

18. 統合開発環境に関する知識 I

1. 科目の概要

ソフトウェア開発を効率的に進めるために用意されている統合開発環境について解説する。統合開発環境の歴史、機能、特徴、使い方などを説明する。また OSS の開発でしばしば利用される代表的な統合開発環境のいくつかについて、その特徴、基本的な機能と導入方法を紹介する。

2. 習得ポイント

本科目の学習により習得することが期待されるポイントは以下の通り。

| 習得ポイント | 説明 | シラバスの対応コマ |
|-----------------------------------|---|-----------|
| I-18-1. 統合開発環境の概要 | 統合開発環境の主な機能や統合開発環境を利用するメリット、統合開発環境を利用したソフトウェア開発手順の概要を解説する。 | 1 |
| I-18-2. さまざまな統合開発環境 | OSS開発で用いられる主な統合開発環境の種類と代表例を紹介する。対応言語、対応フレームワーク、開発用途などの特徴と違いを解説する。 | 2 |
| I-18-3. Eclipseの歴史、開発の背景と特徴 | OSS統合開発環境のひとつであるEclipseについて、開発の背景、歴史、支援団体、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、プラグインによる拡張などの特徴を解説する。 | 3 |
| I-18-4. Eclipseの設定と基本機能 | Eclipseを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、プラグインによる拡張方法、関連アプリケーションの設定などについて具体的に説明する。また基本機能について解説する。 | 6 |
| I-18-5. Eclipseの基本操作 | ワークベンチウィンドウ、プロジェクトの作成、コーディング方法、ビルドと実行など、Eclipseの基本的な操作について解説する。またプラグインによる拡張の利用例など、Eclipseの効果的な活用についても触れる。 | 7 |
| I-18-6. NetBeans IDEの歴史、開発の背景と特徴 | OSS統合開発環境のひとつであるNetBeans IDEについて、開発の背景、歴史、コミュニティ、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、ユーザインタフェースなどについて解説する。 | 4 |
| I-18-7. NetBeans IDEの設定と基本機能、基本操作 | NetBeans IDEを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、関連アプリケーションの設定などについて具体的に説明する。また基本機能と基本的な操作について解説する。 | 8,9 |
| I-18-8. WideStudioの歴史、開発の背景と特徴 | OSS統合開発環境のひとつであるWideStudioについて、開発の背景、歴史、コミュニティ、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、MWT (Multi-platform Widget Toolkit)などの特徴を解説する。 | 5 |
| I-18-9. WideStudioの設定と基本機能 | WideStudioを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、作業環境の管理方法などについて具体的に説明する。また基本機能について解説する。 | 10 |
| I-18-10. WideStudioの基本操作 | プロジェクトの作成、アプリケーションウィンドウ、インスタンスの管理、イベントプロシージャ、プログラムのビルドと実行、デバッグ方法など、WideStudioの基本的な操作について解説する。 | 11 |

【学習ガイダンスの使い方】

1. 「習得ポイント」により、当該科目で習得することが期待される概念・知識の全体像を把握する。
2. 「シラバス」、「IT 知識体系との対応関係」、「OSS モデルカリキュラム固有知識」をもとに、必要に応じて、従来の IT 教育プログラム等との相違を把握した上で、具体的な講義計画を考案する。
3. 習得ポイント毎の「学習の要点」と「解説」を参考にして、講義で使用する教材等を準備する。

3. IT 知識体系との対応関係

「18. 統合開発環境に関する知識 I」と IT 知識体系との対応関係は以下の通り。

| 科目名 | 基本レベル(I) | | | | | | | | | | | 応用レベル(II) | | | |
|--------------------|---------------------|---------------|---------------|----------------|------------------|----------------------|--------------------|-----------------------|---------------------|-------------------------|-----------------------|-------------------------|---------------------------|----------------------------|------------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 18. 統合開発環境に関する知識 I | <統合開発環境によるソフトウェア開発> | <さまざまな統合開発環境> | <Eclipseとは何か> | <NetBeansとは何か> | <WideStudioとは何か> | <Eclipse入門 - セットアップ> | <Eclipse入門 - 基本操作> | <NetBeans入門 - セットアップ> | <NetBeans入門 - 基本操作> | <WideStudio入門 - セットアップ> | <WideStudio入門 - 基本操作> | <EclipseによるJavaプログラミング> | <EclipseによるWebアプリケーション開発> | <NetBeansによるWebアプリケーション開発> | <WideStudioによるWebアプリケーション開発> |

[シラバス : http://www.ipa.go.jp/software/open/oss/download/Mdel_Curriculum_05_18.pdf]

<IT 知識体系上の関連部分>

| 分野 | 科目名 | 基本レベル(I) | | | | | | | | | | | | | 応用レベル(II) | | | |
|--------------------|-----|-----------------------------------|--------------------------------|---------------------------|-----------------------------|------------------------------------|--------------------------|-----------------------------|--|--------------------------------|-------------------------|----------------------------|-----------------------|--------------------------------|-----------------|----|--|--|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | |
| 情報処理系標準事項と情報セキュリティ | 1 | IT-IAS1. 情報保証と情報セキュリティ | IT-IAS2. 基礎的な問題 | IT-IAS3. 運用上の問題 (対策) | IT-IAS4. ホリゾ | IT-IAS5. 攻撃 | IT-IAS6. 情報セキュリティ分野 | IT-IAS7. フォレンジック (情報セキュリティ) | IT-IAS8. 情報の保護 | IT-IAS9. 情報セキュリティサービ | IT-IAS10. 脅威分析モデル | IT-IAS11. 脆弱性 | | | | | | |
| | 2 | IT-SP. 社会的な観点とグローバルなフェッショナルとしての課題 | IT-SP1. プロフェッショナルとしてのコミュニケーション | IT-SP2. コンピュータの歴史 | IT-SP3. コンピュータを取り巻く社会環境 | IT-SP4. チームワーク | IT-SP5. 知的財産権 | IT-SP6. コンピュータの法的問題 | IT-SP7. 組織のIT | IT-SP8. プロフェッショナルとしての倫理的な問題と責任 | IT-SP9. プライバシーと個人の自由 | | | | | | | |
| 応用技術 | 3 | IT-IM. 情報管理 | IT-IM1. 情報管理の概念と基礎 | IT-IM2. データベース関係の概観 | IT-IM3. データアーキテクチャ | IT-IM4. データモデリングとデータベース設計 | IT-IM5. データと情報の管理 | IT-IM6. データベースの応用分野 | | | | | | | | | | |
| | 4 | IT-WS. Webシステムとその技術 | IT-WS1. Web技術 | IT-WS2. 情報アーキテクチャ | IT-WS3. デジタルメディア | IT-WS4. Web開発 | IT-WS5. 脆弱性 | IT-WS6. ソーシャルソフトウェア | | | | | | | | | | |
| ソフトウェアの手法と技術 | 5 | IT-PF. プログラミング基礎 | IT-PF1. 基本データ型 | IT-PF2. プログラミングの基本的構成要素 | IT-PF3. オブジェクト指向プログラミング | IT-PF4. アルゴリズムと問題解決 | IT-PF5. イベント駆動プログラミング | IT-PF6. 再帰 | | | | | | | | | | |
| | 6 | IT-PT. 技術を統合するためのプログラミング | IT-PT1. システム間連携 | IT-PT2. データやり取りと交換 | IT-PT3. 統合的コーディング | IT-PT4. スクリプトプログラミング | IT-PT5. ソフトウェアセキュリティの要領 | IT-PT6. 種々のプログラミング言語の概要 | IT-PT7. プログラミング言語の概要 | | | | | | | | | |
| | 7 | DE-SWE. ソフトウェア工学 | DE-SWE0. 歴史と概要 | DE-SWE1. ソフトウェアプロセス | DE-SWE2. ソフトウェアの要求と仕様 | DE-SWE3. ソフトウェアの設計 | DE-SWE4. ソフトウェアのテストと検証 | DE-SWE5. ソフトウェアの保守 | DE-SWE6. ソフトウェア開発・検証・保守と連携 [18-1-1, 2] | DE-SWE7. ソフトウェアプロジェクト管理 | DE-SWE8. 言語翻訳 | DE-SWE9. ソフトウェアのフォールトトレランス | DE-SWE10. ソフトウェアの構成管理 | DE-SWE11. ソフトウェアの標準化 | | | | |
| | 8 | IT-SIA. システムインテグレーションとアーキテクチャ | IT-SIA1. 要求仕様 | IT-SIA2. 調達/手配 | IT-SIA3. インテグレーション | IT-SIA4. プロジェクト管理 | IT-SIA5. テストと品質保証 | IT-SIA6. 組織の特性 | IT-SIA7. アーキテクチャ | | | | | | | | | |
| システム基盤 | 9 | IT-NET. ネットワーク | IT-NET1. ネットワークの基礎 | IT-NET2. ルーティングとスイッチング | IT-NET3. 物理層 | IT-NET4. セキュリティ | IT-NET5. アプリケーション分野 | IT-NET6. ネットワーク管理 | | | | | | | | | | |
| | 10 | DE-NWK. テレコミュニケーション | DE-NWK0. 歴史と概要 | DE-NWK1. 通信ネットワークのアーキテクチャ | DE-NWK2. 通信ネットワークの設計 | DE-NWK3. LANとWAN | DE-NWK4. クラウドサービスとセキュリティ | DE-NWK5. データのセキュリティと整合性 | DE-NWK6. ワイヤレスコンピューティングとモバイルコンピューティング | DE-NWK7. データ連携 | DE-NWK8. 組み込み機器向けネットワーク | DE-NWK9. 通信技術とネットワーク概要 | DE-NWK10. 性能評価 | DE-NWK11. ネットワーク管理 | DE-NWK12. 圧縮と伸張 | | | |
| | 11 | IT-PI. プラットフォーム技術 | IT-PI1. オペレーティングシステム | IT-PI2. アーキテクチャと機構 | IT-PI3. コンピュータインフラストラクチャ | IT-PI4. デプロイメントソフトウェア | IT-PI5. ファームウェア | IT-PI6. ハードウェア | | | | | | | | | | |
| ソフトウェア開発 | 12 | DE-OPS. オペレーティングシステム | DE-OPS0. 歴史と概要 | DE-OPS1. 並行性 | DE-OPS2. スケジューリングと管理 | DE-OPS3. メモリ管理 | DE-OPS4. セキュリティと保護 | DE-OPS5. ファイル管理 | DE-OPS6. リアルタイムOS | DE-OPS7. OSの概要 | DE-OPS8. 設計の原則 | DE-OPS9. デバイスマネジメント | DE-OPS10. システム性能評価 | | | | | |
| | 13 | DE-CAD. コンピュータのアーキテクチャと構成 | DE-CAD0. 歴史と概要 | DE-CAD1. コンピュータアーキテクチャの基礎 | DE-CAD2. メモリシステムの構成とアーキテクチャ | DE-CAD3. インタフェースと通信 | DE-CAD4. デバイスサブシステム | DE-CAD5. CPUアーキテクチャ | DE-CAD6. 性能・コスト評価 | DE-CAD7. 分散・並列処理 | DE-CAD8. コンピュータによる計算 | DE-CAD9. 性能向上 | | | | | | |
| 複数領域にまたがるもの | 14 | IT-ITF. IT基礎 | IT-ITF1. IT00-100の基礎的なテーマ | IT-ITF2. 組織の問題 | IT-ITF3. IT1. 01の歴史 | IT-ITF4. IT分野 (学科) とそれに関連する分野 (学科) | IT-ITF5. 応用情報 | IT-ITF6. IT分野における数学と統計学の活用 | | | | | | | | | | |
| | 15 | DE-ESY. 組み込みシステム | DE-ESY0. 歴史と概要 | DE-ESY1. 低電力コンピューティング | DE-ESY2. 高信頼性システムの設計 | DE-ESY3. 組み込み用アーキテクチャ | DE-ESY4. 開発環境 | DE-ESY5. ライフサイクル | DE-ESY6. 要件分析 | DE-ESY7. 仕様定義 | DE-ESY8. 構造設計 | DE-ESY9. テスト | DE-ESY10. プロジェクト管理 | DE-ESY11. 並行設計 (ハードウェア、ソフトウェア) | DE-ESY12. 実装 | | | |

4. OSS モデルカリキュラム固有の知識

OSS モデルカリキュラム固有の知識として、実用的な統合開発環境に関する知識がある。Eclipse、NetBeans、WideStudio といった著名な統合開発環境 (IDE) を用いて、IDE の機能や効率的な開発を行う手法を学ぶ。

| 科目名 | 第1回 | 第2回 | 第3回 | 第4回 | 第5回 | 第6回 | 第7回 | 第8回 | 第9回 | 第10回 | 第11回 |
|--------------------|--|-------------------------------|------------------------------------|--|--|--|-------------------|----------------------|------------------------|------------------------|----------------------|
| 18. 統合開発環境に関する知識 I | (1) 統合開発環境によるソフトウェア開発 (2) 統合開発環境の主な機能 | (1) 統合開発環境の種類 (2) 主な統合開発環境 | (1) Eclipse の背景 (2) Eclipse の特徴 | (1) NetBeans の背景 (2) NetBeans IDE の特徴 | (1) WideStudio の背景 (2) WideStudio の特徴 | (1) Eclipse のセットアップ (2) Eclipse の基本操作 | (1) Eclipse の基本操作 | (1) NetBeans のセットアップ | (1) NetBeans IDE の基本操作 | (1) WideStudio のセットアップ | (1) WideStudio の基本操作 |

(網掛け部分は IT 知識体系で学習できる知識を示し、それ以外は OSS モデルカリキュラム固有の知識を示している)

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-1. 統合開発環境の概要 | |
| 対応する コースウェア | 第1回（統合開発環境によるソフトウェア開発） | |

I-18-1. 統合開発環境の概要

統合開発環境の主な機能や統合開発環境を利用するメリット、統合開発環境を利用したソフトウェア開発手順の概要を解説する。

【学習の要点】

- * OSS の統合開発環境は、その多くは GUI で、大規模なプログラムも一人で作成することが出来る。
- * 統合開発環境では、プログラムを単なるソースコードではなく、プロジェクトとして開発をする。

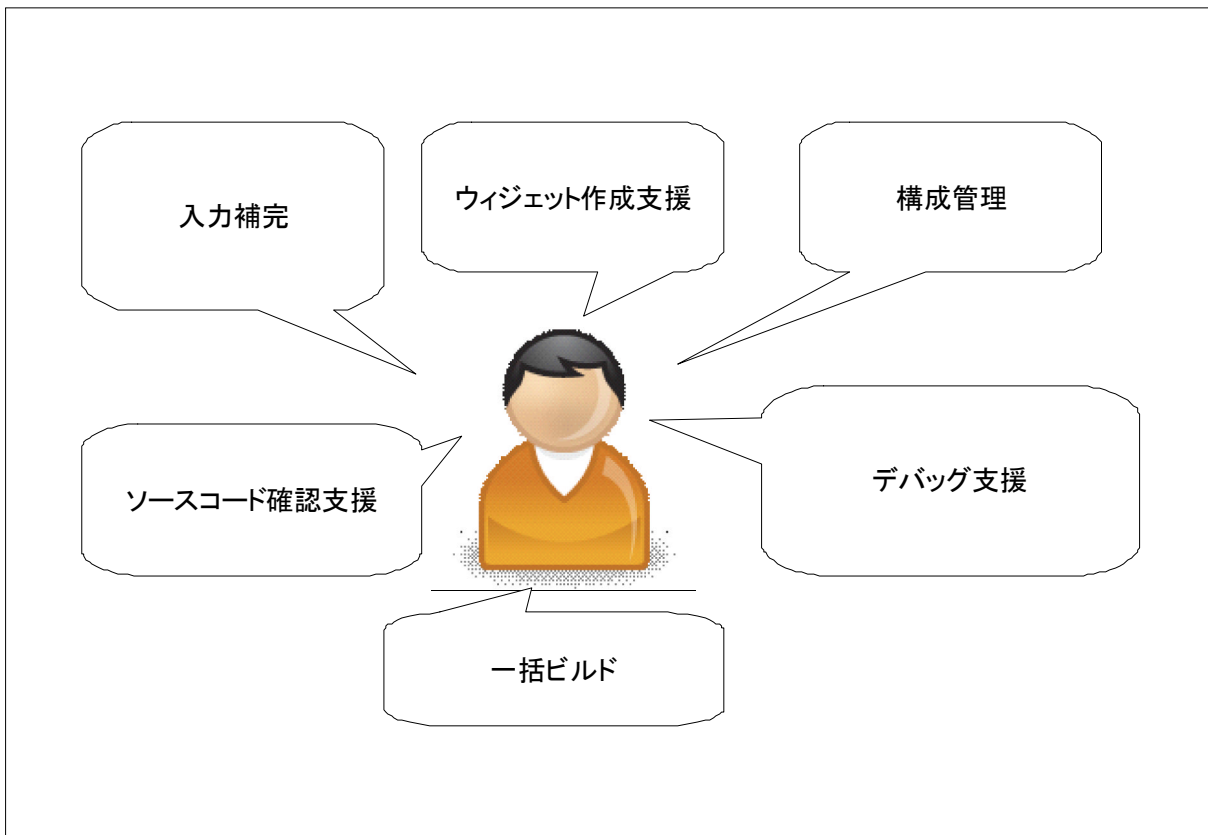


図 I-18-1. 統合開発環境の主な機能

【解説】

1) 統合開発環境の主な機能

- * 構成管理
開発中のソフトウェアのライブラリのディレクトリ構成や各ファイルのバージョンなど、手で管理をすると煩雑なところを、自動で管理してくれる。
- * ウィジェット作成支援
GUI 周りのソースコードを、グラフィカルにウィジェット(GUI の構成要素)を配置するだけで自動生成してくれる。
- * 入力補完
関数やキーワードなど途中まで入力したものを自動で補完したり、入力候補を表示する機能を備えている。
- * ソースコード確認支援
ソースコードの色付けなど、コードを確認する際に便利な補助機能を備えている。
- * 一括ビルド
ビルドをする時、長いコマンドを打つことなく、ボタン一つでビルドができる。
- * デバッグ支援
グラフィカルなデバッグツールを備えている。

2) 統合開発環境のメリット

- * エディタ、コンパイラ、デバッガなどがひとつの GUI インタフェースで扱えるため、開発者の負担が非常に軽くなる。
- * 自動でディレクトリ構成をしてくれるので、ソースコードの管理の手間が省ける。
- * バージョン管理が容易にできる。
- * GUI プログラミングの手間が省ける。

3) 統合開発環境での開発手順

- * プロジェクトの作成
まず、プロジェクトを作る。プロジェクトとは、一つのプログラムを作る際のソースコードやライブラリ、アイコンの画像ファイルなどをひとまとめにした単位のことである。
- * ソースコードの作成
プロジェクトにソースコードのファイルを追加、編集する。
- * ビルド、デバッグ
ビルド、デバッグをして完成させる。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|---------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-2. さまざまな統合開発環境 | |
| 対応する コースウェア | 第 2 回(さまざまな統合開発環境) | |

I-18-2. さまざまな統合開発環境

OSS 開発で用いられる主な統合開発環境の種類と代表例を紹介する。対応言語、対応フレームワーク、開発用途などの特徴と違いを解説する。

【学習の要点】

- * 主な OSS による統合開発環境は、Eclipse、NetBeans、WideStudio などがある。
- * Eclipse は Java 用統合開発環境としてよく使われているが、プラグインを追加することにより、C/C++、Perl、Ruby など多様な言語によるソフトウェア開発にも対応できる。
- * NetBeans は Java、Ruby、C/C++用統合開発環境である。
- * WideStudio は、C/C++を始め、Java、Ruby、Perl、Python、Objective CAML によるデスクトップアプリケーションの開発を効率的に行うことが可能である。

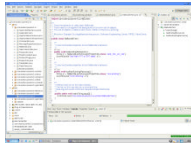
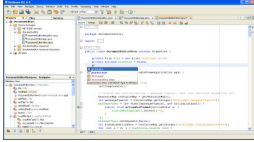
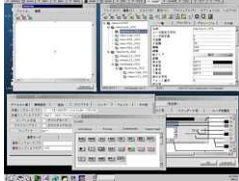
| | |
|--|---|
| <p>Eclipse</p> <ul style="list-style-type: none"> ・プラグインが非常に豊富 ・フレームワーク対応 ・ソースコード作成支援 |  |
| <p>NetBeans</p> <ul style="list-style-type: none"> ・サン・マイクロシステムズによる開発 ・プラグインが利用できる ・GUI開発機能 |  |
| <p>WideStudio</p> <ul style="list-style-type: none"> ・GUIアプリケーションが作りやすい ・マルチプラットフォーム対応 GUIライブラリ採用 ・多言語対応 ・国産OSS(日本語ドキュメントが充実) |  |

図 I-18-2. 代表的な OSS 統合開発環境の特徴

【解説】

1) 主な OSS による統合開発環境

OSS による統合開発環境には、主に次のものがある。

- * Eclipse
- * NetBeans
- * WideStudio
- * J-alcedo

2) Eclipse の特徴

- * プラグインによる機能の拡張
プラグインにより、Java 以外の言語の開発や有用なデバッガの使用、UML 図の作成ができるようになる。
- * フレームワーク対応
Struts や JUnit など、プラグインによりさまざまなフレームワークに対応している。
- * ソースコード作成支援
関数補完機能を備えたエディタを備えている。

3) NetBeans の特徴

- * 多数のプログラミング言語に対応
NetBeans は Java のほかにも、Ruby、C/C++などの開発ができる。
- * GUI 開発機能
NetBeans は、GUIビルダーのみである程度の機能を持ったアプリケーションが開発できるなど、特に GUI 開発に優れている。
- * フレームワーク対応
Struts や Java Server Faces、JSF などの Web フレームワークに対応している。

4) WideStudio の特徴

- * 多数のプログラミング言語に対応
WideStudio は、C/C++をはじめ、Java、Perl、Python、Ruby などを用いて開発ができる。
- * MWT(Multi-platform Widget Toolkit)
WideStudio は MWT ライブラリを採用していることにより、同じソースコードでも違う OS で再コンパイルすることで、その OS のネイティブコードをもったアプリケーションが作成できる。
- * マルチエンコーディング
マルチエンコーディングに対応しているため、ソースコードのテキストエンコーディングを気にする必要がない。
- * 多言語対応
英語、日本語に限らず、韓国語や中国語にも対応している。
- * 国産 OSS
日本で開発がはじまった OSS IDE であるため、日本語のドキュメントが充実している。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|------------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-3. Eclipse の歴史、開発の背景と特徴 | |
| 対応する コースウェア | 第 3 回 (Eclipse とは何か) | |

I-18-3. Eclipse の歴史、開発の背景と特徴

OSS 統合開発環境のひとつである Eclipse について、開発の背景、歴史、支援団体、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、プラグインによる拡張などの特徴を解説する。

【学習の要点】

- * Eclipse は 1998 年に IBM によって開発が始まり、2001 年にオープンソース化された。
- * Eclipse はプラグインを追加することによって C/C++ など Java 以外の他の言語の開発ができるほか、バグパターンの検出やコーディングスタイルのチェックなどのプログラム開発において非常に便利な機能を追加することができる。
- * Eclipse のライセンスは、EPL(Eclipse Public License)である。

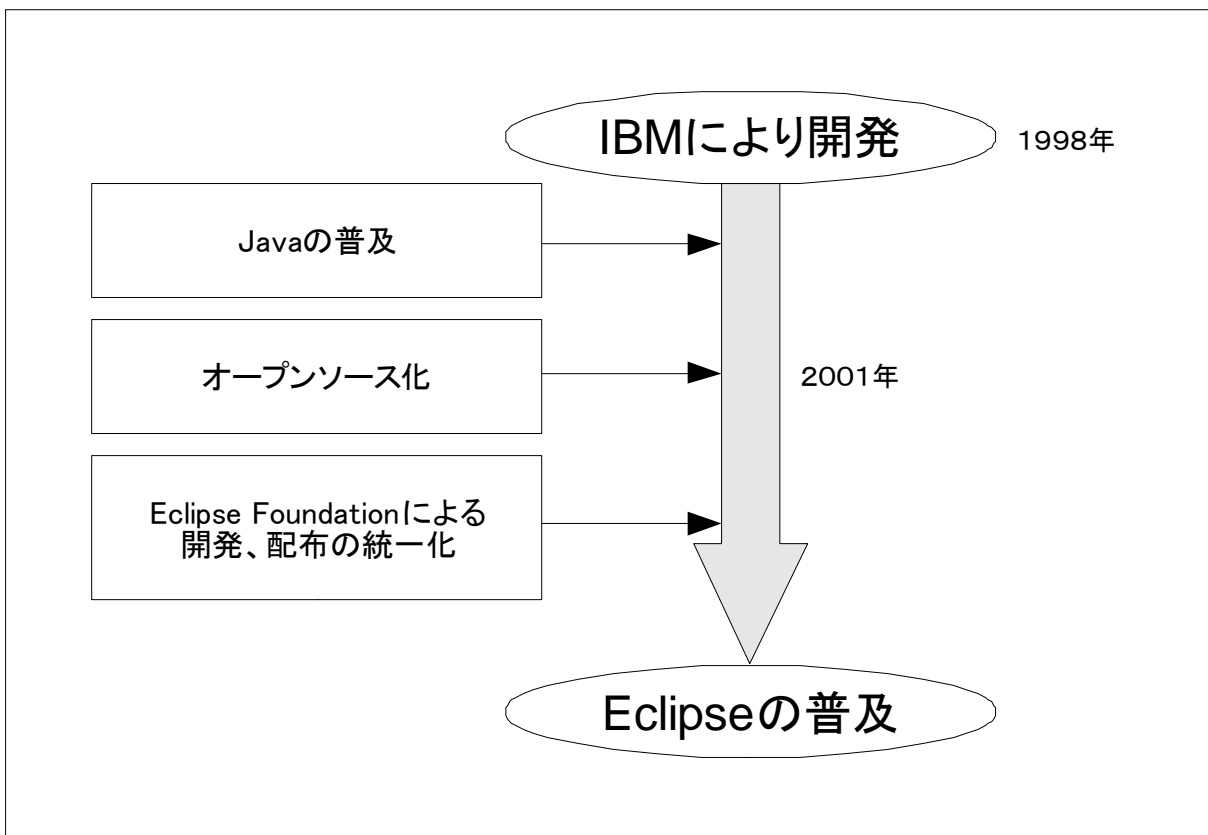


図 I-18-3. Eclipse 普及の概要

【解説】

1) Eclipse の背景と歴史、支援団体

- * 1998 年、Eclipse の開発が始まる。
IBM により、統合開発環境として開発が始まる。
- * 2001 年、オープンソース化
オープンソースコミュニティ Eclipse Board of Stewards が設立され、多くの開発者により、急速に開発が進む。
- * 2004 年、Eclipse Foundation 設立
オープンソースコミュニティが再編され、非営利組織の Eclipse Foundation に Eclipse に関してすべてが委譲される。

2) Eclipse 関連プロジェクト

Eclipse には、次のような関連プロジェクトがある。

- * Business Intelligence and Reporting Tools
- * Data Tools Platform
- * Device Software Development Platform
- * Eclipse Project
- * Eclipse Modeling Project
- * SOA Tools
- * Eclipse Technology Project
- * Tools Project
- * Test and Performance Tools Platform Project
- * Eclipse Web Tools Platform Project

3) Eclipse のライセンス

Eclipse は、Eclipse Public License (EPL)を採用している。これはオープンソースのライセンスのひとつである。

4) Eclipse のアーキテクチャ、機能、プラグインによる拡張

Eclipse はその機能のほとんどがプラグインによって実現されている。Java 開発環境自体もプラグイン化されており、それが標準的に装備されている。

その他、バージョン管理システム CVS との連携や、ユニットテストを行うためのフレームワーク JUnit との連携など、さまざまな機能をプラグインによって追加することができる。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|---------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-4. Eclipse の設定と基本機能 | |
| 対応する コースウェア | 第 6 回 (Eclipse 入門-セットアップ) | |

I-18-4. Eclipse の設定と基本機能

Eclipse を利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、プラグインによる拡張方法、関連アプリケーションの設定などについて具体的に説明する。また基本機能について解説する。

【学習の要点】

- * Eclipse は Eclipse.org のサイトからダウンロードする。また、起動させるには JRE が必須である。
- * Eclipse は、プラグインによる機能の拡張機能を備えている。

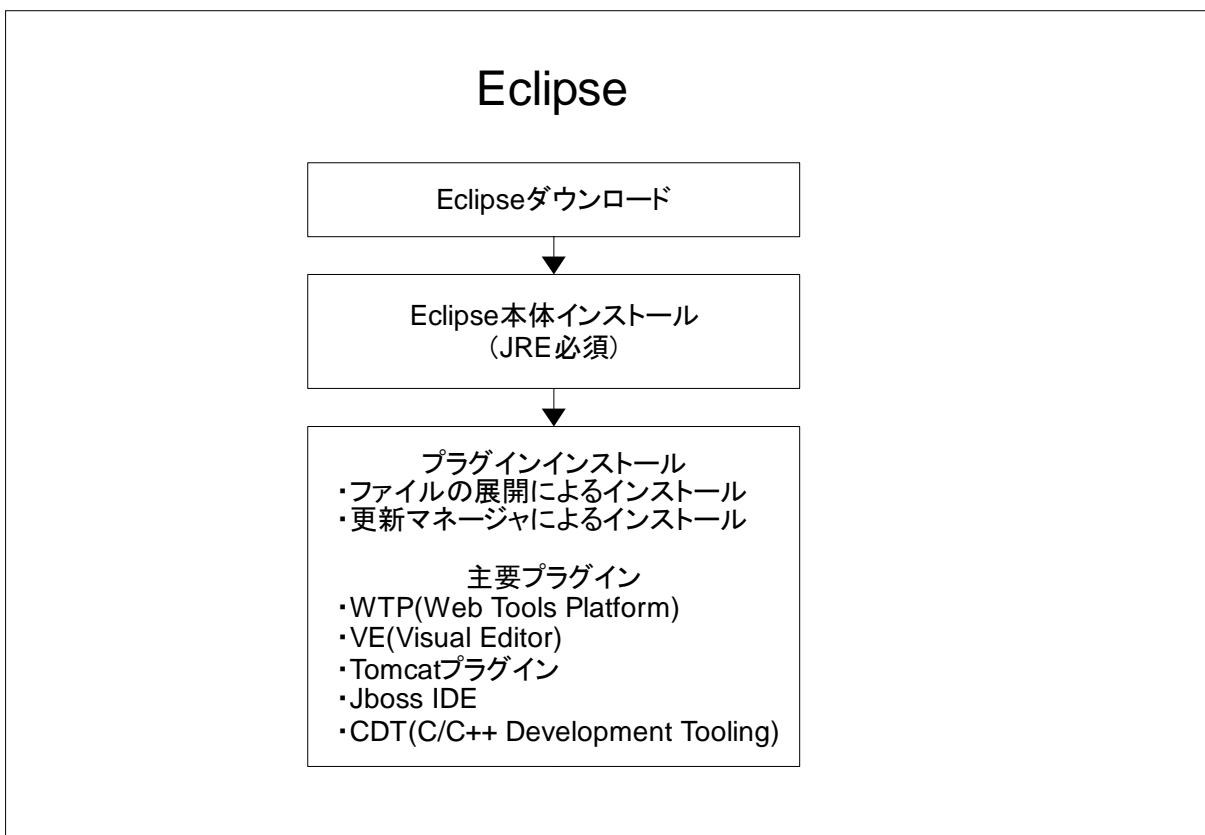


図 I-18-4. Eclipse のインストール方法など

【解説】

1) Eclipse の入手方法

<http://download.eclipse.org/eclipse/downloads/index.php> にて、最新版をダウンロードできる。また、Eclipse を日本語化するために、Language Pack もダウンロードしておくといよい。

2) Eclipse の動作環境とその整備

Eclipse は Java で書かれているため、インストールするには Java の実行環境である JRE(Java Runtime Environment)が必要である。これは以下のページでダウンロードできる。

<http://www.java.com/ja/download/manual.jsp>

なるべく最新版を手に入れることが望ましい。

3) Eclipse のプラグインの拡張方法

Eclipse には数多くのプラグインがあり、それをインストールするには以下の方法をとる。

大きく分けてその方法には 2 種類ある。

* ファイルのコピーによるインストール

多くの場合、zip ファイルで配布されているプラグインを解凍すると、features フォルダと plugins フォルダと説明のファイルが現れる。そのうち、features フォルダの中身を Eclipse 本体の features フォルダに、plugins の中身を Eclipse 本体の plugins フォルダにコピーする。

* Update Site を使ったインストール

プラグインの作者が Update Site を用意している場合、これを使ってプラグインをインストールするとよい。Update Site の登録の仕方は次の通り。ツールバーからヘルプ>ソフトウェア更新>検索およびインストールを選ぶ。次に、インストールする新規フィーチャーを選択する画面になるので希望のフィーチャーを選択し、新規リモートサイト>名前と URL の入力>OK>終了といった流れになる。

4) Eclipse の基本機能

* プラグインによる機能の拡張

* グラフィカルデバッガによるデバッグ、ステップ実行

* バージョン管理システム CVS との連携

* ユニットテストの自動化を行うフレームワークである JUnit との連携

* ビルドシステム Ant との連携

* リファクタリングの実現の支援

* コード編集支援

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|-------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-5. Eclipse の基本操作 | |
| 対応する コースウェア | 第 7 回 (Eclipse 入門-基本操作) | |

I-18-5. Eclipse の基本操作

ワークベンチウィンドウ、プロジェクトの作成、コーディング方法、ビルドと実行など、Eclipse の基本的な操作について解説する。またプラグインによる拡張の利用例など、Eclipse の効果的な活用についても触れる。

【学習の要点】

- * Eclipse を起動したときにあらわれるウィンドウ全体を、ワークベンチウィンドウと呼び、プロジェクトの全工程を統括したビューを提供する。
- * Eclipse では、ソースコードをプロジェクト単位で管理する。

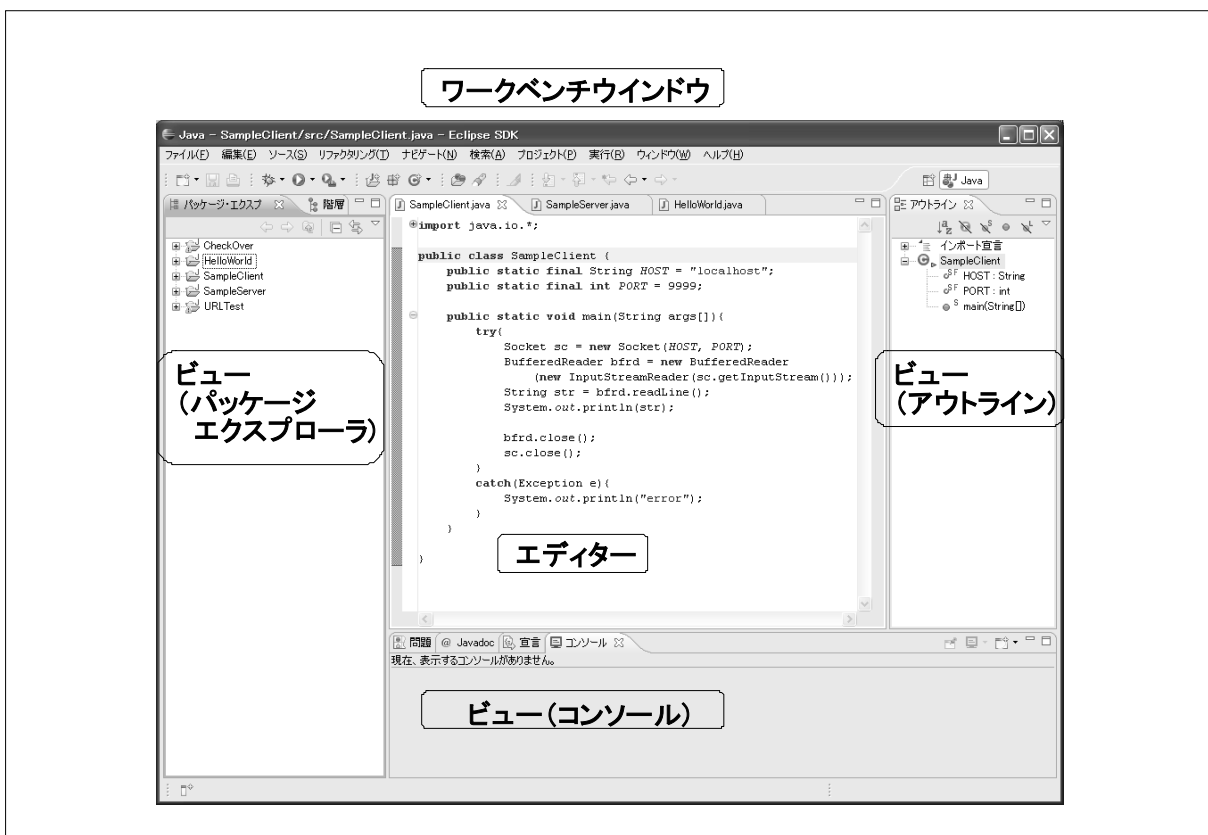


図 I-18-5. Eclipse のワークベンチウィンドウ

【解説】

1) ワークベンチウィンドウ

ワークベンチウィンドウとは、Eclipse を起動したときにあらわれるウィンドウ全体をさす。初期状態では、左側のペインには、パッケージエクスプローラーが、中央上部はテキストエディタ、右側のペインにはアウトライン、下部のペインにはコンソールが表示されている。

2) プロジェクトの作成

新しくプロジェクトを作るには、ファイル>新規>プロジェクトを選び、任意のプロジェクトを選んでプロジェクト名を付ければよい。

3) コーディング支援機能

コーディングをする時は、ソース編集支援機能を使うことができる。例えば、関数名を入力した時には、関数名や引数の補完を行う。そのほかにも、ソース内に明らかな文法の間違ひがある場合は、その行をマークするなどの機能がある。また、メニューバーの実行から、実行を選ぶと、自動でビルドをしてプログラムを実行することができる。

4) プラグインによる機能の拡張

ヘルプ>ソフトウェア更新>検索とインストールを選ぶか、Eclipse 本体のプラグインフォルダにプラグインファイルをコピーすることで、プラグインをインストールすることができる。これにより、Eclipse を機能拡張することができる。以下に、代表的なプラグインを示す。

- * WTP(Web Tools Platform)
Web アプリケーション開発のためのツール群。
- * CDT(C/C++ Development Tooling)
C、C++のための開発ツール群。
- * EPIC Perl プラグイン
Perl のための開発ツール群。
- * Omondo プラグイン
UML のクラス図、シーケンス図など記述するためのツール群。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|-----------------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-6. NetBeans IDE の歴史、開発の背景と特徴 | |
| 対応する コースウェア | 第 4 回 (NetBeans IDE とは何か) | |

I-18-6. NetBeans IDE の歴史、開発の背景と特徴

OSS 統合開発環境のひとつである NetBeans IDE について、開発の背景、歴史、コミュニティ、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、ユーザーインターフェースなどについて解説する。

【学習の要点】

- * NetBeans は、1996 年にチェコの学生によってはじまった Xelfi を元としている。
- * NetBeans はサン・マイクロシステムズ社を中心としたコミュニティで開発されている。
- * NetBeans のライセンスは、CDDL(Common Development and Distribution)である。

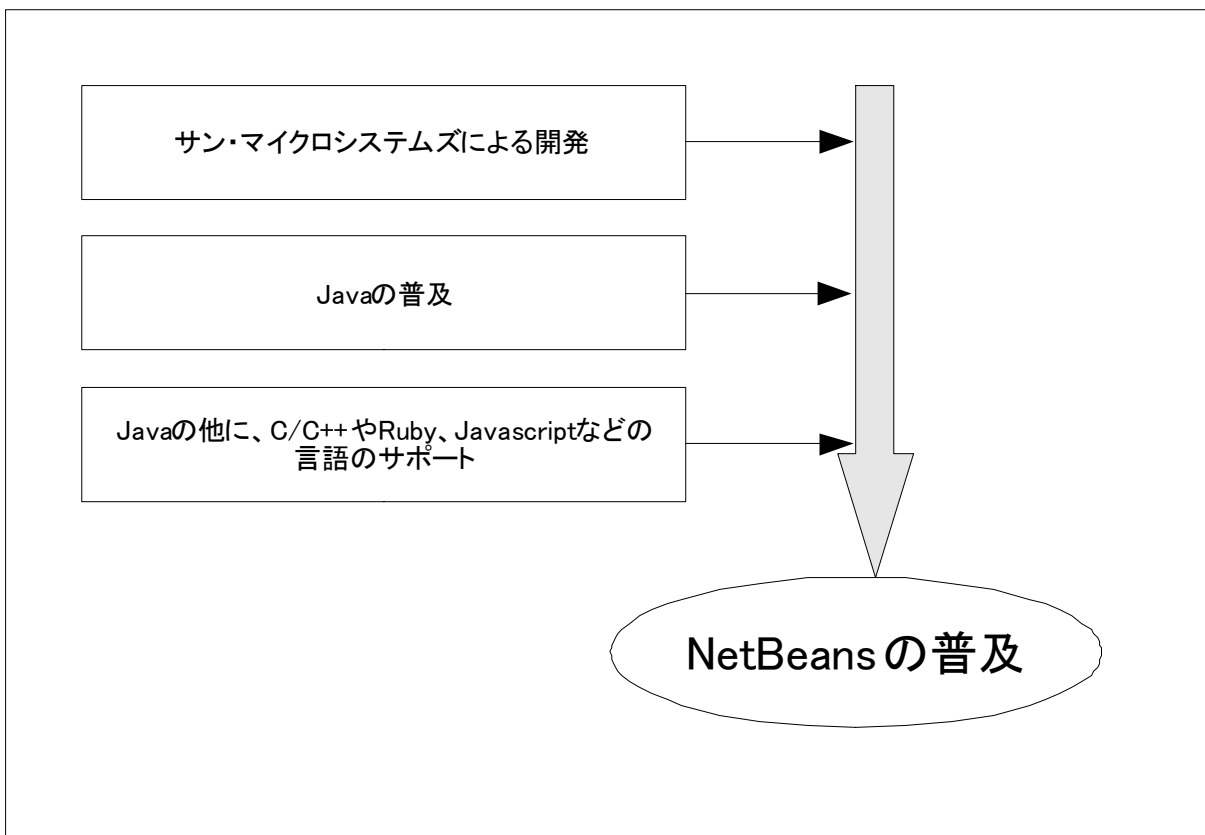


図 I-18-6. NetBeans 普及の概要

【解説】

1) NetBeans の歴史、背景

NetBeans は、1996 年にチェコのチャールズ大学の学生プロジェクトとして始まった。1997 年には会社を設立し、1999 年に NetBeans としてリリースされた。そしてこの年、Sun Microsystems 社に買収され、その翌年の 2000 年には NetBeans はオープンソースプロジェクトとしてリリースされることになった。それ以降は順調に利用者数を伸ばしながら、度重なるアップデートを行っている。

2) NetBeans のコミュニティ

Netbeans.org 内に日本語サイトがあり、メーリングリスト、ドキュメント公開等を行っている。

3) NetBeans のインタフェース

左側にファイル一覧とインスペクタ。中央には GUI 作成ツール。右側にはボタンなどのパレットがある。中央の GUI 作成ツールはソースエディタにもなり、その場合は中央から右側にかけてウインドウが開く。そして一番下にはコンソール出力がある。コンパイル時のエラーなどはこの欄に表示される。

4) NetBeans のライセンス

NetBeans のライセンスは、CDDL(Common Development and Distribution)ライセンスによる。CDDL ライセンスの日本語訳は、下記を参照のこと。

http://sourceforge.jp/projects/opensource/wiki/licenses%2FCommon_Development_and_Distribution_License

5) NetBeans の特徴

* 多数のプログラミング言語に対応

NetBeans は Java のほかにも、Ruby、C/C++などの開発ができる。

* GUI 開発機能

NetBeans は、GUIビルダーのみである程度の機能を持ったアプリケーションが開発できるなど、特に GUI 開発に優れている。

* フレームワーク対応

Struts や Java Server Faces、JSF などの Web フレームワークに対応している。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|--|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-7. NetBeans の設定と基本機能、基本操作 | |
| 対応する コースウェア | 第 8 回 (NetBeans 入門-セットアップ) 第 9 回 (NetBeans 入門-基本操作) | |

I-18-7. NetBeans の設定と基本機能、基本操作

NetBeans IDE を利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、関連アプリケーションの設定などについて具体的に説明する。また、基本機能と基本的な操作について解説する。

【学習の要点】

- * NetBeans は、<http://download.netbeans.org> からダウンロードする。
- * NetBeans をインストールするには、まず JDK をインストールする必要がある。必要なバージョンは <http://www.netbeans.org> にて確認すること。
- * ソフトウェア開発の手順は、まずプロジェクトを作成し、パッケージやクラスの生成をしてソースコードを書き、デバッグを行った上で、ビルドで実行可能なプログラムを生成する。

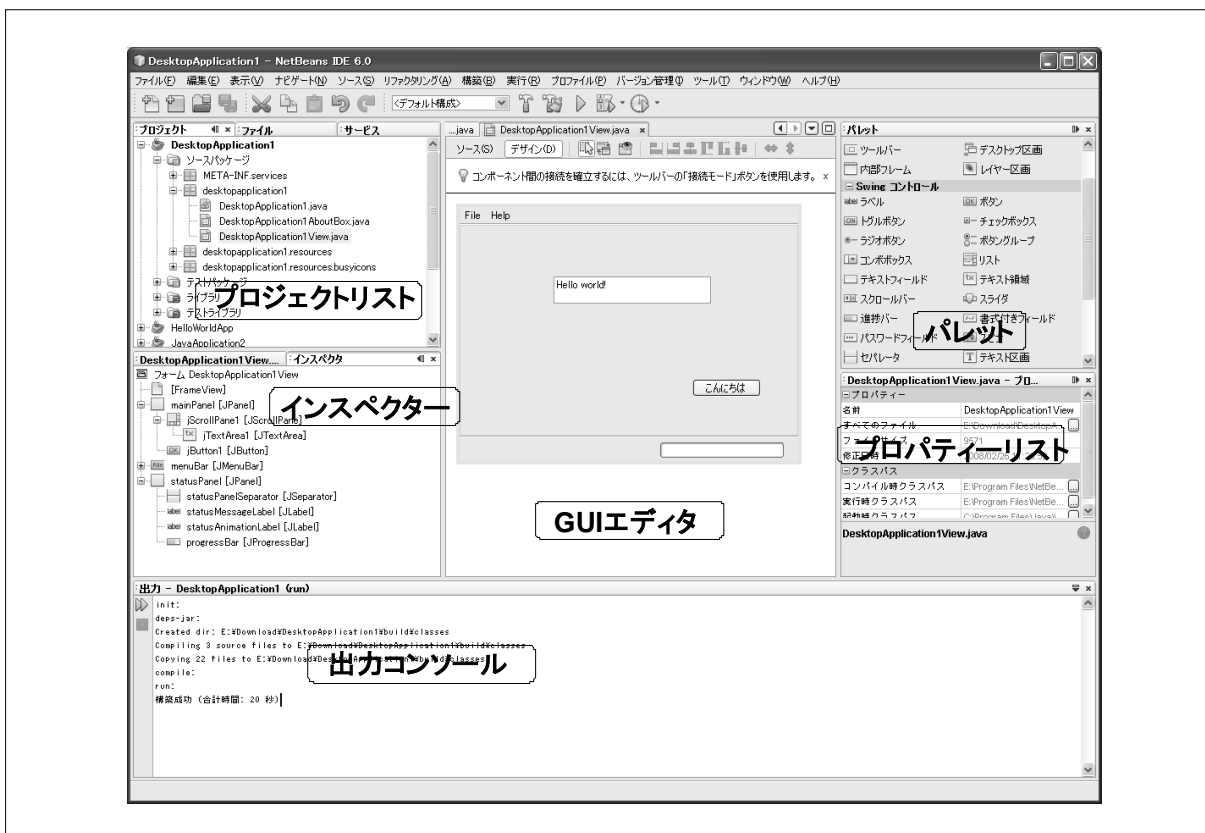


図 I-18-7. NetBeans スクリーンショット

【解説】

1) NetBeans の入手、インストール方法

NetBeans は、<http://download.netbeans.org> からダウンロードする。2008 年 1 月現在の最新版は、NetBeans IDE 6.0 である。

NetBeans をインストールする前に、環境を整備する必要がある。推奨されるハードウェア構成などは、http://www.netbeans.org/community/releases/60/relnotes_ja.html にて紹介されている。

また、ソフトウェア環境としては、J2SE (Java SE Development Kit) JDK 5.0 Update 12 以降 (JDK 6.0 を含む) を必要とする。

インストールする際は、ダウンロードしたファイルの中からインストーラを起動して、NetBeans をインストールしたいファイルパスと、JDK がインストールされているファイルパスを指定して行う。これ以外は特に変更しなくてもよい。

2) NetBeans の基本機能

* ユーザーインターフェースの作成

ウインドウ、メニュー、ツールバーなどのパーツは NetBeans 内にあるので、自分で作る必要はない。プログラマはそれらのパーツにコンポーネントを記述すればよい。

* データの表現と管理

ユーザーに対してグラフなどでデータを表示し、そのデータを操作するためのツールが豊富にある。

* バージョン管理

CVS もしくは Subversion を使うことができる。

3) NetBeans の基本操作

まずプロジェクトを作成する。プロジェクトの作成は、ファイル>新規プロジェクトを選ぶことで行える。次に、開発したいプロジェクトのカテゴリを選び、そのプロジェクトがどのようなプロジェクトなのかを選択して次へボタンを押す。プロジェクト名と保存場所を尋ねられるので、それを指定する。また同時に、そのプロジェクト独自のプロパティを設定する。次に完了ボタンを押すことで、プロジェクトのひな型が出来上がる。プロジェクト作成後は、ソースを記述していく。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|---------------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-8. WideStudio の歴史、開発の背景と特徴 | |
| 対応する コースウェア | 第 5 回 (WideStudio とは何か) | |

I-18-8. WideStudio の歴史、開発の背景と特徴

OSS 統合開発環境のひとつである WideStudio について、開発の背景、歴史、コミュニティ、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、MWT (Multi-platform Widget Toolkit) などの特徴を解説する。

【学習の要点】

- * WideStudio/MWT プロジェクトは、1999 年に発足した。
- * WideStudio は MIT ライセンスである。
- * WideStudio は MWT (Multi-platform Widget Toolkit) という複数の異なるプラットフォームで動作するライブラリを利用しているため、異なるプラットフォーム間でも互換性を持つプログラムを作成することができる。

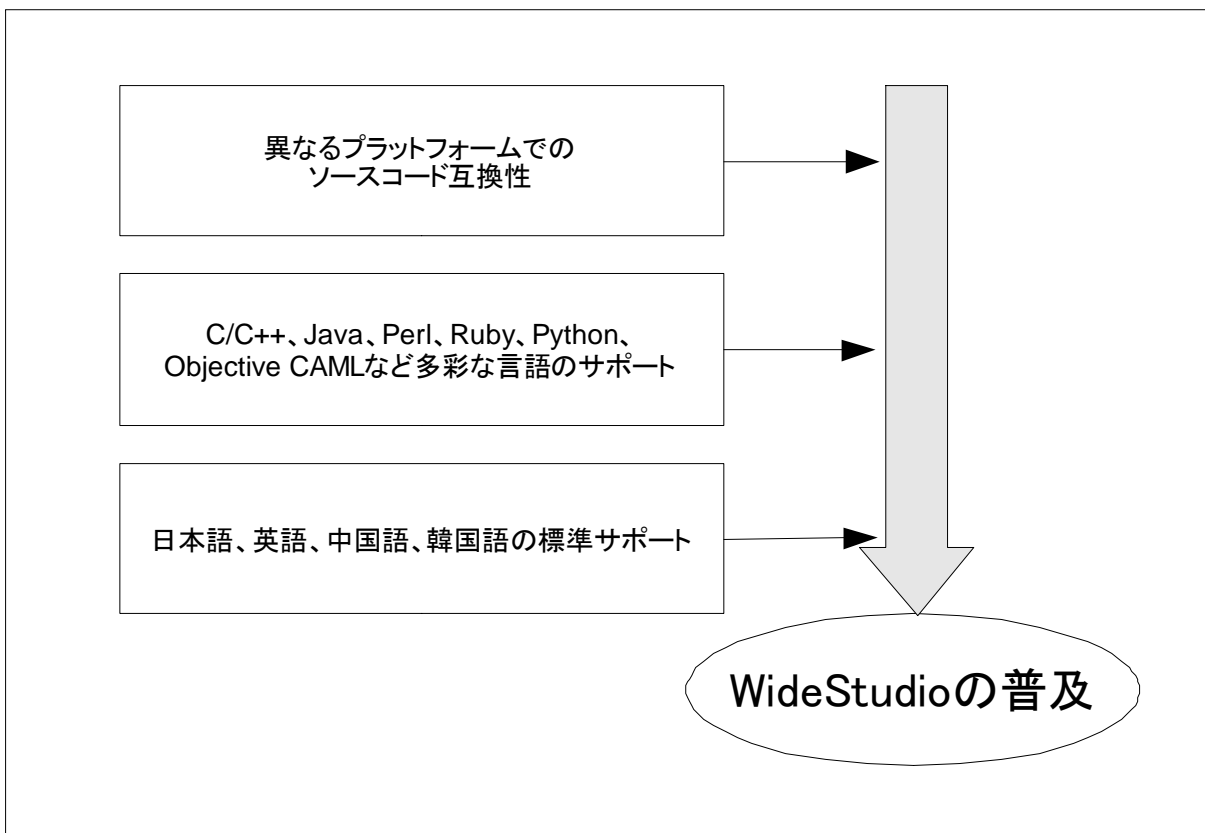


図 I-18-8. WideStudio 普及の概要

【解説】

1) WideStudio の歴史

WideStudio は 1999 年に、平林俊一氏によって開発が始まった。2000 年には α 版が公開され、同年 12 月には Turbo Linux Japan 社の開催するソフトウェアコンテストで個人部門最優秀賞を受賞した。その後も順調に開発は続き、2008 年 1 月現在ではバージョン 3.97 が公開されている。

2) WideStudio のコミュニティ

WideStudio は日本発のオープンソースソフトウェアであり、日本語のコミュニティが発達している。登録者数 1000 人を超すメーリングリストも、日本語で交わされている。

3) WideStudio のライセンス

WideStudio は MIT ライセンスを採用している。そのため、WideStudio を用いて開発したソフトウェアの配布に関してほとんど制限がない。

MIT ライセンスの詳細は、<http://www.opensource.org/licenses/mit-license.html> を参照のこと。

4) WideStudio の特徴

* MWT(Multi-platform Widget Toolkit)

WideStudio 最大の特徴は、MWT (Multi-platform Widget Toolkit)ライブラリを採用していることである。これにより、同じソースコードを再コンパイルすることで、異なる OS でもネイティブコードのアプリケーションを作成できる。OS が異なってもソースコードは同一でかまわないので、異なった OS 毎にアプリケーションを作成する手間が省ける。

* 多くのプログラム言語で開発可能

C/C++を始め、Java、Perl、Python、Ruby などを利用して開発ができる。

* マルチエンコーディング

ほとんどの場合 OS が異なるとデフォルトのテキストエンコーディングも異なり、OS 間でソースコードを共有する場合はその点に注意をする必要があった。しかし、WideStudio はマルチエンコーディングに対応しているので、ソースコードのテキストエンコーディングを気にする必要はない。

* 多彩な GUI 部品

ボタンやスクロールバーの他にも、タイマーやドロウイングエリアといった GUI オブジェクトを備えている。

* 多言語対応

日本語、英語の他にも韓国語や中国語などをサポートしている。

* エディタ

WideStudio は独自のテキストエディタを備えていないので、設定から任意のエディタを選択し、それを用いてソースコードを書く。

| スキル区分 | OSS モデルカリキュラムの科目 | レベル |
|----------------|-------------------------------|-----|
| 開発体系分野 | 18 統合開発環境に関する知識 I | 基本 |
| 習得ポイント | I-18-9. WideStudio の設定と基本機能 | |
| 対応する コースウェア | 第 10 回 (WideStudio 入門-セットアップ) | |

I-18-9. WideStudio の設定と基本機能

WideStudio を利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、作業環境の管理方法などについて具体的に説明する。また、基本機能について解説する。

【学習の要点】

- * WideStudio を入手するには、<http://www.widestudio.org/ja/index.html> を参照、それぞれのプラットフォームにあったものをダウンロードする。
- * WideStudio の作業環境の設定では、利用するエディタやヘルプの閲覧時ブラウザ、その他さまざまな項目が設定できる。



図 I-18-9. WideStudio の設定画面

【解説】

1) WideStudio のダウンロード、インストール方法

WideStudio を入手するには、<http://www.widestudio.org/ja/index.html> に行きそれぞれのプラットフォームにあったものをダウンロードする。

2008 年 1 月現在で対応しているプラットフォームは以下のものである。

- * Windows
- * BTRON
- * Linux
- * FreeBSD
- * MacOSX
- * μ CLinux
- * Solaris
- * ITRON
- * ZAURUS

インストール方法はそれぞれのプラットフォームに依存するが、Windows の場合は、普通に解凍したフォルダ内に入っている setupj.exe を実行すればインストールできる。

2) WideStudio の環境設定

WideStudio では環境設定から使いたいエディタやブラウザ、コンパイラ、実行環境などを簡単に選ぶことができる。環境設定は、オプション > 環境設定から選ぶことができる。また、それだけに限らず同様にオプションから、WideStudio のフォントや色設定なども幅広く変更することができる。

3) WideStudio の基本機能

WideStudio は独自のテキストエディタを備えていないので、WideStudio 上でやることはソースコードを書く作業ではなく、次のようなものになる。

- * プロジェクトの管理
- * アプリケーションウィンドウの作成
- * GUI ツールの配置
- * プロパティの設定
- * プロシージャの管理
- * デバッガによるバグ探し

【解説】

1) プロジェクトの作成

プロジェクトを新規作成するには、ツールバーのプロジェクト>新規プロジェクトを選ぶ。次に、プロジェクト作成ウィザードが表示されるので、英数小文字でプロジェクト名称を入力し、言語エンコーディングタイプを指定する。

2) GUI アプリケーションを作る場合

* 新規プロジェクトの種類

プロジェクト作成ウィザードで、Normal Application を選択する。

* アプリケーションウィンドウの作成

ファイル>新規ウィンドウを選択する。ウィンドウ作成ウィザードが表示されるので、ウィンドウのタイプや名称等を指定して、生成ボタンを押す。

* アプリケーションウィンドウ上へのオブジェクトの配置

表示>オブジェクトボックスを選択することで、アプリケーションウィンドウ上に配置できるオブジェクトの一覧が表示される。これをアプリケーションウィンドウ上にドラッグアンドドロップすることで、ボタンやテキストフィールドなどを配置することができる。

* イベントプロシージャの作成

プロシージャを設定したいオブジェクトを選択し、編集>プロシージャ編集>プロシージャ新規作成を選択する。プロシージャ編集ウィンドウが表示されるので、プロシージャ名称、起動トリガ、起動関数名を指定する。

* 関数の編集方法

プロシージャの編集を選ぶと、テキストエディタが立ち上がるので、そこに必要な処理を記述していく。

3) コンソールアプリケーションを作成する場合

* 新規プロジェクトの種類

プロジェクト作成ウィザードで、Console Application を選択する。

* プロジェクトの設定

プロジェクト>プロジェクトの設定を選び、リンクタブを選択して通常モードの個別ライブラリから -mwindows をはずす。

* 新規ウィンドウ

ファイル>新規ウィンドウでウィンドウを作成する。

* プロシージャの設定

新規プロシージャを選び、トリガを INITIALIZE とし、関数を作る。

* 関数の編集

先に作成したプロシージャを編集して、コードを記述する。

4) アプリケーションのビルド、実行

ファイルが保存されていることを確認して、ツールバーからビルド>ビルドオールを選択すると、アプリケーションがコンパイルされる。終了後、ビルド>実行を選択すると、作成したアプリケーションが実行される。

5) デバッグ

コンパイルエラーが起こった場合には、メッセージボックスの表示を調べる。何行目でどのようなエラーがあったのかが表示されているので、それを手掛かりにプログラムの修正をすることができる。