SWEN20003

Workshop 4, Week 5

Eleanor McMurtry, University of Melbourne

Input in Java

Reading input in Java

- Preferred: the `Scanner` class

```
import java.util.Scanner;

class Program {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNext()) {
            System.out.println("Next: " + scanner.next());
        }
    }
}
```

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)
- `scanner.hasNext()`: returns true if the input has more tokens to read

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)
- `scanner.hasNext()`: returns true if the input has more tokens to read
- `scanner.next()`: returns the next token

Reading input in Java

- **Token:** a single “word” surrounded by whitespace (or the start/end of the string)
- `scanner.hasNext()`: returns true if the input has more tokens to read
- `scanner.next()`: returns the next token
- `scanner.nextline()`: returns all tokens until the next newline

Reading input in Java

- `Scanner` can read from any source, including `Strings`:

```java
import java.util.Scanner;

class Program {
    public static void main(String[] args) {
        String sample = "Hello World! Testing tokenisation...";
        Scanner scanner = new Scanner(sample);
        while (scanner.hasNext()) {
            System.out.println("Next: " + scanner.next());
        }
    }
}
```

Maven

What is Maven?

- Project management tool, similar to `Makefile`s
- Specifies source code structure, compiler settings, libraries...

Why Maven?

Why Maven?

- Allows other developers to run your code regardless of setup

Why Maven?

- Allows other developers to run your code regardless of setup
- Account for operating system or Java version differences

Why Maven?

- Allows other developers to run your code regardless of setup
- Account for operating system or Java version differences
- **Dependency:** another library, project, or piece of software your project relies on

Online dependencies

- **Pros:**
- **Cons:**

Online dependency ecosystems

- **Pros:** easy to use; don’t have to manually find libraries online

- **Cons:** can make your project bigger than it needs to be; awkward for developers with slow Internet

Bagel