THE UNIVERSITY OF MELBOURNE

Lab6（week9）

COMP90041 Programming and software development

Zhe(Zoe) Wang

Writing to a text file

The class PrintWriter is a stream class that can be used to write to a text file

```
import java.io.PrintWriter;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
```

```
PrintWriter outputStream = null;
String filename = "out.txt";
outputStream = new PrintWriter(filename);

outputStream = new PrintWriter(new FileOutputStream(filename, true));  //append to the end
```

```
outputStreamName.close();
```

StreamDemo

PrintWriter outputStream

- Output streams connected to files are usually *buffered*
- When enough data accumulates, or when the method **flush** is invoked, the buffered data is written to the file all at once
- **close** invokes the method **flush**

Reading from text file

- **Scanner *StreamObject* =new Scanner(new FileInputStream(*FileName*));**
- **nextInt** and **nextLine**
- hasNextLine()

InputDemo

IOException

FileNotFoundException

IOException

-use try…catch

Write Object (in binary)

```
import java.io.ObjectOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;

ObjectOutputStream outputStream = null;
String filename = "dogs.dat";

outputStream = new ObjectOutputStream(new FileOutputStream(filename));
outputStream.writeObject(dogs);
```

Read Object

```
import java.io.ObjectInputStream;
import java.io.IOException;
import java.io.FileInputStream;

ObjectInputStream inputStream = null;
String filename = "dogs.dat";

inputStream = new ObjectInputStream(new FileInputStream(filename));
dogs =(Dog) inputStream.readObject();
```

type cast!

ClassNotFoundException

Binary I/O of Object

In addition, the class of the object being read or written must implement the Serializable interface

public class Dog implements Serializable {…}

1.Write a complete Java program that tests whether or not the directory (folder) containing the program also contains a file named `Sally.txt`. The program has no input, and the only output tells whether or not there is a file named `Sally.txt`.

2.Write a complete Java program that asks the user for a file name, tests whether the file exists, and, if the file exists, asks the user whether or not it should be deleted. It then either deletes or does not delete the file as the user requests.

3.Write a complete Java program that finds the longest word in a text file.

4. Write a complete Java program that reads two files and write into a new file.

5.Write a complete Java program that stores text file content line by line into an array.

6.Write a complete Java program that gets a list of all file/directory names from the given.

7.Write a complete Java program that gets specific files by extensions from a specified folder.

8.Write a complete Java program that gets last modified time of a file.