More Sophisticated Behaviour

Technical Support System V1.0

Produced by:

Dr. Siobhán Drohan
Mr. Colm Dunphy
Mr. Diarmuid O'Connor
Dr. Frank Walsh

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

Lectures and Labs

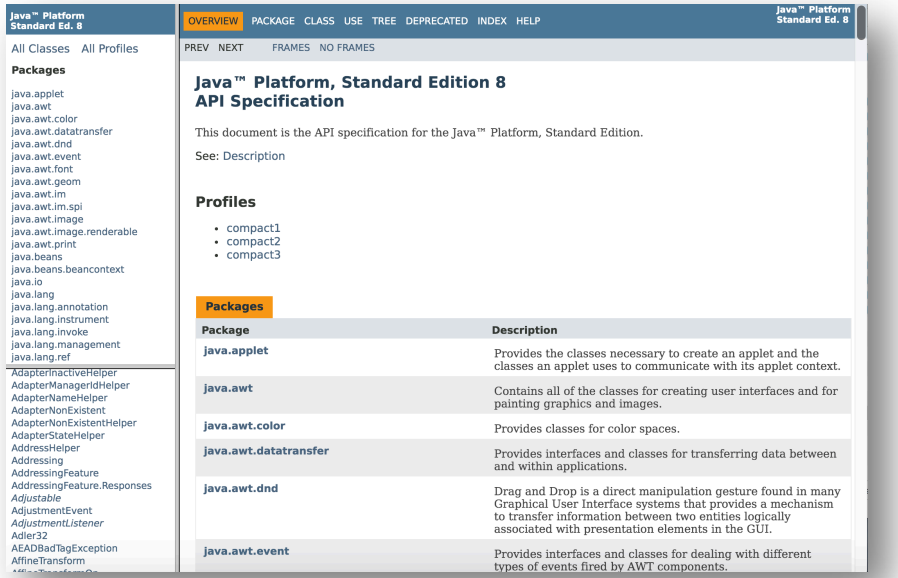
- The Tech Support System lectures and labs are based on examples in Chapter 5 of:
  - Objects First with Java
    - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling

Topic List

1. Recap of Library Classes (**Java's API**).
2. **Interface Vs Implementation**.
3. Technical Support System **V1**:
   - Three Classes:
     1. **InputReader** class
     2. **Responder** class
     3. **SupportSystem** class

The Java class library (**API**)

- **API**: *Application Programmers' Interface*



- 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.

https://docs.oracle.com/javase/8/docs/api/

Using **library classes - import**

- Classes from the library must be imported
  - using an ***import*** statement

```
import javax.swing.*;
```

```
import javax.swing.JOptionPane;
```

  - exception are classes from *java.lang*

- They can then be used like classes from the current project.

Working with the **library**

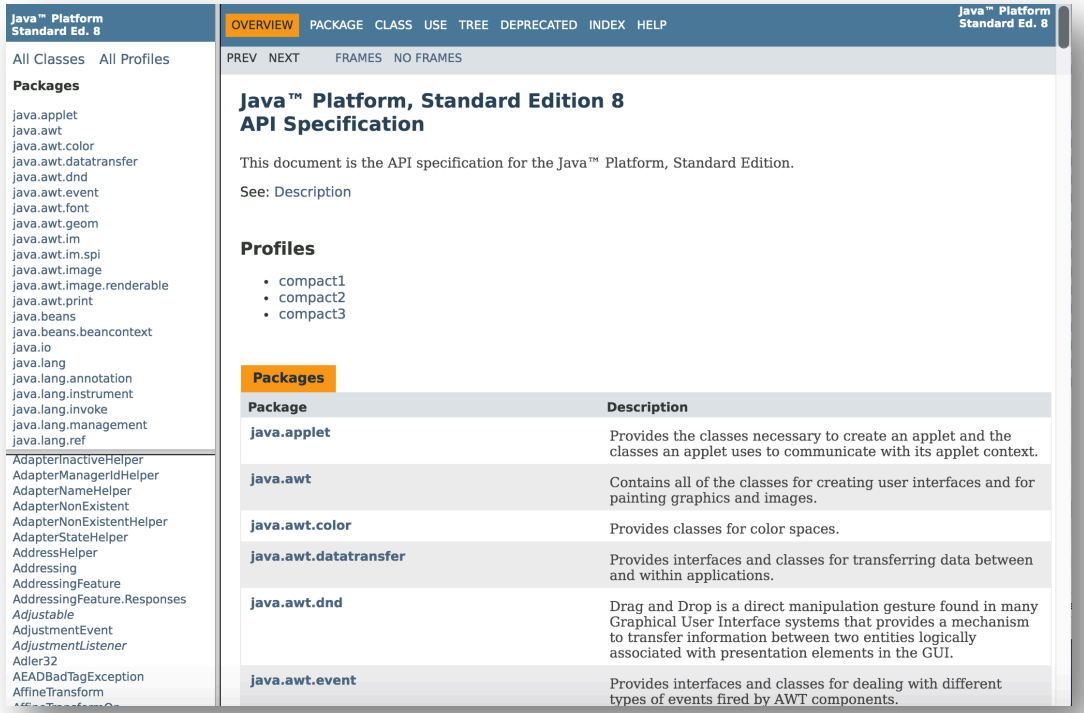
You should:

- know some important packages/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

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

Interface vs implementation

The documentation includes:

- Class **name**;
- Class **description**;
- List of **constructors** and **methods**
- **Return values** and **parameters** for constructors and methods
- Description of the purpose of each constructor and method

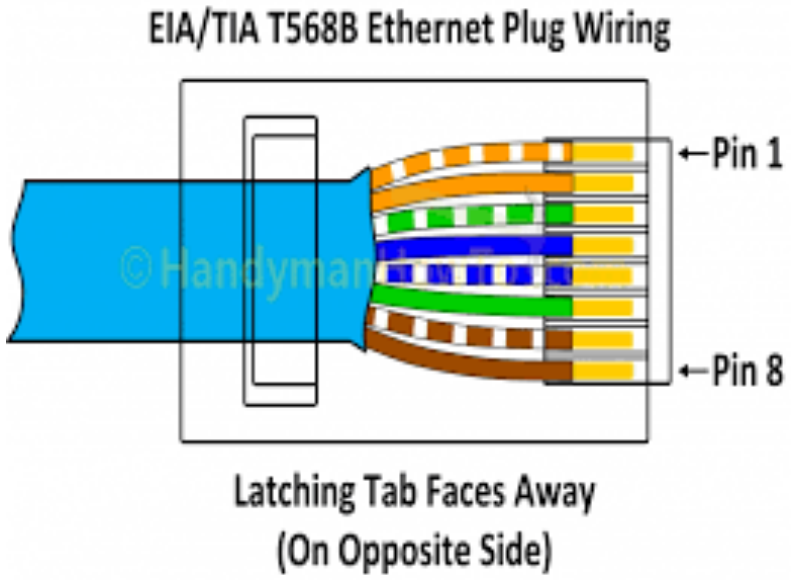
→ **the *interface* of the class**

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



Cat5e Wire Diagram for T568B (Straight Through Cable)				
RJ45 Pin #	Wire Color (T568A)	Wire Diagram (T568A)	10Base-T Signal 100Base-TX Signal	1000Base-T Signal
1	White/Orange		Transmit+	BI_DA+
2	Orange		Transmit-	BI_DA-
3	White/Green		Receive+	BI_DB+
4	Blue		Unused	BI_DC+
5	White/Blue		Unused	BI_DC-
6	Green		Receive-	BI_DB-
7	White/Brown		Unused	BI_DD+
8	Brown		Unused	BI_DD-

Any Questions?