

Java プログラミング入門単位認定型完全 e-learning へ向けての試み ～コンテンツ構築および実践バージョン～

高岡詠子、石井和佳奈
千歳科学技術大学

千歳科学技術大学では、平成 16 年度より採択された現代的教育ニーズ取組支援プログラムの一環として、各科目教員と学生との連携によるコンテンツ作成が行われている。Java プログラミング言語入門を目的とした授業を 18 年度より単位認定型完全 e-learning 形式で行うことが決まり、来年度の授業形態を視野に入れて、17 年度春学期、試験的に e-learning システムを導入して授業を行った。開発されたシステムの詳細、コンテンツ概要と運用、および学生のアンケート結果を紹介し、今後の動向について述べる。

An Experiment for Fully E-Learning School Credit of Introduction to Java Programming Course -Development of contents and practice-

Eiko Takaoka and Wakana Ishii
Chitose Institute of Science and Technology

Abstract

We have developed e-learning contents that cover Java programming basics. We applied our system to department junior year on a trial basis this spring semester. In this paper, we introduce full details regarding our system, brief summary of contents, operational procedures, and report result of course evaluation.

1. はじめに

本論に入る前に、千歳科学技術大学の学部構成およびコンピュータ実習室環境を示す。

本学は平成 10 年 4 月に開学した私立大学であり「光科学部」という名前の学部を 1 つ持つ単科大学である。1 学年の定員数は 240 名である。2 年進学時には学生は「物質光科学科」（以下物質）「光応用システム学科」（以下システム）の 2 学科に分かれる。物質光科学科では「光」を利用する物質・材料の開発に関する知識と応用技術を学ぶ。光応用システム学科では光テクノロジーという新しい視点で、自然科学から社会科学までをカバーした幅広いシステムを扱える知識や技術を学ぶ。

本学のコンピュータ実習室は、PC 教室、LLC 教室の 2 教室が隣り合っており、各教室に 80 名を収容できる。LLC 教室は語学教室も兼ねており、学生用コンピュータの画面は切り替えスイッチで

教卓にあるコンピュータのディスプレイのイメージを学生の 80 台全体に配信できる。2 教室とも Windows XP と Linux OS のデュアル OS であり、学生のアクセス認証およびファイルシステムは統一管理されており、どちらの OS からでも学生個人のファイルシステムにアクセスできるようになっている。

2. 全学的 e-learning プロジェクト

平成 16 年度に採択された現代的教育ニーズ取組支援プログラム、テーマ 6: IT を活用した実践的遠隔教育（e-learning）における全学的 e-learning プロジェクトについて述べる。

2.1. e-learning を介した全学情報教育の共有化

本プロジェクトの柱は、高校での情報導入に伴う大学初年度教育での情報処理の能力の多様化に備え、情報実習系および情報講義系科目全ての

e-learning 化を図ることと、専門教育課程での実践的な IT スキル教育への社会的要請に応え、情報キャリアアップ科目を開設することである。具体的な内容を以下に示す。

2.1.1. 高校のカリキュラム「情報」導入に対する新課程対応

学部 1 年生対象の必修科目「情報メディア実習」に e-learning を導入する。これは、PC マエストロ[1,2,3]を補習教材として利用することを指す。ただし、PC マエストロの開発は平成 15 年度から行っており、運用も平成 16 年度より行っているため、新たな試みではない。

2.1.2. プログラミング教育(実習)に関する全学必修化

2 年生以降に学科ごとに設置されていた情報実習系および情報講義系科目を統合し、全ての科目の e-learning 化を行う。システム 2 年必修秋学期の「コンピュータプログラミング」と物質 3 年必修秋学期の「ホットニクス計算機実習 1」が統合され、全学 2 年必修春学期の「コンピュータプログラミング」として生まれ変わる。また、システム 2 年必修春学期の「情報システム実習」と物質 3 年必修春学期の「ホットニクス計算機実習 2」が統合され、全学 3 年必修春学期に「コンピュータ理工学実習」として生まれ変わる。さらに全学 1 年必修「情報技術概論」から「コンピュータプログラミング」の流れの中で、学生が、情報の基礎知識を学び、その後、C 言語プログラミングの学習へスムーズに入っていけるような授業を行えるよう検討を行う。

2.1.3. システム学科の独自性の強化:「ソフトウェア科目群」の充実

システム学科独自の科目として、システム必修「ソフトウェアデザイン」においては、高度なプログラミング実習を 2 年秋、3 年春のどちらでも履修できるようにし、e-learning による単位認定を図る。これは本プロジェクトの目的の 1 つであり、かなりのウェイトを占める。この科目の担当は筆者である。

また、「情報プロジェクト A・情報プロジェクト B」という新たな科目導入による情報キャリア

アップ教育の確立を行う。どちらも選択科目であり、「情報プロジェクト A」はシステム 2 年秋学期に開講され、情報資格試験対策の授業を行う。「情報プロジェクト B」はシステム 3 年秋学期に開講され、1 プロジェクト 7 名程度(1 教員 3 プロジェクト程度)に対して、13 週を通じて、ソフトウェアの企画・設計・実装・評価を学ぶという情報キャリアアップを目指した科目である。

2.1.4. その他

その他の情報系科目として、全学 2 年必修春学期の「メディアリテラン」およびシステム 3 年選択秋学期の「アルゴリズム応用論」においても e-learning 化を図る。

2.2. コンテンツ作成における学生の階層

本プロジェクトのもう一つの特徴として、e-learning コンテンツの作成に、科目教員と学生の密な連携体制があげられる。実際にコンテンツを作成する学生のスキルは多岐にわたっており、2 年生から 4 年生、修士課程の学生が行っている。新入生は入学と同時に e-learning を使った授業をユーザとして体験する。2 年生になると、プロジェクト・スタッフという形でコンテンツ作成に興味を持つ学生を募集する。プロジェクト・スタッフは、3 年生のメディア・コンサルタントから技能を伝授され、自らが 3 年生になったときには次のプロジェクト・スタッフに教える立場となる。メディア・コンサルタントは、情報系研究室の 4 年生、大学院生から技能を受け継ぎ・・・というつながりができている。

3. ソフトウェアデザイン完全 e-learning 化をめざした 17 年度の試み

前述のとおり、この科目の担当は筆者であり、他の科目はブレンド型 e-learning をめざす中、今回のプロジェクトにおいては唯一完全 e-learning 化を行う科目である。コンテンツ作成は筆者の研究室でおこなっている。本節では、コンテンツの概要と作成のポリシーについて述べる。

3.1. PLESS から Angels

平成 14 年度までに本研究室ではプログラミング言語教育支援システム PLESS(Programming

Language Education Support System)プロジェクト[4]の中で、スタンドアローンのコンテンツである Cmaster, Jmaster[5]の開発を行った。しかし、これらのコンテンツでは今後の e-Learning 教育の場に適応することは難しいということで、16 年度、既存の Cmaster, Jmaster を JSP/Servlet を使って書き直し、eL コンテンツとして再開発するために、プログラミング言語 eL システム Angels(Assistive and Navigation equipped E-Learning System)プロジェクト[6]を始動した。このプロジェクトは、個人の習熟度に合わせて個人的なフィードバックをかけることができ、さらに学習者からの意見を取り入れることによってより学習者にとって魅力的なコンテンツを作成することを目的としたものであった。Angels は、スタンドアローンであった Cmaster, Jmaster を eL コンテンツとしての仕様を満たすように、また課題として残されていた部分を改善するように設計し直し実装を行ったものである。この後、Jmaster on Angels は Java 言語教育完全 e-learning 化を目指して 16 年度末から 17 年度春にかけてさらに形態を変えることとなる。

3.2. コンテンツの構築

3.2.1. 概要

ソフトウェアデザインのコンテンツ作成の基礎となっているのは、Jmaster on Angels における学習法の構築である。学習法を構築する上で、学習者の苦手な項目、およびテキスト等だけではわかりにくい項目のうちどのようなイメージがほしいかというアンケート調査を行い、その結果を踏まえて、Java 言語における言葉の概念、オブジェクト指向言語特有の考え方を中心に理解させるようにシステムの設計を行った[7]。プログラミング初心者にはオブジェクト指向における振る舞い(行動)と状態を理解させるために、クラス、メソッド、フィールドの説明には Macromedia Flash アニメーションを中心に開発を行った。最後にまとめとして実際のプログラムを実行される様子をアニメーションによって学習者に提示することもできる。

本プロジェクトのコンテンツは、SCORM 対応であることが求められており、Jmaster on Angels の講義テキスト、アニメーションを

SCORM 対応にすることから始まり、それに加え、映像、ドリル(演習問題)が追加された。

コンテンツの目次リストを以下に示す。簡条書きの大項目がほぼ 1 回分の授業内容に相当する。1 回の授業につき、それぞれいくつかのテキスト、アニメーション、映像教材、ドリル(演習問題)が用意される。

- 1) オブジェクト指向と Java
- 2) クラスの概念とインスタンス
- 3) クラス・フィールド・メソッド
- 4) 今までのまとめ
- 5) 継承①
- 6) 継承②
- 7) 配列
- 8) 抽象クラス
- 9) インターフェース
- 10) GUI プログラミング

3.2.2. 映像によるオブジェクト指向概念の表現

映像における最大のメリットは、テキストやホワイトボード、コンピュータ等を使っても説明しきれないイメージを伝えられることである。何年間にも渡り、オブジェクト指向言語の授業を行ってきた中で、授業の導入部分における、オブジェクト指向・メッセージパッシング・インスタンスの生成といった概念、そして継承の概念を理解させる方法について試行錯誤を行ってきたが、表現しきれない事柄も多かった。今回、映像コンテンツ作成にあたり、今まで表現しきれなかったこれらのオブジェクト指向特有の概念を表現するために、シナリオ作成の段階からさまざまな工夫を行った。以下、詳細を述べる。

● クラスとオブジェクト

クラスとは？オブジェクトとは？という話をするとき、プログラムソースを使った具体的な説明をする前に概念的なイメージを伝えたいと思った。そこで、今回は、たとえば人オブジェクトをつくる場合にはその型紙になるような設計図(クラス)をつくり、その設計図からものの実体をつくるというイメージを伝えるために、学生何人かを人クラスのオブジェクト(インスタンス)として登場させた。Lesson 1 では、プログラムソースを使わずに、クラスからオブジェクトが生

成され、各オブジェクトは状態（フィールド）と行動（メソッド）を持っているということを理解させる。Lesson2ではLesson1を踏まえ、プログラムソースを使って、インスタンスが生成される様子を理解させる。Lesson3では、さらにプログラムの実行を1行ずつ追いつながら、コンストラクタを使ってインスタンスが生成され、その結果、各インスタンスのフィールドがどう変わるかということまで理解させる。

- **メッセージパッシング**
オブジェクト指向における重要な概念の一つにメッセージパッシングがある。この考え方を理解させるために、メッセージパッシングによるメソッド呼び出しを前述の学生インスタンスのやりとりで表現するなどの演出を行った。クラスとは何か、そしてクラスと実体（インスタンス）との違い、実際にインスタンスが生成されるということはどういうことなのかということを理解させるための映像の例を図1に示す。この映像では、各インスタンスの中にそれぞれフィールド（name, age, occupation）が存在し、メッセージパッシングによって送られたメソッドは、受け取るインスタンスによって処理が異なってくるということも理解できるようなシナリオになっている。
- **変数の種類**
Javaにおけるプリミティブ型変数と参照型変数の違いを理解させ、変数の生成やアクセスにあたりメモリ上でどのような処理が行われるのかを理解させる。また、インスタンス変数を宣言しただけではなぜ実体ができないのか、変数というのはインスタンスが実際にあるメモリへの参照を表しているだけで、実体ではないということなども理解させるための演出を行った。
- **コンストラクタ**
インスタンス生成にあたり new キーワードを伴ってコンストラクタを呼び出す必要性はなんとなくわかっていても、細かい話になってくるとほとんどの学生は混乱する。そこで、コンストラクタの説明のみに1回分の映像を割り当て、コンストラクタの役割やデフ

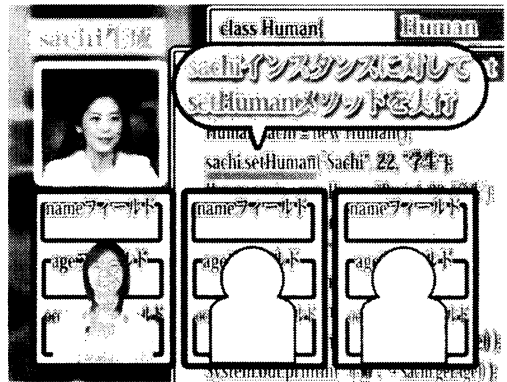


図1：Java映像

ルトコンストラクタ、引数ありのコンストラクタを宣言した場合に引数なしコンストラクタがないとコンパイルエラーが起こることがあるのはなぜか等を理解させるための演出を行った。

- **配列**
配列を使ったコーディング理解の演出のほか、配列宣言と変数の生成に伴うメモリ上の処理を理解させるために「変数の種類」同様の演出を行った。
- **継承**
オブジェクト指向理解においての山場の一つである継承については映像を2回にわたった。継承の概念をアニメーションや複数の図で表し、継承の利点を説明、その後、スーパークラスの持つ情報をサブクラスで引き継ぐ様子等をプログラムを使わずに理解させた後、実際にプログラムを使って、今まであったクラスを継承して新しいサブクラスをつくる方法と、今まであったクラスに共通な部分をまとめて新しくスーパークラスをつくる手法にわたった説明を行う。また、オーバーライドや継承におけるコンストラクタの扱いについても理解させる。継承の説明においても学生インスタンスの映像をベースとしており、スーパークラスからサブクラスへメソッドやフィールドが継承される様子や、コンストラクタは継承されないこと、デフォルトコンストラクタの必要性等、非常に説明が難しい内容についても映像を使うことで一貫した説明を行うことができる。

- その他
その他、映像全体に渡る工夫点をあげる。まず、重要部分をテロップで何度も視覚に訴えることで、耳からと目からと同時に重要な概念を覚えさせることができる。また、すべての映像を通して、クラス、インスタンス、コンストラクタ、メソッド、スーパークラスとサブクラス等の表示、テロップの色、それぞれの関係を表す線や矢印の色などは統一をはかっている。

3.2.3. 映像完成までの製作工程

本コンテンツの映像は、実際の授業を収録したものを編集するというものではなく、本コンテンツ専用シナリオを作成して撮影・編集を行っている。映像の撮影・編集は、本学の映像専門の研究室が担当し、シナリオ作成・出演・編集チェックは筆者・および筆者の研究室が担当している。1本の映像は5分から20分程度のものが12本ある。映像完成までの流れは以下のとおりである。

- シナリオ作成
全体の流れを意識しながら、その回のエッセンスが5分から20分以内に収まるようにせりふを作成する。例プログラムなども用意し、編集者に対するテロップの指示や映像のアニメーション等の指示も記載する。
- 撮影
5分から20分の映像コンテンツの作成にかかる撮影の時間は2時間から4時間である。平成16年秋にスタジオ（メディア・ラボ）が新たにできたため、飛行機の騒音等による中断がなくなった分、それまでよりは3割ほど短縮されたものの、かなりの時間がかかることは事実である。また、スタジオ専用の部屋ではないので照明器具やカメラなどは撮影のたびにセットしなければならず、準備時間・撤収時間を合わせると実際は1,2時間余分にかかっている。
- 編集
編集はAdobe Premiere Proを使って行っているが、まず撮影したカメラからコンピュータ上にデータを移すのにほぼ1日、最初の編集には3~4日の時間を要する。

- 編集チェック
編集チェックは筆者を中心に、コンテンツのテキスト、アニメーション、ドリルの作成者を交えて行う。テロップの語句の誤字脱字とフォントや色のチェックから、インスタンスの生成の動き等がこちらのイメージ通りになっているか、画面の切り替わりのタイミング、プログラムとの整合性などチェック項目はかなり多い。最初のチェックにかかる時間は出来上がった映像の5~10倍ほどの時間を要する。つまり20分の映像ならば2~3時間かかるということである。第1回目のチェックの手直しには1~2日の猶予をみる。2回目以降は直した部分のみのチェックであるが、完成までには平均2,3回の手直しを行う。手直しの量は回によって異なるが、やはり最初の1,2回の導入部分、プログラムの説明、継承の2回分では、手直しに1週間ほどかかっている。
- サーバへアップロード
映像コンテンツは、編集チェックをすべてクリアするとサーバへアップロードされる。

3.2.4. 講義テキスト（教科書）

3.2.1 節で述べた大項目の中にさらに小項目があり、それぞれに講義テキストが用意される。各テキストはHTMLファイルである。テキストだけではイメージを伝えにくい内容についてFlashアニメーションを用意した。アニメーションファイルをテキスト内部に組み込むことはせず、テキストでは対応するアニメーションを見るよう記述するにとどまっており、実際に学生は目次リストから対応するアニメーションをクリックして見ることになる。

3.2.5. アニメーション

テキストで表現しきれないイメージや映像に盛り込めないような細かいプログラムの説明を伝えるために、アニメーションを作成した。テキストの内容を補助するためのものと、長いプログラムの説明を行うためのものがある。プログラムの説明を行うアニメーションでは、プログラムの各行にマウスを合わせると、その行の説明が表示される。

3.2.6. ドリル

各大項目に、理解度を確保するためのドリルを用意した。ドリルは学習者が能動的に進度に合わせて回答できる「演習モード」と、達成率を100%にする期限を与える「課題モード」の両方で回答できる。語句やプログラムの穴埋め問題や、プログラムの実行結果を問うような問題がある。回答方式としては、選択式（各小問ごとに複数選択肢から選ぶ）、パズル方式（大問に複数選択肢を用意し、その中から当てはまるものを空欄へマウスで移動する）がある。実際に言葉を入れるという方法のほうが選択式よりも力はつくのではないかという意見もあり、現在検討中である。また、ヒントを表示することもできる。学生は、1年生の数学でも同じような e-learning コンテンツを使っており、そちらのドリルではヒントを最後まで見ると解答が書いてあるものが多いため、このドリルでもその文化を受けているせいか、最後のアンケートでは、ドリルのヒントに解答を載せてほしいという要望もあった。ヒントの出し方については今後の課題である。

3.3. 運用

これらのコンテンツを実際に使用して17年度春学期に12週間の授業を行った。学生は今年度は基本的に授業に出席するが、授業の中で e-learning の映像やテキストを自主学習し、期日（1週間後：次の授業の前日）までにその回のドリルの達成率を100%にし、プログラミングの課題を提出する。自力で課題やドリルができる学生は授業に出席しなくてもよい。e-learning による学習と平行して、3.2.1 節の 4) と 5) の間と、6) と 7) の間には、プログラミングの実践に関する授業も行った。評価は中間テスト・期末テストの2回であり、どちらもペーパー試験およびオンラインでプログラミングを行う試験を行った。試験内容は前年度 e-learning を使わない授業を行ったときとほぼ同レベルなものにした。ペーパー試験は語句の穴埋めとプログラムの穴埋め問題である。

前年度と比較すると、中間テストに関してはあまり差が出なかったが、期末テストに関してはペーパー試験、オンライン試験ともに本年度の方が成績が良い。今年は、教科書で学んだ語句をドリル

ルを使って理解度を確保できるようにトレーニングできた成果が出ているのではないか。また、アンケートに基づいた学習法の構築に従ってカリキュラムを作成し、複数の教材がうまく連動して学習できた成果といえよう。また、e-learning 閲覧に加えてプログラミングの授業もしっかり行っている効果が現れているのだろう。また、前半からこつこつと教科書を勉強している学生は中間、期末とも試験結果が良いようである。このことから、前半の基礎部分がしっかり理解できていることがかなり重要であることを確認できる。オンライン試験に関しては、前半だけでなく、授業期間全体を通して教科書閲覧度が高い人がやはりいい成績をとっているという傾向がみられる。

前年度との比較結果や教科書閲覧状態と試験との相関関係についての詳細データの報告は次回に行う。

3.4. アンケート結果

授業の最後にとったアンケートの結果について述べる。アンケートの項目は、自分の理解度に関するもの、e-learning に関するもの、コンテンツに関するもの合わせて26項目であるが、紙面の都合上、主なものだけ紹介する。

まず、理解度に関する質問では、図2に示すように授業の前後における理解度はかなりあがった。レベルの上がり方は2レベルアップした割合が39%と最も多く、その90%以上はレベル1からレベル3までアップしている。

次に e-learning に関する質問では、e-learning が①全般②言葉の概念理解③Java の文法④Java のプログラムの流れ⑤プログラムを書く、の5項目について役に立ったかの質問にはそれぞれ83%、79%、71%、69%、60%の学生がはいと答えており、e-Learning で学習意欲が湧いたかの質問には63%がはいと答えていることから、本大学での e-learning の利用価値は学生には認められているといえるだろう。しかし、プログラムを書くという最も重要な項目についての効果は最も低い数値を示している。これは予想されていたことであり、現在提供されている e-learning コンテンツを個人で学習するだけでは、Java やオブジェクト指向に関する知識はについても実際にプログラミ

ングを行う実習にあたる部分がないため、コーディング能力がつかないという問題を解決することが必須である。

授業時間以外で e-learning を使ったかどうかには 84% がはいと答えており、主にどこで使ったかの質問には大学が 50%、家が 38%、両方が 9% であることから、学生は大学でも家でも気軽に e-learning にアクセスできる環境にあることがわかる。また、e-learning を使った主な目的としては講義中に 26%、課題に 50%、復習に 19%、予習に 5% であるにもかかわらず、予習・復習に役立ったかどうかには 88% の学生がはいと答えていることから、学生に

e-learning を根付かせるためには、①ドリルや課題の提出期日をこまめに設定する ②コンテンツの閲覧期間をこまめに設定するなどこちらから半強制的に使用させる運用方法を取り、利用を促進すれば、自主学習での有効利用性もあがると思われる。

次に、e-learning さえあれば、授業時間に先生や TA がいなくても、課題を全て終わらせることができたと感じるかの質問には 図 3 に示す結果となった。図より、現在では先生、TA の存在は学生にとって

かなりの重要度を占めていることがわかり、完全 e-learning になった際にはこれらを補完する仕組みが必須である。

実際にプログラミングを行う際に役立つもの(参考にしたもの)に順位を付けるという質問には、教科書、ドリル、映像、課題、先生の授業、のうち 1 位をつけた割合はそれぞれ 25%、19%、

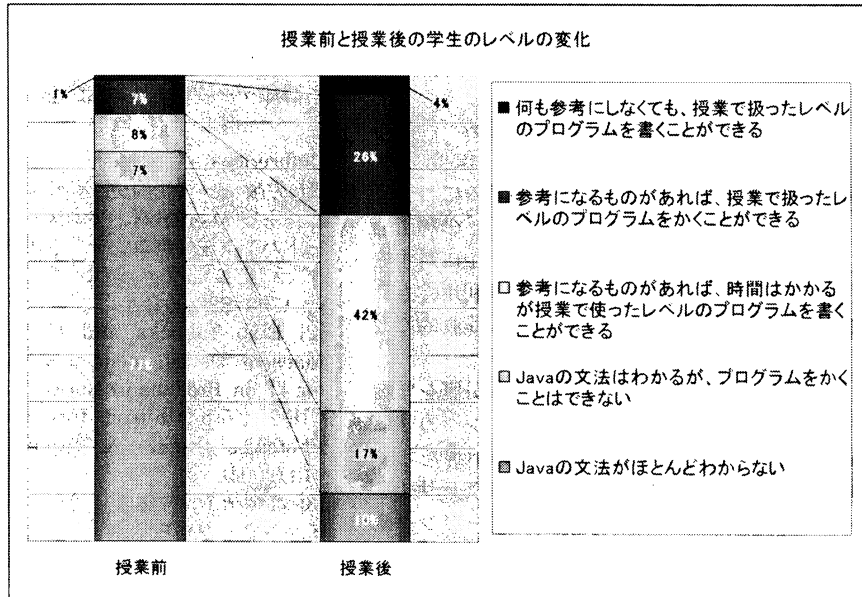


図 2 : 学生の理解度

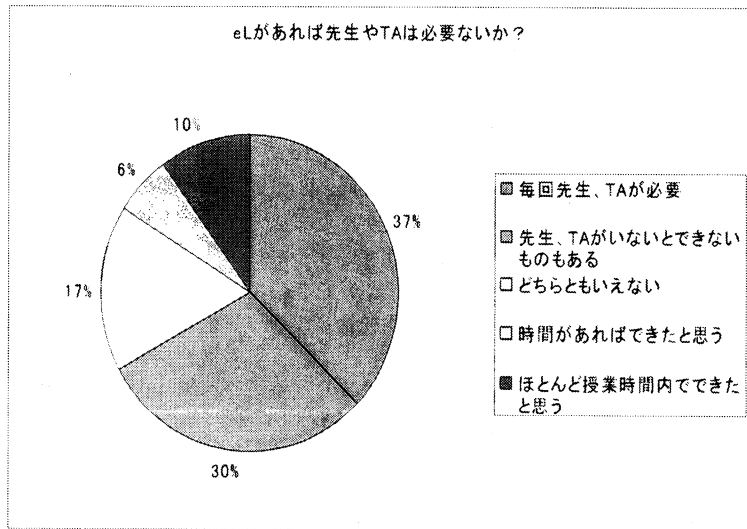


図 3 : eL は授業の代わりになるか

15%. 23%. 18%であり、教科書を選んだ中では1位の割合は37%と最も高い数値を示している。映像に関しては残念ながらあまり役立ったという結果は得られなかったが、映像は先生が授業でプログラムの説明を行う代わりになると思うかの質問には54%がはいと答えていることと、今年度は授業があったので映像を見なくても理解ができたということを考えると、映像の効果については来年度以降も調査を続ける必要があるだろう。

4. 結論、今後の課題と現在の取り組み

Java プログラミング言語 18 年度完全 e-learning 化をめざした 17 年度の試みについて述べた。総じて、本大学での e-learning の利用価値は学生には十分認められており、学生に対する利用促進によって、自主学習での有効利用性も高まるだろう。来年度に向けて、学生が能動的に取り組む仕組みを構築する必要がある。

また、プログラミング言語のような分野を学習する際には、実際に講師からコーディング方法についてのメッセージやヒントなどの指導をしてもらうことが重要である。ソフトウェアデザインを完全 e-learning に移行すれば、スクーリングやメンタによる指導は受けられるとはいえ、学習者は基本的に個人で学習を受けることとなる。以下のような問題点が生じてくるであろう。

- 1) プログラミングの概念の理解ができない
- 2) Java 等の言語自体のインストールができない
- 3) プログラミングの技術が身につかない
- 4) コンパイルエラーが取れない

このような背景を踏まえ、Java のインストールのように学習者が手間とを感じる箇所の簡略化や、学習者がコーディングを行いながらプログラムを作成する手順、コンパイル時に発生するエラー文の読み方と取り除き方を学ぶことができる Java プログラミング初等教育の専用統合環境を構築している。

映像を導入したことによって従来に比べてオブジェクト指向特有の概念をイメージしやすいコンテンツを作成することができた。しかし、今年度は授業を行ったため、映像コンテンツの有効性については有用な結果は得られておらず、来年

度以降の調査が必要である。また、教科書やアニメーションなどのコンテンツと併用した最適なコンテンツづくりが必要であろう。たとえば、映像においては長いプログラムを扱うと字が小さく読めなくなってしまうという欠点がある。したがって、プログラムを使った説明においては該当する教科書やアニメーション等を閲覧できるようにしておく、プログラム全体を見たいときのためにファイル提供する等の各コンテンツ間の連携が必要であろう。

詳細データに関しては次回に続く・・・

References:

- [1] 佐藤 威, 青山朋史, 渡部 彰, 高岡詠子, "PC マエストロ: 映像教材、アニメーション教材連動ブロードバンド配信型コンテンツの構築", 第 45 回プログラミング・シンポジウム報告集, pp.179-182(2004).
- [2] Eiko Takaoka and Hiroyoshi Usui, "PC Maestro: Development of a Multimedia Content for IT on Broadband Networks", Proceedings of CIF'5 (5th Chitose International Forum on Photonics Science & Technology) pp. 112 - 118.(2004).
- [3] 高岡詠子, 佐藤威, "専用エディタを使ってコーディングしながら学ぶ「PC マエストロ HTML バージョン」", 情報処理学会研究報告, 2005-CE-79, 情報処理学会, pp.25-32 (2005).
- [4] 高岡詠子, "プログラミング言語教育補助システム PLESS、日本工学教育協会 平成 15 年度工学工業教育研究講演会講演論文集, pp.61-64(2003).
- [5] 高岡詠子, 齊藤将志, 谷口景介, "Java プログラミング教育補助システム Jmaster の開発と教育効果", 情報処理レターズ, 情報処理学会, pp.381-382(2003).
- [6] 石井和佳奈, 佐々木雅啓, 中川朝博, 辻健史, 澤田英敏, 内海貴博, 高岡詠子, "プログラミング言語 e-learning システム Angels の構築", 情報処理学会第 67 回全国大会講演論文集第 4 分冊 pp.455-456(2005).
- [7] 高岡詠子, "大学におけるデジタルコンテンツ開発の変遷および e-learning プロジェクト", 情報教育シンポジウム SSS2005 プレカンファレンス論文集, pp.46-52(2005).