

More Sophisticated Behaviour

Technical Support System V1.0

Produced Dr. Siobhán Drohan
by: Mairead Meagher



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Lectures and Labs

- This weeks remaining lectures and labs are based on examples in Chapter 5 of:
 - Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling

Topic List

- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class

The Java class library (API)

- Thousands of classes.
- Tens of thousands of methods.
- Many useful classes that make life much easier.
- A competent Java programmer must be able to work with the libraries.
- Documentation of the Java libraries is in HTML format (generated using javadoc comments).
- Readable in a web browser.
- *API: Application Programmers' Interface*

Using library classes

- Classes from the library must be imported using an *import* statement (except classes from *java.lang*).
- They can then be used like classes from the current project.

Working with the library

You should:

- know some important classes by name.
- know how to find out about other classes.

Remember:

- We only need to know the **interface**, not the **implementation**.
- API contains the **interface** description for all library classes.

Topic List

- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class

Interface vs implementation

The documentation includes

- the name of the class;
- a general description of the class;
- a list of constructors and methods
- return values and parameters for constructors and methods
- a description of the purpose of each constructor and method



the *interface* of the class

Interface vs implementation

*The documentation **does not** include*

- private fields (most fields are private)
- private methods
- the bodies (source code) for each method



the *implementation* of the class

Topic List

- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class

Technical Support System V1

- A console based system.
- A textual dialog system i.e. you enter text on the console and the system will provide a response.
- In this version, the system responds with the same String always:
 - “That sounds interesting. Tell me more...”

Technical Support System V1

 Problems @ Javadoc  Declaration  Console 

<terminated> SupportSystem [Java Application] C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (25 Jan 2016, 12:08:33)

Welcome to the DodgySoft Technical Support System.

Please tell us about your problem. We will assist you with any problem you might have. Please type 'bye' to exit our system.

> my computer is broken

That sounds interesting. Tell me more...

> really broken

That sounds interesting. Tell me more...

> help me

That sounds interesting. Tell me more...

> pleaseeeeeeee

That sounds interesting. Tell me more...

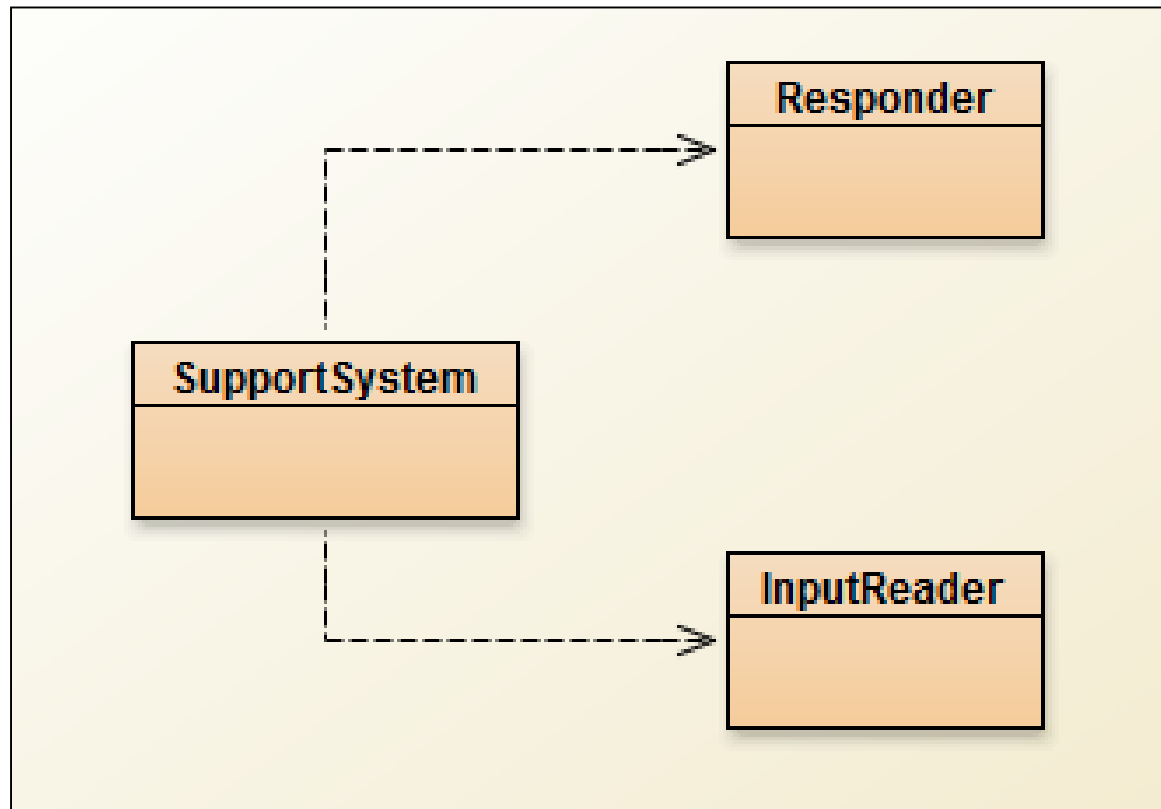
> BETY

That sounds interesting. Tell me more...

> BYE

Nice talking to you. Bye...

Class Diagram



Topic List

- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class

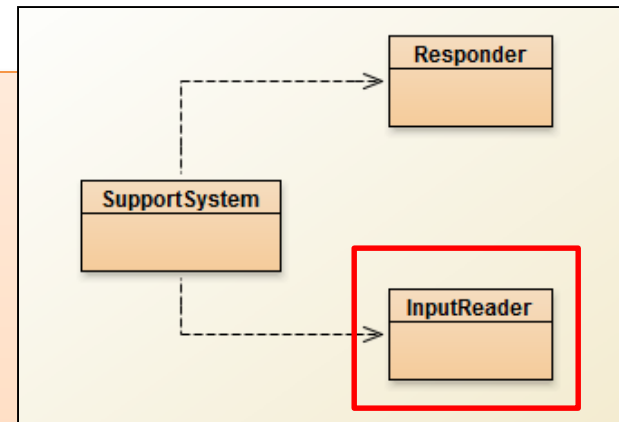
```
import java.util.Scanner;

public class InputReader{

    Scanner input;

    public InputReader(){
        input = new Scanner(System.in);
    }

    /**
     * Read a line of text from standard input (the text terminal),
     * and return it as a String.
     *
     * @return A String typed by the user.
     */
    public String getInput() {
        System.out.print("> ");           // print prompt
        String inputLine = input.nextLine().trim().toLowerCase();
        return inputLine;
    }
}
```



Topic List

- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class


```
public class Responder{
```

```
/**
```

```
 * Construct a Responder - nothing to do
```

```
 */
```

```
public Responder(){  
}
```

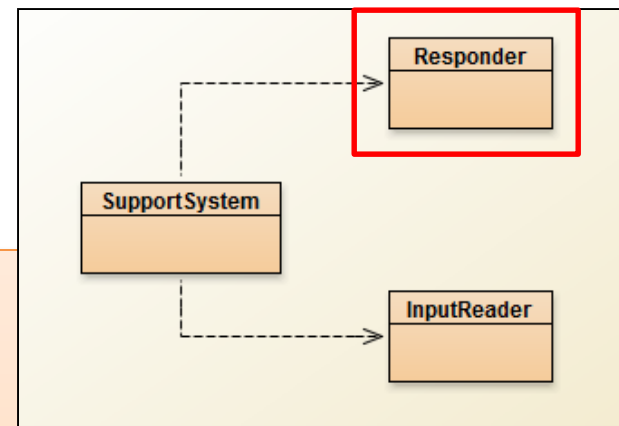
```
/**
```

```
 * Generate a response.
```

```
 * @return  A string that should be displayed as the response
```

```
 */
```

```
public String generateResponse(){  
    return "That sounds interesting. Tell me more...";  
}  
}
```



Topic List

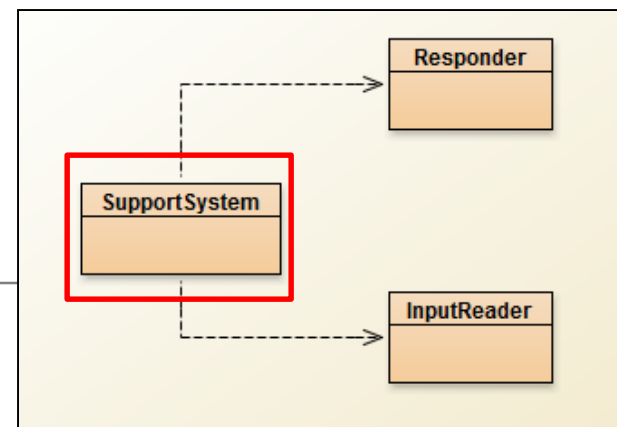
- Recap of Library Classes (Java's API).
- Interface Vs Implementation.
- Technical Support System V1:
 - Overview of the System
 - InputReader class
 - Responder class
 - SupportSystem class

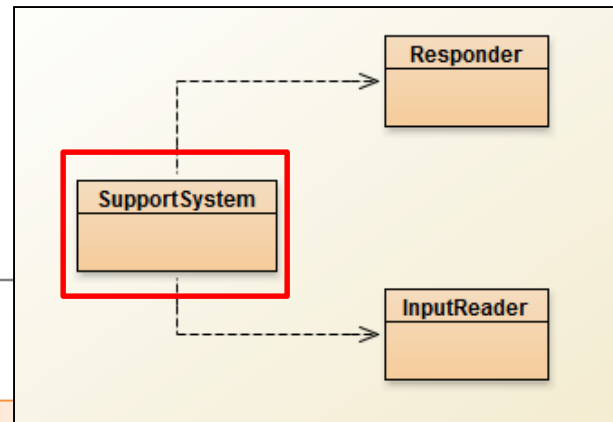
```
public class SupportSystem
{
    private InputReader reader;
    private Responder responder;

    public SupportSystem() {
        reader = new InputReader();
        responder = new Responder();
    }

    public static void main(String[] args){
        SupportSystem app = new SupportSystem();
        app.start();
    }

    public void start(){
        printWelcome();
        String input = reader.getInput();
        while(! input.startsWith("bye")) {
            String response = responder.generateResponse();
            System.out.println(response);
            input = reader.getInput();
        }
        printGoodbye();
    }
}
```





```
private void printWelcome(){
    System.out.println("Welcome to the DodgySoft Technical Support System.");
    System.out.println();
    System.out.println("Please tell us about your problem. We will assist you");
    System.out.println("with any problem you might have. Please type 'bye'");
    System.out.println("to exit our system.");
}

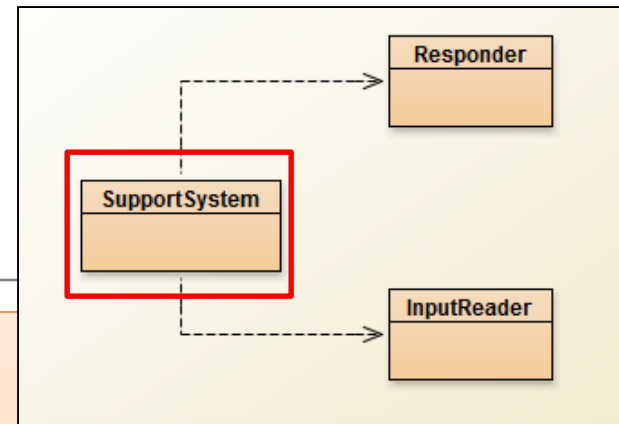
private void printGoodbye(){
    System.out.println("Nice talking to you. Bye...");
}

}
```

Main loop structure

```
public void start(){  
    printWelcome();  
    String input = reader.getInput();  
    while(! input.startsWith("bye")) {  
        String response = responder.generateResponse();  
        System.out.println(response);  
        input = reader.getInput();  
    }  
    printGoodbye();  
}
```

```
Get input  
while(input does not start with "bye"){  
    do something (i.e. print response)  
    Get some new input  
}
```



**Any
Questions?**

