

# プログラミング課題自動チェック機構を活用した 演習課題の分割と事前学習化の試み

伊藤 恵\*1

\*1 公立はこだて未来大学

## An Attempt to Divide Questions and Change Them into Preparation Quizzes Using Automatic Check Mechanism for Programing Exercises

Kei Ito\*1

\*1 Future University Hakodate

情報系大学の初年次教育として実施している Java プログラミング科目において、事前学習を前提とした演習授業を実施している。事前学習の定着のため予習クイズ等を導入しているが、事前学習から演習課題への移行がうまく進まない受講者が後を絶たない。そこで、事前学習から演習への円滑な移行のため、従来より導入しているプログラミング課題の自動チェック機構を活用し、演習課題の分割と一部事前学習化を試みた。その実践結果を報告する。

キーワード: プログラミング学習, 演習課題, 自動チェック

### 1. はじめに

情報系大学・学部において初年次教育の一つとしてプログラミング学習が導入されていることが多い。必修の演習科目として開講されることが多いが、演習であるため実践力が付く一方で、大学入学者の平均学力の低下<sup>(1)</sup>などもあり、必修であることもあいまって十分理解が進まないまま、単に課題をこなすだけで終わってしまう学習者が目立ってきている<sup>(2)</sup>。

著者の所属大学では、事前学習を前提としたプログラミング演習科目が開講されている。授業時間だけで十分な理解が難しい学習者が多いため、事前学習を強化し、事前学習教材に含まれる予習クイズに正解しないと次の教材へ進めない仕組みを導入した。また、演習時には担当教員と Teaching Assistant (以下 TA)とで学習者の質問に対応するが、学習者の理解度に応じた多種多様な質問に効果的に対応できない場合もあることから、演習時の質問対応の質向上のためにプログラミング課題自動チェック機構<sup>(3)</sup>を導入した。この機

構により、学習者がプログラミング課題を提出すると、提出されたプログラムが課題に対して不十分な点をメッセージ表示するほか、自動採点された点数が表示される。これにより担当教員と TA は自動チェックでは分からない質問への対応に注力することができるようになった。

このプログラミング演習科目では、事前学習教材に学習項目の導入資料やそれに関する予習クイズを設定し、演習時には事前学習で扱ったものを応用した課題を課しているが、学習者の学習意欲維持と実践力向上の両立を意図した課題とするため、事前学習と演習課題の難易度調整が難しくなっている。実際に、事前学習はすんなり出来ていても演習課題でつまづいて進まない学習者や、約 3 時間の演習時間内にその日の課題を完成し終わらない学習者が後を絶たない。

そこで事前学習教材や演習課題自体の改善と並行して、演習課題を複数に分割し、さらにその一部を事前学習に含めることを試みたため、その実践結果を報告する。

## 2. 対象科目

### 2.1 対象科目と学習者

著者所属大学で学部2年生向けに開講されている主にJava言語を対象としたプログラミング演習科目「情報処理演習I」を本実践の対象とする。この科目の受講者は1年次にProcessingとC言語による演習科目を受講しているが、プログラミング自体の理解度は必ずしも高いとは言えない。

この科目は学生の所属に応じて前期(4月～7月)に受講する者と、後期(10月～翌1月)に受講する者がいる。また、前期に受講する者は演習用教室のサイズの都合により、学内の所属コースに応じて5クラスに分かれて受講している。受講者は毎年度異なるため厳に公正な比較はできないが、経験的に同じコースの所属であれば年度が異なっても受講者の傾向がよく似ていることから、本報告では毎年度前期に著者が担当している同一所属コースの1クラスのデータを扱う。

### 2.2 演習教材と支援システム

対象とする演習科目の教材は基本的にこれまでの担当教員が作成または更新してきたものである。演習の各回に応じて、予習クイズを含む事前学習用教材と一つ以上の演習課題が用意されている。各回に二つの演習課題を標準としているが、課題の難易度等に応じて課題が一つの回や三つ以上の回もある。課題の題材は受講者の興味を引きそうなものとなるよう設定しつつ、難易度調整のために作るべきプログラムの一部を雛形として与えて、残りを作らせるようにしている。教材の提供等を含め演習科目はすべてオープンソースのコース管理システム Moodle を用いて運用されている。ただし、その一部は著者所属大学用に独自に開発された機能を利用している。

#### 2.2.1 予習クイズ

事前学習教材を単に読み物として提供すると、何も事前学習をしないで、いきなり演習課題に取り組む学習者が増えることから、事前学習を強制するための方策の一つとして予習クイズを導入している。演習一回分の事前学習教材の中に予習クイズを一つ以上設置し、予習クイズに正解しないと次の教材に進むことができ

ず、また、その回のすべての予習クイズに正解していないと演習課題も見ることができない。このため完全自動採点できる問題のみが予習クイズとして使うことができる。

#### 2.2.2 プログラミング課題の自動チェック機構

30～40名程度の受講者に対して、通常、担当教員1名とTA2名が演習中の質問対応を行っているが、受講者の理解度の幅が広く、質問も多岐に渡ることから、プログラミング課題の自動チェック機構を導入している<sup>3)</sup>。

これは個々の演習課題に応じたチェックプログラムを事前に用意しておき、受講者が課題のプログラムを提出すると、即座にチェックプログラムによるチェックを行い、課題を満たせていない点をメッセージ表示するほか、自動チェックできる範囲内で採点も行ってその点数も表示する。この結果表示により受講者は課題をどの程度までこなしているかを随時確認することができる。また、この自動採点の点数は演習課題の点数評価にも利用される。

## 3. 課題の分割と一部事前学習化の試み

対象とする演習科目において、演習課題のプログラムをどこから作り進めるべきかが分からずに作り間違え、それによって演習時間を無駄にする受講者が例年見受けられたことから、演習課題の中に進め方を誘導する説明などを記載していた。しかし、進め方を間違えるタイプの受講者は誘導された箇所すら作り間違え、作ったものが適切かどうか確かめないうまに進んでしまうことで、結果としてはやはり演習時間を無駄にする傾向にあった。そのため、課題の進め方を誘導し、かつ、そこまで進めた分を自動チェックによって確認させるために、演習課題をある程度分割し、複数の課題を通して一つのプログラムを完成させるように、一部課題を更新することとした。

また、対象科目の教授側としては事前学習教材を応用する形で設定してあるはずの演習課題であっても、受講者にとっては大きく乖離して感じられることもしばしばある。そこで演習課題で取り組むべき最初の部分を演習課題から事前学習教材中の予習クイズに移行

させることとした。予習クイズは授業時間外に実施することと、2.2.1 で述べたように予習クイズに正解しないと次の教材に進めないことから、完全に自動採点可能なものである必要があるが、2.2.2 で述べた自動チェック機構による自動採点によって、演習課題の一部であってもそれが可能となった。

### 3.1 2016～2017 年度演習課題の分割と事前学習化

ここでは 2016 年度前期と 2017 年度前期に扱った演習課題の概要を述べる。表 1 は 2016 年度前期と後期および 2017 年度前期を対象の演習科目で扱った課題の一覧である。第一回演習が課題 1-1 と課題 1-2、第二回演習が課題 2-1 と課題 2-2 のように扱っており、2017 年度前期に演習課題の一部を事前学習の予習クイズとした箇所はクイズとしてある。2016 年度後期は対象クラスが異なるため、実践結果の比較には用いないが、2016 年度から 2017 年度への演習課題の段階的更新の途中経過となるため、併載した。

2016 年度前期から 2016 年度後期に課題内容もかなり変更しているが、それと同時にひとまとまりの課題を複数に分割し、受講者が取り組みやすくしている。さらに 2016 年度後期から 2017 年度前期に掛けて、課題内容を一部調整しながら、一部の課題の事前学習化を行っている。

例えば、第四回演習の課題 4-1、課題 4-2 は 2016 年度後期時点で課題を前半課題 4-1 と後半課題 4-2 に分けていたが、2017 年度前期にはその前半の内容をほぼそのまま予習クイズとし、後半だった内容を課題 4-1 に繰り上げた上で、より発展した内容を新たに課題 4-2 として追加した。また、第六回演習の課題 6-1～課題 6-2 については、2016 年度後期時点の課題 6-1、6-2 の内容を共に予習クイズとした上で、課題 6-3、6-4 だった内容を 2017 年度前期には課題 6-1、6-2 にスライドさせた。以降、第 7 回は一部予習クイズ化した上でより発展的な課題を追加、第 8 回は一部予習クイズ化した上で課題の一部省略と発展的内容追加、第 9 回は一部予習クイズ化した上で課題を前にスライド、第 11 回は一部予習クイズ化し、課題を前にスライドさせた上で課題内容の入れ替えも行った。

## 3.2 実践結果

ここでは 2016 年度前期と 2017 年度前期に著者がそれぞれ演習担当したクラスの演習結果を述べる。

### 3.2.1 最初の課題を終えるまでの時間

図 1 は 2016 年度前期に著者が担当したクラスにおいて、その日の最初の課題を完了した者の人数の推移を示したグラフである。横軸が時間で 13:10 が演習開始時間、16:20 が演習終了時間である。縦軸はその時点で最初の課題を終えていた人数であり、受講者全体は 41 名であった。回によって演習課題の難易度が異なるが、比較的進行が早かった回は演習開始から 1 時間程度で多くの受講者が最初の課題を終えているが、課題の難易度が高く、演習課題を一つしか設定されていなかった第 8、9、10 回や最終課題のある第 12 回は完了者数の増加が遅く、演習終了時間になっても半数以上は終えていなかった。

図 2 は 2017 年度前期に著者が担当したクラスの同じグラフである。横軸の時間は図 1 と同様、縦軸の人数も同様だが、受講者全体が 42 名である。課題も受講者も異なるほか、課題一つの粒度が平均的には小さくなっているため、単純に比較することはできないが、最初の課題を完了するまでの時間が大きく短縮されている。

### 3.2.2 最後の課題を終えるまでの時間

図 3 と図 4 は逆にその日の最後の課題を終えた者の人数推移である。その日のすべての課題を終えた人数と同義であるため、課題一つの粒度には影響されない。2016 年度前期は演習終了時間にほとんどの受講者が課題を終えていたのは一回のみで、約半数の回は課題をすべて終えていた受講者は半数以下である。一方、2017 年度前期は約半数の回は演習開始から 2 時間後にはほとんどの受講者がすべての課題を終えている。

### 3.2.3 実践結果の考察

2016 年度と 2017 年度は受講者が異なるため、当然、厳に公正な比較はできない。が、この演習科目を長年担当してきた経験上、同じクラスであれば年度が異なっても受講者の傾向は大きくは変わらず、その影響は少ないと考えられる。

一方、課題が異なることによる影響は相応にある。実際、各クラスの担当教員やこの2年度継続してTAを担当した学生の感触としても、2016年度前期と比べて2017年度前期の課題の難易度は全体としてやや簡単になったと感じている。

対象科目の成績は演習課題の点数とペーパーテストの点数から評価される。演習課題の難易度に直接影響されないペーパーテストの点数は、2016年度前期が100点満点で平均51.0点、2017年度前期が平均51.6点とあまり差がないことから、演習課題の難易度に違いがあったにしろ、同程度の学びが得られていると考えられる。

#### 4. おわりに

事前学習を前提として行っているプログラミング演習科目において、事前学習から演習への円滑な移行のため、従来より導入しているプログラミング課題自動チェック機構を活用し、演習課題の分割と一部事前学習化を試みた。演習課題の内容自体も変わっているため、一概には比較できないが、授業当日の演習課題への取り組み易さが改善される傾向にあることが分かっ

た。今後は、適度な課題分割と事前学習化をしつつ、課題内容をさらに精査し、取り組み易い演習課題でありつつ、多くの受講者がより深く学べる演習教材を目指したい。

#### 謝辞

本研究はJSPS 科研16K04798の助成を受けたものである。

#### 参考文献

- (1) 国立教育政策研究所: “OECD 生徒の学習到達度調査 (PISA)”, <http://www.nier.go.jp/kokusai/pisa/> (2018年1月アクセス)
- (2) 伊藤恵, 椿本弥生, 白石陽, 奥野拓: “プログラミング教育のためのメモ用紙活用における電子ペーパー利用の試み”, 教育システム情報学会, 2016年度第5回研究会, (2017)
- (3) 伊藤恵, 美馬義亮, 大西昭夫: “コース管理システムと授業固有の課題チェック機能のWebサービスによる連携”, 情報処理学会論文誌, Vol.52, No.12, pp.3121-3134 (2011)

表 1 2016～2017 年度演習課題一覧

	2017 前期	2016 後期	2016 前期
課題 1-1	関数の実装(平方根の近似値)	C から Java への書き換え	C から Java への書き換え
課題 1-2	関数の実装と動作確認(1-1 の続き)	数列	数列
課題 2-1	ルーレットホイールセレクション	ルーレットホイールセレクション	C から Java への書き換え
課題 2-2			素数列
課題 3-1	都市のデータを格納するクラス(その1)	都市のデータを格納するクラス(その1)	直方体
課題 3-2	都市のデータを格納するクラス(その2)	都市のデータを格納するクラス(その2)	直方体クラスの利用
クイズ	都市の人口流入シミュレーション(準備)		
課題 4-1	都市の人口流入シミュレーション1	都市の人口流入シミュレーション1	非 static メソッドの作成
課題 4-2	都市の人口流入シミュレーション2	都市の人口流入シミュレーション2	データ格納クラスの作成
課題 5-1	会員カードの機能の実現	会員カードの機能の実現	インスタンスとメソッドの復習
課題 5-2			NVector クラスの拡張
クイズ	簡易データベース(準備)		
課題 6-1	簡易データベース(1/2)	簡易データベース(その1)	継承によるクラス作成
課題 6-2	簡易データベース(2/2)	簡易データベース(その2)	データ格納クラスの拡張
課題 6-3		簡易データベース(その3)	
課題 6-4		簡易データベース(その4)	
クイズ	自治体のオープンデータから統計情報を求める(準備)		
課題 7-1	自治体のオープンデータから統計情報を求める(その1)	自治体のオープンデータから統計情報を求める(その1)	ArrayList クラスの活用
課題 7-2	自治体のオープンデータから統計情報を求める(その2)	自治体のオープンデータから統計情報を求める(その2)	逆ポーランド記法による電卓プログラム
クイズ	簡易テキストマイニング(準備)		
課題 8-1	簡易テキストマイニング(その1)	簡易テキストマイニング(その1)	国別メダル取得数のランキング表示
課題 8-2	簡易テキストマイニング(その2)	簡易テキストマイニング(その2)	
課題 8-3		簡易テキストマイニング(その3)	
課題 8-4		簡易テキストマイニング(その4)	
クイズ	簡易レコメンデーション(準備)		
課題 9-1	簡易レコメンデーション(1)	簡易レコメンデーション(1)	テキスト中の語の出現回数の集計
課題 9-2	簡易レコメンデーション(2)	簡易レコメンデーション(2)	
課題 9-3	簡易レコメンデーション(3)	簡易レコメンデーション(3)	

課題 9-4		簡易レコメンデーション(4)	
課題 10-1	簡易レコメンデーションの完成1	簡易レコメンデーションの完成1	ファイル版 Unigram
課題 10-2	簡易レコメンデーションの完成2	簡易レコメンデーションの完成2	
クイズ	スコアリスト		
課題 11-1	奇数の最大値・最小値	スコアリスト	Web 版 Unigram
課題 11-2	約数さがし	trimmean	Web 版 Unigram の拡張
課題 11-3	距離の総和	距離の総和	
課題 11-4		frameprint	
課題 12-1			サーブレット版 Unigram
課題 12-2			ファイルアップロード対応版 Unigram

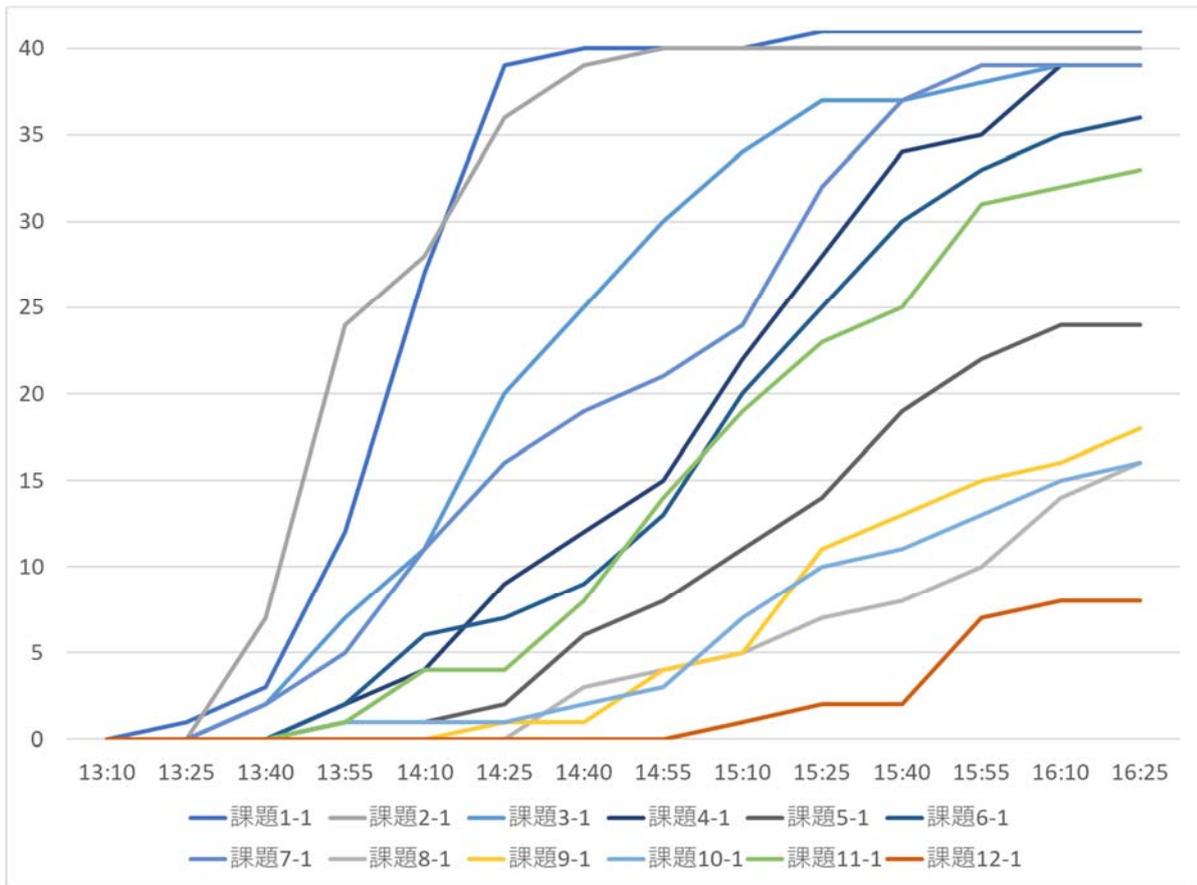


図 1 最初の課題を終えた人数の推移(2016 年度)

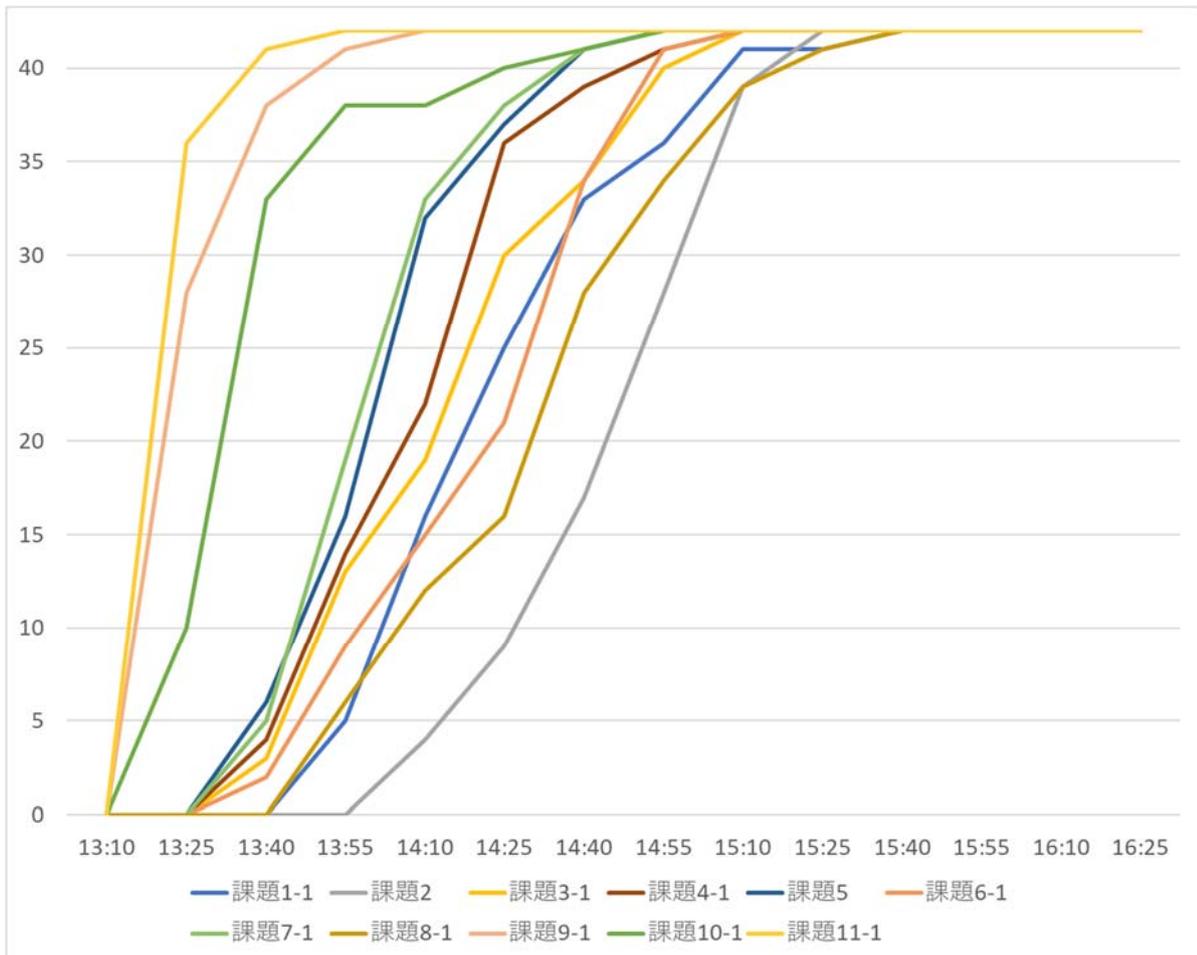


図 2 最初の課題を終えた人数の推移(2017 年度)

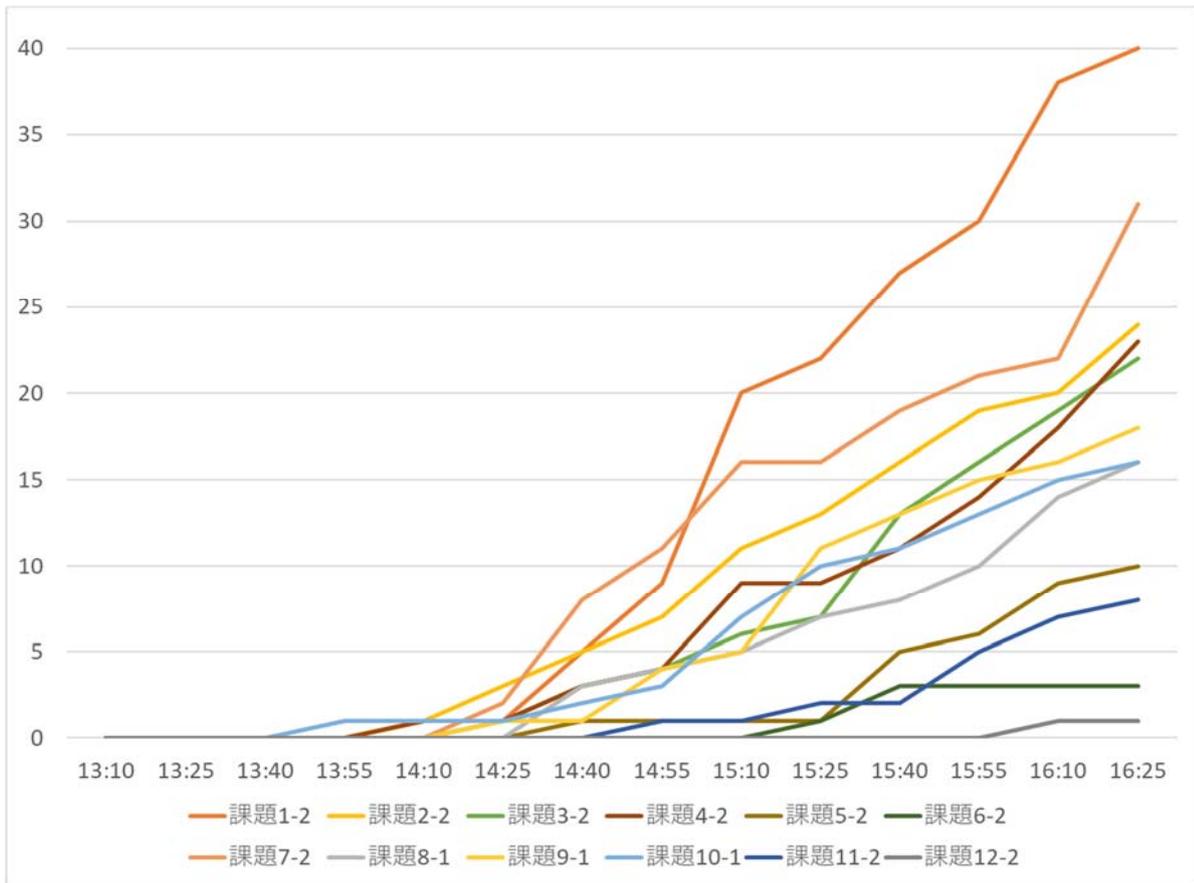


図 3 最後の課題を終えた人数の推移(2016年度)

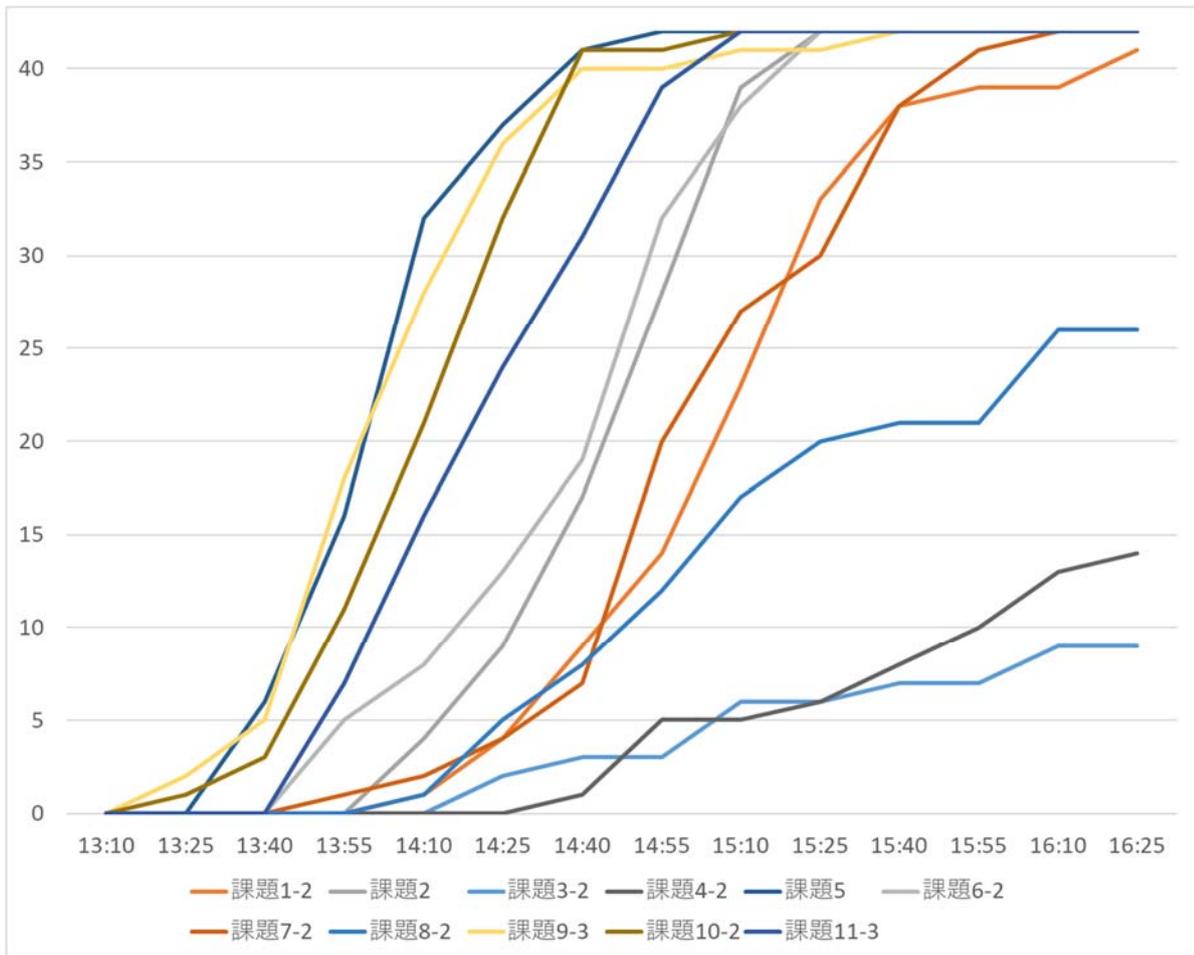


図 4 最後の課題の終了人数の推移(2017年度)