

フルオンライン大学における プログラミング系科目の演習環境

中谷 祐介¹

1. はじめに

サイバー大学（以下、本学）は、すべての授業をインターネットを通じた e ラーニングにより学生に提供する「フルオンライン大学」である。

本学では、学生に対し、ビジネスと IT エンジニアリングの両方の基礎知識を修得するカリキュラムを提供している。IT エンジニアリングに関する科目にはプログラミングに関する演習科目があり、これらの演習科目も e ラーニングによる指導を行っている^{1) 2)}。学生は、このようなプログラミング系の演習科目を受講するとき、各自の所有するコンピュータに演習に必要なソフトウェアをインストールしなければならない。しかし、学生個々のコンピュータ操作のスキルや OS の種類が異なるため、ソフトウェアのインストールを全員同じように行うことは難しい。そのため教員は、学生の演習環境を構築するための指導に時間が費やされてしまう問題がある。そこで、学生がプログラミング系の演習科目を受講する際の演習環境を大学が提供することで、学生が受講に集中しやすい環境を提供することが望まれる。すべての学生の受講環境として統一している環境はウェブブラウザであることを考慮して、ウェブブラウザ上で動くプログラミング環境を提供するサービスを学生に提供することを考える。これらの環境を提供するサービスとして、codigm 社が開発した「Goorm IDE」（以下、goorm）がある。本学ではこの goorm を 2017 年度から導入し、プログラミング系の演習科目において演習環境として利用している。

本稿では、フルオンライン大学におけるプログラミング系科目の演習環境として goorm を導入し科目運営を行っていることを報告する。具体的には goorm の概要を述べ、goorm を利用する科目の内容および運営と goorm を導入したことによる学生の履修状況を報告し、そこから派生する運営上の課題について考察する。

2. 演習環境「goorm」

ここでは、本学が導入している goorm について、その概要と機能を述べる。

¹ サイバー大学 IT 総合学部・教授

2.1. goorm の概要

goorm は、Linux の環境をベースとしたオンラインで利用することができる統合開発環境 (IDE) である。IDE としての機能だけでなく、ユーザ間でファイルを共有する機能や、ユーザ間でチャットを行う機能などを備えている。

学生が goorm を利用するためには、ウェブブラウザ上で大学が指定するウェブサイトにアクセスし、学生ごとに配布されるログイン情報を使用して goorm のシステムにログインを行う。ログインが完了すると、学生が履修する演習科目の一覧が表示される。学生が演習を行う科目を選択すると、その科目の演習環境が起動する。起動後の演習環境の画面を図 1 に示す。

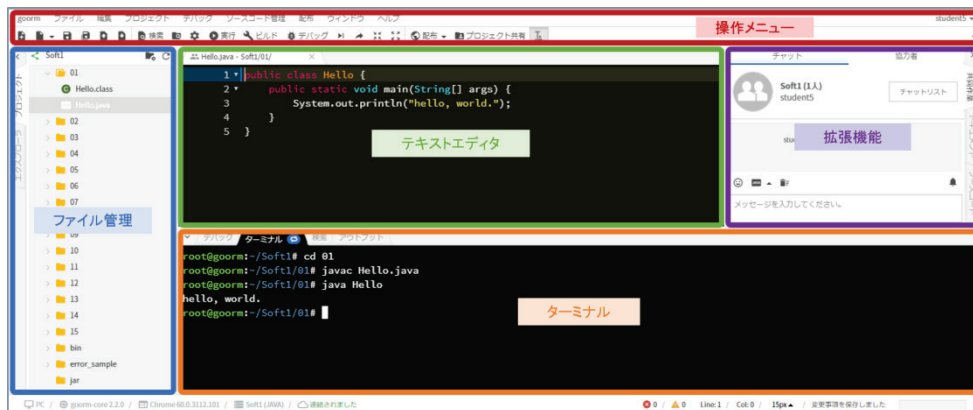


図 1 演習環境「goorm」の画面構成

2.2. goorm の機能

ここでは、図 1 をもとに、goorm の機能を解説する。

2.2.1. 操作メニュー

「操作メニュー」の部分では、演習環境の操作を行うためのメニューが提供される。演習環境の設定や、記述したソースファイルのコンパイル・実行の操作など、さまざまな操作を行うためのメニューが準備されている。goorm 上で作成したファイルをユーザのコンピュータ上にダウンロードする操作や、ユーザのコンピュータ上で作成したファイルを goorm 上にアップロードする操作も、ここから実行することができる。

2.2.2. ファイル管理

「ファイル管理」の部分では、ユーザの演習用ディレクトリの内容がグラフィカルに表示される。ディレクトリ、ファイルの作成、削除や、ファイルの移動などをマウス操作により実行することができる。

2.2.3. テキストエディタ

「テキストエディタ」の部分では、テキストエディタの機能が提供される。一般的なテキストエディタと同様に、プログラミング言語に応じたキーワードの強調表示や、括弧の対応関係の表示など、プログラミングの初心者をサポートする機能も備えている。

2.2.4. ターミナル

「ターミナル」の部分では、UNIX 環境のターミナルの機能が提供される。UNIX のオペレーションを行うために利用するほかに、コマンド操作によるソースファイルのコンパイル、実行を行うことができる。また、複数のターミナル画面が必要な場合は、テキストエディタを表示する部分にターミナル画面を表示することも可能であり、タブの切り替えによってエディタとターミナルの表示を変更することができる。

2.2.5. 拡張機能

「拡張機能」の部分では、IDE としての機能以外の拡張機能が提供される。例えば、ファイル編集の履歴を表示する機能がある。学生があるファイルを編集すると、その段階でのファイルの内容が履歴としてシステム上に保存される。教員は、学生から質問があったときに、その履歴を参照して、学生に指導を行うことができる。また、チャット機能もある。教員と学生との間でチャットを行うことで、ファイルの内容を確認しながらリアルタイムに指導を行うことができる。

3. goorm を利用したプログラミング系科目の運営

ここでは、goorm を利用したプログラミング系科目のうち、Java のプログラミングを学ぶ科目の内容および運営について述べる。また、goorm を導入したことによるその効果について報告する。

3.1. 科目内容と運営

本学では Java によりプログラミングの基本を修得する科目として「ソフトウェア開発論 I」（以下、本科目）を設置している。本科目の概要は以下の通りである。

- 科目名：ソフトウェア開発論 I
- 科目内容：
 - Java によるプログラミングの基本
 - オブジェクト指向の基本
- 対象学生：1～2 年生

本科目では、プログラミングの経験がない学生も考慮して、Java によるプログラミングを基礎から学ぶ科目の構成となっている。本科目の各回の構成を表 1 に示す。本科目は、12 回の講義回と 3 回の演習回で構成している。学期の前半（第 8 回まで）において、Java によるプログラミングの基礎を学び、後半（第 9 回以降）では、オブジェクト指向プログラミングの基礎を学ぶ。

表 1 「ソフトウェア開発論 I」の授業構成

授業回	授業内容	課題
第 1 回～第 3 回	講義	小テスト
第 4 回	演習	レポート
第 5 回～第 7 回	講義	小テスト
第 8 回	演習	レポート
第 9 回～第 11 回	講義	小テスト
第 12 回	演習	レポート
第 13 回～第 15 回	講義	小テスト
期末試験	試験	—

本学では、授業はインターネットを通じて授業コンテンツとして配信している。授業コンテンツには、ビデオを中心とした Video on Demand (VoD) 形式のコンテンツと、ウェブ形式のテキスト情報を中心とした Web Based Training (WBT) 形式のコンテンツが存在する²⁾。

本科目においては、講義回は各回 4 章の授業コンテンツで構成している。第 1 章では VoD 形式により各講義回に学ぶ概要を講義する。第 2 章～第 4 章では、WBT 形式によりコンテンツを提供している。学生は、コンテンツ内のテキストを読み進めると同時に、映像や音声による解説を視聴し学習を進める。各講義回の終わりには「小テスト」を実施し、その回における理解度を確認する。演習回は各回 2 章の WBT 形式のコンテンツで構成している。第 1 章で過去の復習を行い、第 2 章で Java によるプログラミングに関する演習問題を出題する。学生は演習問題に取り組み、結果を「レポート」として提出する。講義、演習内容に関する学生からの質問については、授業内に設置された掲示板や電子メールで受け付ける。学期末には「期末試験」を実施し、全体を通しての理解度を確認する。

本科目では、Java によるプログラミングの演習は goorm 上で行う。goorm では学生 1 人に対して 1 台の UNIX 環境が与えられる。学生は、自身の環境においてディレクトリ、ファイルの管理を行う。プログラミングの際は、goorm (図 1) 上の「ファイル管理」の部分でディレクトリ、ファイル管理を行いながら、「テキストエディタ」の部分でソースファイルの作成、編集を行う。作成したソースファイルは、「ターミナル」の部分において、コマンド入力によりコンパイルし、実行を行う。

3.2. 演習環境の導入による効果

goorm を導入したことによるその効果を検証するため、本科目の運営の面から、導入前と導入後の比較を行う。

まず、演習環境の構築についてである。本科目では、goorm を導入する前は、学生が各自の所有するコンピュータ上に、Java によるプログラミングの環境を構築していた。具体的には、ソースファイルを作成・編集するためのテキストエディタと、Java のソースファイルをコンパイル・実行するための Java SE Development Kit (JDK) をインストールし、パスの設定などを行っていた。すでに述べたように、学生のコンピュータの OS 環境は統一されていない。そのため、ダウンロードするファイル、インストール手順、設定の手順などが学生によって異なり、演習環境を構築するまでの指導に時間を要していた。goorm の導入により大学が学生に演習環境を提供することから、学生は各自で演習環境を構築する必要がなくなった。学生は、ウェブブラウザ上でシステムへのログインの操作を行うことで、すべての学生に共通の演習環境を提供できるようになった。これにより、学生は学期の初めから授業内容に集中できるようになった。担当教員も演習環境の構築のための指導時間が減り、授業運営に注力できるようになった。

次に、プログラミングの指導についてである。goorm を導入する前は、プログラミングに関する質問がある場合、学生はその都度、自身が記述したソースファイルを質問用の掲示板や電子メールに添付した上で、質問文を記述していた。担当教員は学生からの質問を受けると、添付されたソースファイルをダウンロードし内容を確認し、質問に対する回答を行っていた。goorm の導入により、担当教員は学生の演習環境を閲覧することができるようになった。そのため、学生はソースファイルを添付することなく、質問だけを行えるようになった。担当教員は、学生の演習環境を閲覧することで、該当のファイルだけでなく関連するファイルなども参照することができ、指導しやすくなった。また、goorm 上には学生が作成したファイルの履歴が保存されている。そのため、担当教員は学生が作成したソースファイルの作成過程を閲覧しながら適切な指導を行えるようになった。また、各学生の受講の進捗状況を確認することができるため、受講が遅れている学生に対しての具体的な指導も行えるようになった。

このように、goorm の導入により、本科目の運営面において学生、担当教員それぞれにメリットがあり、科目の運営の面においては一定の効果がある。

一方で、goorm を導入したことによる学生の学習効果について検証する。ここでは学習効果の指標として、本科目における学生の合格率を取り上げる。2015 年度秋学期から 2017 年度春学期までの 4 学期における本科目の合格率を集計したものを図 2 に示す。図 2 では、各学期の合格率の下に実際の合格者数を記載している。この集計では、各学期に初めて本科目を受講した学生に対しての集計を行っており、再履修学生は除いている。図 2 の結果から、goorm を導入した 2017 年度春学期の合格率は、導入前の 3 学期と比較して数ポイント改善していることがわかる。

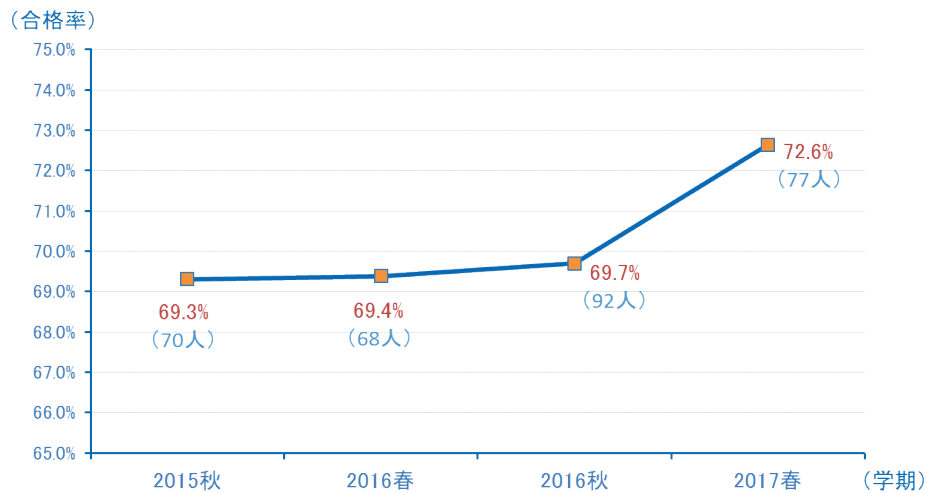


図2 「ソフトウェア開発論Ⅰ」の合格率

3.3. 今後の課題

前節において goorm を導入した効果を本科目の合格率をもとに検証したが、これは1学期分のデータのみである。今後は、数学期にわたり受講データを収集し従来の受講状況や成績と比較し、goorm を導入したことによる学習効果を継続的に調査する必要がある。また、goorm では学生ごとにシステムの利用時間を収集できるため、詳細な演習環境の利用状況を把握することができる。これらのデータを使用して、利用状況と学生の成績との関連、学生の利用状況による受講状況の把握などを行い、授業改善および学生指導の改善に利用することを検討することが課題となる。

4. おわりに

本稿では、フルオンライン大学におけるプログラミング系科目の演習環境について述べた。本学が導入した演習環境である goorm の概要を説明し、Java のプログラミング科目における goorm の導入事例を紹介した。また、goorm を導入したことによる効果について検証を行うとともに、今後の課題について述べた。

参考文献

- 1) 後藤幸功、中谷祐介、“オンライン大学のプログラミング演習科目における受講状況と成績の関係に関する考察”、情報処理学会第78回全国大会論文集、5F-01、2016.
- 2) 中谷祐介、“フルオンライン大学におけるプログラミング系演習科目の運営”、サイバー大学 eラーニング研究、第5号、pp.31-38、2016.