

# STRING/TEXT/FILE MANIPULATIONS

Boaz Kantor  
Introduction to Computer Science  
IDC Herzliya





# Contents

- `java.util.StringBuilder`
- `java.io.File`
- `java.util.Scanner` revisited
- All of the above combined to loop through a text file.



A stylized, dark brown illustration of a plant with several leaves and small, round fruits or buds, positioned on the left side of the slide.

# java.util.StringBuilder



# StringBuilder, the impetus

- A String object is **immutable**. We cannot modify its state.
- To emulate state modification, the String methods create a new object with the modified state, and return its reference.
- We then set our reference variable to point to the new object:  

```
s = s.toUpperCase();
```
- Creating new objects is very inefficient and expensive.
- To work around this disadvantage, Java provides two **mutable** classes:
  - **java.util.StringBuffer** is thread-safe, which makes it irrelevant to our course.
  - **java.util.StringBuilder** was added in JDK 5. It is not thread-safe, which makes faster than StringBuffer.
- In this course we will discuss & use **StringBuilder**.

# String vs. StringBuilder

## String

- Construction:

```
String s = "Hello, ";
```

- Concatenation:

```
s = s + you.getName() + "!";
```

- Printing:

```
System.out.println(s);
```

- Returning a String:

```
return s;
```

## StringBuilder

- Construction:

```
StringBuilder sb = new StringBuilder("Hello, ");
```

- Concatenation:

```
sb.append(you.getName()).append("!");
```

- Printing:

```
System.out.println(sb);
```

- Returning a String:

```
return sb.toString();
```



# When to use which?

- The general rule is:
  - Use a **StringBuilder** when your string manipulation may cause performance reduction.
  - Use a **String** when your string manipulation is local and insignificant in terms of performance.
- In other words, we always use a String, unless when we manipulate it within a loop.



A stylized, dark brown illustration of a plant with several leaves and small, round fruits or buds, positioned on the left side of the slide.

# java.io.File



# java.io.File

- **File** is an object which provides us with file and folder management.
- A **File** object does not provide access the contents of the file. It only provides us with external functionality.
- Read more about the [File API here](#).
- Some common methods:
  - **File(String pathname)**
    - Constructs a new **File** object related to the file/folder denoted by the parameter.
  - **createNewFile()**
  - **delete()**
  - **exists()**
  - **listFiles()**
  - **mkdir()**
  - **renameTo(File dest)**



A stylized, dark brown illustration of a plant with several leaves and small, round fruits or buds, positioned on the left side of the slide.

# java.util.Scanner



# java.util.Scanner revisited

- A Scanner object can parse text by dividing it to words and lines.
- The text may come from various sources:
  - The keyboard

```
Scanner s = new Scanner(System.in);  
System.out.print("Enter your name: ");  
String name = s.nextLine();
```

- A String

```
Scanner s = new Scanner("The quick brown fox jumps over the lazy dog");  
int n = 0;  
while (s.hasNext()) {  
    s.next();  
    n++;  
}  
System.out.println("Number of words: " + n);
```



# java.util.Scanner revisited (continued)

- The text may come from various sources:
  - A file

```
File f = new File("C:\\intro2cs\\textbooks\\Romeo.txt");
Scanner s = new Scanner(f);
int n = 0;
while (s.hasNextLine()) {
    s.nextLine();
    n++;
}
System.out.println("Number of lines in file: " + n);
```



A stylized, dark brown illustration of a plant with several leaves and small, round fruits or buds, positioned on the left side of the slide.

# StringBuilder, Scanner, File, all combined



# Handling text files

- A text file is a file which contains human-readable text.
- We usually handle text files line by line.
- Although not man-understandable, this is a sample text file:

```
But soft! What light through yonder window breaks?  
It is the East, and Juliet is the sun!  
Arise, fair sun, and kill the envious moon,  
Who is already sick and pale with grief  
That thou her maid art far more fair than she.  
Be not her maid, since she is envious.  
Her vestal livery is but sick and green,  
And none but fools do wear it. Cast it off.  
It is my lady; O, it is my love!  
O that she knew she were!  
She speaks, yet she says nothing. What of that?
```



# Looping through the lines

```
String filename = "c:\\intro2cs\\textbooks\\Romeo.txt";
File f = new File(filename);
StringBuilder sb = new StringBuilder();
int numberOfLines = 0;
try {
    Scanner s = new Scanner(f);
    while (s.hasNextLine()) {
        numberOfLines++;
        sb.append(s.nextLine());
        sb.append("\n");
    }
} catch (FileNotFoundException e) {
    System.out.println("Text book " + filename + " not found.");
    return;
}
System.out.println("The Tragedy of Romeo and Juliet / William Shakespeare");
System.out.println(sb);
System.out.println("Total: " + numberOfLines + " lines.");
```



# STRING/TEXT/FILE MANIPULATIONS



Boaz Kantor  
Introduction to Computer Science  
IDC Herzliya