06 Java常用类介绍及作业讲评

宋杰 硕士研究生
songjiesdnu@163.com

北京师范大学教育技术学院
现代教育技术研究所

内容提要

- **6.1 Java常用类**
  - 数据类型类
  - Math类
  - 字符串相关类
- **6.2 作业讲评**
  - 设计思路
  - 示例

宋杰 songjie@163.com

Java类库

- **Java的类库是系统提供的已实现的标准类的集合，是Java编程的API（Application Program Interface），它可以帮助开发者方便、快捷地开发Java程序。**
- **这些系统定义好的类根据实现的功能不同，可以划分成不同的集合，每个集合是一个包，合称为类库。**
- **Java的类库大部分是由它的发明者——SUN公司提供的，这些类库称为基础类库（JFC，**Java Foundation Classes）。
- **API 应用程序编程接口**
  - 面向过程语言 – 函数库（子程序包）
  - 面向对象语言 – 类库

Java类库

- **类库的结构**
  - java.lang　语言基础类库（System、Math、Thread、基本数据类型类）
  - java.util　Java的工具类库(向量、栈、日期)
  - java.io　Java的标准输入输出类库
  - java.applet　用于实现Java Applet小程序的类库
  - java.awt　用于构建图形用户界面的类库
  - java.awt.event　界面用户交互控制和事件响应类库
  - java.net　Java的用于实现网络功能的类库
- **使用JDK Document查看类库**

Object类

- Object类是Java中所有类的超类，在java类的继承关系图上处于根部(root)，所有class的对象(包括数组对象)都继承了Object的方法

宋杰 songjie@163.com

类的介绍/Object

Object()	构造方法
clone()	克隆一个对象
equals(Object obj)	判断本对象与**obj**是否相等
finalize()	被垃圾收集器调用
getClass()	返回对象的**class**
hashCode()	返回对象的**hash code**值
notify()	叫醒一个正在等待的线程
notifyAll()	叫醒所有正在等待的线程
toString()	返回代表这个对象的字符串
wait()	让本线程进入等待
wait(long timeout)	让本线程进入等待
wait(long timeout, int nanos)	让本线程进入等待

数据类型类

- Java的基本数据类型,如int,double,char,long等。利用这些基本数据类型来定义简单的变量和属性十分方便,但是如果需要完成一些基本数据类型量的变换和操作,比如要把一个字符串转化为整数或浮点数,或者反过来要将一个数字转换成字符串,就需要使用数据类型类的相应方法

- 数据类型类与基本数据类型密切相关,每一个数据类型类都对应了一个基本数据类型,它的名字也与这个基本数据类型的名字相似.不同的是数据类型类是一个类,有自己的方法,这些方法主要用来操作和处理它所对应的基本数据类型量

宋杰 songjie@163.com

类的介绍/数据类型类

数据类型类与它所对应的基本数据类型

数据类型类	基本数据类型
Boolean	boolean
Byte	byte
Character	char
Double	double
Float	float
Integer	int
Long	long
Short	short

数据类型类

- **数据类型类**
  - 规定了数据类型的最大值、最小值
  - 构造函数：如new Integer(10);
  - 完成不同数据类型间转换，注意不同的数据类使用的方法会有不同。
  - Double.toString(0.08)、Integer.parseInt(“123”)、Double.ValueOf(“0.08”).intValue ()等，见JDK Doc

Inte

静态属性：
通过类名可以直接访问的属性

- **Integer类：封装了若**
  - 两个属性，这两个静态属性返回int
    - MAX-VALUE： $2^{31}-1$
    - MIN-VALUE： -2^{31}

- **其它数据类型类中的方法与Integer类中的方法相近**

宋杰 songjie@163.com

Integer构造函数

- **public Integer(int value)**
  - 用基本数据类型int生成一个Integer对象
- **public Integer(String s)**
  - 用一个字符串来生成一个Integer对象

宋杰 songjie@163.com

Integer方法

- **Integer类有24方法，其中使用较多的有：**
  - public byte byteValue();
  - public double doubleValue();
  - public int intValue();
  - public long longValue();

- **这些方法把Integer类的对象所对应的int值转换成其它基本数据类型的值**

宋杰 songjie@163.com

与Integer类似的其他类

- **如Double、Float、Byte、Short、Long等也都具有这些方法，可以把这些类的对象所对应的值转换成其它基本数据类型的值**
  - public byte byteValue();
  - public double doubleValue();
  - public int intValue();
  - public long longValue();

宋杰 songjie@163.com

类的介绍/Math

Math类里给出了数学计算所需要的函数，包括：

绝对值	**abs(a)　　这里a可以是int,long,float和double**
三角函数	**sin(a)、cos(a)、tan(a)等**
乘方	**pow(a,b)　a的b次方**
自然对数	**log(a)　　以e为底的对数**
开方	**sqrt(a)　　求a的平方根**
随机数	**random()　[0.0, 1.0)　不小于0.0小于1.0的数**

这些都是静态(static)的方法，用Math.XXX()直接调用

另外它还提供了两个常数e和π

Math.E, Math.PI

类的介绍/Math(1)

```
public class MathTest{

  public static void main(String[] s){

    System.out.println("sin(π/4) is " +
Math.sin(Math.PI/4.0));

    System.out.println("2的4次方是 " + Math.pow(2,4));

    System.out.println("以e为底的e的对数是 " +
Math.log(Math.E));

    System.out.println("81的平方根是 " + Math.sqrt(81));

  }

}
```

```
sin(π/4) is 0.7071067811865475
2的4次方是 16.0
以e为底的e的对数是 1.0
81的平方根是 9.0
```

类的介绍/Math(2)

random()方法的演示程序（涉及到GUI）

```
import java.awt.*;
import java.awt.event.*;

class RandomFrame extends Frame implements ActionListener{
    RandomFrame(String title){
        super(title);
        add(pane = new Panel());
        setPane();
        pack();
        Dimension dm = getSize();
        Dimension ss = getToolkit().getScreenSize();
        setCenter(ss, dm);
```

转下页

类的介绍/Math(2*)

接上页

random()方法的演示程序

```
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });
    }     //构造方法结束


        private void setCenter(Dimension screen, Dimension
frame){
            int x = screen.width/2 - frame.width/2;
            int y = screen.height/2 - frame.height/2;
            setLocation(x,y);
        }
```

转下页

类的介绍/Math(2**)

接上页

random()方法的演示程序

```
private void setPane(){
        tfd = new TextField(20);
        tfd.setFont(new Font("Tahoma", Font.BOLD, 16));
        pane.add(tfd);
        btn = new Button("RandomNumber Generator");
        btn.addActionListener(this);
        pane.add(btn);
}
public void actionPerformed(ActionEvent ae){
        Button b = (Button)ae.getSource();
        if(b == btn){
                String s = Double.toString(Math.random());
                tfd.setText(s);
        }
}
```

转下页

类的介绍/Math(2***)

接上页

random()方法的演示程序

```
        private Button btn;
        private TextField tfd;
        private Panel pane;
}       //RandomFrame class结束

public class MathTest2{
        public static void main(String[] s){
                RandomFrame rf = new RandomFrame();
                rf.show();
        }
}
```

String

- 字符串在Java中是一个特殊的类String,它是一个final class，因此不能被继承

- String被Java的开发者构造得非常接近基本数据类型，换句话说，在很多时候可以像使用基本数据类型一样来使用String类

宋杰 songjie@163.com

类的介绍/String

Constructor	说明
String()	生成一个空字符串
String(char[] chars)	从一个字符数组生成一个字符串
String(char[] chars, int offset, int count)	从一个字符数组的第offset个位置开始取count个字符生成一个字符串

宋杰 songjie@163.com

String声明

- **声明了一个空字符串s的方式**
  - String s;
  - String s = new String();

- **创建一个字符串的方式**
  - String ss = “创建了一个字符串”;
  - String ss = new String(“创建了一个字符串”);

宋杰 songjie@163.com

类的介绍/String

String的成员方法：

String的主要方法	说明
int length()	字符串中字符的个数
char charAt(int index)	返回位于index处的字符
void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)	拷贝字符数 = srcEnd-srcBegin 从dst的dstBegin位置开始放

宋杰 songjie@163.com

类的介绍/String

String的成员方法：

String的主要方法	说明
int indexOf(int ch)	返回第一个ch字符的位置
int indexOf(String s)	返回第一个子串s的位置
boolean equals(Object str)	将当前字符串与str比较
boolean equalsIgnoreCase(String s)	当前字符串与s进行比较,忽略大小写

宋杰 songjie@163.com

特别注意！

- 不可用‘==’比较字符串的字符内容是否相同!

```
String str1 = new String("caterpillar");
String str2 = new String("caterpillar");
System.out.println(str1 == str2);
```

- 要比较两个字符串对象的字符值是否相同，必须使用equals()方法

```
String str1 = new String("caterpillar");
String str2 = new String("caterpillar");
System.out.println(str1.equals(str2));
```

类的介绍/String

String的成员方法：

String的主要方法	说明
int compareTo(String s)	当前字符串大/长时返回大于0的值
String subString(int beginIndex)	取从beginIndex到尾的子串
String subString(int bIdx, int eIdx)	取从bIdx到eIdx的子串

宋杰 songjie@163.com

类的介绍/String

String的成员方法：

String的主要方法	说明
String concat(String s)	将s接在当前字符串的尾部,等于运算符+
String toLowerCase()	获得小写字母版本*
String toUpperCase()	获得大写字母版本*
String replace(char oldCh, char newCh)	用newCh替换oldCh

宋杰 songjie@163.com

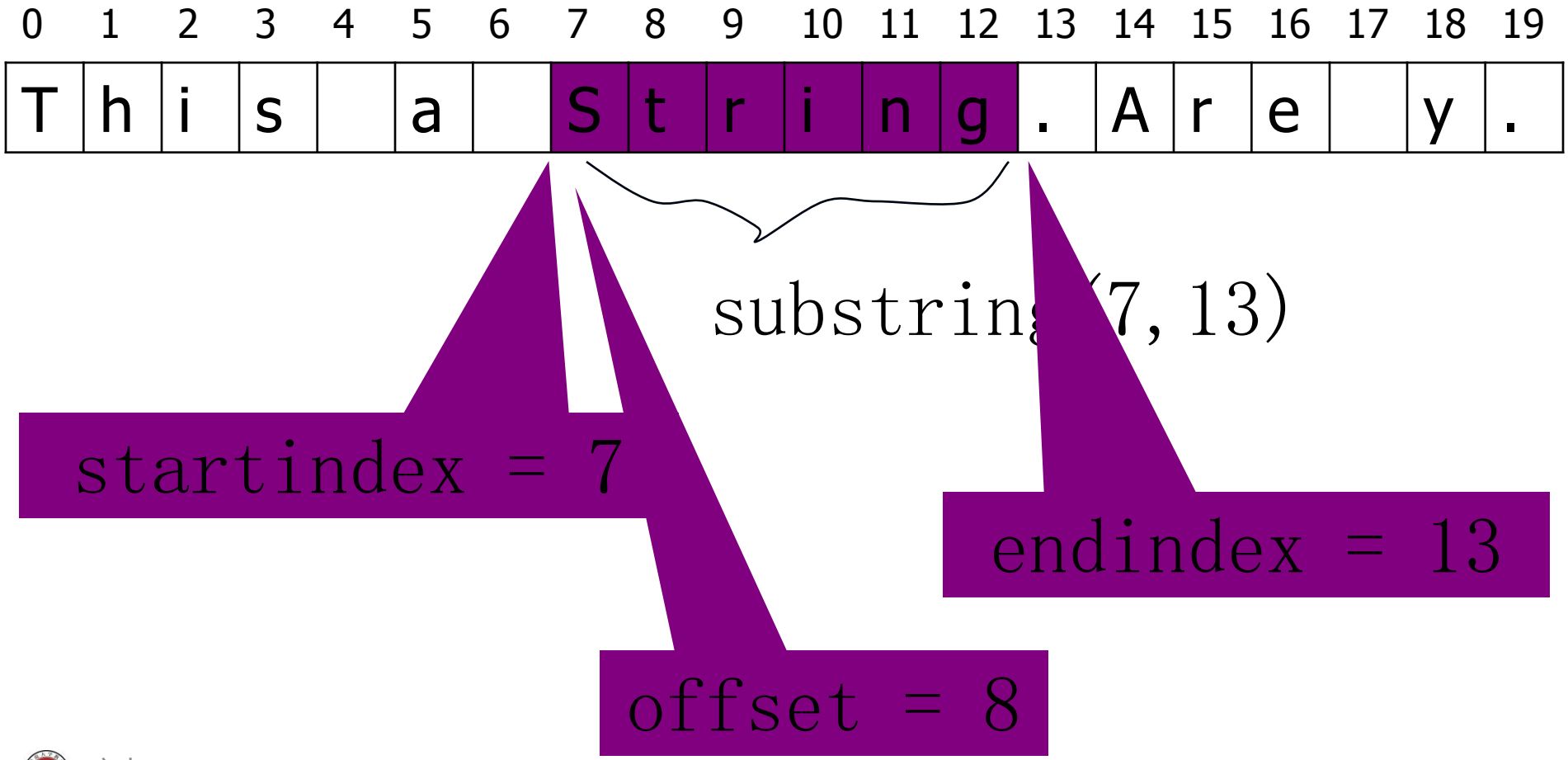
类的介绍/String

String的成员方法：

String的主要方法	说明
String trim()	去掉两端的空格
String valueOf(...)	将其他数据类型的数据转换成字符串
boolean startWith(String s)	查当前字符串是否以字串s开始
boolean endtWith(String s)	查当前字符串是否以字串s结束

宋杰 songjie@163.com

index, offset, 字符在字符串中的位置



StringBuffer

- 自己看JDK或者看书

宋杰 songjie@163.com

为什么要有两个字符串类?

- 如果创建一个字符串，不打算修改它，用String
- 如果创建一个字符串，计划修改它，用StringBuffer
- 详细的比较请参考：
- http://lcell.bnu.edu.cn/do/lcpage?action=view&kold=8712

- 看一个例子吧StringStringBufferTest.java

宋杰 songjie@163.com

StringStringBufferTest.java

```
public class StringStringBufferTest {
public static void stringReplace(String text) {
    text = text.replace('b', 'a');}
public static void bufferReplace(StringBuffer text) {
    text = text.append("er");}
public static void main(String args[]) {
    String textString = new String("bnu");
    StringBuffer textBuffer = new StringBuffer("bnu");
    stringReplace(textString);      bufferReplace(textBuffer);
    System.out.println("textString:"+textString);// ? immutable
    System.out.println("textBuffer
    String myStr="a";          myStr
    System.out.println("myStr:"+n
}
}
```

```
textString:bnu
textBuffer:bnuer
myStr:b
```


宋杰 songjie@163.com

Random

- 该类继承自Object，在java.util包中。
- 构造方法： public Random();

宋杰 songjie@163.com

Random成员方法

public void nextBytes(byte[] bytes)	生成随机字节数据的字节数组
public int nextInt()	生成int类型所表示的范围中的随机数
public int nextInt(int n)	生成0-n(不含n)的随机数
public long nextLong()	生成在long类型所表示的范围中的随机数
public float nextFloat()	生成0.0f-1.0f(不含)之间的随机float数
public double nextDouble()	生成0.0-1.0(不含)之间的随机double数

宋杰 songjie@163.com

StringTokenizer

- StringTokenizer类继承自Object类，用来为字符串构造一个分析器，该类在java.util包中。
- 主要用于将字符串分割的场合。

宋杰 songjie@163.com

StringTokenizer构造方法

- **public StringTokenizer(String str)**
  - 说明：为字符串str构造一个分析器，使用默认的分隔符集合（“ /t/n/r/f"，即：空白字符、制表符、换行符、回车符和换页符）
- public StringTokenizer(String str, String delim)
  - 说明：为字符串str构造一个分析器，字符串参数delim中的所有字符都作为分隔符
- public StringTokenizer(String str, String delim, boolean returnTokens)
  - 说明：为字符串str构造一个分析器，字符串参数delim中的所有字符都作为分隔符；
  - 如果returnTokens为真，则把分隔符也作为子串返回；
  - 如果returnTokens为假，只返回被分隔符隔开的子串。

宋杰 songjie@163.com

StringTokenizer常用方法

- public boolean hasMoreTokens()——测试一个字符串分析器中是否有被分隔符隔开的子串可得到，如果有返回true，否则返回false。
- public boolean hasMoreElements()——与上面方法相同
- public String nextToken()——返回一个字符串分析器中被分隔符隔开的子串（非分隔符）。
- public Object nextElement()——同nextToken()方法。
- public int countTokens()——返回一个字符串分析器中方法nextTokens()能被调用的次数，即有多少个不是分隔符的子串。

例：TestStringToken.java

宋杰 songjie@163.com

作业点评

- 设计思路

- 几个问题
  - 如何接收键盘输入的数据？
    - Scanner类
  - 数据处理的规则？
    - 题目要求中有
  - 如何体现面向对象的思想？
    - 这里主要用到“封装”的思想
  - 对用户输入的异常数据的处理

宋杰 songjie@163.com

作业点评

- 示例
  - 效果展示
  - Student.java

宋杰 songjie@163.com

06 Java常用类介绍及作业讲评

宋杰 硕士研究生
songjiesdnu@163.com

北京师范大学教育技术学院
现代教育技术研究所