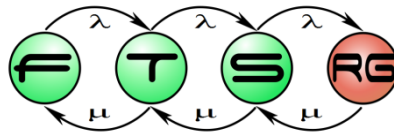


# Bevezető

## Eclipse alapú fejlesztés és integráció



# Adminisztráció

# Adminisztráció

## ■ Órák

- Előadás: K 16-18 IE224
- Labor: CS 8-10 IE224, páros hét (?)
  - Kötelező (katalógus)

## ■ Követelmények

- Házi feladat
- Vizsga

# Házi feladat

- Saját fejlesztés Eclipse környezetben
  - Eclipse plug-in
    - Indokolt esetben RAP/RCP alkalmazás
  - Domain specifikus nyelv
- Témalista a weben
  - Lehet saját is
- Értékelés
  - Függ a választott feladat nehézségétől
  - Felhasznált Eclipse alapú technológiától

# Házi feladat

- Önálló munka
  - Indokolt esetben csapatmunka
    - 2-3 fő, behatárolt részfeladatok!
- Ütemezés
  - 6. hét végére: részletes specifikáció
  - Utolsó oktatási hét: beadás és védés

# Információk

- Tárgy segédanyagai
  - Előadás fóliák
  - Egyéb segédanyagok
  - <http://inf.mit.bme.hu/courses/eafi>
- Eclipse honlap
  - [www.eclipse.org](http://www.eclipse.org)
  - Wiki: [wiki.eclipse.org](http://wiki.eclipse.org)
  - Cikkek: [eclipse.org/resources](http://eclipse.org/resources)
  - További linkek Eclipse oldalakhoz

# Kis ZH

- Név nélkül 😊 - papíron, 5 perc
- Feladatok
  - Értékelj pár szóban a Java jártasságod!
  - Milyen Java fejlesztőrendszereket ismertél meg eddig, mennyire (1-5)
  - Milyen más programozási nyelveket/eszközöket ismersz?
  - Mondanak Neked valamit az alábbi szavak (1-5)?
    - Eclipse plug-in, JDT, meta-modell, DSM, EMF, GEF, GMF, IMP, Xtext, nyelvtanok/elemzők

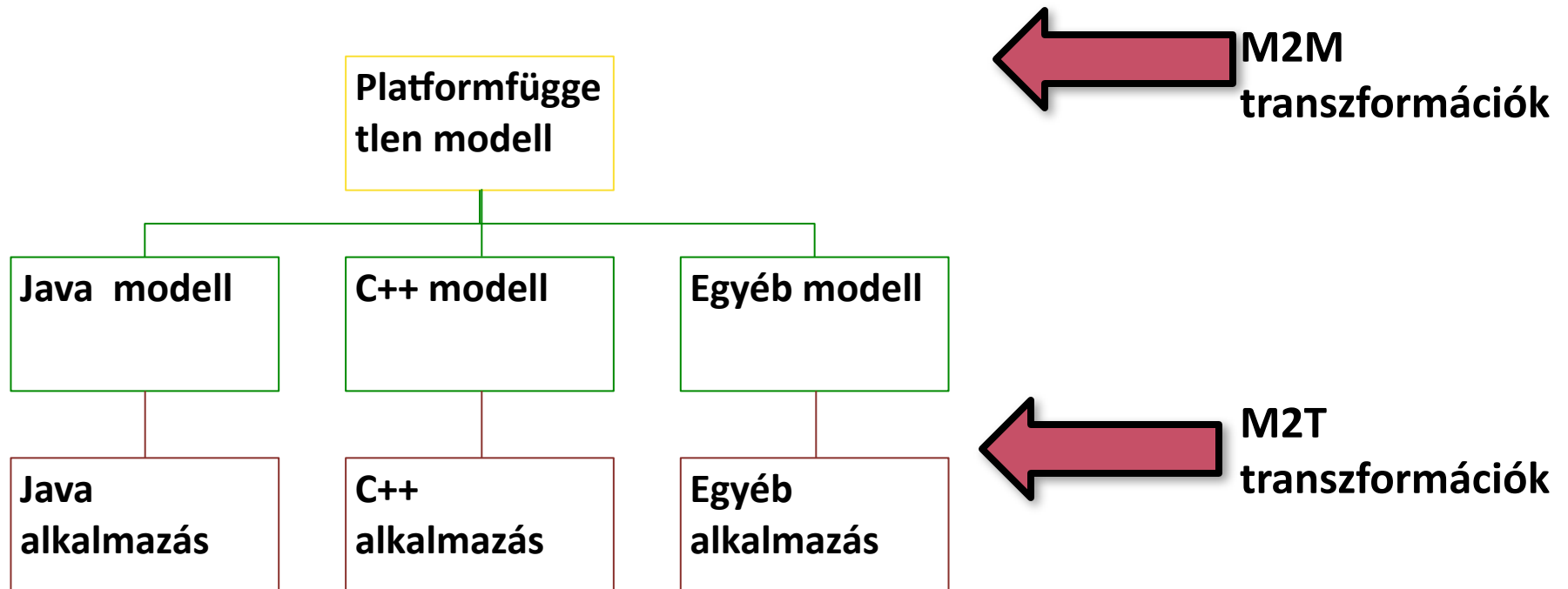
# Eclipse: Nyílt fejlesztőrendszer és platform



# Miért indul ez a tárgy?

- Szakterület-specifikus modellezés
  - (domain-specific modeling)
  - Specifikus modellezési környezetek
    - Eszköztámogatás kritikus
    - Integrációs előnyök
- Cél
  - Bővíthető modellezési környezet (Eclipse)
  - Modellezés és alkalmazásfejlesztés együttesen

# Modellvezérelt architektúra



# Eszköztámogatás

- Modellező környezet
  - Szöveges modellek
  - Grafikus modellek
- Modellek “helyessége”
  - Metamodellezés
  - Validációk
- Modellek átalakítása
  - szöveggé: kód-/konfiguráció generálás;
  - más modell: modelltranszformáció

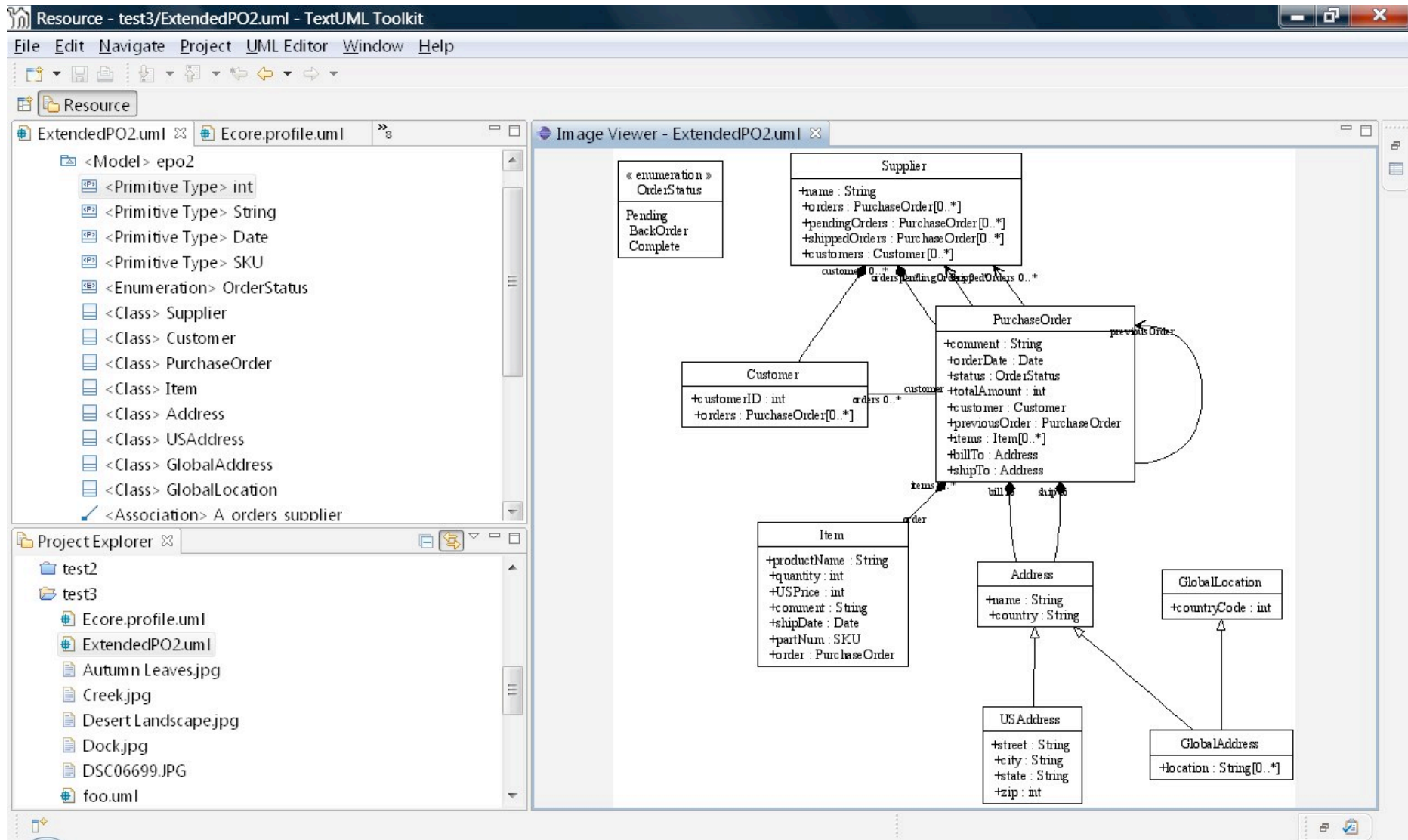
# Eclipse Modeling Project

- Eclipse 3.6 (Helios)
  - 33 projekt és alprojekt
- További projektek
  - Több-kevesebb támogatás
  - Korai fejlesztési fázisban (incubation)
- Jó néhány előkerül a tárgy során

# Eclipse Modeling Project

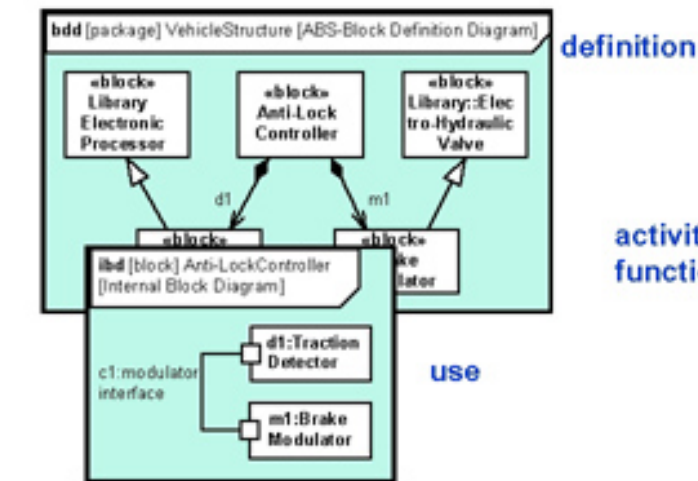
<b>Eclipse Modeling Project</b>	✓	●	🗨️	📄	●	
<b>Amalgam</b>	🥚	●	🗨️	📄	●	?
<b>Agent Modeling Platform (AMP)</b>	🥚	●	🗨️	📄	●	?
<b>Eclipse Modeling Framework (EMF)</b>	✓	●	🗨️	📄	●	?
<b>Modeling Workflow Engine</b>	✓					
<b>Eclipse Model Framework Technology (EMFT)</b>	🥚	●	🗨️	📄	●	?
<b>B3</b>	🥚	●	🗨️	📄	●	
<b>Graphical Modeling Project</b>	🥚					
<b>Graphiti</b>	🥚	●	🗨️	📄		
<b>Generative Model Transformer (GMT)</b>	🥚	●	🗨️	📄	●	
<b>Model-to-Model Transformation</b>	🥚	●	🗨️	📄	●	
<b>Model-to-Text Transformation (M2T)</b>	🥚	●	🗨️	📄	●	?
<b>Model Development Tools (MDT)</b>	🥚	●	🗨️	📄	●	?
<b>Presentation Modeling Framework (PMF)</b>	🥚	●	🗨️	📄	●	
<b>Textual Modeling Framework</b>	🥚		🗨️			

# UML

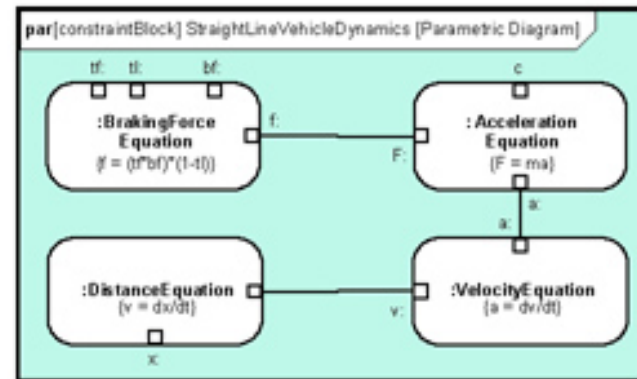
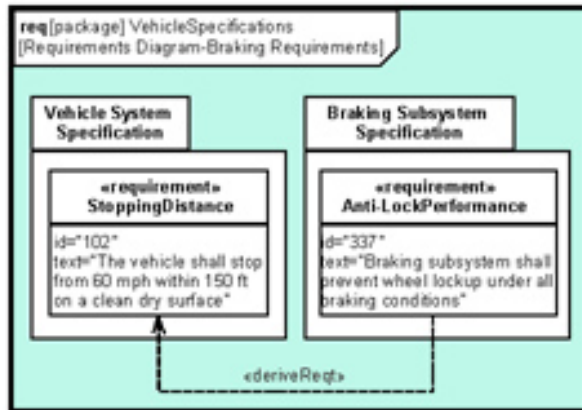
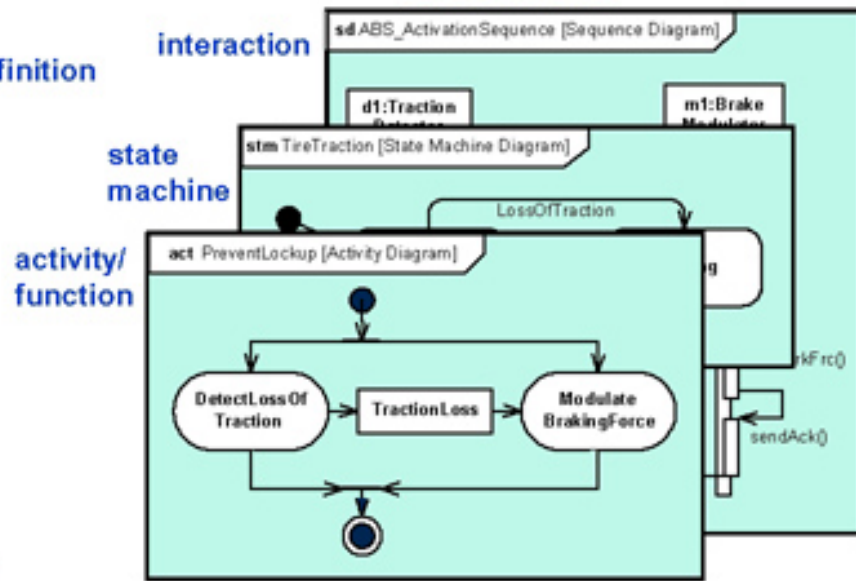


# SysML

## 1. Structure



## 2. Behavior

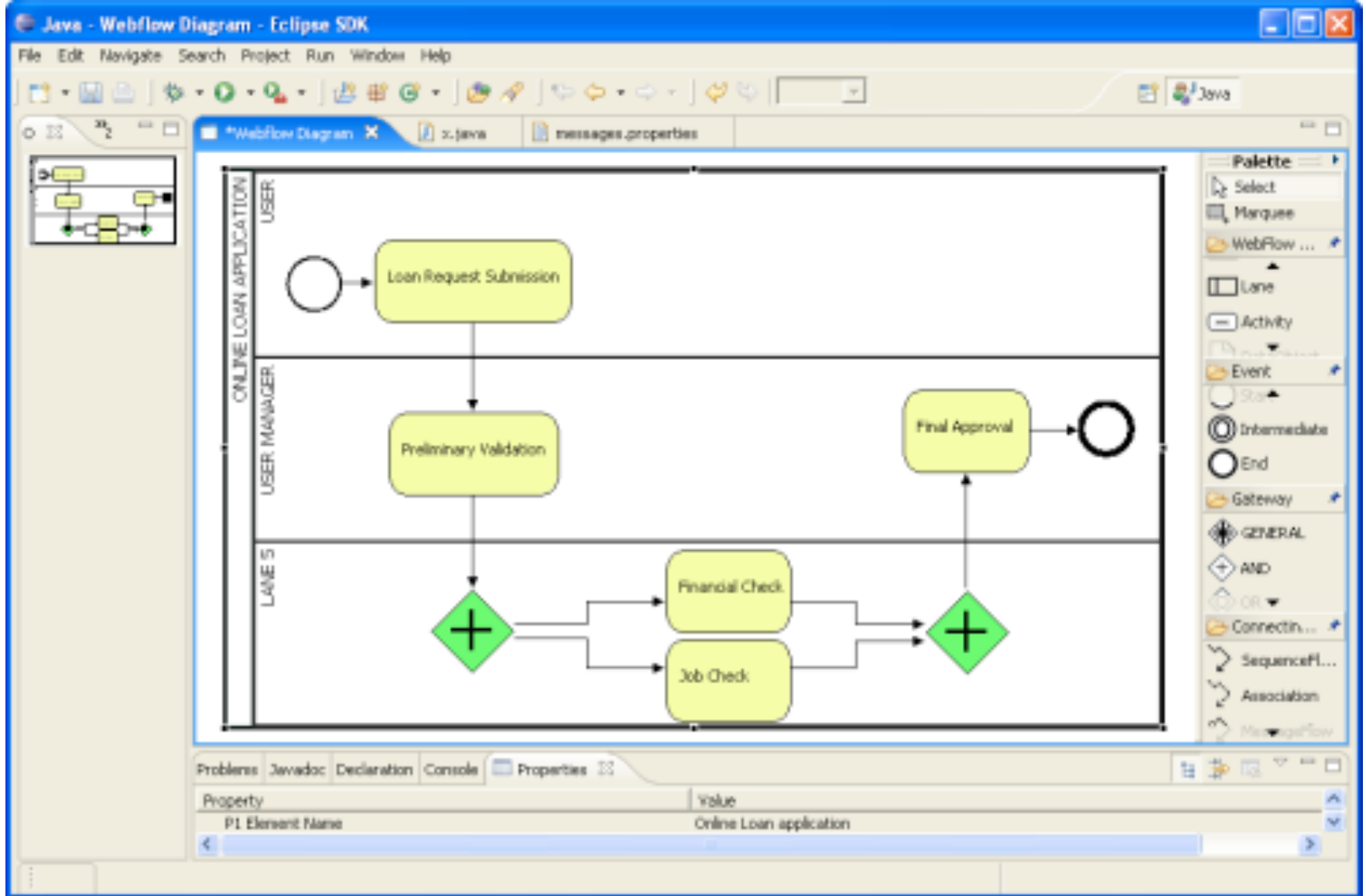


## 3. Requirements

## 4. Parametrics

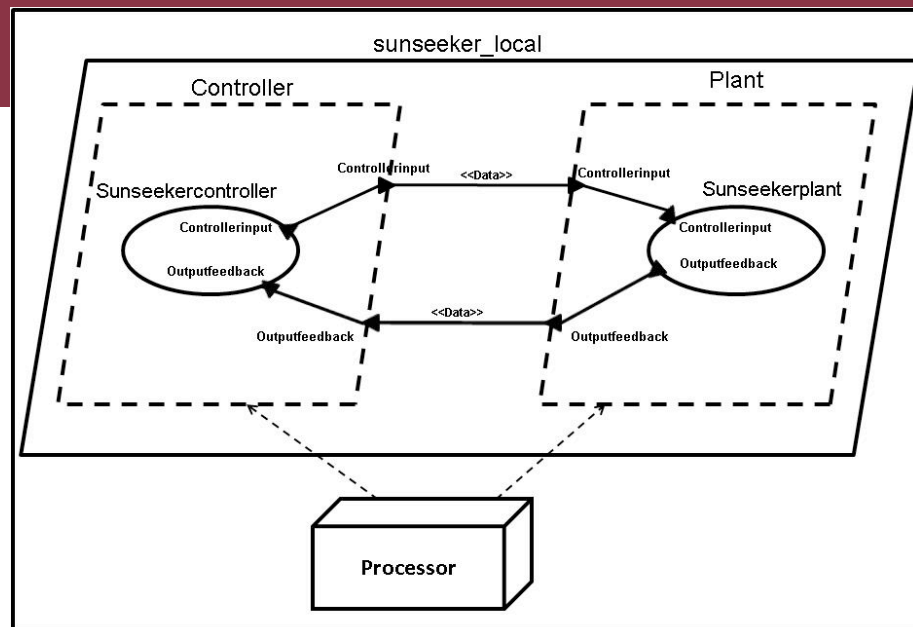
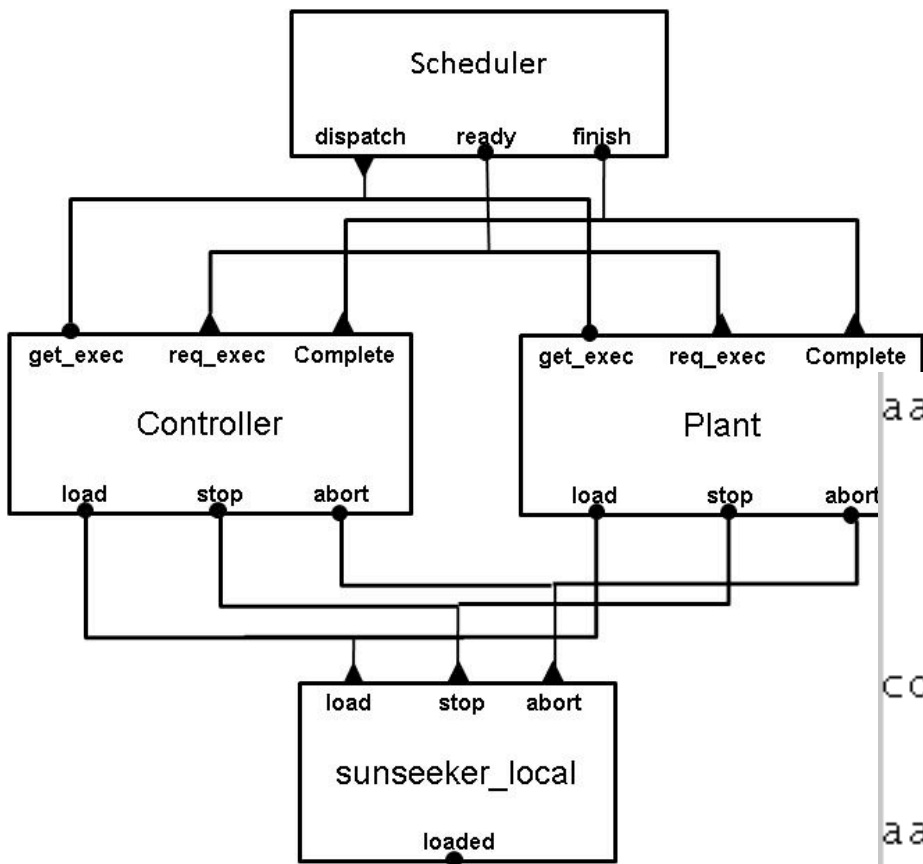
Note that the Package and Use Case diagrams are not shown in this example, but are respectively part of the structure and behavior pillars

# Workflow





# AADL



```

aadl_specification :
aadl_declaration
| cotre_directive
| aadl_specification aadl_declaration ;
| aadl_specification cotre_directive ;

cotre_directive :
IMPORT string_literal ';;';

aadl_declaration :
component_classifier
| system_instance
| property_set ;
    
```

# Eclipse: Áttekintés

# Eclipse: Nyílt fejlesztőrendszer

- Integrált fejlesztői környezet (IDE)
  - Több nyelv (Java, C/C++, PHP, ...)
  - Grafikus szerkesztők (pl. UML editor)
- Erős ipari háttér (IBM, Oracle, ...)
- Nyílt forráskód
- Átgondolt architektúra
  - Moduláris felépítés
  - Tervezési minták (Design patterns) alkalmazása
- Fejlesztés nyelve: Java

# Eclipse: Fejlesztőkörnyezet

The screenshot displays the Eclipse IDE interface. The main editor shows the file `BankAccountTests.java` with the following code:

```
package org.eclipse.samples.banking;

import java.math.BigDecimal;

import org.junit.Test;
import static org.junit.Assert.*;

public class BankAccountTests {

    @Test
    public void testDeposit() throws Exception {
        BankAccount account = new BankAccount();
        account.deposit(new BigDecimal(1000));

        assertEquals("...", account.getBalance(), new BigDecimal(1000));
    }
}
```

A context menu is open over the `account.deposit` call, offering options: "Create method 'deposit(BigDecimal)' in type 'BankAccount'", "Add cast to 'account'", and "Rename in file (Ctrl+2 R direct access)". A yellow tooltip on the right shows the stub for the `BankAccount` class:

```
import java.math.BigDecimal;

public class BankAccount {

    public void deposit(BigDecimal bigDecimal) {
        // TODO Auto-generated method stub
    }
}
```

The Outline view on the left shows the project structure, and the Problems view at the bottom lists three errors:

Description	Resource	Path	Location
Syntax error, insert ";" to complete	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 15
The method deposit(BigDecimal) is undefined for the type BankAccount	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 13
The method getBalance() is undefined for the type BankAccount	BankAccountTests.java	BankingProject/src/org/eclipse/samples/banking/BankAccountTests.java	line 15

The status bar at the bottom indicates: "The method deposit(BigDecimal) is undefined for the type BankAccount | Writable | Smart Insert | 13 : 18".

# Miért Eclipse?

- Erős ipari támogatottság
  - IBM, BEA, Oracle (részben)...
- Sok sikeres fejlesztés a világban (amiből lehet tanulni)
- Átgondolt architektúra
- Több programozási nyelv támogatása
- Több platform támogatása
- Modellezési támogatás

# Eclipse: Alkalmazásplatform

- Jól használható
  - Modellező projektek felhasználhatók
- Komoly ráncfelvarrás várható
- Részletek:
  - Eclipse alapú technológiák c. BSc választható tárgy

# Eclipse – Az első lépések

# Letöltés

- [www.eclipse.org/downloads](http://www.eclipse.org/downloads)
- Ingyenesen letölthető keretrendszer
  - Legtöbb platformra elérhető
- Legfrissebb verzió
  - Eclipse 3.6.0 (hamarosan lesz 3.6.1)



# Letölthető csomagok

- Többféle változat letölthető
  - Eclipse IDE for Java EE Developers
  - Eclipse IDE for Java Developers
  - Eclipse for PHP Developers
  - Eclipse IDE for C/C++ Developers
  - Eclipse for RCP/Plug-in Developers
  - **Eclipse Modeling Tools**
  - ...
- Ha nem megfelelőt választottunk, utólag “javítható” bővítmények telepítésével

# Java környezet

- Java SDK szükséges
  - <http://java.sun.com/javase/downloads/index.jsp>
  - Ajánlott verzió: Java 6
- Keretrendszer
  - Tömörített állomány
  - Bárhova kicsomagolható és futtatható

# Workspace

- Első indításkor workspace helye:
  - Egy könyvtár a merevlemezen
  - A felhasználói adatokat tartalmazza
    - Projektek
    - Mappák
    - Fájlok
    - Beállítások

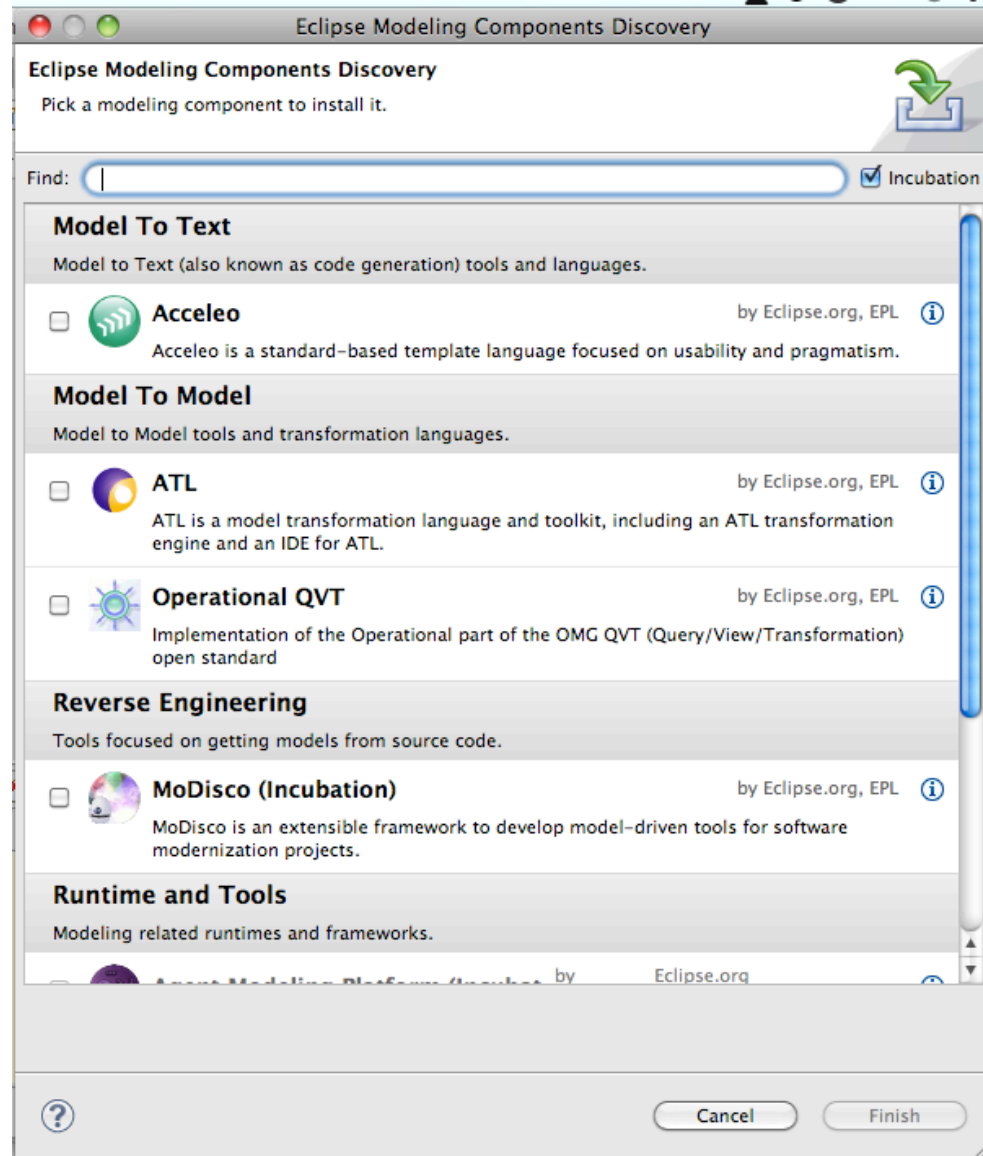
# Első házi feladat

- Eclipse 3.6 telepítése
  - Segíteni fog a félév során 😊

# Eclipse Modeling Tools

- Telepített komponensek
  - Java Developer Tools (JDT)
    - Java fejlesztői eszközök
  - Plug-in Developer Tools (PDT)
    - Támogatás Eclipse bővítmények készítéséhez
  - Eclipse Modeling Platform (EMF)
    - Modellező keretrendszer
  - Mylyn, CVS
    - Csoportmunkatámogatás
  - Help
    - Súgó

# Eclipse Modeling Tools



# Grafikus felület

# Workbench

- Általános felületdefiníció
  - Felhasználó könnyen testre szabhatja
- Elemek
  - Ablak
    - Lap
      - Perspektíva
      - Nézetek
      - Szerkesztők



# Workbench felépítése

Workbench window

The screenshot displays the Eclipse IDE Workbench window. The main editor area shows the code for `ServiceInterface.java`, which defines a `ServiceInterface` with methods `serviceCreated()` and `serviceRemoved()`. The Package Explorer on the left shows the project structure, including the `src` folder and the `ServiceInterface.java` file. The Outline view on the right shows the `ServiceInterface` class and its methods. The Problems view at the bottom shows 0 items. The status bar at the bottom indicates the file is writable and has smart insert enabled.

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

Description	Resource	Path	Location	Type
0 items				

# Workbench felépítése

The screenshot displays the Eclipse IDE interface. The main toolbar is highlighted with a blue box, and a callout bubble points to it with the text "Main toolbar". The toolbar contains various icons for file operations, editing, and development. The main editor area shows a Java file named "ServiceInterface.java" with the following code:

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

The Package Explorer on the left shows the project structure, including the "src" folder and the "ServiceInterface.java" file. The Outline view on the right shows the "ServiceInterface" interface with its methods. The Problems view at the bottom shows 0 items.

# 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 source code for `ServiceInterface.java`, which defines a `ServiceInterface` with two methods: `serviceCreated()` and `serviceRemoved()`. The Outline view on the right shows the `ServiceInterface` and its methods. A red speech bubble with the text "Workbench Site" points to the Package Explorer.

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

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

Description	Resource	Path	Location	Type

# Workbench Site

- Egy ablak egy lapot tartalmaz
- Lap
  - Fő tartalom
    - Szerkesztők
    - Nézetek
  - Elrendezés perspektíva alapján

# 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

# Perspektívák

- Nézetek funkcionális csoportja
- Elrendezést is rögzít
- Általában feladat-centrikus összeállítás
- Példák
  - Java Development
  - Plug-in Development
  - Debug
  - Team Synchronizing

# Workbench felépítése

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with the file `ServiceInterface.java` selected. The main editor window shows the content of `ServiceInterface.java`, which is enclosed in a blue rectangular box. Two callout boxes with a dark red background and white text point to the editor area. The left callout box contains the text "Szerkesztők (0..N)", and the right callout box contains "Megnyitott fájlok". The bottom of the IDE shows the Problems, Javadoc, and Declaration views, which are currently empty.

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

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

Szerkesztők (0..N)

Megnyitott fájlok

Description	Resource	Path	Location	Type
0 items				

Writable Smart Insert 1 : 1

# Szerkesztők

- Szerkesztési feladatokra
  - Tipikus: fájl szerkesztő
    - Egy vagy több fájl
  - DE: bármi fölé lehet szerkesztő
    - Pl. adatbázistartalom
    - Pl. verzió összehasonlítás (compare editor)
- Lehet
  - Szöveges (pl. Java editor)
  - Űrlap-alapú
  - Grafikus (pl. UML editor)
  - Összetett (több lap, laponként különböző)



# 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` containing several Java files, including `ServiceInterface.java`. The main editor window shows the source code of `ServiceInterface.java`, which defines a public interface with two methods: `serviceCreated()` and `serviceRemoved()`. The Outline view on the right shows the hierarchy of the interface. The Problems view at the bottom is empty, indicating no errors or warnings.

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

public interface ServiceInterface {

    public void serviceCreated();

    public void serviceRemoved();

}
```

Nézetek (0..N) csoportokban

Description	Resource	Path	Location	Type
0 items				

# Workbench felépítése

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project structure with the following hierarchy:

- hu.optxware.eclipsecourse.rcpdemo.bl.simple
- hu.optxware.eclipsecourse.rcpdemo.gui
  - JRE System Library [JavaSE-1.6]
  - Plug-in Dependencies
  - src
    - hu.optxware.eclipsecourse.rcpdemo.gui
      - Activator.java
      - Application.java
      - ApplicationActionBarAdvisor.java
      - ApplicationWorkbenchAdvisor.java
      - ApplicationWorkbenchWindowAdvisor.java
      - BookListView.java
      - BookListViewContentProvider.java
      - Perspective.java
      - ServiceInterface.java
      - ServiceManager.java
    - hu.optxware.eclipsecourse.rcpdemo.gui.comr
      - ExitHandler.java
  - icons
  - META-INF
  - OSGI-INF
  - build.properties
  - plugin.xml
  - splash.bmp
- hu.optxware.eclipsecourse.rcpdemo.model.simple

The editor window shows the content of ServiceInterface.java:

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

The Outline view on the right shows the following structure:

- hu.optxware.eclipsecourse
  - ServiceInterface
    - serviceCreated() : void
    - serviceRemoved() : void

A callout box with the text "Projekt hierarchia" points to the ServiceInterface.java file in the Package Explorer.

The Problems view at the bottom shows a table with the following columns: Description, Resource, Path, Location, and Type.

Description	Resource	Path	Location	Type
items				

Projekt  
hierarchia

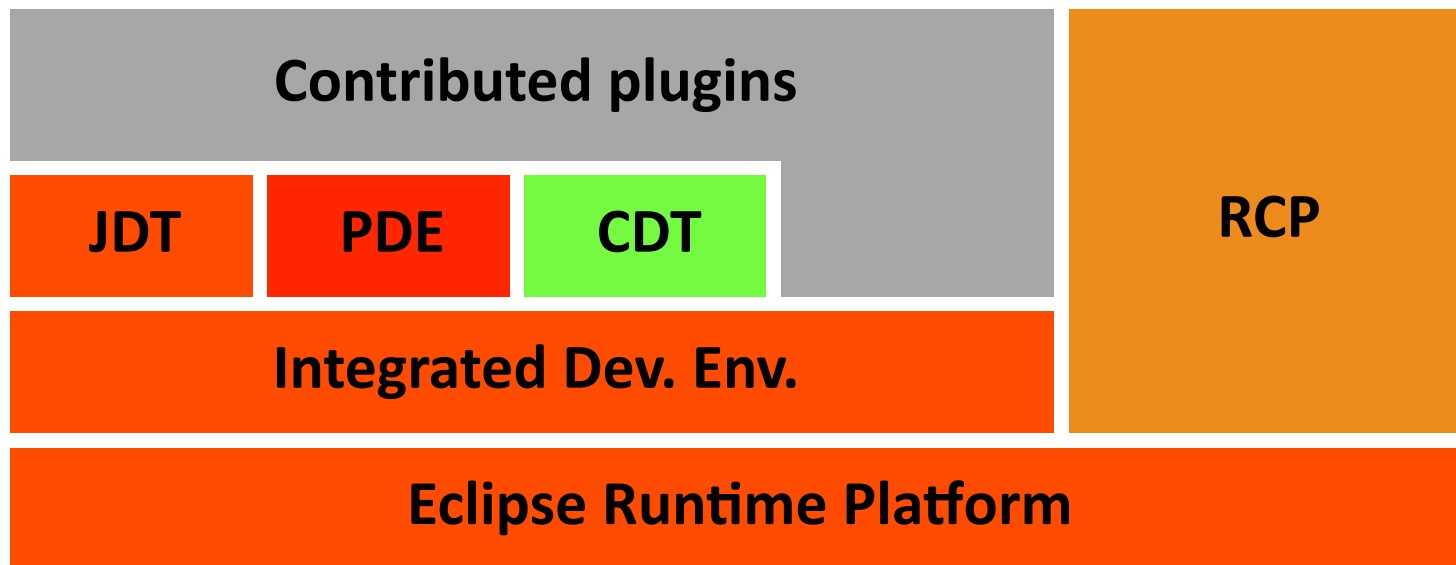
# Nézetek

## ■ Kiegészítő funkciók

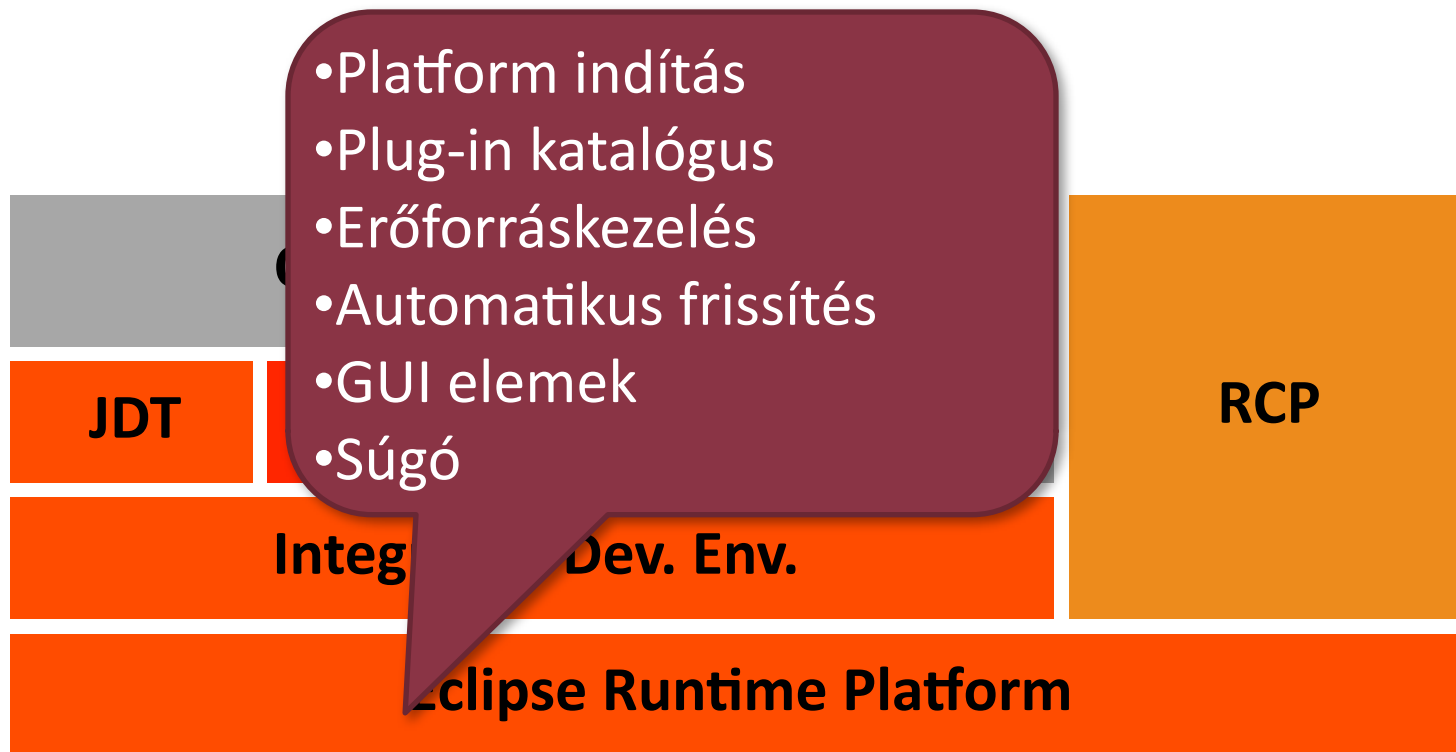
- Statikus információmegjelenítés
- Lehet globális
  - Erőforrásfa (Project Explorer)
  - Üzenetsor (Error Log)
- Vagy függhet a kijelölt elemektől
  - Tulajdonságok nézet (Properties – kijelölt elem tulajdonságai)
  - Áttekintő (Outline – aktuális editor tartalmának szerkezete)

# Eclipse architektúra

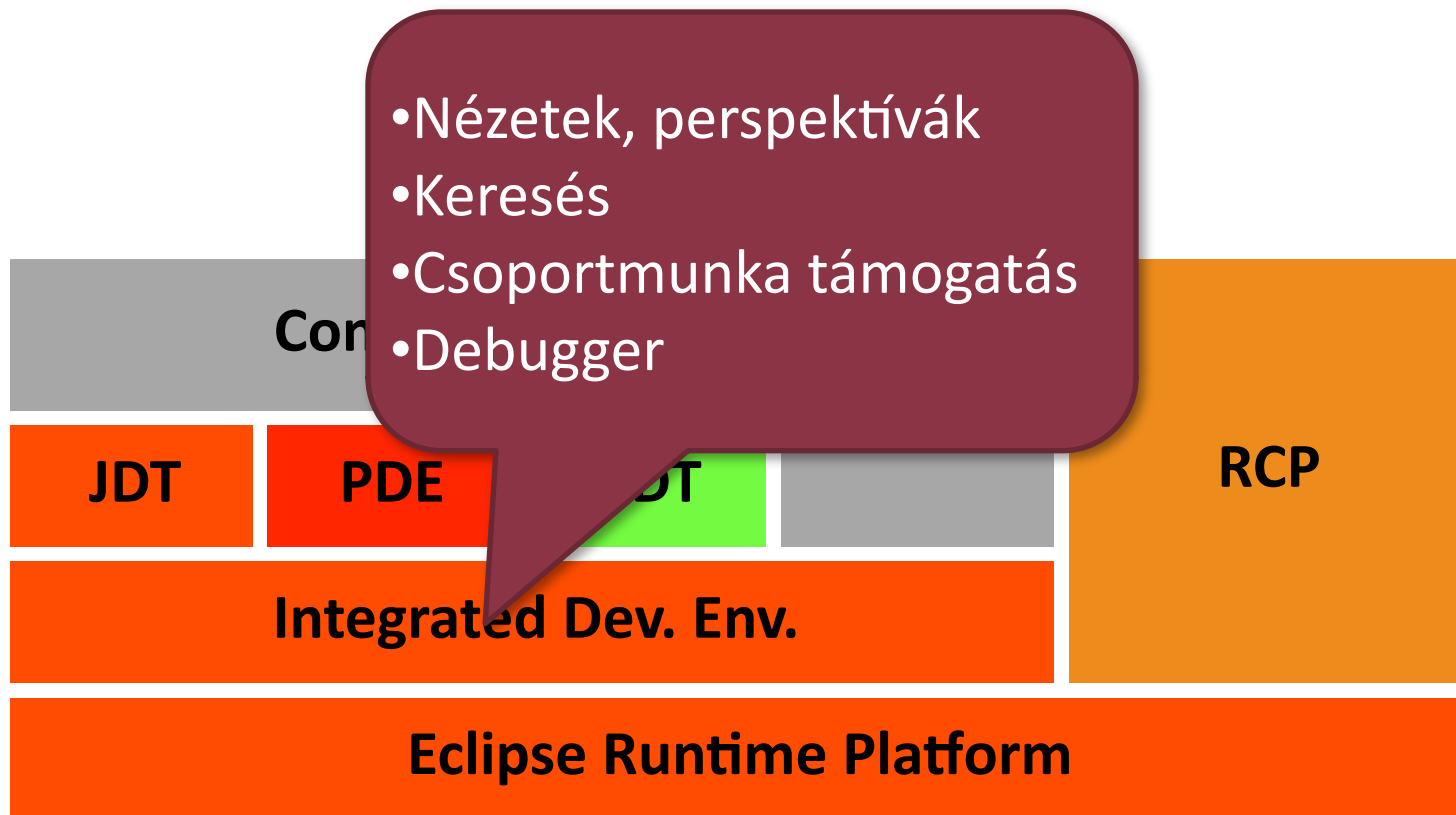
# Architektúra



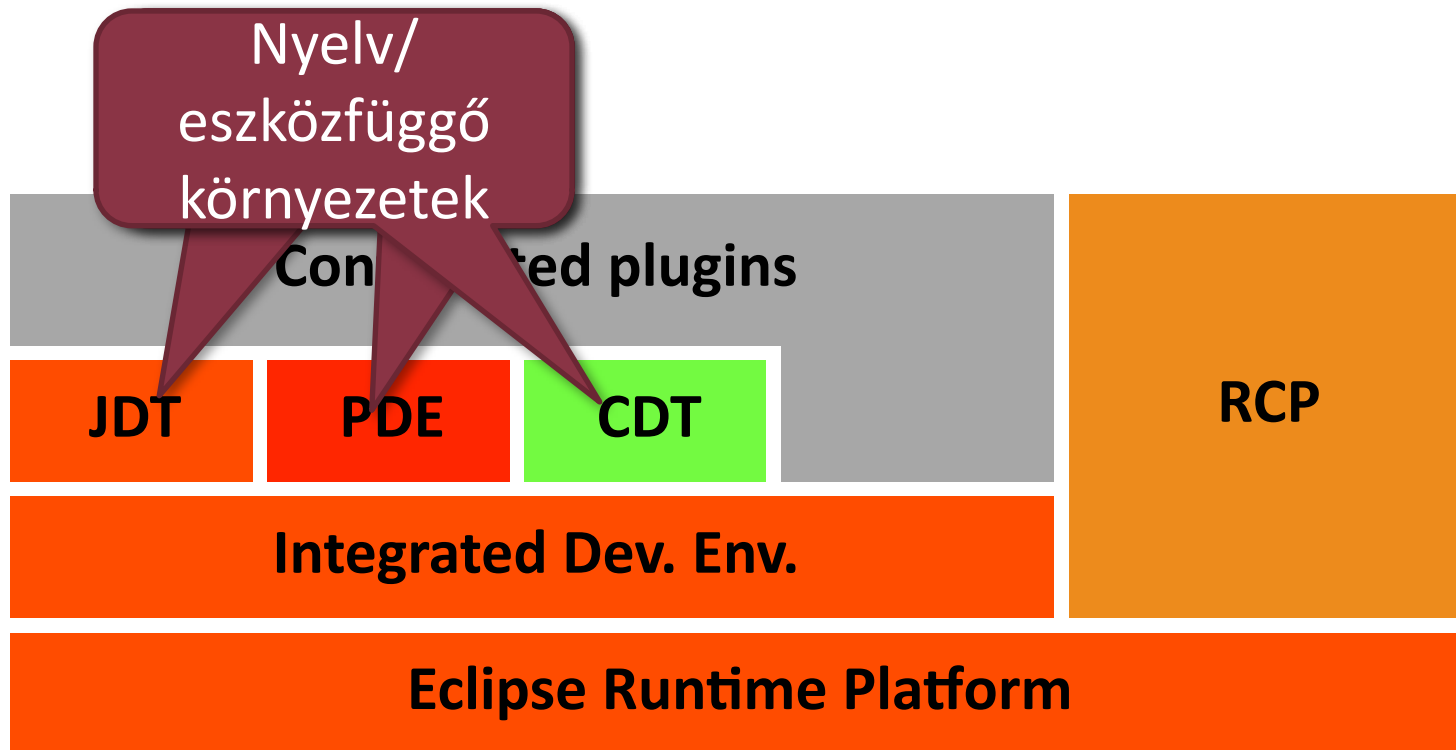
# Architektúra



# Architektúra

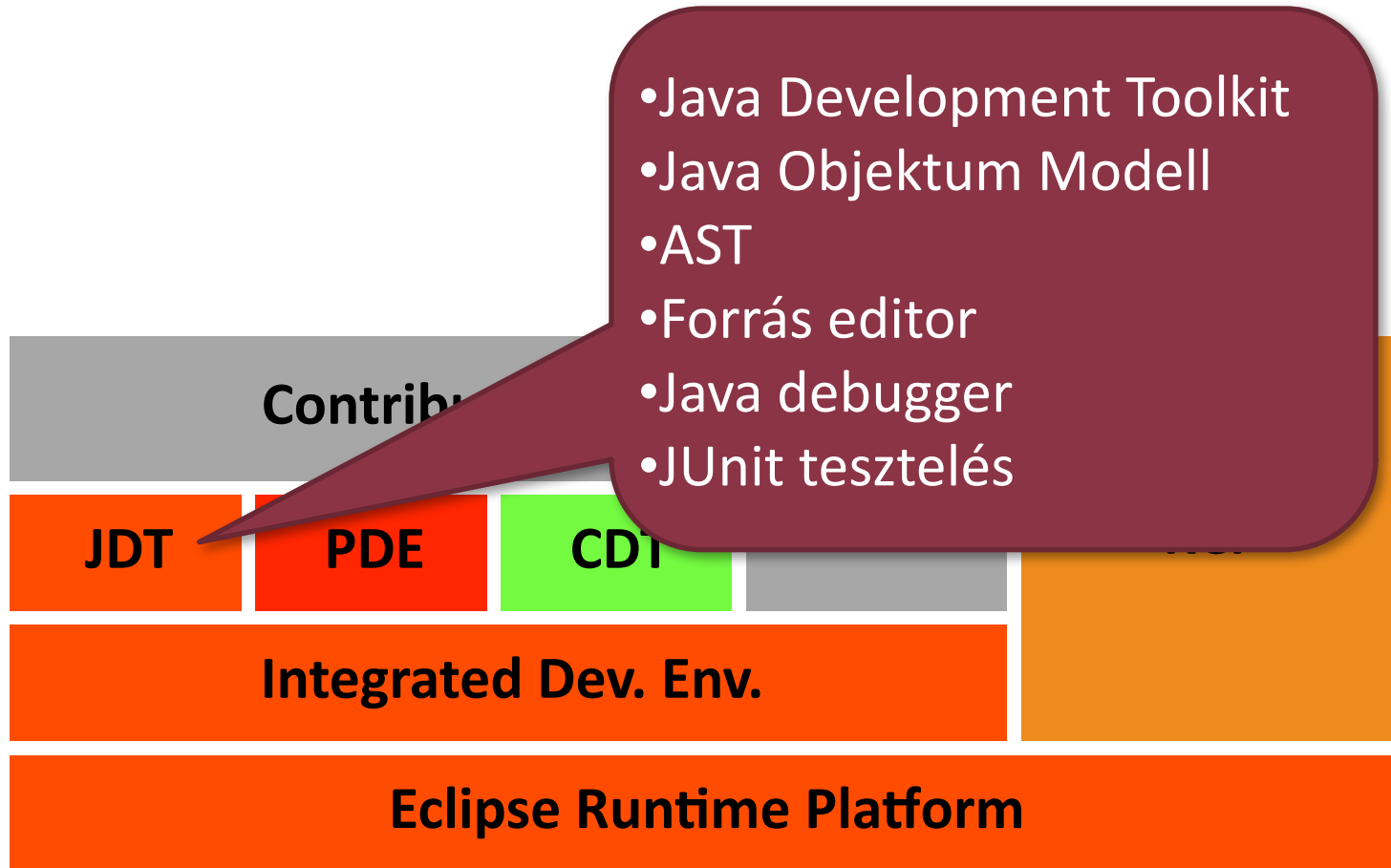


# Architektúra



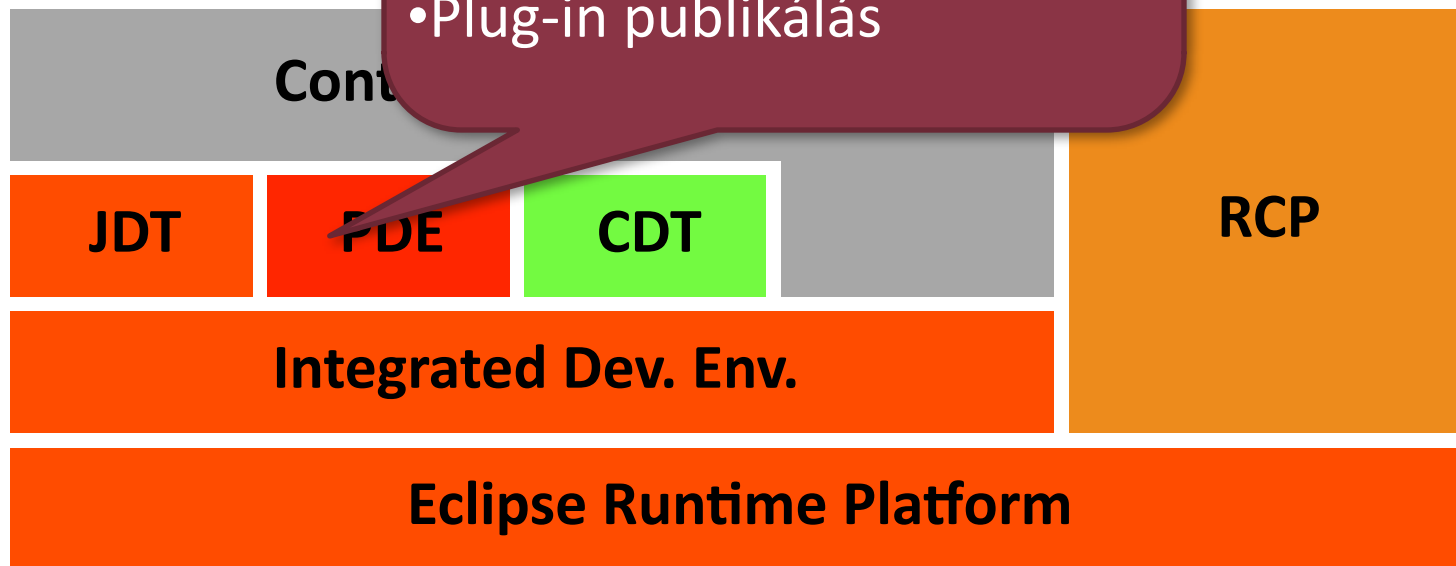


# Architektúra



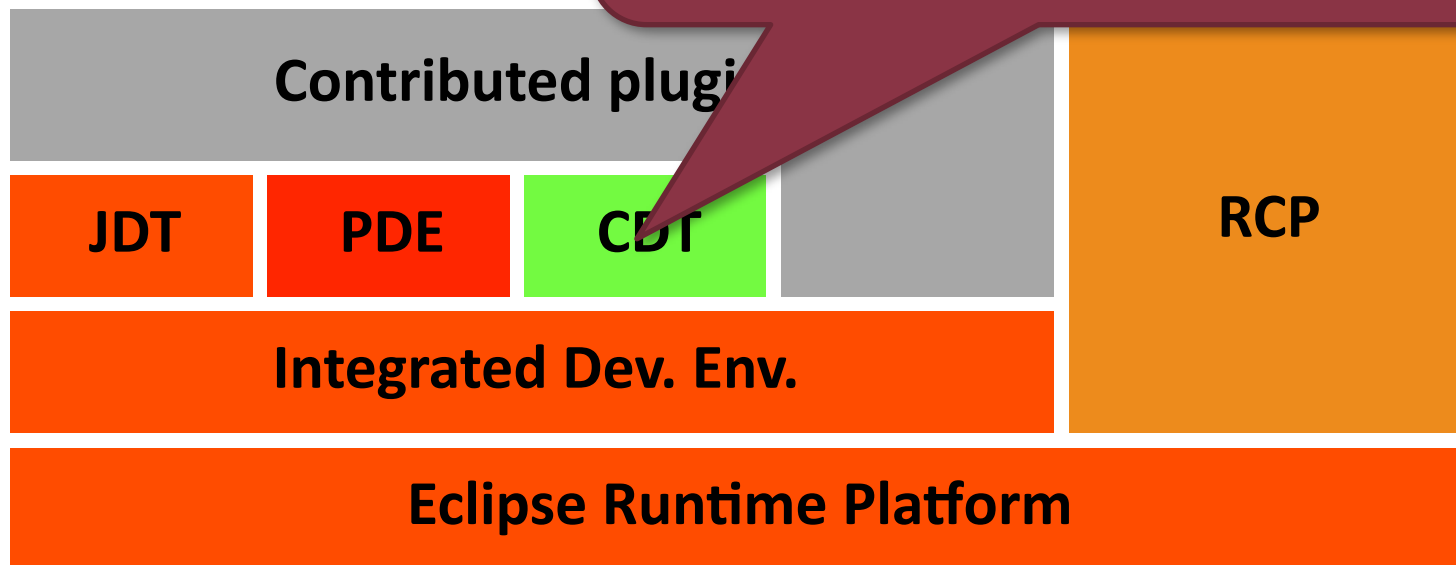
# Architektúra

- Saját plug-inek fejlesztése
- Futtatás támogatás
- Plug-in definíció készítés
- Plug-in publikálás



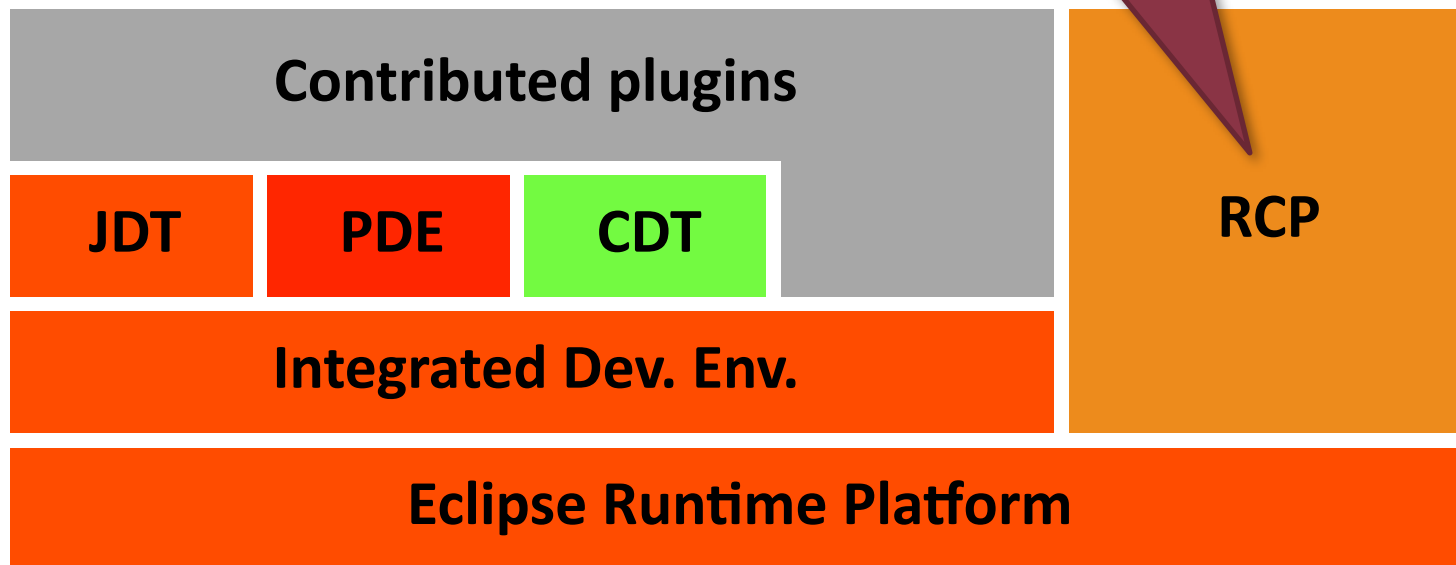
# Architektúra

- C/C++ fejlesztés
- Külső fordító és debug támogatás
- Make támogatás



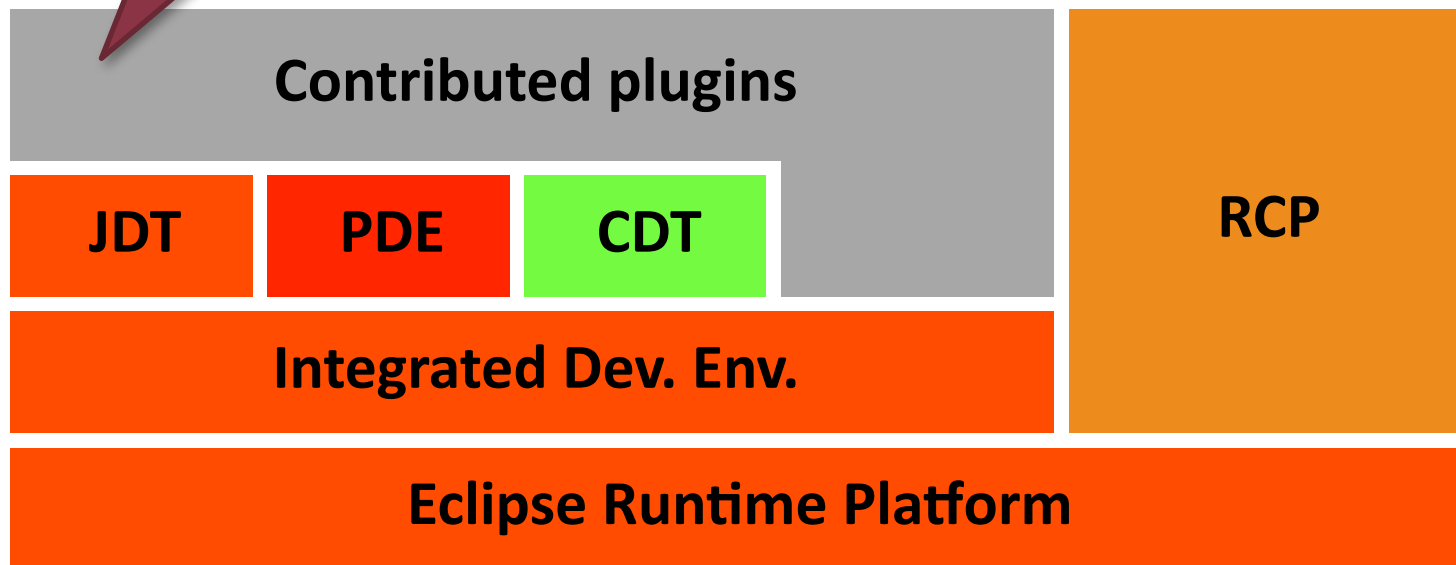
# Architektúra

- Alkalmazások
- Runtime Platform felett



# Architektúra

- Saját plug-inek
- Szabad a pálya



# Összefoglalás

- Eclipse fejlesztés
  - Plug-inek készítése
  - Java nyelven
  - Cél:
    - eszközök összehangolása