

Design Patterns

MSc in Communications Software

Produced
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



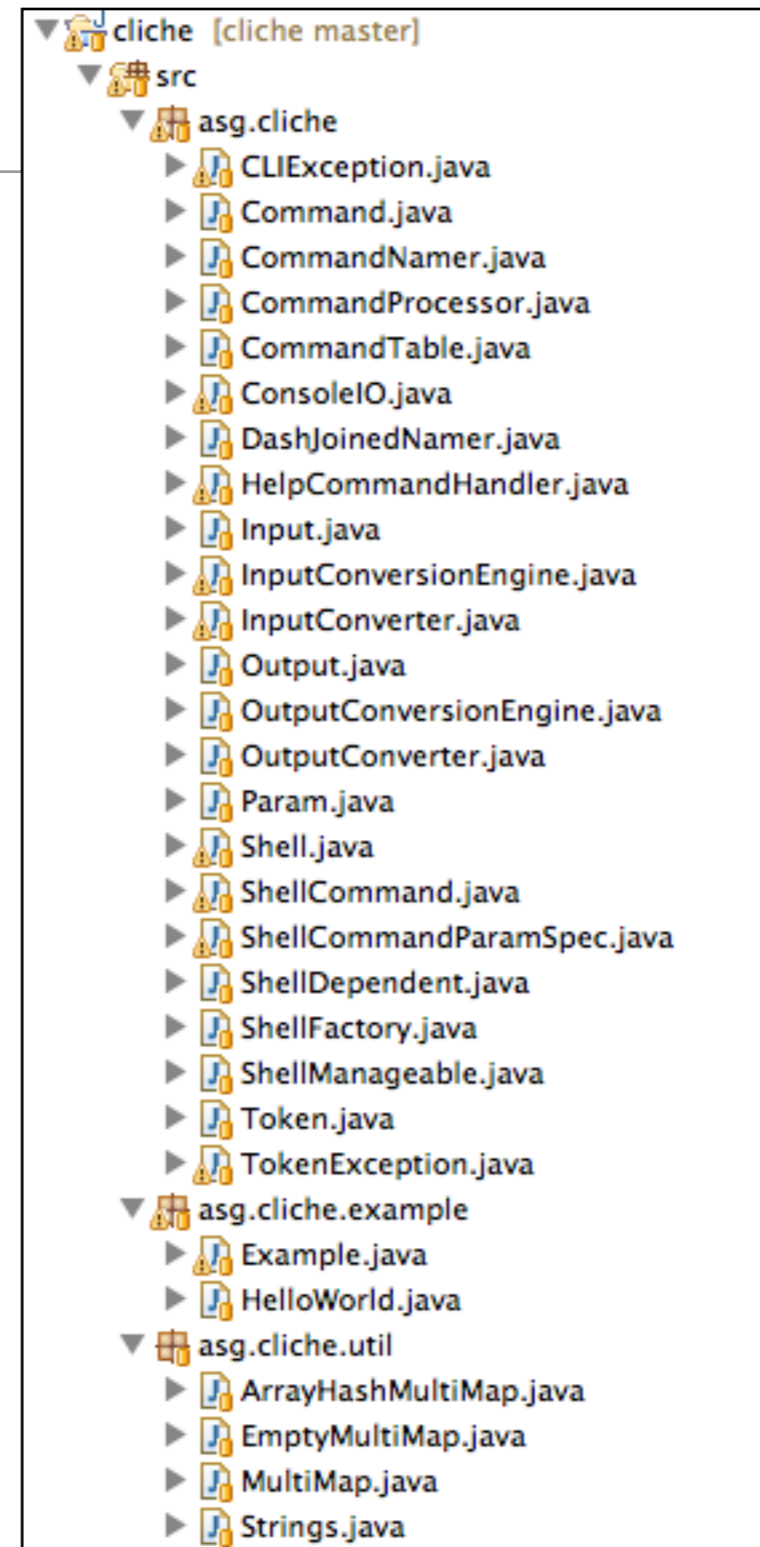
Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



Pacemaker Command

Cliche Library

- Open Source Library for quickly implementing Command Line applications
- Parses command line based on annotated class
- Directly invokes correct method



Cliche Example

```
public class HelloWorld
{
    @Command
    public int add(int a, int b)
    {
        return a + b;
    }

    @Command
    public String echo(String s)
    {
        return s;
    }

    public static void main(String[] params) throws IOException
    {
        ShellFactory.createConsoleShell("hello", null, new HelloWorld()).commandLoop();
    }
}
```

```
hello> ?list
abbrev name      params
a      add      (p1, p2)
h      hello    ()
hello> hello
Hello, World!
hello> ?help add
Command: add
Abbrev: a
Params: (p1, p2)
Description: add(p1:int, p2:int) : int
Number of parameters: 2
p1      int
p2      int

hello> add 45 34
79
hello> a 45 34
79
hello> exit
```

Pacemaker commands (subset)

```
public class PacemakerShell
{
    //...
    public PacemakerShell()
    {
        paceApi = new PacemakerAPI();
        pacemaker = new PacemakerService (paceApi, new AsciiParser());
    }

    @Command(description="List all users details")
    public void listUsers () throws Exception
    {
        System.out.println(pacemaker.getUsers());
    }

    @Command(description="Create a new User")
    public void createUser (@Param(name="first name") String firstname, @Param(name="last name") String lastname,
        @Param(name="email") String email, @Param(name="password") String password) throws Exception
    {
        println (pacemaker.createUser(firstname, lastname, email, password));
    }

    @Command(description="Delete a User")
    public void deleteUser (@Param(name="id") Long id)
    {
        println (pacemaker.deleteUser(id));
    }
}
```

Transcript

- Cliche library reads and interprets command lines..
- ... and triggers call directly to annotated methods:

```
@Command(description="Delete a User")
public void deleteUser (@Param(name="id") Long id)
{
    println (pacemaker.deleteUser(id));
}
```

```
Welcome to pacemaker-console - ?help for instructions
pm> help
[delete-user, help, undo, list-users, redo, create-user]
pm>
pm> cu homer simpson homer@simpsin.com secret
+-----+-----+-----+-----+-----+
| ID | FIRSTNAME | LASTNAME | EMAIL | PASSWORD |
+-----+-----+-----+-----+-----+
| 1 | homer | simpson | homer@simpsin.com | secret |
+-----+-----+-----+-----+-----+
pm> cu marge simpson marge@simpson.com secret
+-----+-----+-----+-----+-----+
| ID | FIRSTNAME | LASTNAME | EMAIL | PASSWORD |
+-----+-----+-----+-----+-----+
| 2 | marge | simpson | marge@simpson.com | secret |
+-----+-----+-----+-----+-----+
pm> lu
+-----+-----+-----+-----+-----+
| ID | FIRSTNAME | LASTNAME | EMAIL | PASSWORD |
+-----+-----+-----+-----+-----+
| 1 | homer | simpson | homer@simpsin.com | secret |
| 2 | marge | simpson | marge@simpson.com | secret |
+-----+-----+-----+-----+-----+
pm> du 1
pm> lu
+-----+-----+-----+-----+-----+
| ID | FIRSTNAME | LASTNAME | EMAIL | PASSWORD |
+-----+-----+-----+-----+-----+
| 2 | marge | simpson | marge@simpson.com | secret |
+-----+-----+-----+-----+-----+
pm>
```

Cliche / Strategy

- Modify Cliche and install 'strategy' to intercept command dispatching mechanism
- This strategy will be called for all commands, and will replace existing dispatching mechanism
- "ShellCommand" object passed + list of parameters read from the console

```
public interface CommandProcessor
{
    public void doCommand(ShellCommand command, Object[] parameters);
}
```

PacemakerShell

```
public class PacemakerShell implements CommandProcessor
{
    //...

    public PacemakerShell()
    {
    }

    @Override
    public void doCommand(ShellCommand command, Object[] parameters)
    {
        // Handle all commands here...
    }

    public static void main(String[] args) throws Exception
    {
        PacemakerShell main = new PacemakerShell();
        CommandSpecifications commandSpecs = new CommandSpecifications();

        Shell shell = ShellFactory.createConsoleShell("pm", "Welcome to pacemaker-console - ?help for instructions", commandSpecs, main);
        shell.commandLoop();
    }
}
```

- Create a shell and pass a ‘CommandProcessor’ (main above) + ‘commandSpecs’
- Commands will be parsed and passed to ‘doCommand’ method for fulfillment.

CommandSpecifications

- CommandSpecifications now an empty
- Annotations introspected by Cliche to guide command parsing.
- Dispatching of commands now handled via CommandProcessor strategy

```
public class CommandSpecifications
{
    @Command(description="List all users details")
    public void listUsers () throws Exception
    {}

    @Command(description="undo last command")
    public void undo () throws Exception
    {}

    @Command(description="redo last command")
    public void redo () throws Exception
    {}

    @Command(description="Create a new User")
    public void createUser (@Param(name="first name") String firstname, @Param(name="last name") String lastname,
                           @Param(name="email") String email, @Param(name="password") String password) throws Exception
    {}

    @Command(description="Delete a User")
    public void deleteUser (@Param(name="id") Long id)
    {}

    @Command(description="Help")
    public void help ()
    {}
}
```



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE

