Design Patterns and Frameworks	Exercise Sheet No. 12
DiplInf. Florian Heidenreich	Software Technology Group
INF 2080	Institute for Software and Multimedia Technol-
http://st.inf.tu-dresden.de/teaching/dpf	ogy
	Department of Computer Science
	Technische Universität Dresden
	01062 Dresden

## **OSGi and Design Patterns**

## Task 1: The OSGi Framework

Download Eclipse Classic 3.5.1 from http://www.eclipse.org and install ObjectTeams (OT) via the update manager of Eclipse. More information on installing OT can be found on the OT download site.<sup>1</sup>

Read the tutorial of Lars  $Vogel^2$  and download the exercise Equinox runtime from the course website. You have to register the target runtime in the preferences as depicted in the following screenshot.

		Preferences		
targ	0 1	Farget Platform	\$.\$. ₹	
♥Plug-in Development Target Platform		Add, edit and remove target definitions. Th as the target platform which workspace plu against. New definitions are stored locally, n the workspace. Target Definitions:	e active target definition will be used g-ins will be compiled and tested but they can be moved to a project	
		☑ ॏDPF (Active) ☐ ● Running Platform	Reload	
			Add	
			(Remove)	
			Move	
?			Cancel OK	

1a)

Now create a new OT Plug-in Project dpf.osgi for the OSGi framework Equinox. Make sure you add org.eclipse.core.runtime to the plug-in dependencies.

Create the following plugin.xml in the plug-in's root folder.

<sup>&</sup>lt;sup>1</sup>http://objectteams.org/distrib/install\_1.3.html

 $<sup>^{2}</sup> http://www.vogella.de/articles/OSGi/article.html$ 

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
   id="app"
   point="org.eclipse.core.runtime.applications">
    <application
      cardinality="singleton-global"
      thread="any"
      visible="true">
      <riin
        class="dpf.osgi.Apparat">
      </run>
    </application>
  </extension>
</plugin>
```

Implement an application using the following code.

```
import org.eclipse.equinox.app.IApplication;
import org.eclipse.equinox.app.IApplicationContext;
public class Apparat implements IApplication {
    @Override
    public Object start(IApplicationContext context) throws Exception {
      Person hans = new Person("Hans");
      Person karl = new Person("Karl");
      hans.identify();
      karl.identify();
      return null;
    }
    @Override
    public void stop() { }
}
```

Now create a second OT Plug-in Project dpf.osgi.base without generating an Activator and which only consists of the Person class outlined below.

```
public class Person {
  public String name;
  public Person(String name) {
    this.name = name;
  }
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public void identify() {
    System.out.println(name);
  }
}
```

Now create a new OSGi run configuration, make sure that all bundles are activated and the Enable OT/Equinox setting is enabled.

type	filter text	G	Select All	
B	indles	Start Level	Auto-Start	
	Workspace	4.7.4.	4.6.4.	Deselect All
	aprospi (1.0.0.qualifier)	default	default	
	def and (1.0.0 qualifier)	default	default	Add Working Set
	Tavast Blatform	default	default	
2	arget ration	default	default	Add Required Bundles
3	b org eclipse core jobs (3.4.100 v20090429-1800)	default defa	default	
3	horg eclipse core runtime (3.5.0 v20090525)	default	true	Restore Defaults
1	Corg eclipse equipox app (1.2.0 v20090520-1800)	default	default	
1	org.eclipse.equinox.common (3.5.1.835x v20090807-1100)	2	true	
1	areclipse.equinox.preferences (3.2.300.v20090520-1800)	default	default	
1	p. org.eclipse.equinox.registry (3.4.100.v20090520-1800)	default	default	
1	apeorg.eclipse.osgi (3.5.1.R35x_v20090827)	-1	true	
1	porg.objectteams.otequinox (1.3.2.200909280002)	default	default	
1	worg.objectteams.otequinox.hook (1.3.2.200909280002)	default	false	
<b>v</b>	org.objectteams.otequinox.runtime (1.3.2.200909280002)	default	false	
-	🐌 org.objectweb.asm (3.1.0.v200803061910)	default	default	Only show selected bundle
			*******	15 out of 20 selected
🗹 In	clude optional dependencies when computing required bundles			
A N	ld new workspace bundles to this launch configuration automatically			

Run the application. The commands  $\tt help, apps$  and  $\tt startApp$  on the OSGi console should get you started.

Additional hints can be found at  $^3$  and  $^4$ .

## Task 2: The OSGi Framework

## 2a)

Try out and understand the ObjectTeams Observer pattern example which can be found under File  $\Rightarrow$  New... $\Rightarrow$  Examples...

 $<sup>^3 \</sup>rm http://trac.objectteams.org/ot/wiki/OtEquinox/HowtoCreateOTEquinoxAppFromOTApp Section 1 <math display="inline">^4 \rm http://trac.objectteams.org/ot/wiki/OtEquinox$ 

and application of	the standard Obse	rver pattern with	
iva.			5-6-
8			0
nple nus Example			
Pattern Example			
tem Example			
	n and application of ava.	and application of the standard Obse ava. nple nus Example Pattern Example them Example the Example	n and application of the standard Observer pattern with ava. nple nus Example Pattern Example the Example the Example

The ObjectTeams Language Reference<sup>5</sup> provides detailed explanation on roles in ObjectTeams.

```
2b)
```

In this subtask we are going to extend the example of Task 1. Create an ObjectTeams Equinox Plug-in project dpf.osgi.ext (without generating an Activator).

Import the exported package of dpf.osgi.base at dpf.osgi.ext and the packages of dpf.osgi.base and dpf.osgi.ext at dpf.osgi.

Now integrate the Team below in the new plug-in and change the implementation of IApplication in the dpf.osgi plug-in.

```
public team class University {
    public void register(Person as Student ersti, int matrikel) {
        ersti.matrikel = matrikel;
    }
    public class Student playedBy Person {
        studentIdentify <- before identify;
        public int matrikel;
        public void studentIdentify() {
            System.out.println("Matrikel: "+matrikel);
        }
    }
}</pre>
```

 $<sup>^{5}</sup> http://object$ teams.org/def/1.2/OTJLDv1.2-current.pdf

```
import org.eclipse.equinox.app.IApplication;
import org.eclipse.equinox.app.IApplicationContext;
public class Apparat implements IApplication {
 @Override
 public Object start(IApplicationContext context) throws Exception {
   Person hans = new Person("Hans");
   Person karl = new Person("Karl");
   System.out.println("--No context--");
   hans.identify();
   karl.identify();
   University u = new University();
   u.activate();
   u.register(hans, 123);
   u.register(karl, 345);
   System.out.println("--Uni active-----");
   hans.identify();
   karl.identify();
   u.deactivate();
   System.out.println("--Uni inactive---");
   hans.identify();
   karl.identify();
   return null;
 }
 @Override
 public void stop() { }
7
```

To start this application some changes need to be integrated in the Eclipse run configuration.

First, register the extension point org.objectteams.otequinox.aspectBindings in the plugin.xml.<sup>6</sup> It should look like this:

```
<extension point="org.objectteams.otequinox.aspectBindings">
<aspectBinding
icon="platform:/plugin/org.objectteams.otdt.ui/icons/ot/calloutbinding_obj.gif">
<basePlugin
icon="platform:/plugin/org.eclipse.pde.ui/icons/obj16/plugin_obj.gif"
id="dpf.osgi.base">
</basePlugin>
<team
activation="ALL_THREADS"
class="dpf.osgi.ext.University"
icon="platform:/plugin/org.objectteams.otdt.ui/icons/ot/team_obj.gif">
</team
activation="ALL_THREADS"
class="dpf.osgi.ext.University"
icon="platform:/plugin/org.objectteams.otdt.ui/icons/ot/team_obj.gif">
</team>
</team>
</team>
</team>>
```

Additionally, add these parameters to the VM arguments:

```
-Declipse.ignoreApp=true

-Dosgi.noShutdown=true

-XX:+UnlockDiagnosticVMOptions

-XX:+UnsyncloadClass

-Dot.equinox=0

-Dosgi.hook.configurators.include=org.objectteams.eclipse.transformer.hook.HookConfigurator

-Dosgi.framework.extensions=org.objectteams.eclipse.transformer.hook
```

 $<sup>^{6}</sup> http://trac.objectteams.org/ot/wiki/OtEquinox/AspectBinding$ 

2c)

Extend the previous solutions with the role **professor**. A professor can be identified through his group membership.

```
---Uni inactive----

...

---Uni active-----

Matrikel: 123

Hans

Matrikel: 345

Anja

Peter is professor of the database group

---Uni inactive---

...
```

2d)

Extend the solution with lectures, which are held by professors and attended by students. Implement a method which returns all lectures attended by a student.