

public class Demo\_13 {

public static void main(String[] args) {

Runtime runtime = Runtime.getRuntime();

int number = runtime.availableProcessors();

System.out.println("处理器的个数：" + number);

System.out.println("空闲内存总量：" + runtime.freeMemory() / 1024 / 1024 + "MB");

System.out.println("最大可用内存量：" + runtime.maxMemory() / 1024 / 1024 + "MB");

System.out.println("内存总量：" + runtime.totalMemory() / 1024 / 1024 + "MB");

}

}

6-14:

package Src.com.learn.Array;

public class Demo\_14 {

public static void main(String[] args) {

System.out.println("计算-1.0的绝对值：" + Math.abs(-1.0));

System.out.println("大于5.6的最小整数：" + Math.ceil(5.6));

System.out.println("小于-4。2的最大整数：" + Math.floor(-4.2));

System.out.println("对-4.6进行四舍五入：" + Math.round(-4.6));

System.out.println("2.1和-2.1中的较大值：" + Math.max(2.1,-2.1));

System.out.println("2.1和-2.1中的较小值：" + Math.min(2.1,-2.1));

System.out.println("大于或者等于0.0小于1.0的随机数：" + Math.random());

System.out.println("1.57的正弦值：" + Math.sin(1.57));

System.out.println("4的平方根" + Math.sqrt(4));

System.out.println("2的3次方" + Math.pow(2,3));

}

}

6-15:

package Src.com.learn.Array;

import java.io.IOException;

public class Demo\_15 {

public static void main(String[] args) {

Runtime runtime = Runtime.getRuntime();

try {

runtime.exec("notepad.exe");

} catch (IOException e) {

throw new RuntimeException(e);

}

}

}

6-16

; package Src.com.learn.Array;

import java.util.Random;

public class Demo\_16 {

public static void main(String[] args) {

Random ra = new Random();//不传入种子

// 随机生成10个0到100的整数

for(int i = 0;i < 10; i++)

{

System.out.println(ra.nextInt(100));

}

}

}

6-17:

package Src.com.learn.Array;

import java.util.Random;

public class Demo\_17 {

public static void main(String[] args) {

// 传入种子

Random random = new Random(13);

// 随机生成10个0到100之间的数

for(int i = 0;i < 10; i++)

{

System.out.println(random.nextInt(100));

}

}

}

6-18:

package Src.com.learn.Array;

import java.util.Random;

public class Demo\_18 {

public static void main(String[] args) {

Random random = new Random();

System.out.println("生成boolean类型的随机数：" + random.nextBoolean());

System.out.println("生成float类型的随机数：" + random.nextFloat());

System.out.println("生成double类型的随机数：" + random.nextDouble());

System.out.println("生成int类型的随机数：" + random.nextInt());

System.out.println("生成0到100之间的随机数：" + random.nextInt(100));

System.out.println("生成long类型随机数：" + random.nextLong());

}

}

6-19:

package Src.com.learn.Array;

import java.math.BigInteger;

public class Demo\_19 {

public static void main(String[] args) {

BigInteger b1 = new BigInteger("1234111111111111111111152121671118121291");

BigInteger b2 = new BigInteger("123456891");

System.out.println("b1和b2的和：" + b1.add(b2));

System.out.println("b1和b2的差：" + b1.subtract(b2));

System.out.println("b1和b2的积：" + b1.multiply(b2));

System.out.println("b1和b2的商：" + b1.divide(b2));

System.out.println("b1和b2的较大值：" + b1.max(b2));

System.out.println("b1和b2的较小值：" + b1.min(b2));

// 创建BigInteger数组接收b1除以b2的商和余数

BigInteger[] bigIntegers = b1.divideAndRemainder(b2);

System.out.println("商：" + bigIntegers[0] + "余数：" + bigIntegers[1]);

}

}

6-20:

package Src.com.learn.Array;

import java.math.BigDecimal;

import java.math.RoundingMode;

public class Demo\_20 {

public static void main(String[] args) {

BigDecimal b1 = new BigDecimal("0.001");

BigDecimal b2 = BigDecimal.valueOf(0.009);

System.out.println("b1和b2的和：" + b1.add(b2));

System.out.println("b1和b2的差：" + b1.subtract(b2));

System.out.println("b1和b2的积：" + b1.multiply(b2));

System.out.println("b1和b2的商：" + b1.divide(b2,3, RoundingMode.HALF\_UP));

System.out.println("b1和b2的较小值：" + b1.min(b2));

System.out.println("b1和b2的较大值：" + b1.max(b2));

}

}

【实验成绩】