Using scanners directly to read files

COMP 102

Victoria University of Wellington

Files: line-by-line or token-by-token

If a file is formatted by line

- Eg, each "item" is described by a sequence of values on a single line

- It is simplest to read every line into a List of lines, and process with a for-each loop

```
List<String> allLines = Files.readAllLines( Path.of(fname) );
for (String line : lines){
    Scanner scan = new Scanner (line);
    ….
```

```
973 biscuits  27 33 15 4 9
731 cake 3 5 2
189 fruit 54 2 83 96
446 beans 1 3 2 5 3 4 7 2 5 1
```

If a file is a sequence of tokens, and the lines don't mean anything

- Eg, long sequence of words, with arbitrary line breaks.
- Eg, long sequence of numbers

```
Once upon a time
there was a chicken
who lived on a little
farm in a tiny village,
away  out in the far
country, beyond the
```

- It is simplest to put a single Scanner around the whole file:

```
Scanner scan =  new Scanner( Path.of(fname) );
 …  scan.next() ….  scan.nextDouble()….
```

Summing all the numbers in a file

```
/** Return the sum of all the numbers in a file, ignoring the non-numbers */
public double addNumbers(String fname){
    try {
        Scanner scan = new Scanner( Path.of(fname) );
        double total = 0;
        while (scan.hasNext() ) {
            if (scan.hasNextDouble() ) {
                total = total + scan.nextDouble();
            }
            else {
                scan.next();
            }
        }
        scan.close(); //Need to close the file to release it
        return total;
    } catch (IOException e) { UI.printf("File failure %s\n", e);}
}
```

Files with headers

- A data file may contain a header before the bulk of the data

  => need to read header first before reading rest of data

```
try {
    Scanner scan = new Scanner(Path.of(recordFileName ) );
    String name = scan.nextLine();
    int sID = scan.nextInt();
    String deg = scan.next();
    int count = scan.nextInt();
    for (int c = 0; c < count; c++){
        String code = scan.next();
        String grade = scan.next();
        int year = scan.nextInt();
        // process data
    }
    scan.close();
} catch (IOException e) { UI.println("File error: " + e); }
```

record-300765432.txt

```
Jo Miro
300765432
BSc
23
COMP102 A 2021
ENGR121 B+ 2021
COMP103 A- 2021
ENGR123 A- 2021
NZSL101  B+ 2021
COMP261 A- 2022
MATH261 B 2022
SWEN221 A+ 2022
   :
```

Files with multiple sets of data

```
try {
    Scanner scan = new Scanner(Path.of(recordFileName ) );
    while (scan.hasNext()){
        String name = scan.nextLine();
        int sID = scan.nextInt();
        String deg = scan.next();
        int count = scan.nextInt();
        for (int i=0; i < count; i++){
            String code = scan.next();
            String grade = scan.next();
            int year = scan.nextInt();
            // process data
        }
    }
    scan.close();
} catch (IOException e) { UI.println("File error: " + e); }
```

student-records.txt

```
Jo Miro
300765432
BSc
23
COMP102 A 2021
ENGR121 B+ 2021
COMP103 A- 2021
ENGR123 A- 2021
  :
SWEN221 A+ 2022
Jake Muskle
300765433
BA
16
```

Files with headers: passing a Scanner

- Can call another method to read the remaining data

  => must pass the Scanner to the method

```
    Scanner scan = new Scanner(Path.of(recordFileName ) );
    while (scan.hasNext()){
        String name = scan.nextLine();
        int sID = scan.nextInt();
        String deg = scan.next();
        int count = scan.nextInt();
        this.processRecord(scan, count, name, sID, deg);
    }
    scan.close();

 public void processRecord(Scanner sc, int ct, String n, int ID, String deg){
    for (int i=0; i < ct; i++){
        String code = sc.next();
          // process data
    }
```

student-records.txt

```
Jo Miro
300765432
BSc
23
COMP102 A 2021
ENGR121 B+ 2021
COMP103 A- 2021
ENGR123 A- 2021
 :
SWEN221 A+ 2022
Jake Muskle
300765433
BA
16
```

Passing an open Scanner

- You can pass an open Scanner to a method
- The method can read some data from the Scanner
  - starts from wherever the Scanner had got up to
  - leaves the Scanner ready for other code to read from where it left off.

```
/** Reads and processes ct  courses from the Scanner */
public void processRecord(Scanner sc, int ct, String n, int ID, String deg){
    for (int i=0; i < ct; i++){
        String code = sc.next();
          // process data
    }
}
```

student-records.txt

```
Jo Miro
300765432
BSc
23
COMP102 A 2021
ENGR121 B+ 2021
COMP103 A- 2021
ENGR123 A- 2021
 :
SWEN221 A+ 2022
Jake Muskle
300765433
BA
16
```