

# Using Classes

Scanner Class

# Lecture Contents

- Review
  - Using Classes
  - String Class
- Scanner Class
  - Instantiating
  - Basic Use
  - Methods
  - A Comment About User Input

# Using Classes – Review

- Math Class

```
Math.PI  
Math.sqrt(2.0)
```

- System Class

```
System.out.println("Hello World!");
```

- Notice we are using the `class` – remember class names are in Pascal Case (Upper Camel Case).

# String Class – Review

- These are (almost) equivalent ways to declare and initialize a string:

```
String s = "Hello world!";
```

```
String s;  
s = "Hello World!";
```

```
String s = new String("Hello World!");
```

```
String s;  
s = new String("Hello World!");
```

# String Class – Review

- We can call methods from the String class:

```
String s1 = "Hello World!";  
String s2 = s1.toUpperCase();  
  
System.out.println(s2);
```

HELLO WORLD!

# class Scanner

- A simple text parser
  - Often used for input from the keyboard
  - Can parse primitive types and strings
- Unlike `Math` and `String` classes, requires import:  

```
import java.util.Scanner;
```

# Scanner Class - Instantiating

- For the scanner class, we will need to create an *instance* of the class.
  - Compare to declaration and initialization of an integer and a String:

```
int myNum = 5; // primitive type
```

```
String s = "Hello World!"; //String object  
int len = s.length()
```

```
Scanner input = new Scanner(System.in);  
int age = input.nextInt();
```

# Scanner Class – Basic Use

- Basic use

```
import java.util.Scanner;
public class Repeat {
    private static Scanner input = new Scanner(System.in);
    public static void main(String args[]) {
        String s;
        System.out.println("Type Something then press <enter>");
        s = input.nextLine();
        System.out.println("You typed: \"" + s + "\"");
        input.close();
    }
}
```

# Scanner Class – Methods

```
String Scanner.nextLine()  
String Scanner.next()  
int Scanner.nextInt()  
double Scanner.nextDouble()
```

```
boolean Scanner.hasNext()  
boolean Scanner.hasNextInt()  
boolean Scanner.hasNextDouble()
```

# nextLine()

- Returns the next line of text as a `String`
  - Stops at the the next newline character.
  - The newline character is read and discarded.

```
System.out.println("Type something then press <enter>");  
String s = input.nextLine();  
System.out.println("You typed: \"" + s + "\"");
```

```
Type something then press <enter>  
Hello there!  
You typed: "Hello there!"
```

# next()

- Returns the next **token** as a String
  - Stops at the the next whitespace character (space, tab, newline).
  - The remainder of the line, including any newline character will remain in the read buffer.

```
System.out.println("Type something then press <enter>");  
String s = input.next();  
System.out.println("You typed: \"" + s + "\"");
```

```
Type something then press <enter>  
Hello there!  
You typed: "Hello"
```

# nextInt()

- Returns the next line of text
  - Attempts to parse the next characters as an integer
  - The remainder of the line, including any newline character will remain in the read buffer.

```
System.out.println("Enter a number then press <enter>");  
int i = input.nextInt();  
System.out.println("You entered: " + i);
```

```
Enter a number then press <enter>  
4  
You entered: 4
```

# nextInt()

- Returns the next line of text
  - Attempts to parse the next characters as an integer
  - **An exception will be thrown if the next token is not an integer!**

```
System.out.println("Enter a number then press <enter>");  
int i = input.nextInt();  
System.out.println("You entered: " + i);
```

```
Enter a number then press <enter>
```

```
hello
```

```
Exception in thread "main"
```

```
java.util.InputMismatchException at
```

```
java.base/java.util.Scanner.throwFor(Scanner.java:943)
```

```
...
```

# hasNextInt()

- Returns `true` if the next token can be interpreted as a valid `int`
  - No modification of the read buffer

```
System.out.println("Enter a number then press <enter>");
if(input.hasNextInt()) {
    int i = input.nextInt();
    System.out.println("You entered: " + i);
} else {
    System.out.println("Hey! That wasn't an integer!");
}
```

```
Enter a number then press <enter>
Hello there!
Hey! That wasn't an integer!
```

# nextDouble() and hasNextDouble()

- nextDouble() parses the next token and returns it as a double value.
- hasNextDouble() returns true if the next token can be interpreted as a valid double

```
System.out.println("Enter a number then press <enter>");
if(input.hasNextDouble()) {
    double d = input.nextDouble();
    System.out.println("You entered: " + d);
} else {
    System.out.println("Hey! That wasn't a number!");
}
```

```
Enter a number then press <enter>
5e21
You entered: 5.0E21
```

```
Enter a number then press <enter>
5
You entered: 5.0
```

# Scanner Class – Methods

```
String Scanner.nextLine()  
String Scanner.next()  
int Scanner.nextInt()  
double Scanner.nextDouble()
```

```
boolean Scanner.hasNext()  
boolean Scanner.hasNextInt()  
boolean Scanner.hasNextDouble()
```

# Using Classes

Scanner Class