

Bài 11

Làm việc với FileIO

Company: DEVPRO VIỆT NAM

Website: devpro.edu.vn

Design by Minh An

1

Nội dung

- File
- Stream

Design by Minh An

2

File

- Lớp `java.io.File` chứa các phương thức cho phép người lập trình thực hiện các thao tác xử lý với các tệp tin, thư mục.

`exists()` kiểm tra xem file có tồn tại hay không

`getName()` lấy tên file (input-file.txt)

`getParent()` lấy đường dẫn thư mục của file

`getPath()` đường dẫn đầy đủ

`isDirectory()` kiểm tra xem là thư mục hay không

`isFile()` kiểm tra xem là file hay không

`length()` cỡ file (byte)

`createNewFile()` tạo ra file mới

`delete()` xóa file

`list()` lấy tên file, thư mục chứa trong đường dẫn

`mkdir()` tạo thư mục

`renameTo(File dest)` đổi tên file

Design by Minh An

3

File

```
import java.io.File;

public class MainClassFiles {
    public static void main(String []args) {
        File path = new File("D:\\09_DevPro");
        System.out.println("Exist:" + path.exists());
        System.out.println("Name: " + path.getName());
        System.out.println("Can execute: " + path.canExecute());
        System.out.println("Parent: " + path.getParent());
        System.out.println("Is file: " + path.isFile());
        System.out.println("Is directory: " + path.isDirectory());

        File[] list = path.listFiles();
        for (File x : list) {
            System.out.println(x.getAbsolutePath());
        }
    }
}
```

Design by Minh An

4

Stream

- Các Stream thường được sử dụng là:
 - FileInputStream: được sử dụng để đọc dữ liệu từ các file
 - FileOutputStream: được sử dụng để ghi dữ liệu ra các file
 - FileReader: đọc ký tự
 - BufferedReader: đọc chuỗi
 - FileWriter: ghi ký tự
 - BufferedWriter: ghi chuỗi
 - DataInputStream
 - DataOutputStream
 - ObjectInputStream
 - ObjectOutputStream

Design by Minh An

5

FileOutputStream

```
public class FilesExam01 {  
    public static void main(String[] args) {  
        FileOutputStream fout = null;  
        try {  
            fout = new FileOutputStream("D:\\TestFile\\input1.txt");  
            for (char c = 'a'; c <= 'z'; c++) {  
                fout.write(c);  
            }  
        } catch (IOException e) {  
            System.out.println(e.getMessage());  
        }  
        finally {  
            try {  
                fout.close();  
            } catch (IOException e) {  
                System.out.println(e.getMessage());  
            }  
        }  
    }  
}
```

Design by Minh An

6

FileInputStream

```
FileInputStream fin = null;
try {
    fin = new FileInputStream("D:\\TestFile\\input1.txt");
    int x = fin.read();
    while (x != - 1) {
        System.out.println((char)x);
        x = fin.read();
    }
} catch (IOException e) {
    System.out.println(e.getMessage());
} finally {
    try {
        fin.close();
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
}
```

Design by Minh An

7

DataOutputStream

```
public static void writeData() {
    int a = 1000;
    long b = -32001;
    double c = 25.325;
    try {
        String path = "D:\\XuLyTep\\ViDu01\\data.bin";
        FileOutputStream fos = new FileOutputStream(path);
        DataOutputStream dos = new DataOutputStream(fos);
        dos.writeInt(a);
        dos.writeLong(b);
        dos.writeDouble(c);
        dos.close();
        fos.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Design by Minh An

8

DataInputStream

```
public static void readData() {
    int n = 0;
    long m = 0;
    double d = 0;
    try {
        String path = "D:\\XuLyTep\\ViDu01\\data.bin";
        FileInputStream fis = new FileInputStream(path);
        DataInputStream dis = new DataInputStream(fis);
        n = dis.readInt();
        m = dis.readLong();
        d = dis.readDouble();
        fis.close();
        dis.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println("n = " + n);
    System.out.println("m = " + m);
    System.out.println("d = " + d);
}
```

Design by Minh An

9

ObjectOutputStream

```
1 package vn.devpro.day10.read_write_object;
2 import java.io.Serializable;
3 public class Students implements Serializable{
4     int code;
5     String name;
6     int age;
7     String gender;
8     public Students(int code, String name, int age, String gender) {
9         super();
10        this.code = code;
11        this.name = name;
12        this.age = age;
13        this.gender = gender;
14    }
15    public Students() {
16        super();
17    }
18    @Override
19    public String toString() {
20        return "Students [code=" + code + ", name=" + name + ", age=" + age +
21            ", gender=" + gender + " ]";
22    }
23 }
24
```

Design by Minh An

10

ObjectOutputStream

```
public class DataOutputDemo {  
  
    public static void writeStudentObject(Students s) {  
        String path = "D:\\XuLyTep\\ViDu01\\student.bin";  
        FileOutputStream fos;  
        ObjectOutputStream oos;  
        try {  
            fos = new FileOutputStream(path);  
            oos = new ObjectOutputStream(fos);  
            oos.writeObject(s);  
            System.out.println("Student object was written");  
            oos.close();  
            fos.close();  
        } catch (IOException e) {  
            System.out.println("An error has occurred");  
        }  
    }  
}
```

Design by Minh An

11

ObjectInputStream

```
public static Students readStudentObject() {  
    Students s = new Students();  
    String path = "D:\\XuLyTep\\ViDu01\\student.bin";  
    try {  
        FileInputStream fis = new FileInputStream(path);  
        ObjectInputStream ois = new ObjectInputStream(fis);  
        s = (Students)ois.readObject();  
        ois.close();  
        System.out.println("read is ok");  
        //fis.close();  
        return s;  
    } catch (IOException e) {  
        System.out.println("Loi doc file");  
    } catch (ClassNotFoundException e) {  
        System.out.println("Loi mo file");  
    }  
    return null;  
}
```

Design by Minh An

12

Đọc file text với BufferedReader & FileReader

```
public class BufferedReaderUtils {  
    public static void main(String[] args) {  
        try {  
            FileReader path =  
                new FileReader("D:\\TestFile\\input1.txt");  
            BufferedReader buff = new BufferedReader(path);  
            String line;  
            while ((line = buff.readLine()) != null) {  
                System.out.println(line);  
            }  
            buff.close();  
        }  
        catch (FileNotFoundException e1) {  
            e1.printStackTrace();  
        }  
        catch (IOException e2) {  
            e2.printStackTrace();  
        }  
    }  
}
```

Design by Minh An

13

Ghi file text với BufferedWriter & FileWriter

```
try {  
    FileWriter path =  
        new FileWriter("D:\\TestFile\\poem.txt");  
    BufferedWriter buff = new BufferedWriter(path);  
    buff.write("Trong dam gi dep bang sen\\n");  
    buff.write("La xanh bong trang lai chen nhi vang\\n");  
    buff.write("Nhi vang bong trang la xanh\\n");  
    buff.write("Gan bun ma chang hoi tanh mui bun\\n");  
    buff.close();  
} catch (IOException e1) {  
    e1.printStackTrace();  
}
```

Design by Minh An

14

Bài tập

- Tạo một file input.txt theo định dạng: Dòng 1 ghi số nguyên dương n , n dòng tiếp theo mỗi dòng ghi một số thực (n và các số thực nhập vào từ bàn phím).
- Đọc dữ liệu từ file input.txt, dữ liệu được ghi vào biến nguyên n và mảng số thực a (số nguyên dương ở dòng đầu tiên lưu vào biến n , các số thực ở các dòng tiếp theo lưu vào mảng a).
- Sắp xếp mảng a theo chiều tăng dần, ghi dữ liệu từ mảng a ra file output.txt theo định dạng: Dòng đầu tiên ghi chiều dài của mảng, các dòng tiếp theo ghi giá trị các phần tử mảng.
- Đọc dữ liệu từ file output.txt và hiển thị ra màn hình.

Design by Minh An