Files

Table of Contents

Reading files

Boilerplate for reading plaintext files

```java
import java.io.FileReader;      // low level file for simple character reading
import java.io.BufferedReader;  // higher level file object that reads Strings
import java.io.IOException;

public class Program {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {
            String text;
            while ((text = br.readLine()) != null) {
                // do stuff with text
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- `BufferedReader` is a wrapper that encompasses `FileReader`, allowing you to manipulate files
  - well suited to large files and fast processing

Boilerplate: reading plaintext with `Scanner`

- can also use `Scanner`, allowing you to parse lines into tokens, read as integers, …

```
try (Scanner scanner = new Scanner(new FileReader("test.txt"))) {
    while (scanner.hasNextLine()) {
        // do stuff
    }
}
```

Reading CSV files

```
String[] columns = text.split(",");
```

Writing files

Boilerplate for writing plaintext files

```
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.IOException;

public class Program {
    public static void main(String[] args) {
        try (PrintWriter pw = new FileWriter("test.txt")) {
            pw.println("Hello World");
            pw.format("Test a %s and an integer %d", "string", 10);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```