

Telelogic
Tau[®]

Java チュートリアル



Telelogic

Tau[®]

Java チュートリアル



本書は、Telelogic Tau バージョン 4.2、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます

© Copyright IBM Corporation 1997, 2008.

著作権表示

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

Copyright © 2008 by IBM Corporation.

IBM 特許権

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について 実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 106-8711
東京都港区六本木 3-2-12
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムと その他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、製造元に連絡してください。

Intellectual Property Dept. for Rational Software |
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

保証の不適用

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示 もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

機密情報

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

追加の法的通知が、本書で説明するライセンス付きプログラムに付随する「プログラムのご使用条件」に含まれている場合があります。

サンプル・コードの著作権

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・

プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM 対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

IBM の商標

IBM および関連の商標については、www.ibm.com/legal/copytrade.html をご覧ください。これは、IBM が現在所有する米国における商標の最新リストです。以下は、International Business Machines Corporation の米国およびその他の国における商標です。

このページには、IBM が使用しているすべてのコモン・ロー商標は掲載されていません。IBM が販売している製品は多数あるため、コモン・ロー商標のうち、最も重要な商標のみを掲載しております。このページに商標が掲載されていなくても、それは IBM がその商標を使用していないということではなく、その製品が現在販売されていない、または関連する市場で、その製品が重要ではないということの意味するものではありません。

他社の商標

Adobe、Adobe ロゴ、PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows 2003、Windows XP、Windows Vista および / またはその他の Microsoft 製品は、Microsoft Corporation の米国およびその他の国における商標または登録商標です。

Pentium は、Intel Corporation の商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

目次

著作権表示	iv
はじめに	1
概要	1
チュートリアルの目的.....	1
作業を開始する前に	1
本書で使用する表記規則.....	2
Tau について	2
ツールのヒント.....	3
Tau の起動	3
Tau プロジェクトの保存.....	3
Tau の終了	4
プロジェクト ファイルとディレクトリ.....	4
モデル要素名.....	4
モデル ビュー	5
レッスン 1：Java プロジェクトの作成	7
レッスンの目的.....	7
演習 1：Java プロジェクトの作成.....	7
タスク 1a：Java プロジェクトを作成する.....	7
タスク 1b：標準モデル ビューを設定する	9
演習 2：Java パッケージの作成	9
タスク 2a：デフォルトのパッケージ名を変更する.....	9
タスク 2b：新しいパッケージを作成する	10
レッスンのまとめ.....	10
レッスン 2：ユース ケース図の作成	11
レッスンの目的.....	11
演習 1：ユース ケース図の作成	11

タスク 1a：ユース ケース図を作成する	12
タスク 1b：ユース ケース図の名前を変更する	12
タスク 1c：アクターとユース ケースを追加する	12
タスク 1d：関連を追加する	13
レッスンのまとめ	13
レッスン 3：アクティビティ図の作成	15
レッスンの目的	15
演習 1：アクティビティ図の作成	16
タスク 1a：UML 設定を構成する	16
タスク 1b：アクティビティ図を作成する	16
タスク 1c：アクティビティ ノードを描画する	17
レッスンのまとめ	17
レッスン 4：クラス図の作成	19
演習 1：クラス図の作成	19
タスク 1a：クラス図を作成する	19
タスク 1b：クラスを描画する	19
タスク 1c：属性と操作を追加する	20
レッスンのまとめ	20
レッスン 5：Java コードの生成	21
このレッスンの目的	21
演習 1：Java コードの生成と編集	21
タスク 1a：Java パッケージをエクスポートする	21
タスク 1b：compute 操作のコードを追加する	22
タスク 1c：更新したモデルを表示する	23
タスク 1d：Java コードをコンパイルする	24
タスク 1e：アプリケーションを実行する	24
レッスンのまとめ	24
まとめ	25
技術サポートおよびドキュメント	27
サポートへのお問い合わせ	27
製品サポート	27
その他の情報	27

索引29

はじめに

概要

このチュートリアルでは、Java コーディング環境で Tau 製品を使用する基本的な方法について学習します。また、要求分析およびプロジェクト実装の概念についてもご紹介します。このチュートリアルでは、ある都市に在住する高齢者の人口増加予測を算出するシンプルなアプリケーションをモデリングします。人口増加予測を行うことで、都市計画担当者は新築する高齢者センターの適正規模を決めることができます。このチュートリアルで使用する計算例は、フィボナッチ数列に基づいています。フィボナッチ数列の詳細については、http://en.wikipedia.org/wiki/Fibonacci_number（日本語は <http://ja.wikipedia.org/wiki/フィボナッチ数>）を参照してください。

チュートリアルの目的

このチュートリアルを終了すると、以下の作業ができるようになります。

- ◆ Tau インターフェイスの使用
- ◆ Java 実装プロジェクトの作成
- ◆ ユース ケース図の作成
- ◆ アクティビティ図の作成
- ◆ クラス図の作成
- ◆ Java コードの生成と編集
- ◆ Java アプリケーションのコンパイルと実行


作業を開始する前に

このチュートリアルのレッスンを行うためには、Sun Microsystems 社の J2SE 5.0 Development Kit をインストールする必要があります。このキットは http://java.sun.com/javase/downloads/index_jdk5.jsp から入手できます。

注記：開発キットをインストールした後で、システム環境変数 PATH を、インストールフォルダ内の bin ディレクトリの **javac.exe** を指すように正しく設定してください。

本書で使用する表記規則

本書では、以下の表記規則を使用します。

- ◆ **太字**は文中で強調する文字列に使用します。例：
 - <プロジェクト名>.ttp という名前のプロジェクトファイル。
- ◆ Courier フォントはパス名、システムメッセージ、表示どおりに入力する文字列を表します。例：
 - 出力ウィンドウに Animation session terminated というメッセージが表示されます。
 - [プロジェクト名] ボックスで、<project name> と入力してデフォルトのプロジェクト名を変更します。
 - 関数名 show を入力して、Enter キーを押します。
- ◆ メニュー選択などの GUI オブジェクトおよびコントロールの名前には [] を使用します。例：
 - [デフォルトのモデル ビュー] ドロップダウン リストボックスから、[Java View] を選択します。
 - [描画] ツールバーの [アクティビティ フロー終了] シンボル  をクリックします。
 - Tau ブラウザが表示されない場合は、[表示] > [ブラウザ] を選択します。
- ◆ 説明で初めて触れる概念については、「」で表します。

Tau について

Telelogic Tau は、標準に準拠した Model Driven Development™ (MDD™ : モデル駆動型開発) 機能を提供して、企業 IT アプリケーション向けの複雑なシステムや堅牢なソフトウェアの設計をサポートします。Tau が提供する要件駆動型の反復的アプローチと、幅広いエラーチェック機能、自動化されたシミュレーション機能によって、開発者の生産性は、初期段階の要件分析から最終段階の文書化およびシステム展開までにわたって大きく向上します。

Tau には、開発者が使いやすいツール環境で主な実現技術を駆使できるように、さまざまな機能が用意されています。Tau を使うことで、システム/ソフトウェア開発、テスト実施についてシームレスで効率的な環境が整います。具体的には、以下の作業を実行できます。

- ◆ 分析：システム要件の定義、分析、検証を行うことができます。
- ◆ 設計：アーキテクチャの仕様決定と設計を行うことができます。

- ◆ 実装：Tau を使ってコードの自動生成、ビルド、実行ができます。

ツールのヒント

Tau には、プロジェクトでの作業を効率化できる機能があります。本書では、この機能が使用できる場所では、必ずその方法を紹介します。

キーボード ショートカット

キーボードショートカット (**Alt + X**、**Ctrl + H** など) を使用して、Tau インターフェイス内を簡単に移動できます。使用できるショートカットの一覧は、オンラインヘルプのトピック「エディタのショートカット」を参照してください。

自動配置

Tau の自動配置機能を使用して、描画エリアに一連の要素を簡単に追加できます。この機能は、アクティビティ図を描画するときに便利です。この機能の詳細については、オンラインヘルプのトピック「シンボルの追加」を参照してください。

Tau の起動

Windows

Windows で Tau を起動するには、[スタート] > [すべてのプログラム] > [Telelogic] > [Telelogic Tau <バージョン番号>] を選択します。

Linux および Solaris

Linux および Solaris (日本語版ではサポートされません) で Tau を起動するには、以下のコマンドを入力します。

```
<インストールパス>/bin/tau
```

Tau プロジェクトの保存



Tau のプロジェクトを保存するには、メインメニューバーから、[ファイル] > [すべて保存] を選択します。プロジェクトの変更を自動保存するように設定するには、以下の手順を行います。

1. メインメニューバーから、[ツール] > [オプション] を選択します。

2. [保存] タブの [自動バックアップ] パネルで [有効にする] チェックボックスを選択し、自動保存の間隔を指定します (例: 5 分ごと)。
3. [OK] をクリックします。

Tau の終了

Tau を終了するには、以下の手順を行います。

1. 作業内容を保存します。以下のいずれかの手順を行います。
 - ◆ **Ctrl + S** キーを押す。
 - ◆ [保存] ボタン  をクリックして、作業内容を保存する。
2. [ファイル] > [終了] を選択するか、[閉じる] ボタン  をクリックします。

注記

ワークスペースの要素の左側に赤いバーが付いている場合は、Tau を終了する前に作業結果を保存する必要があります。

プロジェクト ファイルとディレクトリ

Tau では、プロジェクトディレクトリに以下のファイルとサブディレクトリが作成されます。

- ◆ プロジェクトファイル - <プロジェクト名>.ttp。
- ◆ ワークスペース - <プロジェクト名>.ttw。ワークスペースではプロジェクトを作成し、その作業を行うことができます。
- ◆ モデルファイル - <ファイル名>.u2。UML ダイアグラム、パッケージ、コード生成の構成など、プロジェクトのユニット ファイルが含まれます。

モデル要素名

モデル要素の名前は、以下の規則に従う必要があります。

- ◆ クラス名は、「System」のように大文字で始める。
- ◆ 操作とメソッドは、「restartSystem」のように小文字で始める。
- ◆ 単語が連結されている場合は二番目以降の単語の先頭文字を大文字にする。

モデル ビュー

Tau には 1 つのモデルをさまざまな面から見て構築するためのビュー機能があります。各ビューがそれぞれ異なる角度から見たモデルの特性を表現しています。Tau では、いくつかのビューでモデルを構築できます。それぞれのビューでは、モデルの抽象的な特性を表します。モデルビューの詳細については、オンラインヘルプのトピック「ビュー」を参照してください。

レッスン 1：Java プロジェクトの作成

Tau で作成した Java プロジェクトには、UML ダイアグラム、Java パッケージ、ライブラリ、アドイン、があります。さらに、Java のバージョンに応じたコード生成用の構成データも含まれます。Tau は、指定された場所に、「プロジェクト ファイル」を含むディレクトリを作成します。新規プロジェクト用に指定した名前は、プロジェクト ファイルやディレクトリにも使用され、Tau ブラウザのプロジェクト階層の最上位に表示されます。

レッスンの目的

このレッスンでは、Tau で新しい Java プロジェクトを作成し、プロジェクト ワークスペースを構成し、このチュートリアル の演習に必要な Java パッケージを作成します。以下の概念について学習します。

- ◆ プロジェクトの構成設定
- ◆ プロジェクトのワークスペース
- ◆ プロジェクトのディレクトリ

演習 1：Java プロジェクトの作成

この演習では、Tau で新しい Java プロジェクトとワークスペースを作成します。

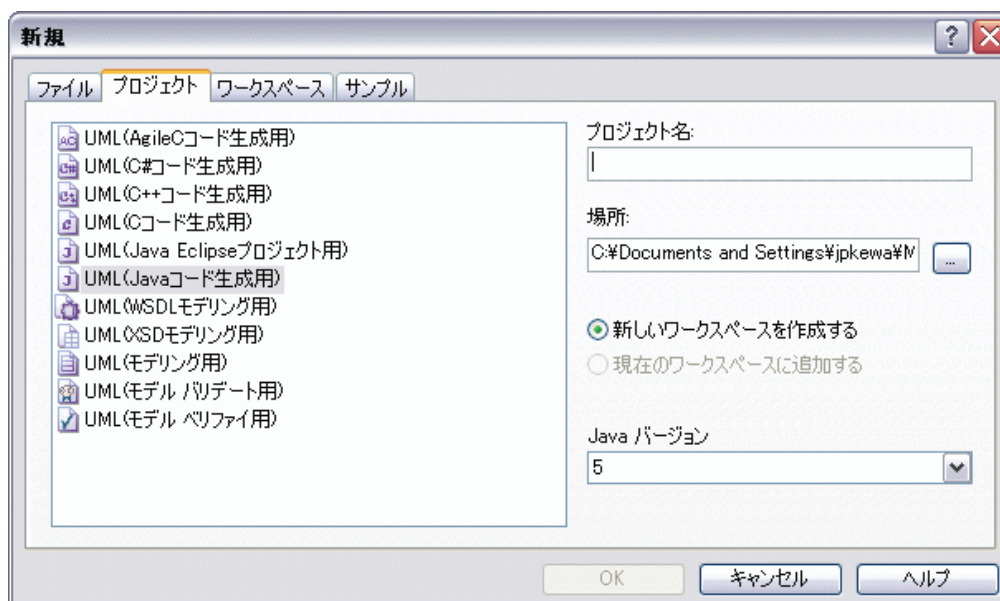
タスク 1a：Java プロジェクトを作成する

Java プロジェクトを作成するには、以下の手順を行います。

1. [スタート] > [すべてのプログラム] > [Telelogic] > [Telelogic Tau 4.0] を選択します。
2. 以下のいずれかの手順を行います。
 - ◆ **Ctrl + N** キーを押す。
 - ◆ メイン ツールバーから、[ファイル] > [新規] を選択する。

レッスン 1 : Java プロジェクトの作成

- ◆ Tau ウェルカムページで、[新規プロジェクト] パネルにある [進む] をクリックする。
- 3. [新規] ダイアログボックスの [プロジェクト] タブで、[UML(Java コード生成用)] を選択します。
- 4. [プロジェクト名] ボックスに、「FibonacciNumber」と入力します。
- 5. [場所] ボックスに、新しいディレクトリ名を入力するか、既存のディレクトリを参照します。
- 6. デフォルト設定の [新しいワークスペースを作成する] オプションを受け入れます。
- 7. [Java バージョン] ドロップダウンリストボックスから、バージョン「5」を選択します。下図に、[新規] ダイアログボックスを示します。



- 8. [OK] をクリックします。
- 9. [Developer ウィザード] のデフォルト設定を受け入れ、[次へ] をクリックし、[完了] をクリックします。新しいプロジェクトとワークスペースが作成され、出力ウィンドウに、以下のメッセージが表示されます。

Java サポートのロード中…

完了

注記：プロジェクトの作成と必要な Java ファイルのロードに要する時間は、メモリー使用量や CPU の速度によって異なります。通常は、このプロセスに数秒かかります。

タスク 1b : 標準モデル ビューを設定する

このチュートリアルでの演習を完了するには、標準モデル ビューでプロジェクトの作業を行います。標準モデル ビューに切り替えるには、以下の手順を行います。

1. Tau ブラウザで、**.ttp** プロジェクト ファイルを選択します。
2. 以下のいずれかの手順を行います。
 - ◆ **Alt + 4** キーを押す。
 - ◆ メニューバーから、[表示] > [モデル ビューの再構成] を選択する。
3. [ビューの再構成] ダイアログボックスで、[Standard View] を選択します。
4. [OK] をクリックします。

演習 2 : Java パッケージの作成

この演習では、ワークスペースに 2 つの Java パッケージを作成します。最初のパッケージは、アプリケーション全体の概要を伝える要求分析の結果を表現したダイアグラム用です。これらのダイアグラムは、アプリケーションの全体的な概要を示します。2 つ目のパッケージはクラス図用です。これから行うレッスンでは、分析パッケージ内にユースケース図とアクティビティ図を作成し、クラス パッケージ内にクラス図を作成します。チュートリアルの後半では、モデルから Java ファイルを生成するときにクラス パッケージを「エクスポート」します。

注記 : デフォルトでは、プロジェクトを作成すると、プロジェクトと同名の Java パッケージが 1 つプロジェクトに追加されます。このレッスンの最初のタスクで、プロジェクトに含まれているこのパッケージの名前を変更し、次のタスクで、2 つ目の Java パッケージを作成します。

タスク 2a : デフォルトのパッケージ名を変更する

Java パッケージ名を変更するには、以下の手順を行います。

1. [モデルビュー] で、「FibonacciNumber」パッケージを選択します。
2. **F2** キーを押します。
3. 「Analysis」と入力します。

タスク 2b : 新しいパッケージを作成する

新しい Java パッケージを作成するには、以下の手順を行います。

1. [モデル ビュー] で、[Model] を右クリックして、[新規] > [パッケージ] を選択します。
2. [モデル ルート要素の作成] ダイアログボックスで、[要素名] フィールドに「Implementation」と入力します。[プレゼンテーションの作成] ダイアログボックスが表示されます。
3. [キャンセル] をクリックして、[プレゼンテーションの作成] ダイアログボックスを閉じます。このチュートリアルでは、パッケージ図を作成する必要はありません。
4. メイン メニューバーから、[ファイル] > [すべて保存] を選択します。

レッスンのまとめ

このレッスンでは、プロジェクトを作成し、プロジェクトに Java パッケージを追加しました。次のレッスンに進み、ユース ケース図を作成してプロジェクトの作業を開始します。

レッスン 2：ユース ケース図の作成

「ユース ケース図」は、あるシステムの振る舞いと機能を、外部ユーザーとのやりとりとともに表現するダイアグラムです。ユース ケース図はシステムの動作とその利用者也表現します。

レッスンの目的

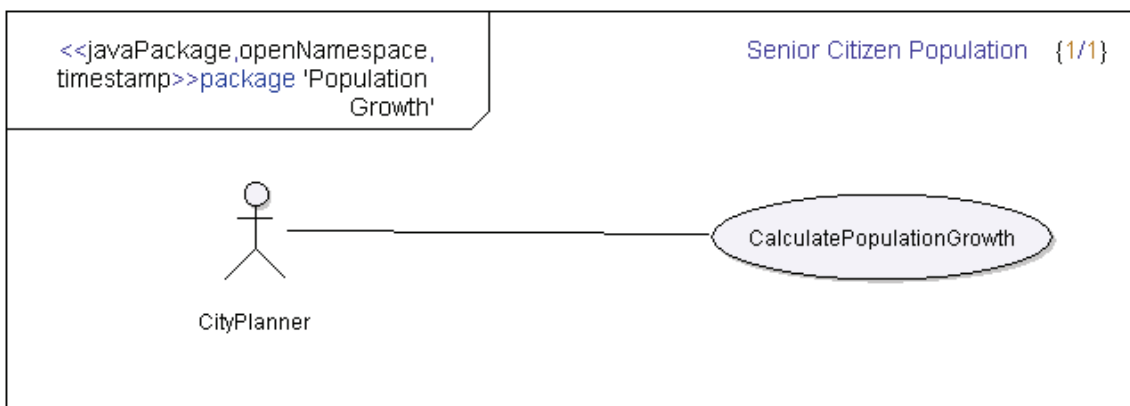
このレッスンでは、モデリング対象のアプリケーション用に、シンプルなユース ケース図を作成します。このレッスンでの作業を通して、ユース ケース図の描画方法のほか、以下の要素についても学びます。

- ◆ アクター
- ◆ ユース ケース
- ◆ 関連

演習 1：ユース ケース図の作成

下図に、この演習で作成するユース ケース図を示します。

ユース ケース図




タスク 1a：ユース ケース図を作成する

ユース ケース図を作成するには、以下の手順を行います。

1. 「Analysis」を右クリックします。
2. [新規] > [ユース ケース図] を選択します。ワークスペースにユース ケース図が作成され、描画エリアに表示されます。

タスク 1b：ユース ケース図の名前を変更する

このタスクでは、プロパティ エディタを使用して、ユース ケース図の名前を変更します。以下の手順を行います。



1. 「ユースケース図 1」を右クリックし、[プロパティ] を選択します。
2. [プロパティの編集] ダイアログボックスで、[名前] フィールドに設定されているデフォルト名を「Senior Citizen Population」に変更します。
3. [閉じる] ボタン  をクリックして、[プロパティの編集] ダイアログボックスを閉じます。

タスク 1c：アクターとユース ケースを追加する

このタスクでは、ダイアグラムにアクターとユース ケースを追加します。アクターは、システムとやりとりする外部の要素です。ユース ケースは、システムの機能を説明し、ユーザーがシステムとやりとりする目的を示します。

このダイアグラムのアクターは、ある都市に居住する高齢者の人口増加予測データを収集する目的でシステムを使用する都市計画担当者です。ユース ケースは、フィボナッチ アルゴリズムを使用して人口増加予測を算出するシステムを表現します。このデータに基づいて、担当者は新築予定の高齢者センターの適正な規模を決定できます。

ダイアグラムにアクターとユース ケースを追加するには、以下の手順を行います。

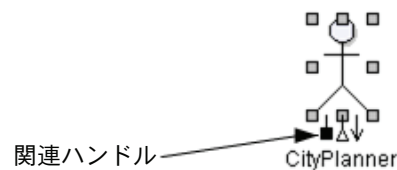
1. [描画] ツールバーの [アクター] シンボル  をクリックし、次に描画エリアをクリックします。描画エリアにアクター要素が作成されます。
2. デフォルト名を「CityPlanner」に変更します。
3. [描画] ツールバーの [ユース ケース] シンボル  をクリックします。描画エリアにユース ケース要素が作成されます。
4. デフォルト名を「CalculatePopulationGrowth」に変更します。

タスク 1d：関連を追加する

関連ラインは、ユース ケース図の2つの要素間の関係を示します。このタスクでは、都市計画担当者とこのアプリケーションのユース ケースの間のやりとりを示す関連ラインを描画します。
[描画] ツールバーの [関連] シンボルを使用するか、「CityPlanner」要素の関連「ハンドル」を選択して、関連ラインを追加できます。


ハンドルを使用して関連ラインを描画するには

1. 描画エリアで、「CityPlanner」を選択します。
2. 下図に示すように、「CityPlanner」要素の下にある関連「ハンドル」をクリックします。



3. 「CalculatePopulationGrowth」ユース ケース要素内の任意の場所をクリックします。2つの要素を接続する関連ラインが作成されます。

[関連] シンボルを使用して関連ラインを描画するには

1. [描画] ツールバーの [関連] シンボル  をクリックします。
2. 「CityPlanner」の右端と「ComputePopulationGrowth」の左端をクリックします。2つの要素を接続する関連ラインが作成されます。
3. メインメニューバーから、[ファイル] > [すべて保存] を選択します。

描画結果は、[ユース ケース図](#)に示すようになっているはずです。

レッスンのまとめ

このレッスンでは、ユース ケース図を作成しました。また、ユース ケース図の以下の要素を理解しました。

- ◆ アクター
- ◆ ユース ケース
- ◆ 関連

次のレッスンに進み、アクティビティ図を作成します。

レッスン 3：アクティビティ図の作成

アクティビティ図は、一連のアクティビティの表現から特定のタイプの振る舞いを導き出すために使用します。アクティビティ図は、矢印で相互に接続された複数のアクティビティ項目で構成されます。矢印は、ダイアグラム内のアクティビティフローの方向を示すために使用されます。

レッスンの目的

このレッスンでは、アクティビティ図を作成します。このアクティビティ図が表すのは、都市計画担当者がアプリケーションを使用して将来の高齢者人口増加を算出する場合に発生する一連のアクティビティです。担当者は、アプリケーションを起動し、人口増加の算出に必要なデータを入力します。この入力データは、将来に向かっての経過年数です。アプリケーションは、入力した年数データに基づいた人口増加の予測値を返します。

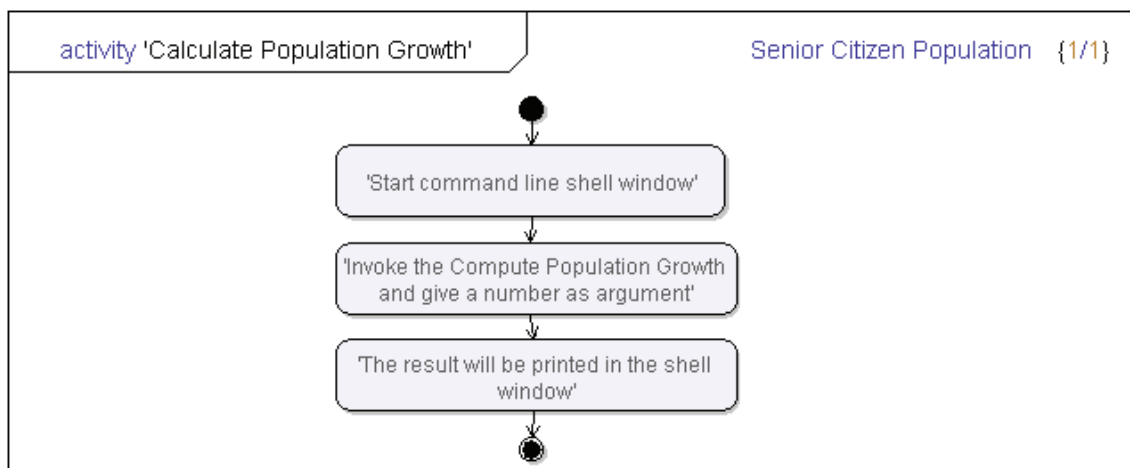
Tau を使用してアクティビティ図を作成し、以下の要素を描画する方法を学習します。

- ◆ 初期ノード
- ◆ アクションノード
- ◆ アクティビティライン
- ◆ 終了ノード

演習 1：アクティビティ図の作成

下図に、この演習で作成するアクティビティ図を示します。

アクティビティ図



タスク 1a：UML 設定を構成する

このタスクでは、上図に示すような[アクティビティ図](#)の要素を描画できるように、UML 編集の設定を構成します。デフォルトで、Tau はアクティビティ図の要素を水平に描画するよう構成されています。上図のように要素を描画できるようにするため、以下の手順に従って設定を垂直方向に変更します。

1. メニューバーから、[ツール] > [オプション] > [UML] を選択し、[UML 詳細編集] タブを選択します。
2. [UML 詳細編集] タブの [アクティビティ図] パネルで、[自動作成の方向] ドロップダウンリストボックスから [垂直] を選択します。
3. [OK] をクリックします。

タスク 1b：アクティビティ図を作成する

アクティビティ図を作成するには、以下の手順を行います。




1. Tau ブラウザで、[Model]、「Analysis」パッケージの順で展開します。
2. [新規] > [アクティビティ図] を選択します。ワークスペースにアクティビティ図が作成され、描画エリアに表示されます。

3. [プロパティの編集] ダイアログボックスで、[名前] フィールドに設定されているデフォルト名を「Senior Citizen Population」に変更します。
4. [保存] ボタンをクリックして、作業内容を保存します。

タスク 1c：アクティビティ ノードを描画する

ノードは、アクティビティフロー内の振る舞いの特定の単位を表現します。このタスクでは、ダイアグラム内に、初期ノード、3つのアクションノード、および終了アクティビティノードを描画します。ノードは、担当者がアプリケーションを起動して、人口増加予測のための年数データを入力したときに発生する振る舞いの一単位を表現しています。

アクティビティノードを描画するには、以下の手順を行います。

1. [描画] ツールバーの [初期ノード] シンボルをクリックします。描画エリアに初期ノード要素が作成されます。
2. [初期ノード] を選択している状態で、**Shift + スペースバー**を押して、ポップアップメニューの [アクティビティ/アクション] シンボルを3回クリックすると、3つのアクションノードが追加できます。アクションノードを追加するたびに、ノード間に「アクティビティフロー矢印」が自動作成されます。矢印はダイアグラム内のフローの方向を示します。
3. 続けて **Shift + スペースバー**を押し、[描画] ツールバーの [アクティビティ終了] シンボルをクリックします。
4. 各アクティビティノードをクリックし、[アクティビティ図](#)に示すように名前を入力します。
5. メインメニューバーから、[ファイル] > [すべて保存] を選択します。

これで、アクティビティ図の描画が完了しました。描画結果は、[アクティビティ図](#)に示すようになっているはずです。

レッスンのまとめ

このレッスンでは、アクティビティ図を作成しました。また、アクティビティ図の以下の要素について学習しました。

- ◆ 初期ノード
- ◆ アクションノード
- ◆ 終了ノード

次のレッスンに進み、クラス図を作成します。

レッスン 4：クラス図の作成


クラス図はシステム内の要素のタイプ、および要素間のやりとりと関係を示します。クラスの関係は通常、依存ラインまたは関連ラインで示されます。

演習 1：クラス図の作成

この演習では、クラス図を作成し、クラス図内にクラスを描画します。クラスにはモデリング対象のアプリケーションについての操作と属性が含まれます。


タスク 1a：クラス図を作成する

クラス図を作成するには、以下の手順を行います。

1. Tau ブラウザで、[Model] を展開します。
2. 「Implementation」パッケージを右クリックして、[新規] > [クラス図] を選択します。ワークスペースにクラス図が作成されます。
3. [プロパティの編集] ダイアログボックスで、[名前] フィールドに設定されているデフォルト名を「Calculate」に変更します。
4. [閉じる] ボタン  をクリックして、[プロパティの編集] ダイアログボックスを閉じます。

タスク 1b：クラスを描画する

クラスを描画するには、以下の手順を行います。

1. [描画] ツールバーの [クラス] シンボル  をクリックし、描画エリアの任意の場所をクリックします。ダイアグラムにクラスが追加されます。
2. クラスのデフォルト名を「Fib」に変更します。

タスク 1c：属性と操作を追加する

このタスクでは、前のタスクで作成したクラスに属性と操作を追加します。ここで追加する「属性」は、アプリケーションのはたらきで算出される結果の値を保持します。「操作」は、数値を算出する動作を表します。

「Fib」クラスに属性と操作を追加するには、以下の手順を行います。

1. 「Fib」クラスを選択します。
2. クラスボックスの中央のパネルにカーソルを置き、属性のテキストボックスに「`result:int`」と入力します。
3. クラスボックスの下部のパネルにカーソルを置き、操作のテキストボックスに「`compute(n:int):int`」と入力します。
4. メインメニューバーから、[ファイル] > [すべて保存] を選択します。

レッスンのまとめ

このレッスンでは、クラス図を作成しました。また、クラス図の以下の要素について学習しました。

- ◆ 属性
- ◆ 操作

次のレッスンに進み、「Fib」クラスから Java コードを生成します。

レッスン 5：Java コードの生成

このレッスンの目的

このレッスンでは、ここまで作成したモデルからソースコードを生成し、そのソースコードを編集します。以下の操作を学習します。

- ◆ Java コードの生成
- ◆ コードの手動追加
- ◆ Java アプリケーションの実行

演習 1：Java コードの生成と編集

この演習では、前のレッスンで作成した「Fib」要素から Java コードを生成します。その後、生成したコードを「compute」操作に手動で追加します。Java コードを生成するには、生成先として指定するディレクトリにパッケージをエクスポートします。Java コードを含むファイルは、このディレクトリに作成されます。

タスク 1a：Java パッケージをエクスポートする

このタスクでは、「Implementation」パッケージを Java のソースコードにエクスポートします。以下の手順を行います。

1. Tau ブラウザで「Implementation」を選択します。
2. メニューバーから、[Java] > [エクスポート] > [パッケージ] を選択します。
3. [フォルダの参照] ダイアログボックスで、以下のいずれかの手順を行います。
 - ◆ [OK] をクリックして、プロジェクトディレクトリにデフォルトフォルダ「Implementation」を作成する。
 - ◆ スクロールバーを使用して別のディレクトリを指定し、[新しいフォルダの作成] をクリックする。

「Implementation」パッケージから生成される Java コードを含む「Fib.java」という名前のファイルが作成されます。

タスク 1b : compute 操作のコードを追加する

このタスクでは、[レッスン 4 : クラス図の作成](#)で作成した compute 操作の本体であるコードを追加します。compute 操作のコードを手動で追加するには、以下の手順を行います。

1. Tau ブラウザで、「Calculate」ダイアグラムをダブルクリックして開きます。
2. 描画エリアで、「Fib」クラスを右クリックして、[ソースコードの編集] を選択します。
[Fib.java] タブの下にエディタが開きます。

3. 以下の行を削除します。

```
int result;  
int compute( int n);
```

4. 削除した行に、以下のコードを挿入します。

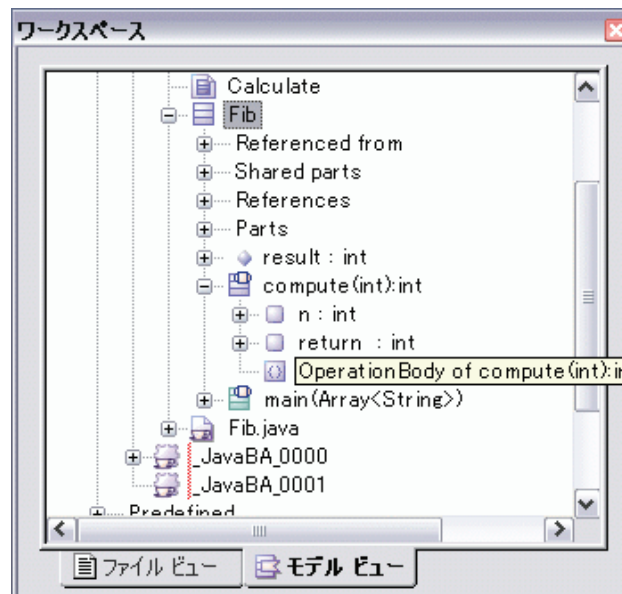
```
int result;  
int compute( int n ){  
    if(n == 0)  
        result = 0;  
    else if(n == 1)  
        result = 1;  
else  
    result = compute(n - 1) + compute(n - 2);  
return result;  
}  
public static void main(String[] argv)  
{  
    Fib myFib = new Fib();  
    myFib.compute( new Integer(argv [0]).intValue() );  
    System.out.println( "The result is " + myFib.result );  
}  
}
```


タスク 1c : 更新したモデルを表示する

このタスクでは、前のタスクでコードを追加して更新したモデルを表示します。デフォルトで、Tau はソースコードの編集時にモデルを自動更新するように構成されています。作業内容を保存すると、Tau はプロジェクト内の更新されたソースファイルを検出し、モデルを更新します。

前のタスクで行った更新内容を表示するには、以下の手順を行います。

1. Tau ブラウザで、[モデル ビュー] タブを選択します。
2. 「Implementation」パッケージディレクトリで「Fib」を展開します。下図に示すように、`compute(int):int` 操作に `operationBody` 要素が追加されています。`operationBody` は、前のタスクで追加したコードです。



タスク 1d : Java コードをコンパイルする

このタスクでは、ソースコードをコンパイルします。コードをコンパイルするには、以下の手順を行います。

1. Tau ブラウザで、「Fib」クラスを右クリックして、[コンパイル] を選択します。
2. 出力ウィンドウの [スクリプト] タブに、`javac` コンパイル コマンドが正しく表示されることを確認します。

注記

[スクリプト] タブに出力される表示内容は、オペレーティング システムによって異なります。

タスク 1e : アプリケーションを実行する

アプリケーションは、前のタスクで追加したコードによって実装された人口増加予測の値を算出するアルゴリズムを実行します。アプリケーションを実行するには、以下の手順を行います。

1. コマンド ウィンドウを開き、プロジェクト ディレクトリに移動します。
2. プロジェクト ディレクトリで、以下のコマンドを入力します。

```
java Implementation/Fib [n]
```

ここでは、[n] はこれからの経過年数を表す整数です。

3. **Enter** キーを押します。この結果は、指定した年数後に予測されるこの都市の高齢者数です。たとえば、「22」と入力した場合、22 年後にその都市に住む高齢者数はおよそ 254 人と予測されます。

レッスンのまとめ

このレッスンでは、Java コードを生成し、手動でコードを追加して、アプリケーションを実行しました。以下の操作を学習しました。

- ◆ Java パッケージのエクスポート
- ◆ ソースコードの編集
- ◆ モデルのテスト

まとめ

このチュートリアルでは、Java 環境で Telelogic Tau を使用し、UML モデリングを行う手法について紹介しました。チュートリアルで演習を行うことで、Tau 製品を理解していただけだと思います。以下の操作方法について学習してきました。

- ◆ プロジェクトの作成
- ◆ 描画ツールとショートカット キーの使用
- ◆ ダイアグラムの描画
- ◆ Java パッケージのエクスポート
- ◆ コードのコンパイル
- ◆ モデルへの外部コードの追加

これらのタスクの実行方法を学習したことで、Telelogic Tau の基礎知識を得ることができました。これから本製品を使用して UML モデリングプロジェクトの作業を継続していけば、さらにスキルを高め、製品に対する知識を深めることができるでしょう。

技術サポートおよびドキュメント

サポートへのお問い合わせ

Telelogic 製品のサポートと情報は、Telelogic サポートサイトから IBM Rational Software Support に移行中です。この移行期間中は、サポートの連絡先がお客様によって異なります。

製品サポート

- 2008 年 11 月 1 日より前に Telelogic 製品を取引されたお客様は、Tau サポート ウェブサイトをアクセスしてください。製品情報の移行後に、IBM Rational Software Support site に自動で転送されます。
- 2008 年 11 月 1 日より前に Telelogic 製品のライセンスをお持ちではなかった新規のお客様は、[IBM Rational Software Support site](#) をアクセスしてください。

お客様サポートにお問い合わせいただく前に、問題を説明するために必要な情報をご用意ください。IBM ソフトウェアサポート担当員に問題を説明する際には、担当員が迅速に問題を解決できるように、問題の具体的な内容と必要な背景情報をすべて伝えてください。あらかじめ以下の情報をご用意ください。

- ◆ 問題発生時に使用していたソフトウェアとそのバージョン
- ◆ 問題に関連したログ、トレース、メッセージなど
- ◆ 問題を再現できるかどうか。再現できる場合はその手順
- ◆ 回避策があるかどうか。ある場合は、その回避策の内容

その他の情報

Rational ソフトウェア製品、ニュース、イベント、その他の情報については、[IBM Rational Software Web site](#) をご覧ください。

下表に、このチュートリアルで取り上げている重要な概念について、さらに詳しく説明しているオンラインヘルプのトピックの一覧を示します。

ヘルプ トピック	参照情報
UML と Java	Java 環境における、Tau を使用した UML モデリングについての全般的な情報。
ダイアグラムの操作	ダイアグラムの作成、保存、出力、その他の一般的なダイアグラム操作に関する情報。
UML 言語ガイド	UML 言語構成要素およびモデル要素をすべて網羅した概説。

索引

J

Java 言語
コードの編集 21
生成コード 21
パッケージ 9

L

Linux 3

S

Solaris
Tau の起動 3

T

Tau 2
技術サポート 27
起動 3
終了 4
ドキュメント 27
閉じる 4
プロジェクト 4

W

Windows
Tau の起動 3

あ

アクティビティ図
作成 16

え

エクスポート
Java パッケージ 21

か

カスタマ サポート 27

き

技術サポート 27
起動
Linux Tau 3
Tau 3
Windows Tau 3

く

クラス図
作成 19

し

終了
Tau 4

そ

操作 20

た

ダイアグラム
アクティビティ 16
クラス 19
ユース ケース 12

つ

追加
アクター 12
関連ライン 13
コード 22
操作 20
属性 20
ツールのヒント 3

て

ディレクトリ 4

と

ドキュメント 27

閉じる
Tau 4

な

名前
要素 4
名前の変更 9, 12

は

パッケージ
作成 9
名前の変更 9

ひ

標準モデル ビュー
構成 9
開く
Tau 3

ふ

ファイル 4
プロジェクト 4

作成 7
ディレクトリ 7
ファイル 7
保存 3

ほ

本書で使用する
表記規則 2

も

モデル
ビュー 5
要素名 4

ゆ

ユース ケース図 11
作成 12

よ

要素
名前 4