

Modulo I- Introdução a Plataforma Eclipse

Prof. Ismael H F Santos

Considerações Gerais

- **Objetivo:** *Discutir os principais conceitos relacionados a Plataforma de Desenvolvimento Eclipse e suas facilidades para a Programação JAVA*
- **A quem se destina :** *Alunos e Profissionais que desejem aprofundar seus conhecimentos sobre Orientação a Objetos e sua aplicação em uma linguagem totalmente orientada a Objetos.*

Ementa

■ Introdução a Plataforma Eclipse

- Introdução
- Arquitetura da Plataforma
- Componentes da Plataforma
- JDT
- PDE

Bibliografia

- *Eclipse project briefing materials*
 - <http://www.eclipse.org/eclipse/presentation/eclipse-slides.html>
- *Eclipse User-Guide – projeto Hotwork*
 - <http://hotwork.sourceforge.net/hotwork/manual/eclipse/eclipse-user-guide.html>

POO-Java

Introdução ao
Eclipse



“The Eclipse Platform is an IDE for anything, and for nothing in particular.”

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

5

O que é o Eclipse

- O Eclipse é uma plataforma para a integração de ferramentas de desenvolvimento.
- Iniciativa *Open Source*.
- Arquitetura extensível baseada no uso e desenvolvimento de *plug-ins*.
- Público Alvo:
 - Desenvolvedores de Aplicação
 - Desenvolvedores de Ferramentas de Desenvolvimento
 - Pesquisadores

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

6

Objetivos do Projeto Eclipse

- Prover uma plataforma aberta para ferramentas de desenvolvimento de aplicações.
 - Independência de sistema operacional
- Facilitar integração de ferramentas, mesmo que oriundas de fornecedores diferentes
 - Adicionar novas ferramentas a produtos já instalados
- Neutralidade de linguagens
 - Não possui restrição quanto ao tipo de conteúdo
 - HTML, Java, C, JSP, EJB, XML, GIF, ...
- Suporte a funcionamento com e sem GUI
- Portabilidade
- Atrair desenvolvedores de ferramentas

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

7

Histórico do Eclipse

- Origens
 - A linha Visual Age de IDEs da IBM estava precisando de um reposicionamento comercial e uma refatoração da sua arquitetura
 - Esta refatoração deu origem ao Eclipse como software livre e sua arquitetura de plug-ins
 - O Eclipse se tornou a base da nova linha de IDEs da IBM, o WebSphere Studio
 - Também é a base da nova linha de ferramentas da Rational (XDE), encampada pela IBM

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

8

Histórico do Eclipse

■ Origens (cont.)

- Infra-estrutura para o desenvolvimento de ferramentas de desenvolvimento para qualquer plataforma e linguagem
 - Java é apenas a “linguagem de sistema” do Eclipse, assim como C é a “linguagens de sistema” do VB
 - Em vez do Swing, é utilizada a biblioteca SWT para componentes visuais
 - Foco no apoio à codificação (automação, refatoração, extreme programming) em vez da construção visual

Histórico do Eclipse

■ Independência da IBM

- A Eclipse Foundation se tornou este ano uma entidade jurídica independente da IBM, e seu conselho diretor hoje é presidida por um executivo da Oracle
- Este fato estimulou outras grandes empresas como a SAP a se tornarem parte da fundação
- Amplo mercado de plug-ins de terceiros, como o SOFIA, Genuitec (MyEclipse), TruStudio (Python), W4T, etc

Histórico do Eclipse

1999

Abril - Início do Eclipse dentro da OTI / IBM responsáveis pelos produtos IBM VisualAge (Smalltalk, Java, J2ME).

2000

Junho - Lançamento do Eclipse Tech Preview

2001

Março - Início do <http://www.eclipsecorner.org/>

Junho - Eclipse 0.9

Outubro - Eclipse 1.0

Novembro - IBM doa base de código do Eclipse, Eclipse Platform, JDT e do PDE
- Lançamento do <http://www.eclipse.org/>

2002

Junho - Eclipse 2.0

Setembro - Eclipse 2.0.1

Novembro - Eclipse 2.0.2

2003

Março - Eclipse 2.1

Junho - Eclipse 2.1.1

Novembro - 2.1.2

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

11

Quem Contribui?

FUJITSU THE POSSIBILITIES ARE INFINITE



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

12

POO-Java

Arquitetura da
Plataforma

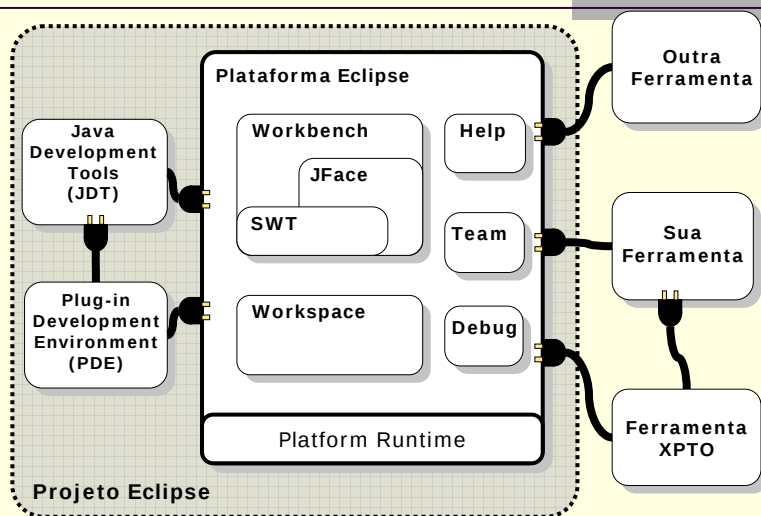


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

13

Visão Geral da Arquitetura

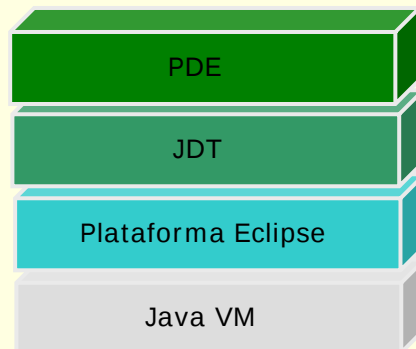


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

14

Representação da Plataforma em Camadas



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

15

Arquitetura Eclipse

■ Plug-in

- Menor bloco de construção do Eclipse. Unidade mínima de funcionalidade.
- Exemplos: **editor HTML**, **Ação para criar arquivos zip**
- Uma aplicação complexa pode ser baseada em vários plugins; a maior parte do próprio Eclipse é implementada por plugins

■ Pontos de Extensão

- Entidades definidas para o agrupamento de contribuições. Um plugin declara pontos de extensão e extensões para pontos de extensão de outros plugins
- Exemplo: ponto de extensão para preferências referentes a interface com usuários de um editor

■ Extensão

- Uma contribuição
- Exemplo: **preferência específicas de um editor HTML**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

16

Arquitetura Eclipse

- Cada plug-in:
 - **Contribui** com 1 ou mais pontos de extensão
 - Opcionalmente, **declara** novos pontos de extensão
 - **Depende** de outros plug-ins
 - **Contém** bibliotecas Java e outros arquivos
 - **Pertence** ao seu próprio diretório de plug-ins
 - Detalhes sobre o plug-in são escritos no arquivo plugin.xml no diretório raiz do plug-in.
- Runtime da Eclipse Plataform
 - Estrutura micro-kernel. Toda a funcionalidade é suprida por plug-ins.
 - Ao iniciar Runtime descobre todos os plug-ins registrado

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

17

Inventários - Manifest

- Plugins explicitam seus pontos de extensão, extensões e relações com outros plugins através de um arquivo de inventário (manifest) no formato XML
- Com base nos arquivos de inventário, o sistema cria uma listagem dos plugins disponíveis e suas interconexões durante a inicialização
- Os plugins só são ativados quando efetivamente solicitados

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

18

Plug-in Manifest

plugin.xml

```
<plugin
  id = "com.example.tool"
  name = "Example Plug-in Tool"
  class = "com.example.tool.ToolPlugin">
<requires>
  <import plugin = "org.eclipse.core.resources"/>
  <import plugin = "org.eclipse.ui"/>
</requires>
<runtime>
  <library name = "tool.jar"/>
</runtime>
<extension
  point = "org.eclipse.ui.preferencepages">
  <page id = "com.example.tool.preferences"
    icon = "icons/knob.gif"
    title = "Tool Knobs"
    class = "com.example.tool.ToolPreferenceWizard"/>
</extension>
<extension-point
  name = "Frob Providers"
  id = "com.example.tool.frobProvider"/>
</plugin>
```

Plug-in identification

Other plug-ins needed

Location of plug-in's code

Declare contribution this plug-in makes

Declare new extension point open to contributions from other plug-ins

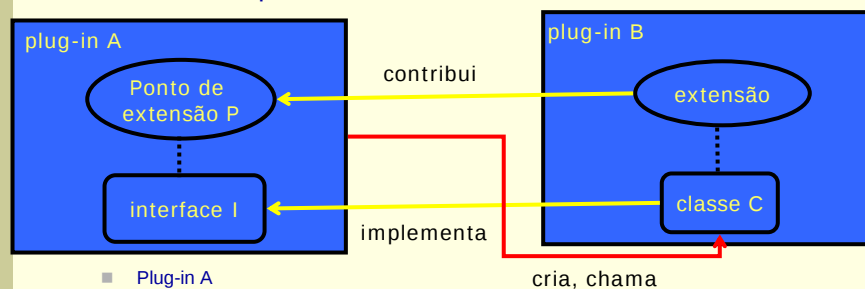
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

19

Arquitetura de plug-ins Eclipse

■ Cenário Típico



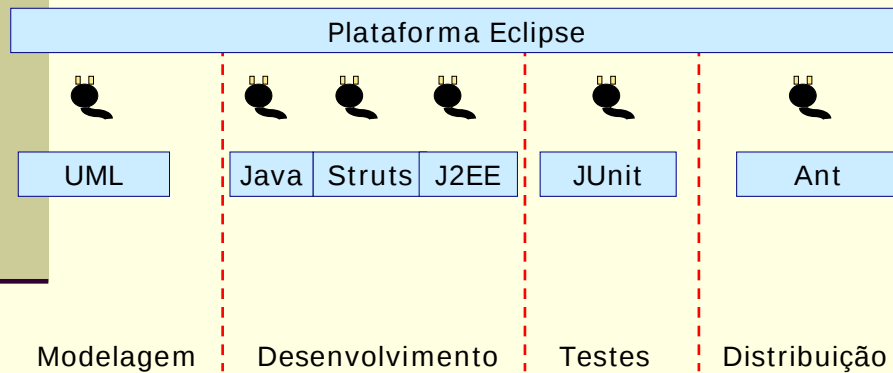
- Plug-in A
 - Declara o ponto de extensão P
 - Declara a interface I de acordo com P
- Plug-in B
 - Implementa a interface I com a classe C
 - Contribui com a classe C para o ponto de extensão P
- Plug-in A instancia C e chama os métodos de I

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

20

Cenário de Utilização de Plug-ins

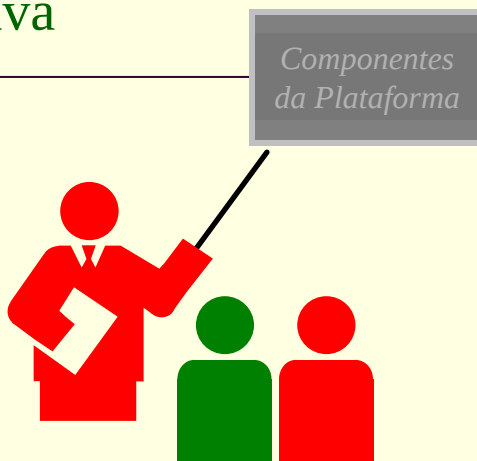


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

21

POO-Java



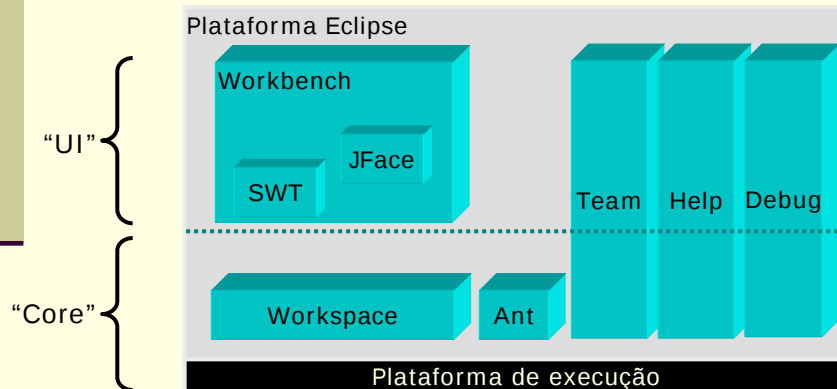
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

22

Componentes da Plataforma

- A Plataforma é a base comum para todos os componentes.
- Consiste em vários componentes chaves:



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

23

Workspace

- Ferramentas lêem, criam, modificam e removem recursos da área de trabalho
- Recursos
 - Projetos, pastas e arquivos
- Possibilidade de trabalhar com vários projetos em paralelo.
 - Organização:
 - Por usuário
 - Por diretórios no sistema de arquivos

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

24

Workspace Component

- Tools operate on files in user's **workspace**

- Workspace holds 1 or more top-level **projects**
- Projects map to directories in file system
- Tree of **folders** and **files**
- {Files, Folders, Projects} termed **resources**
- Tools read, create, modify, and delete resources in workspace
- Plug-ins access via workspace and resource APIs



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

25

Projetos no Eclipse

- Um projeto é um conjunto de arquivos em um diretório raiz
- Um projeto tem uma espécie (*nature*), à qual corresponde uma configuração do ambiente
- Projetos, arquivos e diretórios que os compõem são encarados como “recursos”
- Pode-se criar marcadores com listas de tarefas, pontos de parada de depuração etc.; plugins podem definir tipos de marcadores
- A forma preferida de acrescentar recursos de construção é via scripts Ant

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

26

Alterações de recursos

- Mudanças nos recursos que compõem um projeto são descritas por uma árvore de alterações de recursos (*resource deltas*)
- Alterações em lote geram uma única árvore
- Plugins podem ser informados de mudanças em algum recurso
 - coerência de operação e de interface com o usuário entre plugins
 - construção parcial (*incremental build*)

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

27

Workbench (Bancada)

- Interface do usuário da plataforma Eclipse
 - Fornece uma estrutura para a interação de ferramentas com os usuários
- Componentes do Workbench
 - Perspectivas
 - Visões
 - Editores
 - Oferece suporte ao trabalho em equipe
 - CVS
 - Padrão
 - Outros

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

28

Workbench

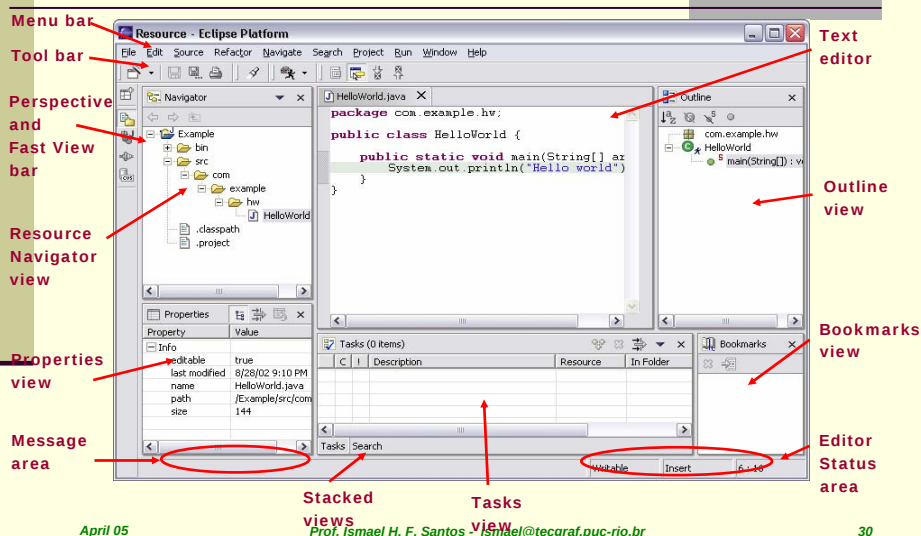
- Composta de visores e editores; diferentes perspectivas correspondem a diferentes organizações de visores e editores
- Editores acrescentam ações aos menus e barras de ferramentas da bancada
- Visores fornecem diversas informações ao usuário sobre os recursos que estão sendo editados
- visores, editores, perspectivas e ações podem ser acrescentados ao sistema por plugins

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

29

Workbench Terminology



SWT

- Interface independente de plataforma; implementação em Java + JNI
- Usa biblioteca padrão do sistema; quando isso não é possível, emulação
- Interface de programação portátil + interface de usuário consistente com o sistema
- Permite extensões não-portáteis, como ActiveX em Windows)
- Pode ser usada em outros projetos

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

31

JFace

- Conjunto de classes para tarefas comuns de GUI, como assistentes, preferências, diálogos etc.
- **Ações (*actions*)**: definição abstrata de um comando a ser incluído na interface sem definir onde (nome, ícone, dica etc.)
- **Visores (*viewers*)**: adaptadores que implementam o padrão observer para alguns elementos SWT, como listas, árvores e tabelas

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

32

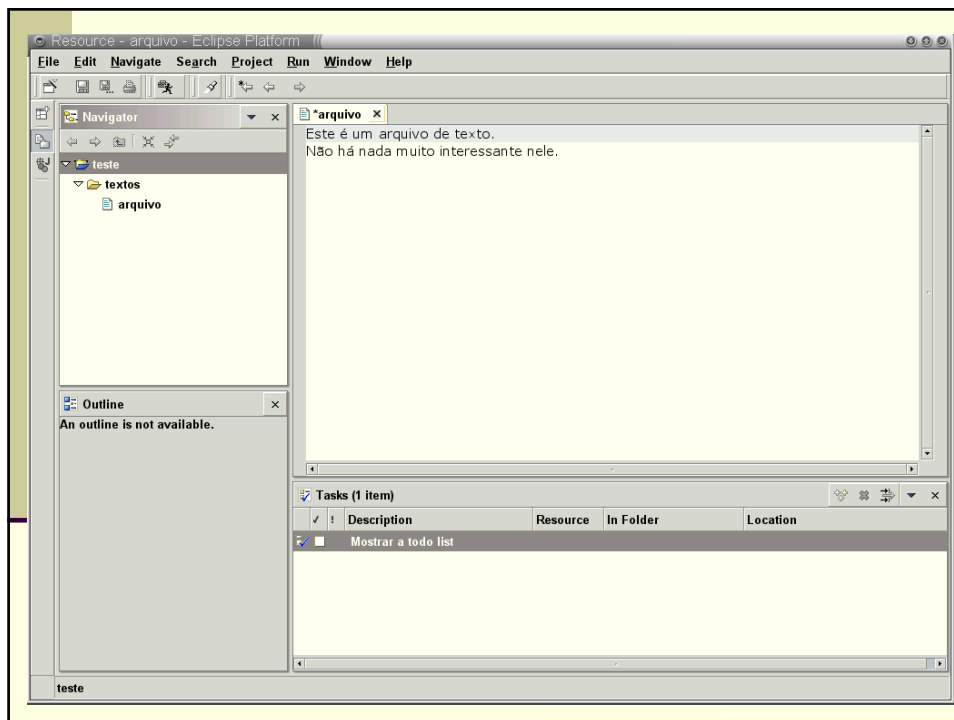
JFace APIs

- Image and font registries
- Dialog, preference, and wizard frameworks
- Structured viewers
 - Model-aware adapters for SWT tree, table, list widgets
- Text infrastructure
 - Document model for SWT styled text widget
 - Coloring, formatting, partitioning, completion
- Actions
 - Location-independent user commands
 - Contribute action to menu, tool bar, or button

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

33



Help

- Plugins podem acrescentar documentação ao sistema
- Documentos em HTML
- Estrutura das relações entre documentos em XML
- O plugin define onde os tópicos de ajuda devem ser inseridos na árvore de documentação
- Servidor HTTP embutido no Eclipse

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

35

Help Component

- Help is presented in a standard web browser



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

36

Debug Component

- **Launch configurations**
 - How to run a program (debug mode option)
- **Generic debug model**
 - Standard debug events: suspended, exit, ...
 - Standard debug actions: resume, terminate, step, ...
 - Breakpoints; Expressions and Source code locator
- **Generic debug UI**
 - Debug perspective
 - Debug views: stack frames, breakpoints, ...
- **Example: JDT supplies Java launcher and debugger**
 - Java debugger based on JPDA
- **Debug mechanisms available to other plug-ins**

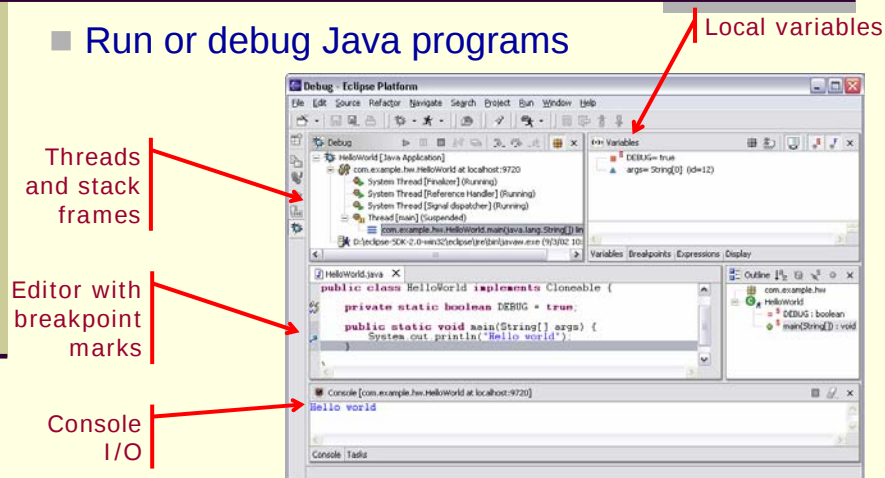
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

37

Eclipse Java Debugger

- **Run or debug Java programs**



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

38

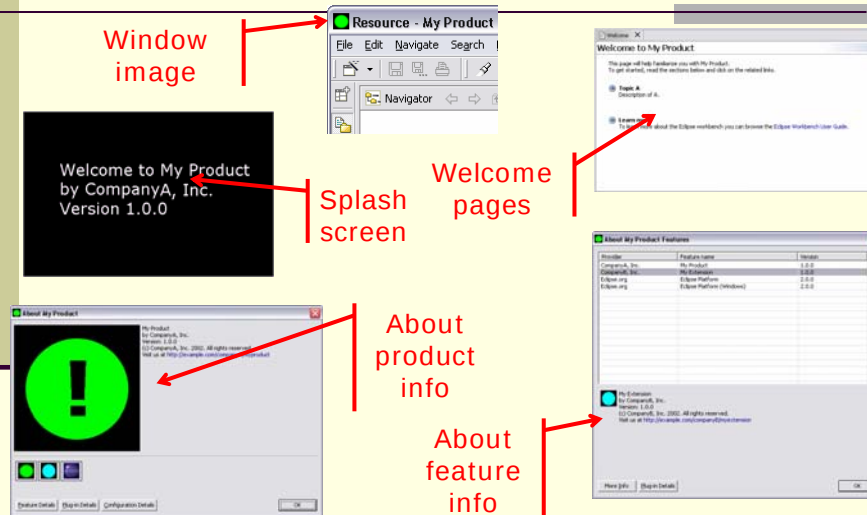
Eclipse Java Debugger

- **Run Java programs**
 - In separate target JVM (user selectable)
 - Console provides stdout, stdin, stderr
 - Scrapbook pages for executing Java code snippets
- **Debug Java programs**
 - Full source code debugging
 - Any JPDA-compliant JVM

Eclipse Java Debugger

- **Debugger features include**
 - Method and exception breakpoints
 - Conditional breakpoints; Watchpoints
 - Step over, into, return; run to line
 - Inspect and modify fields and local variables
 - Evaluate snippets in context of method
 - Hot swap (if target JVM supports)

Product Information



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

41

Product Information

- Primary feature controls product information
 - Splash screen
 - Window image
 - About product info
 - Initial welcome page
 - Default perspective
 - Preference default overrides
- All features can provide
 - Welcome page
 - About feature info

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

42

Eclipse Platform - Summary

- Eclipse Platform is the nucleus of IDE products
- Plug-ins, extension points, extensions
 - Open, extensible architecture
- Workspace, projects, files, folders
 - Common place to organize & store development artifacts
- Workbench, editors, views, perspectives
 - Common user presentation and UI paradigm
- Key building blocks and facilities
 - Help, team support, internationalization, ...

**Eclipse is a universal platform for
integrating development tools**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

43

POO-Java

*JDT – Java
Development
Tools*



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

44

JDT - Java Development Tools

- Estado da arte em ambientes de desenvolvimento Java.
- Construído sobre a Plataforma Eclipse.
 - Implementado como um conjunto de Plug-ins.
 - Utiliza as APIs e pontos de extensão da Plataforma.
- Incluído na distribuição do Eclipse.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

45

JDT - Pontos Fortes

- Perspectiva Java
 - Concentra as funcionalidades necessárias a um desenvolvedor java.
- Facilidades para escrever código:
 - Method Completion
 - Refactoring
 - Quick Fixes
 - Code Templates
 - Formatador de Código
 - Dentre outras facilidades...

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

46

JDT (*Java Development Tool*)

- Classes etc. apresentados em diretórios correspondentes a pacotes
- Navegação em termos de elementos específicos da linguagem, como pacotes, tipos, métodos, atributos etc.
- Editor com características otimizadas
- Visor de estrutura do arquivo
- Funções para refatoração
- Depurador integrado

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

47

Implementação

- Construtor parcial usa uma árvore de alterações de recursos juntamente com um grafo de dependências armazenado persistentemente para reconstruções otimizadas
- Sistema mantém uma árvore de elementos Java para navegação pelos editores e visores; essa árvore é construída e carregada em memória em partes, em função da necessidade

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

48

Implementação

- Interface de usuário e infra-estrutura separados
- Define a espécie “projeto Java”, o construtor parcial Java e marcadores para depuração
- Define a perspectiva Java, visores de pacotes e de hierarquia de tipos, assistentes para a criação de classes, pacotes, interfaces etc.
- Define preferências, como classpath, opções de formatação do editor Java etc.

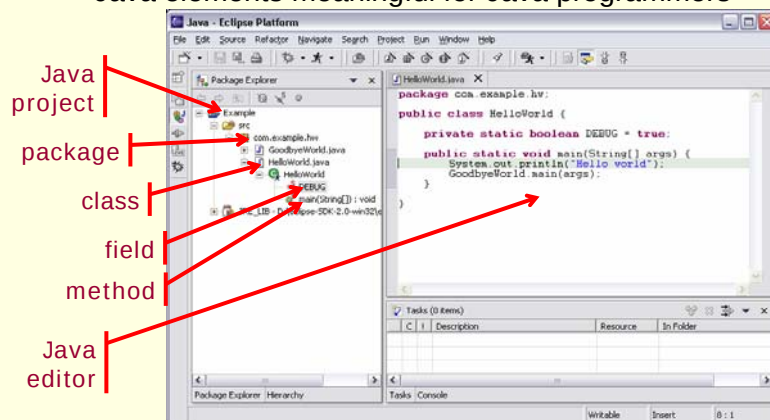
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

49

Java Perspective

- Java-centric view of files in Java projects
 - Java elements meaningful for Java programmers



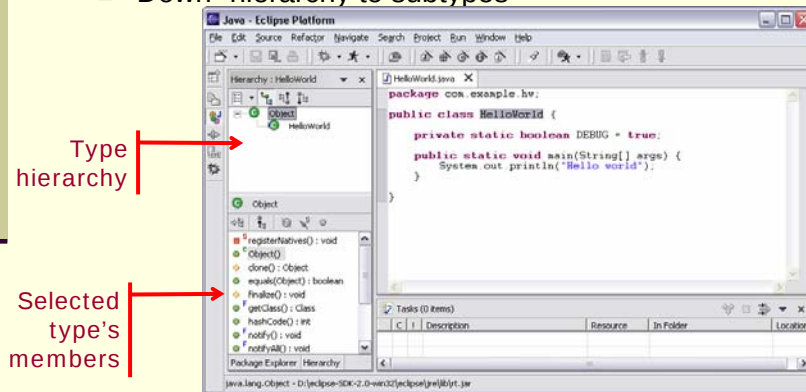
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

50

Java Perspective

- Browse type hierarchies
 - “Up” hierarchy to supertypes
 - “Down” hierarchy to subtypes



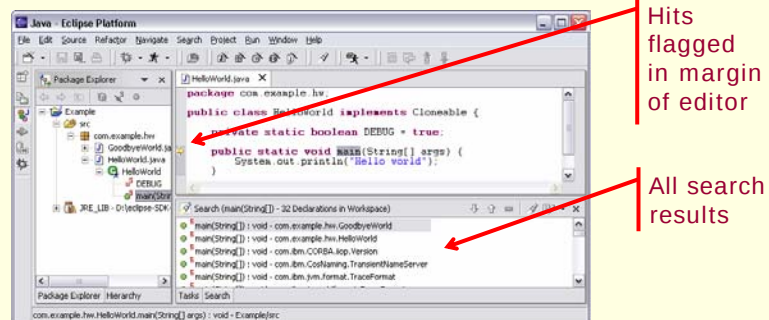
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

51

Java Perspective

- Search for Java elements
 - Declarations or references
 - Including libraries and other projects



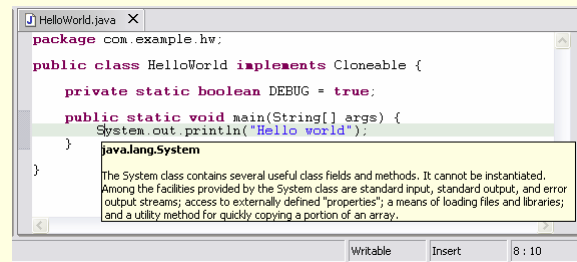
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

52

Java Editor

- Hovering over identifier shows Javadoc spec



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

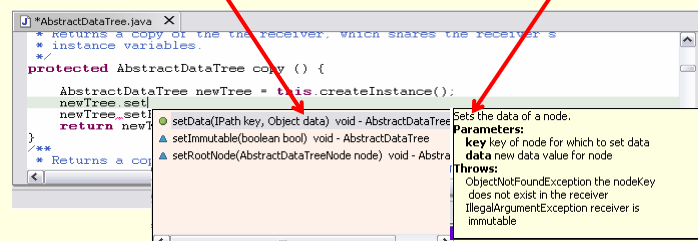
53

Java Editor

- Method completion in Java editor

List of plausible methods

Doc for method



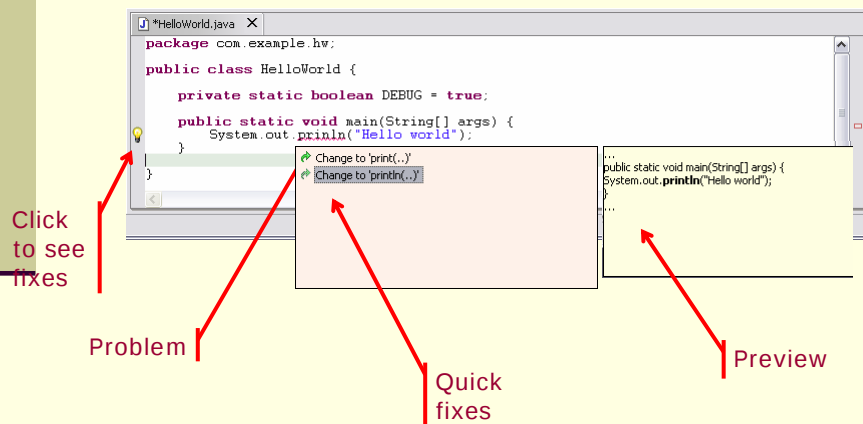
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

54

Java Editor

- On-the-fly spell check catches errors early



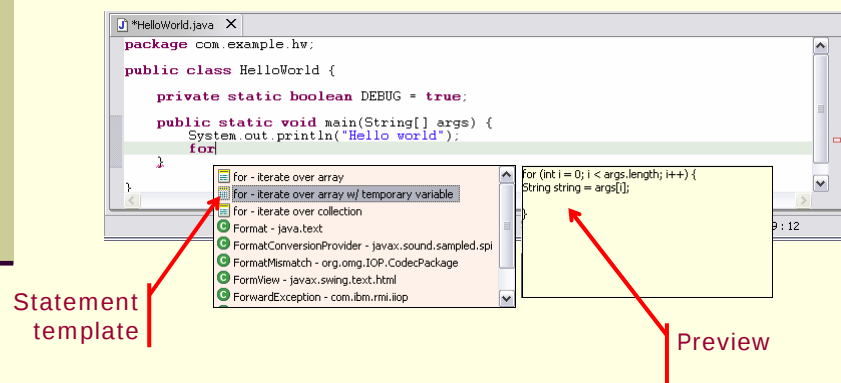
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

55

Java Editor

- Code templates help with drudgery



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

56

Java Editor

■ Java editor creates stub methods

Method stub insertion for anonymous inner types

```
void someMethod() {  
    Runnable r = new Runnable()  
}
```

Runnable() Anonymous Inner Type

Method stub insertion for inherited methods

```
public class TestSuite implements Test  
  
private Vector fTests= new Vector(10);  
private String fName;  
  
clone() Object - Object  
equals(Object obj) boolean - Object  
finalize() void - Object  
hashCode() int - Object  
TestSuite - junit.framework
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

57

Java Editor

■ Java editor helps programmers write good Java code

Variable name suggestion

```
public TestResult run() {  
    TestResult  
    run(result)  
    return res  
}  
/**  
 * Runs the test  
 */  
public void run()  
    result.run()  
}  
/**  
 * Runs the bare
```

result - TestResult
testResult - TestResult

JavaDoc code assist

```
/**  
 * Runs the bare test sequence  
 * @exception  
 *  
 *  
 */  
public void runBare() throws Exception, RuntimeException {  
    setUp();  
}
```

Exception
RuntimeException

Argument hints and proposed argument names

```
boolean expected, boolean actual  
assertEquals(expected, actual);
```

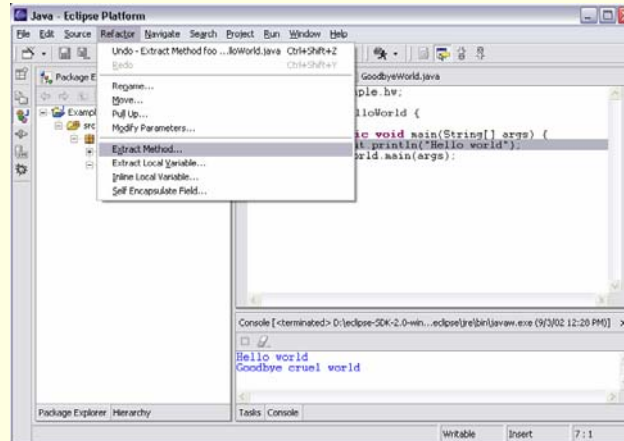
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

58

Refactoring

- JDT has actions for refactoring Java code



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

59

Refactoring

- Refactoring actions rewrite source code
 - Within a single Java source file
 - Across multiple interrelated Java source files
- Refactoring actions preserve program semantics
 - Does not alter what program does
 - Just affects the way it does it
- Encourages exploratory programming
- Encourages higher code quality
 - Makes it easier to rewrite poor code

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

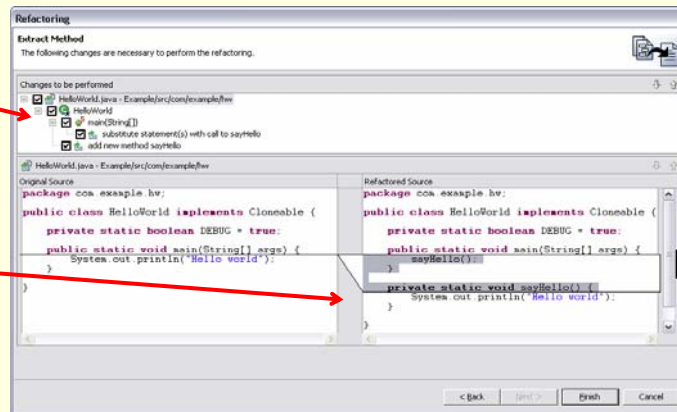
60

Refactoring

- Full preview of all ensuing code changes
 - Programmer can veto individual changes

List of
changes

“before”
vs. “after”



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

61

Refactoring

- Growing catalog of refactoring actions
 - Organize imports
 - Rename {field, method, class, package}
 - Move {field, method, class}
 - Extract {method, local variable, interface}
 - Inline {method, local variable}
 - Reorder method parameters
 - Push members down
 - ...

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

62

Eclipse JDT - Summary

- JDT is a state of the art Java IDE
- Java views, editor, refactoring
 - Helps programmer write and maintain Java code
- Java compiler
 - Takes care of translating Java sources to binaries
- Java debugger
 - Allows programmer to get inside the running program



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

63

POO-Java

*PDE – Plugin
Development
Environment*



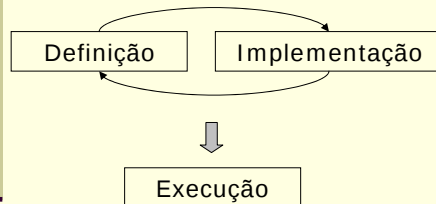
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

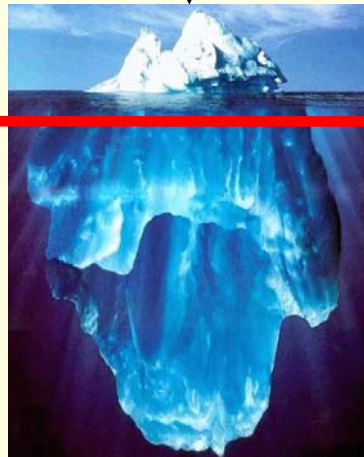
64

Desenvolvendo um plug-in

- Definições Declarativas (manifest)
 - Plugin.xml



- Implementação procedural
 - Java



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

65

Introdução ao PDE

- Ambiente de desenvolvimento de *plug-ins* dentro do *workbench* do Eclipse
- Também é um *plug-in*, construído usando os recursos da própria plataforma e o JDT
- Permite a definição de um *plug-in project*, que agrega os elementos do *plug-in* (arquivo *manifest*, código-fonte, *gifs*)
- Possibilita a construção, compilação, depuração, teste e empacotamento de um *plug-in*

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

66

Introdução ao PDE (cont.)

- Ferramenta para edição do arquivo de *manifest*
- Possui *wizards* para a geração de diversas extensões:
 - actions, xml editor, help content, editor multi-page, wizard para novo arquivo, preferences, property page, views
- Suporte ao *deployment* de um *plug-in*
 - num arquivo .zip ou atualização automática via Eclipse (*feature*)

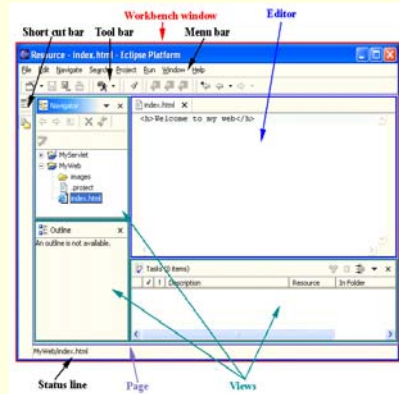
Desenvolvendo com o PDE

- Forma Básica de definir um *plug-in*:
 - (1) Definir no arquivo de manifesto (plugin.xml), os pontos de extensão que o *plug-in* irá contemplar
 - (2) Definir uma subclasse de alguma classe da plataforma Eclipse de acordo com a extensão sendo definida
 - (3) Implementar métodos herdados destas classes juntamente com código de lógica de funcionamento do *plug-in*

Desenvolvendo com o PDE

■ Exemplos de pontos básicos de extensão do *Workbench*

- `org.eclipse.ui.views`
- `org.eclipse.ui.viewActions`
- `org.eclipse.ui.editors`
- `org.eclipse.ui.editorsActions`
- `org.eclipse.ui.popMenus`
- `org.eclipse.ui.actionSets`
- `org.eclipse.ui.actionSetPart` Association
- `org.eclipse.ui.perspectives`
- `org.eclipse.ui.perspectivesE`xtensions



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

69

Eclipse project briefing materials.

Copyright (c) 2002, 2003 IBM Corporation and others. All rights reserved. This content is made available to you by Eclipse.org under the terms and conditions of the Common Public License Version 1.0 ("CPL"), a copy of which is available at <http://www.eclipse.org/legal/cpl-v10.html>

The most up-to-date briefing materials on the Eclipse project are found on the eclipse.org website at <http://eclipse.org/eclipse/>

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

70