

Computer Science 313e - More I/O

The only way to learn a new programming language is by writing programs in it.
-B. Kernighan and D. Ritchie

Reading Assignment:

Finish reading chapter 8 (Arrays chapter) in Morelli

Feedback Questions:

1. Write a Java code segment that returns the character at index 3 in previously defined String variable `myString`.

2. Write a Java code segment that sorts a previously declared and initialized array `myArray`. The elements in `myArray` are of type double.

Example:

```
public class TestFileIO
{
    public void static main(String[] args) throws IOException
    {
        // Read a line from the file called "test.txt" and print it to the screen

        BufferedReader in = new BufferedReader(new FileReader("test.txt"));
        String str = in.readLine();
        System.out.println(str);
    }
}
```

Next let's extend our program so it reads lines from our file and prints them to the screen until the end of the file is reached.

File I/O

1. Input from a File

In package java.io:

Recall: `BufferedReader` class

`readLine()` method - read one line of chars and return them in a string

`FileReader` class

- allows us to read text from a file

- filename is argument of constructor

Note: `readLine()` returns null when it tries to read the end of file character

```
public class TestFileIO
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader in = new BufferedReader(new FileReader("test.txt"));
        String str;

        while((str = in.readLine()) != null)
            System.out.println(str);
    }
}
```

Exercise: Given a file "add.txt" that contains two doubles (each on its own line), read the numbers and print the numbers and their sum to the screen.

Ex:

add.txt

2.5

1.3

Output to the screen: $2.5 + 1.3 = 3.8$

More File I/O

2. Output to a text file

In java.io package:

PrintWriter class

- writes various data types by converting them to their string representations

Methods (v can be of any type):

- print(v) : prints string representation of v

- println(v): prints string representation of v followed by a new line

BufferedWriter class

- buffers output (more efficient)

FileWriter class

- writes text to a file. The filename is the argument to the constructor.

2nd optional argument: flag indicating what to do if
file already exists

true --> append text to end of file

false --> overwrite file

Example: Write your name to a file called "myName.txt".

```
import java.io.*;
```

```
public class Names
```

```
{
```

```
    public static void main(String[] args) throws IOException
```

```
    {
```

```
        String me = "Mary Eberlein";
```

```
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(  
            "myName.txt")));
```

```
        out.println(me);
```

```
        out.close();
```

```
    }
```

```
}
```

Exercise: Write a program that appends your age and your major to the file
"myName.txt".

Example myName.txt file:

Mary Eberlein

22

mathematics