



# Java 核心技术

第十一章 Java 文件读写

第三节 文本文件读写

华东师范大学 陈良育



# 文本文件读写(1)

- 文件类型
  - 一般文本文件(若干行字符构成的文件), 如txt等
  - 一般二进制文件, 如数据文件dat
  - 带特殊格式的文本文件, 如xml等
  - 带特殊格式二进制文件, 如doc,ppt等
- 文件是数据的一个容器(口袋)
- 文件可以存放大量的数据
- 文件很大, 注定Java只能以流形式依次处理



# 文本文件读写(2)

- 从Java角度理解
  - 输出：数据从Java到文件中，写操作
  - 输入：数据从文件到Java中，读操作
- 文本文件读写
  - 输出文本字符到文件中
  - 从文件中读取文本字符串



# 文本文件读写(3)

- 写文件

- 先创建文件，写入数据，关闭文件
- `FileOutputStream`, `OutputStreamWriter`, `BufferedWriter`
- `BufferWriter`
  - `write`
  - `newLine`
- `try-resource` 语句，自动关闭资源
- 关闭最外层的数据流，将会把其上所有的数据流关闭
- 查看例子 `TxtFileWrite.java`



# 文本文件读写(4)

- 读文件

- 先打开文件，逐行读入数据，关闭文件
- FileInputStream, InputStreamWriter, BufferedReader
- BufferedReader
  - readLine
- try-resource 语句，自动关闭资源
- 关闭最外层的数据流，将会把其上所有的数据流关闭
- 查看例子TxtFileRead.java



# 文本文件读写(5)



- 总结

- 理解节点类、转换类和装饰类的联合用法
- 尽量使用try-resource 语句，自动关闭资源



# 代码(1) TxtFileWrite.java

```
import java.io.*;

public class TxtFileWrite {
    public static void main(String[] args) {
        writeFile1();
        System.out.println("=====");
        //writeFile2(); // JDK 7及以上才可以使用
    }
```



## 代码(2) TxtFileWrite.java

```
public static void writeFile1() {
    FileOutputStream fos = null;
    OutputStreamWriter osw = null;
    BufferedWriter bw = null;
    try {
        fos = new FileOutputStream("c:/temp/abc.txt"); // 节点类
        osw = new OutputStreamWriter(fos, "UTF-8"); // 转化类
        //osw = new OutputStreamWriter(fos); // 转化类
        bw = new BufferedWriter(osw); // 装饰类
        // br = new BufferedWriter(new OutputStreamWriter(new
        // FileOutputStream("c:/temp/abc.txt")))
        bw.write("我们是");
        bw.newLine();
        bw.write("Ecnuers.^^");
        bw.newLine();
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        try {
            bw.close(); // 关闭最后一个类，会将所有的底层流都关闭
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```





## 代码(3) TxtFileWrite.java

```
public static void writeFile2() {  
    //try-resource 语句, 自动关闭资源  
    try (BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("c:/temp/abc.txt")))) {  
        bw.write("我们是");  
        bw.newLine();  
        bw.write("Ecnuers.^^");  
        bw.newLine();  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```



## 代码(4) TxtFileRead.java

```
import java.io.BufferedReader;

public class TxtFileRead {
    public static void main(String[] args) {
        readFile1();
        System.out.println("=====");
        //readFile2(); //JDK 7及以上才可以使用
    }
```



## 代码(5) TxtFileRead.java

```
public static void readFile1() {  
    FileInputStream fis = null;  
    InputStreamReader isr = null;  
    BufferedReader br = null;  
    try {  
        fis = new FileInputStream("c:/temp/abc.txt"); // 节点类  
        isr = new InputStreamReader(fis, "UTF-8"); // 转化类  
        //isr = new InputStreamReader(fis);  
        br = new BufferedReader(isr); // 装饰类  
        // br = new BufferedReader(new InputStreamReader(new  
        // FileInputStream("c:/temp/abc.txt")))  
        String line;  
        while ((line = br.readLine()) != null) // 每次读取一行  
        {  
            System.out.println(line);  
        }  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    } finally {  
        try {  
            br.close(); // 关闭最后一个类，会将所有的底层流都关闭  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```



## 代码(6) TxtFileRead.java

```
public static void readFile2() {  
    String line;  
    //try-resource 语句, 自动关闭资源  
    try (BufferedReader in = new BufferedReader(new InputStreamReader(new FileInputStream("c:/temp/abc.txt")))) {  
        while ((line = in.readLine()) != null) {  
            System.out.println(line);  
        }  
    }  
    catch(Exception ex)  
    {  
        ex.printStackTrace();  
    }  
}
```





谢谢!