
Using scanners to read files

COMP 102

Victoria University of Wellington

Files and Scanners

If a file has lines, each with several values in it:

- Wrap each line from the file in a Scanner, and
- Read the values from the Scanner.

```
List<String> allLines = Files.readAllLines(Path.of("image.pxm"));
for (String line : allLines ){
    Scanner scan = new Scanner (line);
    UI.setColor(new Color (scan.nextInt(), scan.nextInt(), scan.nextInt()));
    UI.fillRect(x, y, 2,2);
    x = x+2;
    if (x > RIGHT) {
        x = LEFT;
        y = y+2;
    }
}
```

image.pxm

```
25 53 201
240 2 150
100 250 0
240 220 220
....
```

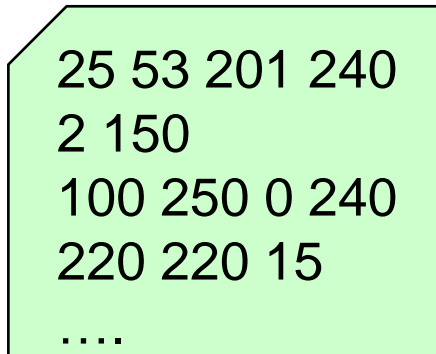
Files and Scanners

If a file has lines, each with varying number of values in it:

- Wrap each line from the file in a Scanner, and
- Read the values from the Scanner.

```
List<String> allLines = Files.readAllLines(Path.of("numbers.txt"));
for (String line : allLines){
    Scanner scan = new Scanner (line);
    double max = Double.NEGATIVE_INFINITY;
    while (scan.hasNextDouble()){
        double num = scan.nextDouble();
        if (num > max) {
            max = num;
        }
    }
    UI.println(max);
}
```

numbers.txt



```
25 53 201 240
2 150
100 250 0 240
220 220 15
....
```

Files and Scanners

- If each line has fixed number of values of different types:

```
try {
```

```
    List<String> allLines = Files.readAllLines(Path.of("flights.txt"));
```

```
    for (String line : allLines){
```

```
        Scanner scan = new Scanner (line);
```

```
        int PLU = scan.nextInt();
```

```
        String product = scan.next();
```

```
        double price = scan.nextDouble();
```

```
        ..... // do something with the values
```

```
    }
```

```
} catch (IOException e) { UI.println("File failure: " + e); }
```

fruit.txt

```
4447 quince 11.45
4430 pineapple 6.82
4041 red-plum 5.99
4416 D'Anjou-pear 5.44
4011 Banana 2.99
```

Reading files line by line

If items have a varying number of values:

May need loop within each line:

```
/**Adds up sales of item on each line of a file */
```

```
public void addCounts(){
    List<String> lines = Files.readAllLines(Path.of("data.txt"));
    for (String line : lines) {
        Scanner sc = new Scanner(line);
        int code = sc.nextInt();
        String item = sc.next();
        int lineTot = 0;
        while (sc.hasNextInt()) {
            lineTot = lineTot + sc.nextInt();
        }
        UI.printf("%s (%d): %d\n", item, code, lineTot);
    }
}
```

```
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
```

Processing values from a line

```

try {
    List<String> lines = Files.readAllLines(Path.of(filename));
    for ( String line : lines ) {
        Scanner sc = new Scanner(line);
        double left = sc.nextDouble();
        double top = sc.nextDouble();
        double wd = sc.nextDouble();
        double ht = sc.nextDouble();
        String shape = sc.next();
        int r = sc.nextInt();
        int g = sc.nextInt();
        int b = sc.nextInt();

        UI.setColor( new Color (r, g, b) );
        if (shape.equals("Oval") ) { UI.fillOval(left, top, wd, ht); }
        else { UI.fillRect(left, top, wd, ht); }
    }
} catch (IOException e) { UI.println("File failure: " + e); }

```

Diagram.txt

```

50.0 20.0 10.3 7.8 Oval 25 53 201
75.0 100.2 16.9 12.0 Rect 240 2 150
304.0 28.7 25.0 51.5 Oval 100 250 0
...

```

extracting all the values on the line

Do something
with all the
values