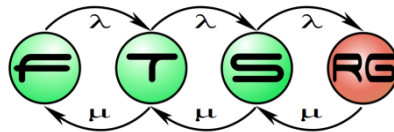


Az e4 projekt



e4 platform

- Eclipse platform új változata
 - Újabb technológiák
 - Modellezés
 - OSGi szolgáltatások

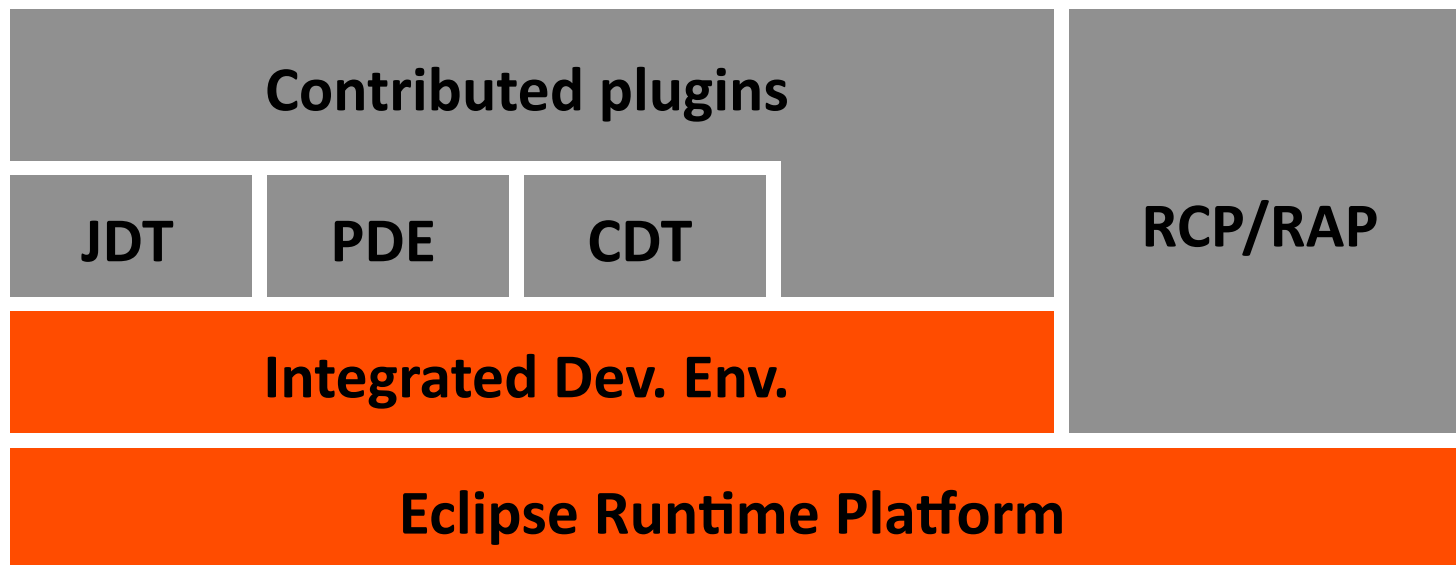
Szószedet

- e4: Eclipse.org alprojekt
 - Incubation
 - Nem termék
- Eclipse 4.0
 - Kiadás
 - e4 technológiákra építve

Eclipse architektúra

Ismétlés

Architektúra



Workbench felépítése

Workbench window

The screenshot displays the Eclipse IDE interface. The top window title is "Java - hu.optxware.eclipsecourse.rcpdemo.gui/src/hu.optxware.eclipsecourse.rcpdemo.gui/ServiceInterface.java - Eclipse SDK - /Users/stampie/Document...". The Package Explorer on the left shows a project structure with a package named "hu.optxware.eclipsecourse.rcpdemo.gui" containing several Java files, including "ServiceInterface.java". The central Editor window shows the code for "ServiceInterface.java", which defines a "ServiceInterface" with methods "serviceCreated()" and "serviceRemoved()". The Outline view on the right shows the "ServiceInterface" class and its methods. The Problems view at the bottom is empty, showing "0 items". The status bar at the bottom indicates "Writable", "Smart Insert", and "1 : 1".

```
ServiceInterface {  
    serviceCreated();  
    serviceRemoved();  
}
```

Description	Resource	Path	Location	Type

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The main toolbar is highlighted with a blue box. A callout bubble with a dark red background and white text points to the toolbar, containing the text "Main toolbar". The IDE shows a Java project with a package explorer on the left, a central editor window displaying the code for `ServiceInterface.java`, and an outline view on the right. The bottom status bar shows "Writable", "Smart Insert", and "1 : 1".

Main toolbar

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {
    void serviceCreated();
    void serviceRemoved();
}
```

Description	Resource	Path	Location	Type

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project structure with a package named `hu.optxware.eclipsecourse.rcpdemo.gui`. The main editor window shows the `ServiceInterface.java` file with the following code:

```
package hu.optxware.eclipsecourse.rcpdemo.gui;  
  
public interface ServiceInterface {  
    void serviceCreated();  
    void serviceRemoved();  
}
```

The Outline view on the right shows the `ServiceInterface` interface with its methods: `serviceCreated() : void` and `serviceRemoved() : void`. A red callout box with the text "Workbench Site" points to the `ServiceInterface.java` file in the Package Explorer.

At the bottom of the IDE, the Problems view is visible, showing 0 items. The status bar at the very bottom indicates "Writable", "Smart Insert", and "1 : 1".

Workbench felépítése

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure for 'hu.optxware.eclipsecourse.rcpdemo.gui'. The main editor window shows the 'ServiceInterface.java' file with the following code:

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

A blue box highlights the Java icon in the top toolbar, and a red callout bubble points to it with the text "Perspektíva választó".

The Problems view at the bottom shows 0 items.

Description	Resource	Path	Location	Type

Workbench felépítése

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with the `ServiceInterface.java` file selected. The main editor window shows the code for the `ServiceInterface` class, which is a public interface with two methods: `serviceCreated()` and `serviceRemoved()`. The Outline view on the right shows the class hierarchy, including the `ServiceInterface` class and its methods. The Problems view at the bottom shows 0 items.

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

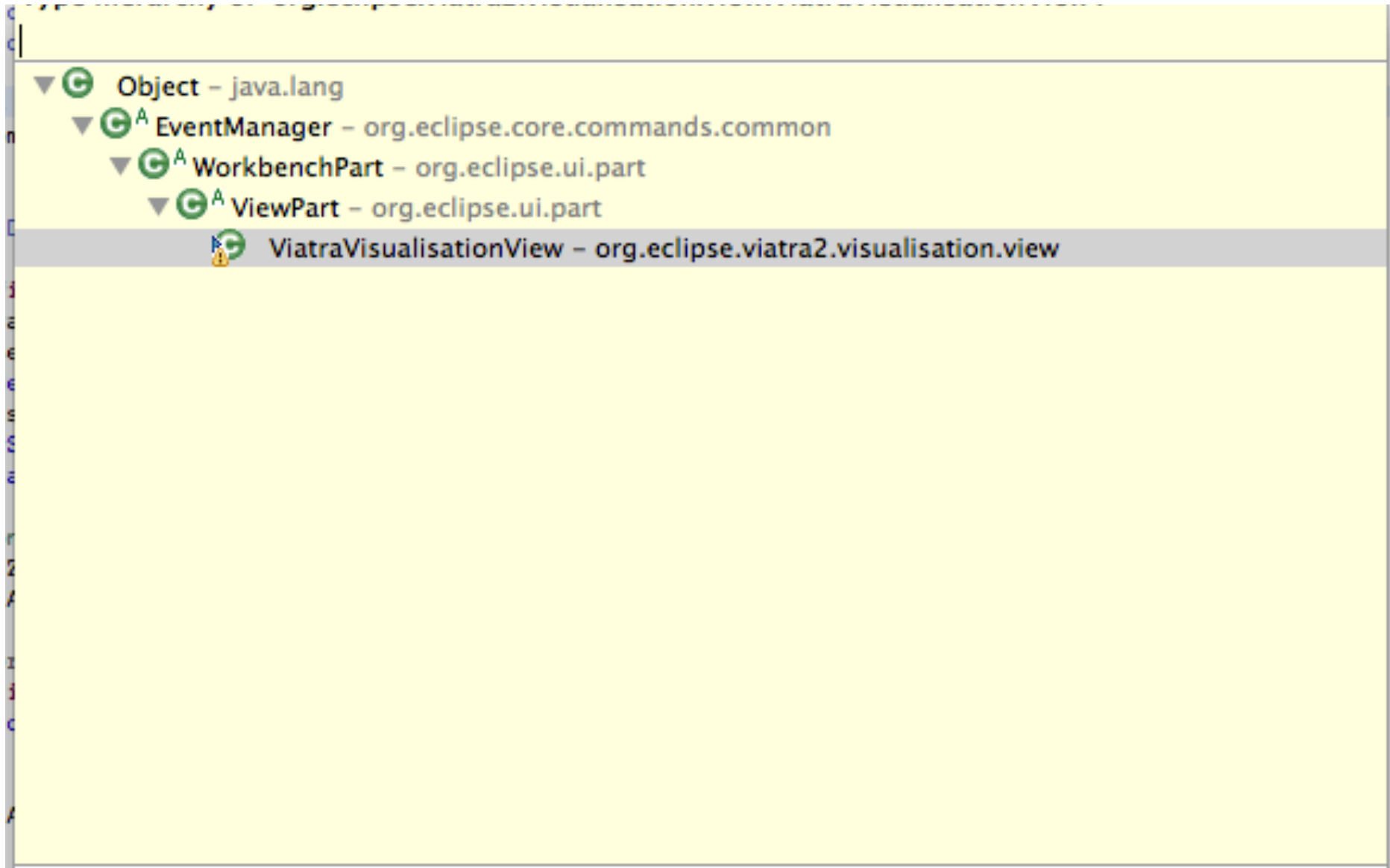
    public void serviceCreated();

    public void serviceRemoved();

}
```

Nézetek (0..N)
csoportokban

Implementáció



Workbench felépítése

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with several Java files. The main editor window shows the code for `ServiceInterface.java`. A blue rectangular box highlights the editor area. Two callout boxes with a dark red background and white text point to the editor:

- A callout box on the left contains the text: **Szerkesztők (0..N)**
- A callout box on the right contains the text: **Megnyitott fájlok**

```
package hu.optxware.eclipsecourse.rcpdemo.gui;

public interface ServiceInterface {

    public void serviceCreated();






    public void serviceRemoved();

}
```

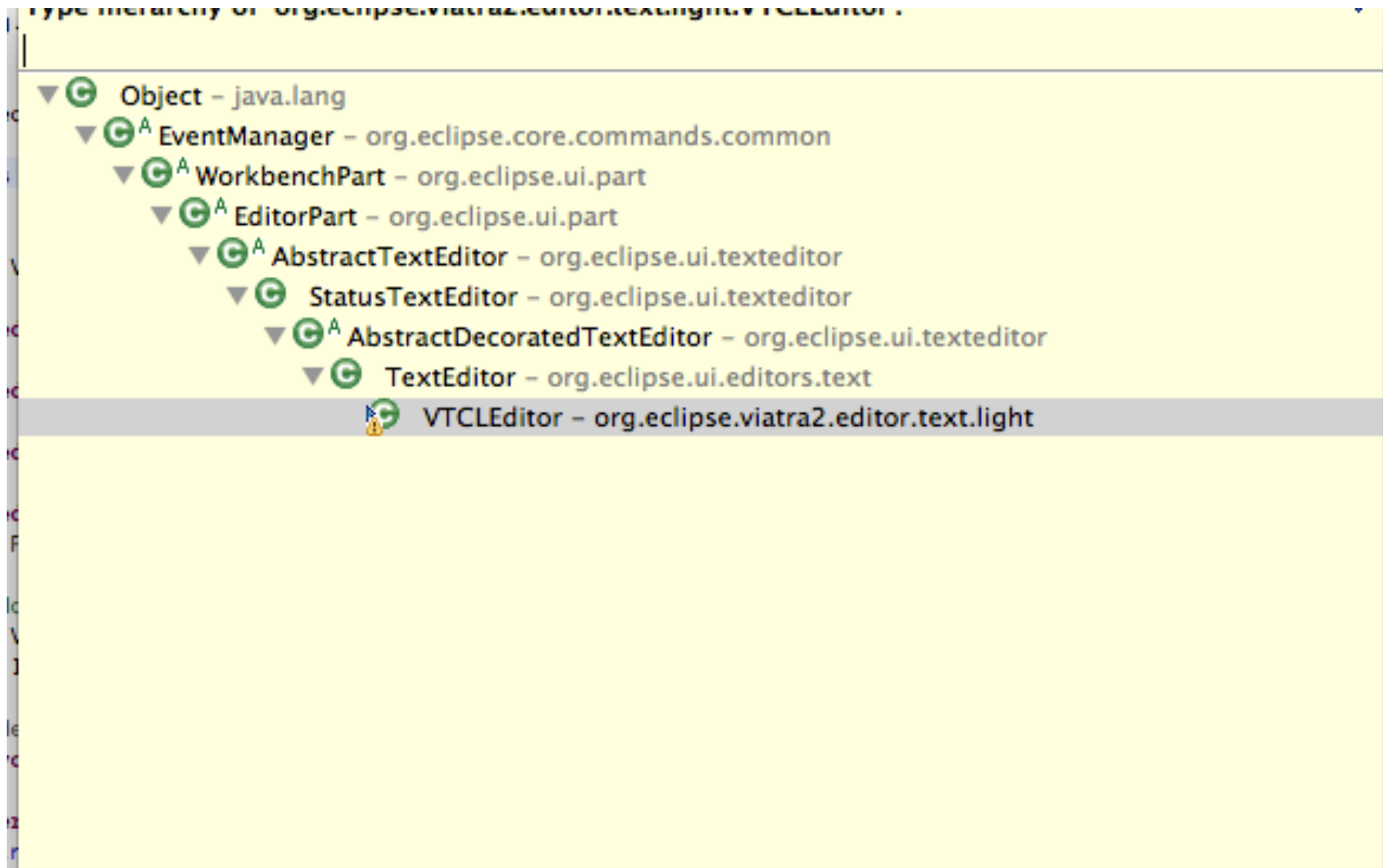
At the bottom of the IDE, there is a table with the following structure:

Description	Resource	Path	Location	Type
0 items				

Implementáció

- ▼  Object - java.lang
 - ▼  ^AEventManager - org.eclipse.core.commands.common
 - ▼  ^AWorkbenchPart - org.eclipse.ui.part
 - ▼  ^AEditorPart - org.eclipse.ui.part
 -  ^AViatraTreeEditor - org.eclipse.viatra2.treeditor

Implementáció – Szöveges editor



Platform szolgáltatások

- Szolgáltatások elérése: Singleton objektumok
 - PlatformUI.getWorkbench()
 - Platform.getExtensionRegistry()
 - ResourcesPlugin.getWorkspace()
 - ...

Beállítások elérése

```
IPreferenceStore store =  
    IDEWorkbenchPlugin.getDefault()  
        .getPreferenceStore();  
boolean saveBeforeBuild =  
    store.getBoolean  
        (SAVE_BEFORE_BUILD);
```


Súgó hozzárendelése widgethez

```
IWorkbenchHelpSystem helpSystem =  
    getSite()  
        .getWorkbenchWindow()  
            .getWorkbench()  
                .getHelpSystem();  
helpSystem.setHelp(control, "some_context_id");
```

Problémák

- Bonyolult elérés
 - Tesztelés nehézkes -> mock testing
- Szoros függőség
 - Ismerjük plug-in belső felépítését!
 - Platform nehezen fejleszthető!
 - Kompatibilitás
- Párhuzamos adatstruktúrák
 - ViewRegistry, EditorRegistry, Context

e4 platform

Modeled Workbench, Dependency Injection

■ Források:

- http://wiki.eclipse.org/E4/UI/Modeled_UI
- <http://wiki.eclipse.org/images/c/c3/PartsAndServices2010.zip>
- <http://www.eclipse.org/e4/resources/contexts.pdf>

Eclipse 4.0 SDK Early Adopter Release

Rest of Platform:
Debug,
Text,
Team,
...

JDT:
Java Editor,
Java Debugging,
Refactoring,
...

PDE:
Manifest Editor,
Launching,
API Tooling,
...

Workbench (Compatibility, provides 3.x APIs)

Modeled UI, CSS styling, Dependency Injection, Application Services

Eclipse 4
Application Platform

Equinox



EMF Core



SWT, JFace



Java Virtual Machine

OS (Windows, Mac, Linux)

XWT

- <http://wiki.eclipse.org/images/a/ab/XWT.pdf>