YAO ZHAO

2019-2-19

CS209A SPRING2019

CS209 - LAB1

Key Content

- File
- I/O Stream
- Charsets
- Simple text processing

Before exercise

1. Class File basic usage

```
public static boolean createFile(String destFileName) {
    File file = new File(destFileName);
    if (file.exists()) {
        System.out.println("Create single file " + destFileName + " fail,target file already exists! ");
        return false;
    }
    if (destFileName.endsWith(File.separator)) {
        System.out.println("Create single file " + destFileName + " fail, target file cannot be a directory! ");
        return false;
    }
    // Check if the directory where the target file is located exists
    if (!file.getParentFile().exists()) {
        // if the directory where the target file is located doesn't exist, create
        // its' parent directory.
        System.out.println("directory where target file is located doesn't exist, create its' parent directory! ");
        File parentFile = file.getParentFile();
        parentFile.mkdirs();
        if (!file.getParentFile().mkdirs()) {
            System.out.println("Create directory where target file is located fails! ");
            return false;
        }
    }
    // Create target file
    try {
        if (file.createNewFile()) {
            System.out.println("Create single file " + destFileName + " success! ");
            return true;
        } else {
            System.out.println("Create single file " + destFileName + " fail! ");
            return false;
        }
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("Create single file " + destFileName + " fail! " + e.getMessage());
        return false;
    }
```

Invoke above method and observe result.

2. Deeper understanding I/O

Please see < JAVA IO.pdf >

3. Charsets

(1) Char vs actual value

Run the following code:

```
char c = '赵';
int value = c;
System.out.printf("%s\n",c);
System.out.printf("%X\n",value);
```

Observe the result.

(2) Transform from different charset

Run the following code:

```
String str = "赵耀"; // UTF-16
try
{
	byte[] bytes1 = str.getBytes("GBK"); // or GBK
	for (byte b : bytes1) {
		System.out.printf("%2X ", b);
	}
	System.out.println();
	byte[] bytes2 = str.getBytes("UTF-16");
	for (byte b : bytes2) {
		System.out.printf("%02X ", b);
	}
	System.out.println();

	byte[] bytes3 = str.getBytes("UTF-16BE");
	for (byte b : bytes3) {
		System.out.printf("%02X ", b);
	}
	System.out.println();

	byte[] bytes4 = str.getBytes("UTF-16LE");
	for (byte b : bytes4) {
		System.out.printf("%02X ", b);
	}
	System.out.println();
}catch(UnsupportedEncodingException e){

	e.printStackTrace();
}
```

Observe the result.

PS: UTF-16:赵-8D75 耀-8000, GB:赵-D5D4 耀-D2AB

Exercise

Please write a program named “CharsetConvertor” that converts a file to a new specified charset, then generates the output file in the form of hexadecimal characters which base on the content with new charset.

Name requirements for the output file: arg[0] and arg[2] should be concatenated using “_”.

For example, here is a file named “sample.txt”, which charset is GB18030, target charset is UTF-8.

If run the command:

```
java    CharsetConvertor sample.txt GB18030 UTF-8
```

The output file should be like this: