ISSN 1345-0093

愛知大学情報メディアセンター紀要

Vol.31/No.1 2022.March

COM

情報メディアセンター利用案内

◇サービス時間〈月～土曜日〉（都合により変更する場合があります。掲示およびホームページをご覧下さい。）

豊橋校舎

区　　分	曜日	420教室 （オープンアクセスルーム）	メディアゾーン （図書館）　※1	413教室・421教室・ 423教室・523教室	SEサービス
講義期間	月～金	9：10～19：00	9：00～19：50	講義利用のみ※2	9：00～17：40
講義期間以外	月～金	9：10～17：00	9：00～17：50		
通年	土	9：10～13：00	10：00～16：50		なし

※1　メディアゾーンは，豊橋図書館の運用日程に準じます。
※2　講義期間の木曜と土曜，及び，講義期間以外の期間は全て，413教室を開放教室として使用します。

名古屋校舎

区分	厚生棟4F　※3 メディアゾーン受付	講義棟7F メディアカウンター	SEサービス
講義期間 （7・1月以外）	（月～金） 8：50～21：00 （土） 8：50～17：00	（月～金） 8：50～20：00 （土） 8：50～17：00	（月～金） 8：50～18：30
講義期間 （7・1月）	（月～土） 8：50～21：00	（月～金） 8：50～20：00 （土） 8：50～17：00	
講義期間以外	（月～土） 8：50～17：00	閉室	（月～金） 8：50～17：00

※3　メディアゾーンは，名古屋図書館の運用日程に準じます。

車道校舎

区分	受付（3階）	メディアゾーン	K802，K804	SEサービス
講義期間	（月～金） 8：50～18：30	（月～金） 8：50～21：30	講義利用のみ	（月～金） 9：30～18：30
講義期間以外	（月～金） 8：50～17：00	（月～金） 8：50～19：00		

■センター閉室日　／　日曜・祝日・夏期休暇期間（8月土曜日）・年末年始・創立記念日（11/15）・
入試期間ただし，祝日授業日は開室

はじめに

情報メディアセンター長　岩田　員典

昨年度は新型コロナウイルスの影響で皆様が多忙なこともあり発刊できませんでしたが，本年度は愛知大学情報メディアセンター紀要COM46号を無事発刊できました。まだまだ新型コロナウイルスの状況が予断を許さない状況であり，お忙しい中投稿していただいた執筆者の方々をはじめ，編集委員や発刊に関わっていただいた方々にお礼を申し上げます。今号も4件の投稿を頂きました。多くの方々にご愛読いただければ幸いです。

さて，前述のように新型コロナウイルスは，2019年末に発見され，2020年1月に日本で初感染が確認されてから瞬く間に感染が広まり，いまだに収束の兆しが見えません[1]。そのような状況において，本学でも2020年度授業開始が5月にずれ込み，さらにはオンラインでの授業となりました。すべてをオンラインで実施するというのは，学生だけでなく教職員も未経験の状況でしたが，他大学と比べても比較的トラブルが少なくオンライン授業を開始・運営できたのは，関係各所の教職員の方々，学生の皆さん，保護者の方々の御協力と御理解があったからに他なりません。この場を借りて改めてお礼申し上げます。

手探りの中で始まったオンライン授業だったため，さまざまな苦労や表には出てきていない小さなトラブルは多数あったと思います。私の場合も，オンデマンド授業用の資料の準備には多大な時間を要し，春学期は連日朝から晩まで作成をしてなんとか間に合うという自転車操業状態でした[2]。また，実習科目においては，普段は学生の進捗度合いを見ながらこちらから声をかけてフォローをしていたのですが，それができなかったため最終課題の得点が全体的に低いという問題も生じました。ただし，どの科目においても課題の提出率は高く，単位の取得率は高かったように思われます。

一方，学生にオンライン授業について尋ねてみると，教員側と同様に賛否両論だと感じました。特にオンデマンド型については，自分のペースで好きなときに学習でき，理解できなかったところは再学習できるという点がよかったようです。ただし，自分でスケジューリングするのが難しく課題をため込み大変だったという声もありました。

---

[1] 本巻頭言執筆中の2021年12月上旬において，日本での流行はかなり押さえ込めているが，感染力の高いオミクロン株の登場により再流行の危険性が高まっている。

[2] 秋学期の分は夏休みに半分ぐらいは作成できていたのでなんとかなった。

このようにオンライン授業についてはさまざまな意見が見受けられますが，新型コロナウイルスの流行を機に強制的に導入に踏み切らされたのは，大学としては変革の機会だったのかもしれません。社会全体を見てもこの機にオンライン化されたものは多く，オンラインミーティングやオンラインで業務の実施などもますます増えていくと言えます。そのため，大学時代にこれらのシステムに触れたことは今後の糧になると思います。

今号への投稿はいずれもそのようなオンライン授業に関わりの深いものとなっております。是非ご一読頂ければと思います。

最後になりましたが，2020年9月30日を以て情報メディアセンター所長の第1期目の任期を皆様のおかげで無事終えることができました。そして，新型コロナウイルスの影響下で体制を変えるのは望ましくないと言うことで，力不足ながら引き続き情報メディアセンター所長を務めさせていただくことになりました。今後も事務スタッフやICT委員の皆さんの御協力を得ながら努めて参りたいと思っております。よろしくお願いいたします。

目　　次

はじめに ……情報メディアセンター長：岩田　員典

論文

情報リテラシーのための統合システム構築と運用－ Microsoft Excel・Word
自動採点システムを含む － …… 松井　吉光，岩田　員典 …… 1

JARファイルを対象とする自動採点機能を有した
Javaプログラミング実習用システムの構築と運用 …… 毛利　元昭 …… 11

情報教育実践報告

Moodleワークショップを利用した学生による相互評価の実践 …… 岩田　員典 …… 37

研究ノート

2019年度と2020年度におけるMoodleアクセス状況調査 …… 森野　誠之 …… 51

センターだより

ICT委員会　会議報告 …… 61

情報メディアセンター主催行事 …… 62

Moodle（LMS）運営業務報告（2019・2020年度） …… 63

ICT委員会構成員 …… 69

情報メディアセンター沿革・歴代所長 …… 70

編集後記 …… 71

原稿募集要項

執筆要項

情報リテラシーのための統合システム構築と運用
― Microsoft Excel・Word 自動採点システムを含む ―

松井　吉光（法学部）・岩田　員典（経営学部）

要旨

情報技術の発展にともない，今日においてはワープロソフトウェアや表計算ソフトウェアを取り扱う技術は必要不可欠である。このような時代背景においてPCの操作に不慣れな学生に，これらの技術を身につけさせるためには繰り返し問題を解きながら理解を深めてもらうのが有効である。しかし，これを実施するには多数の練習問題を用意し，それを採点し，採点結果を学生に提示することが必要である。この方法を受講者数が多い初年次対象の情報リテラシー科目で実施する場合，問題の準備・採点・結果の提示のために教員の負担が大きくなりすぎ実施が困難である。そこで，本論文ではこのような教員の負担を軽減するとともに，受講者の学習効果を上げることを目指したOffice Open XML形式のExcel・Word自動採点統合システムの構築について述べる。さらに受講者の課題の正答率の上昇から本システムの有効性についても言及する。

キーワード：情報教育，e-learning，自動採点

1. はじめに

近年の情報技術の発展にともない，大学や企業においては電子メールの利用をはじめとしてワープロソフトウェア・表計算ソフトウェアなどの技術が当然求められる時代となっている。さらには，これらのソフトウェアを活用できるスキルは情報リテラシーの一部として考えられるようになり，この活用は大学や企業においてできて当たり前という状況にある。したがって，これらを利用できる最低限の技術を身につけることが肝要である。特にワープロソフトでは，単に文字を修飾するだけでなく，レポートなどを作成するための文書の構成ができるようになることが求められる。また，表計算ソフトでは関数を利用したデータの分析や，グラフによるデータの可視化の技術が重要視される。そして，PC（Personal Computer）に不慣れな学生がこれらの技術を身につけるためには，繰り返し類似の問題を解きパターンになれることが必要である。

これを実行するための課題となるのが，どのように問題を用意し，その問題

をどのように採点するかである。特に初年次の学生向けに情報リテラシー科目を設定し，受講者が大多数の場合は問題の採点だけでも教員の大きな負担となる。また，学習効果を高めようと問題数を増やすと受講者の学習意欲を維持するのが困難である。そこで，教員の負担を減らしつつ，学生へ採点結果を伝えるためのタイムラグを減らすため，採点結果や間違いを即座に指摘するExcel・Wordの自動採点プログラムを開発した。さらには，学生の学習意欲を高めるために問題の達成度やランキングを表示する機能も含んだ統合システムとして構築し運用している。本論文ではこの統合システムについて述べ，課題の正答率の上昇から本システムによる学習成果の評価も行う。

2. 背景

本学では2006年度から1年生の情報リテラシーレベルを向上させることを目的として情報リテラシー･入門（前期），応用（後期）を設定している。この科目を設定したのは，当初文系の学生にはPCの操作に不慣れな学生が多かったためであった。その後PCの普及が進み徐々にその傾向が弱まったかに思われたが，近年はスマートフォンやタブレットの普及によりPCを所持していない学生もまた増えており，10年前よりもPCに慣れていない学生が増えていることからも重要な科目となっている。そのため，本科目は新入生のほぼ全員が受講することで，約1500名の受講者がいるという超大人数科目となっている。前期に開講している情報リテラシー・入門では，受講者は約1500名であり担当教員は10名であるため，平均して約150名の学生を担当していることになる。したがって，学生一人あたりが100題の課題を解くとすると15万題もの解答が提出されることになる。例えば，一問あたりの採点に30秒かかるとすると一人の学生の全解答を採点するのに50分かかる。したがって，全ての解答の採点には教員一人あたりおおよそ125時間費やすことになる。また，採点の順番が最後になった学生にその結果を提示するまでの時間のずれが大きくなる。それにより，採点結果を示されたときには学生が問題の内容を忘れてしまっている可能性がある。

このような状況に対応するために2006年度の前期においては，本科目は完全な自学自習形態とし，市販教材を用いた課題提出による単位取得という方式を採用した。しかしながら，従来のe-Learningに関する市販教材は，画面の内容を読解する講義型であり，課題試験も知識を択一形式で出題するのみである。そのため，受講者としては，その場で正答すればよいという一過性の学習になり，必要最低限の技術が身につかないという問題が生じた。

そこで，市販教材では情報リテラシーレベルの向上は困難であると判断し，2006年度の後期に対面型演習形式で自学自習のできるe-Learningシステムの独自開発に取り組んだ[1]。このシステムではMicrosoft Officeを利用して学生が作成した解答を自動採点し，その結果を保存する。これにより，教員の負担を減らしつつ，学生へ採点結果を伝えるためのタイムラグを減らすことができている。

ただし，このシステム開発時にはExcel，WordのファイルはMicrosoft Officeの独自形式であったため，ExcelはHTMLファイルとして保存したものを，WordはWord XMLファイルとして保存したものを採点する採点プログラムを利用していた。また，このシステムは単に成績を保存するシステムであった。その後，Office Open XML[2, 3]に対応した採点プログラムを含むシステムに改良した[4]。本論文ではこのシステムを基に，学生の学習意欲を高めるために問題の達成度やランキングを表示する機能も含んだ統合システムについて説明する。

3. 関連システム

本論文で開発したシステムと関連が深いと考えられるMicrosoft Officeの自動採点ソフトはファイル形式がOffice Open XMLになってからは主に以下の2種類に分類できる。

- VBAを用いてマクロにより採点
- 外部プログラムによりXMLを解析することにより採点

VBAを用いた方式はそのファイルのみで採点ができるという利点がある。一方で，外部プログラムを利用する方法は，解答例から採点基準ファイルを作成するという手段をとることができるためVBA方式よりも問題を多数用意しやすいという利点がある。

大曽根[5]，中村[6]や渡邊[7]は最初のVBAを用いる方法を採用している。一方，石川ら[8]はVisual Basic.Netを利用した後者の方式となっているが，2006年のためXMLを解析しての採点ではない。我々は多数の問題を用意する必要性もあり，外部プログラムで採点する手法を採用している[9, 10]。

4. 自動採点統合システム

4.1 システムの構成と基本動作

本システムはExcel, Wordを利用して学生が作成した課題を自動採点し，その結果を保存・管理する統合システムである。その概要は以下の通りである（図1）。

1. ユーザはシステムにブラウザ経由でアクセス
2. システムのインターフェイスを介して表示された課題を適宜ダウンロード

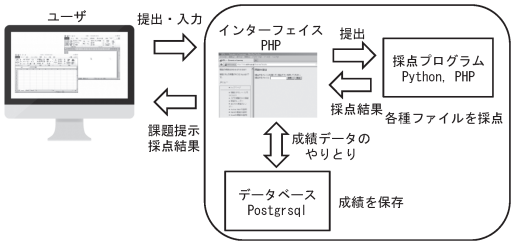


図1　システムの構成

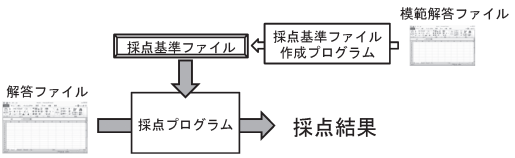


図2　採点基準ファイルと採点

3. ユーザは問題をダウンロードしたPCで解答を作成
4. 解答をシステムにアップロード。システムはアップロードされたファイルを自動採点エンジンによって採点し，その結果をデータベースに保存
5. 採点結果をブラウザに表示

また，本システムはタイピングについても課題を用意しており，タイピングに関してはブラウザに入力する方式になっている[11]。

4.2　Excel・Word採点プログラム

自動採点プログラムはExcel，Wordで作成されたファイルを採点することができる。採点に際しては，設問ごとに採点基準ファイルを用意することでプログラムを修正することなくさまざまな設問を採点できる。さらに採点基準ファイルは，模範解答ファイルから半自動で作成することができる[9, 10]（図2）。ここで半自動と記述しているのは，別解などは別途設定が必要だからである。これらの機能を活かしExcelを124題，Wordを126題を用意することができている。

統合システムとしてこの自動採点プログラムは4.1で説明したように，アップロードされた解答ファイルを採点する。つまり，ユーザが設問ファイルをダウンロードし，そのファイルに解答を入力する。そして，解答ファイルをアップロードすることで採点が行われる。採点結果は即座にユーザにブラウザを通じて提示される。これらの処理のためにインターネットに接続された環境が必要である。

ただし，インターネットに接続する必要があるのは採点や成績の管理を統合システムによって行っているためである。採点プログラムや必要なファイル一式を導入したPCならばスタンドアロンで採点をすることもできる。

4.3　学習効果と意欲の向上のための機能

受講者が自己学習をしやすくするために，PDFや動画による解説を公開する仕

組みを備えている（図3）。これらは全ての設問に対して準備する必要はなく，教員側が説明を追加したい問題に限ることができる。また，公開方法は後述（4.4）の設問IDをファイル名として設定して，所定のディレクトリに配置するだけである。本学では設問の種類ごとに解説や動画を用意している。

受講者の学習意欲を維持するために図4や図5のように課題の達成率やランキングを表示する機能も備えている。達成率については，本学で開講している情報リテラシー・入門，応用の単位取得条件の一つになっているため，常に学生が気にしていることもあり簡単に見られるようにしている。一方でランキングは達成率に基づいて全受講者中の順位を表示するようになっている。これにより他の受講者との進捗度合いを比較することができ，学習意欲の維持に効果が期待できる。

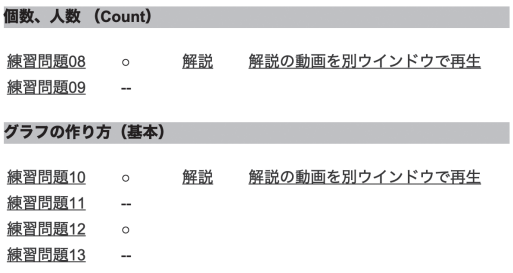


図3　PDFと動画の公開

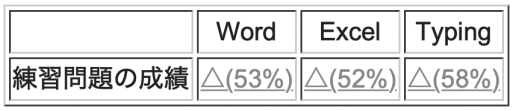


図4　達成率の確認

図5　達成率によるランキング

4.4　講義担当者向け機能

本システムは，講義で利用することも想定して以下の機能を有する。

1. 設問作成・編集
2. 出題期間管理
3. ログの参照，成績の参照とダウンロード
4. 出席管理
5. 不正防止

設問作成機能は4.2で説明したように模範解答ファイルを基に半自動で作成が可能である。特にExcelについては別解の設定なども本システム経由で行うことができる[10]。また，タイピングに関しても解答例となるテキストファイルをアップロードするだけである。

これら設問は「α -x-yy-zz」のように設問IDを付けて管理している。ここで「α」は「T」，「E」，「W」のいずれかとなり，それぞれタイピング，Excel，Wordの課題であることを表している。

「x」は本学では出題するコースを表現

するのに利用しており，「0」ならば入門の練習問題，「1」ならば応用の練習問題となっている。

「yy」は設問番号に関わる値である。ただし，これは管理上の設問番号であり受講者に表示する番号は別途設定が可能である。

また，「zz」は類題番号である。これは，「yy」で指定された設問番号に対して複数の設問を用意した場合に利用する。例えばExcel01という設問があった場合に，受講者AにはE-0-01-00を割り当て，別の受講生BにはE-0-01-01と別の設問（類題）を割り当てることができる。この設定により，ユーザごとに異なる設問を出題することもできる。

この設問IDは出題期間，出席管理，成績にも関わる。その際には「zz」の値は使わずに，「α -x-yy」（例えばE-0-01）を使う。このように管理することで，どの類題を解いたかに関係なく設問番号に対して解答したと管理することが可能になっている。

2の出題期間管理機能は，練習問題の出題期間を講義の実施に合わせてクラスごとに設定することができ，設定された日時になると自動で設問にアクセスできるようになる。以前は講義の実施回に合わせて順番に設問を手動で公開していた。この手動での設定は最も進捗が早いクラスにあわせた公開となっていたため，休講などで進行が遅くなっているクラスでは，授業に全く関係ない設問まで公開されているという問題が生じていた。そのため，クラスごとに設問の公開開始日時を設定できるようにした。また，現時点では利用していないが，公開終了日時も指定することができる。

ログと成績は，システムから全員・クラス・ユーザIDなどを指定してそれぞれ確認することができる。また，成績はCSV（Comma Separated Values）ファイルとしてダウンロードすることもできる。

4の出席管理は，「実施期間」，「IPアドレスによる制限」，「アクセスコード」を設定することで，授業に来ている受講者のみアクセスできる設問を用意することができる。これにより，その設問を提出しているかどうかで出席を管理することが可能である。

不正防止機能は，各設問に対する機能と，出席に関わる機能の2つに分けられる。各設問に関する不正防止機能は，他の受講者から解答ファイルをもらって提出するということを防ぐために実装している。本学では出題されている練習問題の半分以上を提出することが単位取得条件の一つとなっている。しかし，練習問題の数が多く締め切りに間に合わない場合に，他の受講者からファイルをもらって提出するという不正行為が散出していた。そこで，ユーザID・設問IDなどを基に計算したハッシュ値を設問ファイル

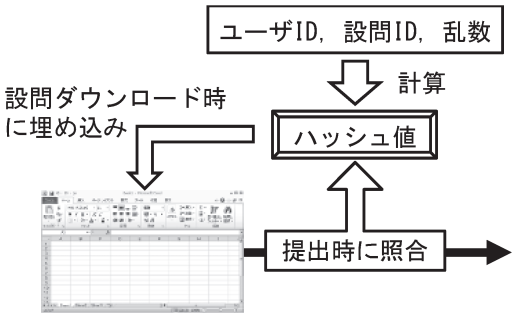


図6　ハッシュ値の埋め込みと照合

中に埋め込み，解答ファイルを提出した際にその値を照合することで他の受講者からファイルを受け取っていないかどうかを判別するようにしている（図6）。

また，情報リテラシー･入門，応用は単位取得条件として原則欠席回数は1回以下と定めている。そのため，授業に来られない受講者が，出席している友人に代理で提出を頼むということがあった。そこで，授業時間中に同じIPアドレスから複数の受講者による出席用の設問提出があった場合に，不正行為が行われた可能性があるとして講義担当者にメールで通知する機能を実装した。これにより，代理出席を検出できるようになっている。

5. 採点プログラムと運用実績の評価

本システムは以下の構成により運用しており，本章の評価もこのシステムでおこなった。

CPU：Intel Xeon E3-1270 v6 3.80GHz

メモリ：8GB

OS：CentOS Linux 7.9.2009

各種ソフトウェア：

Apache 2.4.6, PostgreSQL 9.6.4

Python 3.6.2, PHP 7.3.3

PhpSpreadsheet 1.6.0

5.1　Excel・Word採点時間

Excelを124題，Wordを126題をそれぞれ10回採点した時間を計測した。時間の計測はコマンドライン上から「date+%s%N」を使用して，採点実行前と実行後の時刻の差により求めている。そのため，前述のApacheとPostgreSQLはこの採点時間計測には関係ない。また，Excelの採点にPHPとPhpSpreadsheetを利用し，Wordの採点にはPythonを利用している。

採点時間の平均値，標準偏差と中央値を表1に示す。なお，計測時はns（10^{-9}秒）で計測しているが，結果を見やすくするためにμs（10^{-6}秒）で記載している。

表1　採点プログラムによる採点時間

Excel	平均値	41.41 μs
	標準偏差	2.163 μs
	中央値	41.16 μs
Word	平均値	14.72 μs
	標準偏差	0.1381 μs
	中央値	14.71 μs

これらの結果から分かるように本採点プログラムは充分高速に採点できていることが分かる。

5.2　運用における学習効果の評価

本学の情報リテラシー・入門では第2回の授業でExcel・Wordの実力テスト（事前テスト）を実施している。これは，受講生がExcelやWordを受講前にどの程度活用できるのかを把握するためである。この事前テストは，その学期で教える予定の内容について出題している。したがって，もしも受講者の大半が事前テストで満点を取るようならば実習内容を見直す必要がある。ただし，これまでそのような状況は生じていない。

また，学期の最後には同じ内容で事後テストを行っている。これを行うことで，受講者がExcelとWordに関する知識と技能を概ね身につけることができたかどうかを判定することができる。

なお，2020年度はコロナウィルスのため，これらのテストが実施できなかったため2015年度～2019年度の5年分のデータを用いて分析をおこなった。このデータの受講者データ7548名のうち事前・事後テストの両方を受けた受講者（Excel 6355名，Word 6481名）を対象としている。

受講生の全体の習熟度の伸びを評価するためにHakeによる規格化ゲイン（Normalized Gain）[11]を用いる。規格化ゲインは事前テストの正答率の平均値を<%*Pre*>，事後テストの正答率の平均値を<%*Post*>とすると以下のように定義される。

$$G := \frac{(<\%Post>-<\%Pre>)}{(100-<\%Pre>)}$$

これは受講者の学力の伸びを示す指標として提案され，広く用いられている。この指標の最大値は<%*Post*>が100％のときであり，G=1である。Excelに関しての規格化ゲインを表2に，Wordに関しての規格化ゲインを表3に示す。これらの表の値から，平均点は事前テストと事後テストで上昇し，規格化ゲインも非常に高い値を示している。また，標準偏差が事前テストはいずれも大きな値を示していたのが事後テストでは大幅に小さくなっている。これは，事前テストでは全くもしくはほとんど解けなかった受講者が，事後テストでは完答できていることを示している。

授業ではExcelやWordについて講義を行っているため，この結果がすべて本システムによる成果とはいえない。しかし，本システムがなければ，いずれの年度においても1500名超の受講者に対して多数の設問を提供することは難しく，習熟度がここまで高まらなかった可能性がある。その面から見ても本システムは評価できる。

表2　Excelの規格化ゲイン

	事前テスト	事後テスト	G
平均値	65.90	98.62	0.9597
標準偏差	41.20	9.351	
中央値	90.48	100	

表3　Wordの規格化ゲイン

	事前テスト	事後テスト	G
平均値	6964	97.33	0.9122
標準偏差	28.76	28.76	
中央値	73.68	100	

6. まとめと今後の課題

本論文では教員の負担を軽減するとともに，受講者の学習効果を上げることを目指した自動採点統合システムの構築ついて述べた。本システムの評価として，ExcelとWordの採点にかかる時間を計測し，教員が採点するよりも非常に高速に採点できることを示した。さらに受講者の正答率の上昇を規格化ゲインにより評価した。それにより，本システムが受講生の習熟度向上に貢献していることを示し，その有効性について考察した。

今後の課題としては，開発環境と運用環境の統一と配布の仕組みの確立がある。これまでにも，ライブラリなどのアップデートのタイミングにより，開発環境と運用環境で利用しているバージョンが異なるという問題が生じることがあった。また，本学だけでの利用から学外機関にも利用してもらうためには配付方法の確立も重要である。これらの問題を解決するためにDocker上で動作するようにシステム全体を変更し，GitLabにより公開する予定である。

謝辞

本研究の一部は，愛知大学名古屋一般教育研究室事業「HITs用学習コンテンツの作成および研究」によるものである。

注

本論文で使用しているIntel Xeonは，アメリカ合衆国及びその他の国におけるIntel Corporation又はその子会社の登録商標または商標である。Microsoft, Microsoft Office, Word, Excel は，Microsoft Corporationのアメリカ合衆国及びその他の国における登録商標または商標である。CentOSは，CentOS ltd.の登録商標または商標である。

参考文献

[1] 岩田員典，功力由紀子，齋藤毅，谷口正明，長谷部勝也，松井吉光，古川邦之，「Excel，Word 自動採点システムHITsの構築と運用」，愛知大学 情報メディアセンター紀要「COM」，Vol.20，No.1，第35号，2010/3，pp.11-23

[2]「Office（2007）Open XML ファイル形

式の概要」，2015/6，
https://docs.microsoft.com/ja-jp/previous-versions/office-development/aa338205 (v=office.12) ?redirectedfrom=MSDN, 2021/2/26 参照
[3] “Standard ECMA-376 Office Open XML File Formats”, 2016/12, https://www.ecma-international.org/publications-and-standards/standards/ecma-376/, 2021/2/26 参照
[4] 岩田員典，松井吉光，長谷部勝也，谷口正明，池森均，梅垣敦紀，齋藤毅，澤田貴行，土橋喜，中尾浩，西本寛，古川邦之，毛利元昭，「情報リテラシーのためのWord，Excel自動採点システムの構築と運用」，平成25年度教育改革ICT戦略大会，2013/9，pp.294-295
http://www.juce.jp/archives/taikai_2013/d-14.pdf
[5] 大曽根匡，関根純，丹保歩子，「Excel用標準テストのための自動採点プログラムの開発とその実施結果の報告」，専修大学情報科学研究，Vol.35，2015/3，pp.19-41，
https://senshu-u.repo.nii.ac.jp/?action=repository_uri&item_id=7846&file_id=32&file_no=1
[6] 中村邦彦，「Microsoft Office課題の自動採点プログラム」，香川大学経済学部研究年報，51，2012/3，pp.1-43，
http://shark.lib.kagawa-u.ac.jp/kuir/file/27706/20150809142518/AN00076010_51_1_43.pdf
[7] 渡邊光太，「Microsoft excelによる試験採点システムの作成」，城西情報科学研究，Vol.21，No.1，2011/3，pp.37-46，
https://libir.josai.ac.jp/il/user_contents/02/G0000284repository/pdf/JOS-InfoBull-2105.pdf
[8] 石川千温，中村永友，渡邊慎哉，小池英勝，「学習者の自学自習を支援するエクセル用自己採点ツールの開発」，2006PCカンファレンス，2006/8，pp.321-322，
https://gakkai.univcoop.or.jp/pcc2006/papers/pdf/185.pdf
[9]長谷部勝也，松井吉光，谷口正明，「HITsにおけるWord文書の採点プログラム2019年度版の開発」，愛知大学　一般教育論集，第56号，2019/3，pp.31-39
[10] 岩田員典，「Office Open XML形式のExcelファイル自動採点システム」，愛知大学情報メディアセンター紀要「COM」，Vol.29，No.1，第44号，2019/3，pp.1-15
[11] 松井吉光，「HITsにおけるTyping練習用採点プログラムの改良」，愛知大学情報メディアセンター紀要「COM」，Vol.30，No.1，第45号，2020/3，pp.15-22
[12] Richard R. “Hake”, American Journal of Physics, Vol.66, Issue 1, 1998, pp.64-74

JARファイルを対象とする自動採点機能を有した Javaプログラミング実習用システムの構築と運用

毛利　元昭（愛知大学経営学部）

要旨

プログラミングの実習系授業の実施において，担当教員側の負担軽減や学習者側でのテストの重要性から，自動採点機能を有した実習用システムやウェブサービスを利用するケースは多い。しかし，既存のそれらはソースコードを提出させる形にしている。そのような形式を単純に用いた場合，学習効果の低下やシステム負荷の増大を招く恐れがある。それらを抑制すべく，筆者が本学経営学部で担当しているプログラミング系授業では，JARファイルを対象とするJavaプログラミング実習用システムを独自に構築・運用している。本システムは実行可能JARファイルを受講者のローカルな開発環境で作成させることを前提とするが，本稿では，なぜ本システムによって問題を抑制できるのか，どのように設計・実装・実現しているのかについて説明するとともに，他の様々な機能について説明する。また，受講者の反応を中心に本システムの運用の状況について述べる。

キーワード：自動採点，プログラミング学習，Java，GUI，PHP，シェルスクリプト

1. はじめに

プログラミングの実習系授業の実施において，担当教員側の負担は大きい。資料・教材作成と講義だけでなく，課題の細かな出題，受講者の課題進捗状況の把握，採点とフィードバック，質問やトラブルへの対応などが求められる。しかも，その多くが個別対応と即時性を求められ，授業中の担当教員とアシスタントは，講義を中断して対応に追われ続けることになる。情報工学や情報科学を専攻していない初心者50～60人規模の複数クラスを単独で担当するような場合，授業を正常に実施するには何らかの効率化・自動化が必要と言える。

また，プログラミングの学習において，作成したプログラムが想定通りの動作を行うものであるかを確認すること，いわゆるテストは必須と言える。テストデータと期待される結果を担当教員側で用意できる場合は多いが，簡単なコーディングでも苦戦する初心者にとって，テストを正しく実施することは容易でない。開発環境に付属するテスト用のツールなどを利用してもらうおうとしても，

さらなる学習コストを必要とするため，やはり初心者には難しい。

こういった背景から，プログラミングの実習系授業の実施において，自動採点機能を有した実習用システムを授業担当者が構築・運用するケースがある。例えば，名古屋工業大学のCAPES[1)]や早稲田大学のWaseda Online Judge[2)]がある。同様に，近年には様々なプログラミング学習用ウェブサービスが開始され，そのほとんどは課題提示と自動採点（オンラインジャッジ）機能を有し，自身の課題進捗状況を保存・把握できるものもある。簡易的なプログラム開発環境を有しているものもある。日本人の利用者が多いものとしては，前述のAizu Online Judge[3)]やProgate[4)]，paiza[5)]などがあり，世界的にはAtCoder[6)]が有名である。しかし，授業で用いられる実習用システムは授業に閉じたものであるため，学外者には利用できない。また，プログラミング学習用ウェブサービスは内容のアレンジができない。そのため，筆者も本学経営学部で担当しているプログラミング系授業の実習用システムを独自で構築・運用することとした。

ここで，既存の多くの実習用システムやプログラミング学習用ウェブサービスは，ソースコードを提出させる形にしている。具体的には，ウェブページ上に穴埋め式のソースコードと条件や期待される働きが提示され，解答欄に解答を記入あるいは作成したソースコードのファイルを投稿し，文字列やプログラムの実行結果を正答例のものと比較することで，点数化や合否判定を行う。しかし，このような方法を単純に用いた場合には下記のような問題が生じる。

1. 受講者がプログラムの全体像を把握・コントロールする機会の減少
2. 受講者のコーディングに関する自由度の減少
3. コンパイルや実行の機会の減少
4. コンパイル不可能な状態での提出の増加
5. システム負荷の増加

これらの問題を抑制するために，筆者はローカルの開発環境にて作成された実行可能JARファイルを提出させることを前提とした，自動採点機能を有するプログラミング実習用システム[注1)]を考案・構築した。本稿では，主にそのシステム設計・構成や運用の状況を説明する。

2. 受講者に実行可能JARファイルを提出させる意義

JARとはJava Archiveの略であり，Javaのクラスファイルやデータファイ

注1) ソースコードや課題バンク等は下記のURLにて公開している。

https://halo.aichi-u.ac.jp/~mouri/works/programming_training_system_java.zip

ルなどをZIP圧縮でひとまとめにした形式である。実行可能JARファイルとは，実行用のマニフェストファイルを内に含んだJAR形式のファイルである。Javaがインストールされた環境であれば，「java -jar」コマンドとともにファイルを指定することでプログラムを実行できる。

本システムにおいて特徴的な点は，受講者のローカル環境にて実行可能JARファイルを作成させてから，そのファイルを提出させることである。このような形式をとる自動採点システムは，筆者が調べた限りでは見当たらなかった。既存のプログラミング実習用のシステムでよくみられるのは，ブラウザ上にソースコードを入力させる形である。しかしこの形は，第1章で挙げたような問題を抱えている。ここでは，実行可能JARファイルを提出させることで，なぜそれらの問題が解決されるのかを述べる。

自動採点がプログラムの実行を伴わない形であれば，「正解となる文字列を入力する」ことと等価である。ソースコードに対してリッチな自動採点を行う場合，字句解析・構文解析とパターンマッチングなどを利用することとなる。このような形は，計算機リソースと時間が必要になることに加え，自動採点スクリプト等の作成に高い技術力と多大な労力が必要になる。筆者を含め，言語処理などの分野を専門としない者には荷が勝ちすぎる状況と言える。他方で，簡易的な自動採点を行う場合，空白や改行を除いて比較差分を評価することとなる。このような形は，受講者のコーディングに関する自由度がほぼ無い。答案が記号・単語あるいは短文になるような工夫がなされていることが前提であり，「デバッグせよ」「○○ができるように改良せよ」のような，プログラムの全体像を把握・コントロールさせるような大まかな指示による課題の答案を，実質的に採点できない。つまり，応用力を養うことが目的の課題の出題が困難である。

自動採点がプログラムの実行を伴う形であれば，入出力の関係は「仕様」つまり厳密にしつつ，コーディングに関する自由度は高くできる。よって，応用力を養うことが目的の課題でも自動採点可能なものが作りやすい。しかし，ソースコードを提出させる形であれば，サーバ側でコンパイルが必要となる。コンパイルには計算機リソースと時間が必要であり，システム負荷を増大させるため，多数の答案提出が重なった場合などにリクエストのタイムアウトが生じやすくなる。タイムアウトした場合には採点結果を正しく示せないため，答案の再提出が必要となり，その結果，さらなるシステム負荷が生じることとなる。それ以上に問題なのは，実行を受講者の主体で行う必要がないため，受講者からコンパイルや実行の機会を奪うことにつながる点である。

プログラミングにおいて自らの手によるテストは重要であり，学習という観点からも試行錯誤は重要である。そもそもコンパイル不可能なソースコードが提出される可能性も大きく，デバッグしにくい状況は避けるべきである。なお，近年にはブラウザ上で利用できるクラウド開発環境も現れてきたが，導入が困難な上に様々な制約やリソースの問題を抱えており，授業での利用は現時点では考えていない。

一方で，JARファイルを提出させることを前提とした場合，話は変わってくる。筆者の授業では，開発環境のEclipse[注2)]を受講者に利用させている。このEclipseは，実行可能JARファイルの作成（エクスポート）に際し暗に実行を必要とする。エクスポートには起動構成が必要なのだが，ソースコードから実行させることで，起動構成が自動生成されつつコンパイルが行われるのである。よって，受講者は実行とそれに付随するコンパイルを，提出のたびに自然に繰り返す[注3)]ことになる。つまり，データ入力を含め実行の様子を主体的に確認することにつながる。また，コンパイル時にエラーが生じた場合はエクスポートされないため，コンパイル不可能な状態での提出が防止できる。もちろん，Eclipseが有するハイライト機能や入力補完機能など，リッチな開発環境としての機能をローカルのリソースで利用するため，コードの誤りを避けつつデバッグがしやすい上に，システム負荷は総合的に小さい。

これらのメリットを成立させる要因は，JARファイルがOSに依存せず実行できる点である。Javaは仮想マシン（VM）上で中間言語ファイルを実行する仕組みをとっており，受講者のローカル環境（Window，Mac，Linux問わず）で作成された実行可能JARファイルであっても，多くの場合サーバ側（本システムの場合はLinux）でそのまま実行できる。なお，JavaFX[注4)]など外部ライブラリが必要な場合やVMのバージョン違いによっては若干の工夫が必要であるが，現行の演習課題およびシステムでは解決済みである。具体的には，受講者にはEclipse上でJava SE-1.8の形式でプロジェクトを作成・実行・エクスポートしてもらい，サーバ側では実行時にオプ

注2) Javaの主要な統合開発環境。日本語化ツールPleiadesとともに無償利用できる。

注3) 実際には，起動構成が一旦生成されたらその後のエクスポートの際に実行は不要であるが，受講者にそれを知らせないことで実行が習慣化される。

注4) Java VM上で動作するGUIライブラリ。現在はJavaに同梱されなくなったため，OpenJFXを導入・利用する形となり，初心者にとっては利用・保守し難くなっている。

ションを付け加えている。

実行可能JARファイルによるメリットは他にもある。「javap[注5)]」コマンドを用いてJARファイルを解析することで，プログラムの構造を確認できる点である。コンパイル後，つまり字句解析・構文解析とそれらの整理が行われた後であり，ソースコードと比べて表現の差異がある程度吸収されるため，構造が同じであれば正答例と比較しやすい。すなわち，入出力の関係だけでなく，プログラムの構造も採点の対象にできる。もちろん，採点者には「javap」コマンドに関する知識や解析結果の読解・利用技術が必要となるが，複雑な状況を求めなければ対応できるケースも多い。筆者の場合，標準的には，クラスやメソッドの情報などを抽出し必要に応じて正答例のものと比較する形で自動採点の際に利用している。本システムでの具体的な利用方法については第5章で説明する。

3. システム設計

3.1 システム要求

JARファイルを対象とする自動採点機能を有したJavaプログラミング実習用実習用システムを構成するにあたり，当初から考慮したシステム要求は以下の通りである。

1. 履修者がウェブブラウザからアクセスでき，課題の表示や提出，評価の確認もウェブブラウザ上で可能
2. 利用にはログインを必要とし，テスト用ユーザを除いて学内の認証サービスにて認証
3. 記述式の課題も表示や提出ができ，正答例を別途用意することで自動採点にも対応
4. JARファイルが提出された場合，ファイルの解析および実行を自動で行い，その結果をファイルとして保持
5. 自動採点の場合，提出から結果の表示までを20秒以内とし，その間に60人程度がアクセスしてもシステムダウンが発生しない（タイムアウトは許容）
6. 管理者の画面から，提出されたファイルや答案の確認，採点の修正や手動採点が可能
7. 管理者の画面から，クラスごと・授業の回ごとに課題の公開・提出期間等の設定が可能

前述の要求が満たせるように，ハードウェアの選定と環境構築およびシステム設計を行っている。現行機のハードウェアと環境は表1の通りである。なお，当初のマシンはスペックやバージョンが現行

[注5)] JavaのSDKに付属する，バイトコードで書かれたクラスファイルを解析・逆アセンブルして人間が理解できる文字列に変換するツール。

表1　システムのハードウェアと環境

本体	PC/AT互換機
CPU	Core i5 760 2.80GHz
RAM	4 GB
ストレージ	HDD 320GB
ネットワーク	1000Base-TX
OS	Lubuntu 20.04 64bit
ウェブサーバ	Apache 2
サーバ側処理	PHP 7, Bash 5
ブラウザ側処理	jQuery 3
データベース	Maria DB 10
Java環境	OpenJDK 16, OpenJFX
画面キャプチャ	Xvfb, ImageMagick

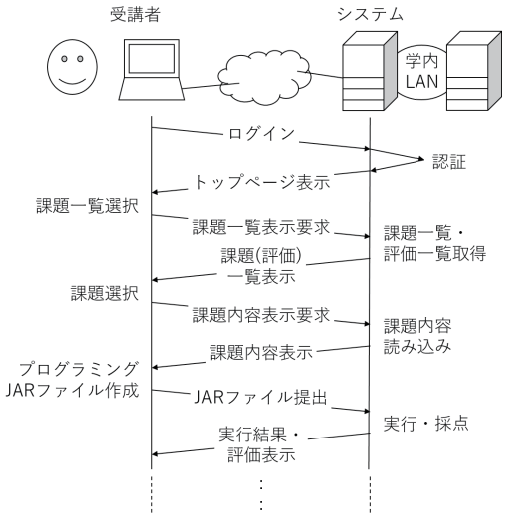


図1　受講者によるシステム利用の流れ

より大きく劣るが，HDDの障害を除き，運用中の過負荷によるシステムダウン等は発生しなかった。

受講者によるシステムの利用の流れは図1のようなものを想定した。まずログイン認証を行い，受講者として登録されていればトップページが表示される。トップページには各機能や資料へのリンクが示されており，受講者が課題一覧を選択すると，その時点に出題されている課題の一覧が表示される。このとき，進捗状況を把握できるよう，各課題の評価を記号化したものが各課題の内容へのリンクとして表示される。受講者が課題を選択すると，それに対応した課題内容が表示される。受講者は表示された条件やサンプルプログラムをもとにプログラミングを行い，ローカル環境にて実行可能JARファイルを作成する。受講者が課題内容の画面からJARファイルを提出すると，サーバ側で実行や採点が行われ，実行結果や評価が表示される。その後，受講者はページを戻ったり課題一覧を表示させたりして，課題への再挑戦や次の課題への挑戦を行っていく。

前述の要求に基づくシステムの具体的な要件および追加機能は詳細多岐にわたるため，必要に応じて後述する。

3.2　フォルダ構成

本システムはウェブサービスとして運用するため，ウェブ上から直接アクセスできる表層部と，そうでない内部に分かれたフォルダ構成とした。概要は図2の通りである。

表層部に関しては，PC内の著者用の

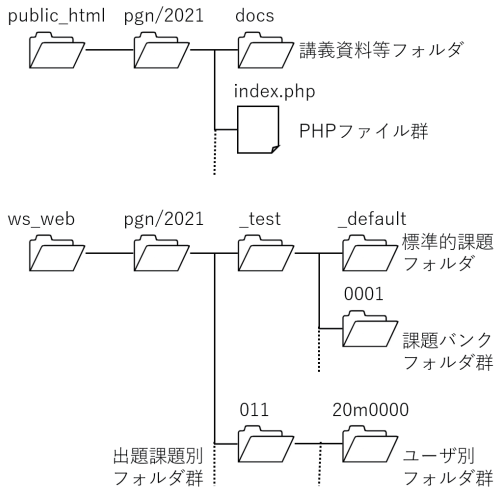


図2　システムのフォルダ構成

ホームフォルダにあるフォルダ「public_html」内に「授業略称・年度」のフォルダが用意してあり，その下にPHPファイル群を置いてある。「index.php」がトップページとなっており，そこから各ページへのリンクをたどる形となる。また，簡便のため，講義資料等は「docs」というフォルダに別途置き，トップページにもリンクを載せている。表層部とそれに付随する各機能は主にPHPの形でモジュール化して用意するが，自動採点そのものはPHPでは行わない。内部による処理結果を受けて，結果を表示するにとどめる。なお，PHPファイル群については第章で詳細を述べる。

内部に関しては，これもホームフォルダ下のフォルダ「ws_web」内に「授業略称･年度」のフォルダが用意してあり，その下に出題課題別のフォルダ群と，各課題内容と採点用スクリプトがフォルダ群（課題バンク）として収められた「_test」がある。

出題課題別のフォルダ群は，出題する課題を設定する際に，その課題の番号で自動作成される。例えば，第02回の授業の3つ目の課題であれば「023」となる。この出題課題別フォルダの下にさらに受講者ごとのフォルダを自動作成し，提出されたJARファイルやその実行結果，自動採点のログ，その他のファイルなどを，それぞれ指定の名前で保存する形である。

課題バンクにおいては1つの課題につき1つのフォルダとしており，課題の識別子をフォルダ名としている。それぞれのフォルダに，データベースへの登録用設定ファイル，課題の説明，ソースコード，正答例，自動採点スクリプト，その他のファイルなどを指定の名前で収めてある。また，課題バンクとは別に「_default」フォルダを用意してあり，ここには標準的な課題の説明と自動採点スクリプトをタイプ別で収めてある。課題バンクのフォルダ内に説明やスクリプトのファイルが存在しなければ，これらが利用される。なお，自動採点スクリプトについては第5章で詳細を述べる。

課題バンクフォルダ内には，主に下記のファイルが収められている。

● setting.txt

データベースへの登録用設定ファイル

- direction.txt
  課題の説明
- figure.png
  説明の補足用の画像
- code.txt
  ソースコード（提示用）
- answer.txt
  正答例
- answer.png
  画像を提出させる課題用の正答例
- test_cmd.sh
  自動採点スクリプト
- input*.dat（*は任意の数値）
  自動実行時の入力
- output*.dat（*はinputに対応する数値）
  自動実行時の想定出力（比較用）
- output_jp*.txt（*は数値）
  javapによる解析結果の抜粋（比較用）
- diff_res*.log（*は数値）
  実行結果に対する想定差分
- その他，配布ファイルなど

3.3　データベース構成

データベースは授業略称・年度で作成し，その中に下記の名称・項目を持つテーブル群を用意する。

- users　　ユーザ情報テーブル
  - u_id　　学内アカウント名
  - u_name　氏名
  - cls_id　クラス識別子
  - grd_id　権限
- p_bank　　課題バンクテーブル
  - p_id　　課題識別子
  - type　　課題タイプ
  - title　　課題タイトル
  - dist　　配布ファイル名（任意）
- p_link　　各授業の設定テーブル
  - lnk_id　授業の識別子（cls_id+l_id）
  - cls_id　クラス識別子
  - l_id　　授業の回
  - t_open　授業開始日時
  - t_view　課題公開日時
  - t_late　課題締切日時
  - p_id1～p_id8
    各出題課題の課題バンク内識別子
- evals　　評価テーブル
  - u_id　　学内アカウント名
  - s010～s159
    各出題課題の評価点数

各テーブルおよび各項目の意味の詳細は，以降の項でそれぞれ述べる。

3.3.1　users

ユーザ情報を管理するテーブルである。主キーとなるユーザ識別子を学内アカウント名と同一とするのは，本システムが学内のシステムであり，学内の別サーバで認証を行うためである。また，テスト用の特別なユーザを除き，パスワードやそのハッシュ値は記録しない。

クラスごとに時限が異なるためクラス識別子を，受講者と担当教員では利用できる機能を分けるため権限をそれぞれ項目として持たせてある。担当教員が進捗

などを把握しやすくするため，受講者には本システムを通して氏名を登録させる。

3.3.2 p_bank

課題バンクを管理するテーブルであるが，実質的にインデックスである。主キーとなる識別子は前述のフォルダ名に対応しており，他の項目はフォルダ内の設定ファイルに記述された情報が登録されている。メンテナンス性とデータベースの負荷の観点から，各課題の詳細はここに登録していない。

課題タイプは下記の通りである。

1. JARファイルを受け取るのみ
2. JARファイルを受け取って自動採点（テキストベースで自動実行）
3. JARファイルを受け取って自動採点（起動時にスクリーンショット）
4. テキストデータを受け取るのみ
5. テキストデータを行ごとに自動採点
6. PNGファイルを受け取るのみ
7. ドキュメントを受け取るのみ
8. ZIPファイルを受け取るのみ

課題を出題する際に内容を把握しやすくするため，タイトルを項目として持たせてある。また，課題によってはファイルを配布するため，そのファイル名を登録できるようにしてある。ただし，そのファイルの実体は課題バンクフォルダに収めてあることとする。

3.3.3 p_link

各授業の設定を管理し，課題バンクを紐づけするテーブルである。主キーとなる識別子はクラスと授業回（2桁）から構成されているため冗長であるが，検索の観点からクラス識別子と授業回も項目として持たせてある。

授業の開始日時は，実質的に，その回の課題の提出が可能となる日時である。それとは別に，課題の公開日時を設けたのは，予習を促すためである。また，学期の終盤に用意した総合的な課題などは，早めに公開することでモチベーションにつながることが期待され，それを可能とするためである。課題の締切日時は，受付を停止するというよりは，以降を遅延提出として扱う日時である。

1つの授業に最大8つの課題を設定できるよう，課題バンク内の識別子を登録するための項目を8個設けてある。上限を8にした理由は，難易度にもよるが，課題数が多すぎると受講生が対応しきれずモチベーションを下げる恐れがあるためと，現在までの授業実施において不足していないためである。

3.3.4 evals

各出題課題の評価点数を管理するテーブルである。主キーとなるユーザ識別子を除き，項目名は先頭の"s"に続いて授業回（2桁）とその中での課題の順番からなる。現在の実装では，クラスを分け

ずひとまとめで扱っている。また，項目は後から追加していく形をとっている。

1つの授業で設定できる課題は8つまでとしたが，それとは別に，授業ごとに出欠を0番，感想を9番とし，その状況も登録対象としている。現状では，出欠に関しては「無断欠席」「欠席」「遅刻」「出席」を区別し，感想に関しては「短文」「長文」を区別している程度である。

4. PHPファイル群

PHPファイル群は，大きく分けて以下の4つに分けられる。

1. ユーザ情報および認証関連
2. 課題および評価の提示関連
3. 課題の提出および採点関連
4. 管理関連

それぞれについて，以降の節で述べる。

4.1 ユーザ情報および認証関連

該当するファイルは以下の通りである。

- index.php
  トップページ
- config.php
  サイト内の共通設定，セッションの維持，アクセス制限など（サイト内のほぼ全てのPHPファイルから冒頭で呼び出し）
- add_u.php
  受講者の初回登録ページ
- edit_u.php
  ユーザ情報の確認・更新ページ
- update_u.php
  ユーザ情報の更新
- login.php
  ログイン認証，データベースからのユーザ情報の読み込み
- logout.php
  ログアウト

これらのファイルが担当する機能は論旨に大きく関わらないため，説明は割愛する。

4.2 課題および評価の提示関連

該当するファイルは以下の通りである。

- show_eu.php
  課題（評価）一覧ページ
- show_p.php
  課題内容ページ
- show_p9.php
  授業への意見・感想を入力するページ
- disp_png.php
  内部に保存された画像の表示
- dl_file.php
  内部に保存された提出ファイルのダウンロード
- dl_file_d.php
  配布ファイルのダウンロード

この中のshow_ls.phpとshow_p.phpについて，以降の項で説明する。

4.2.1 show_eu.php

現在出題されている全ての課題内容へのリンク一覧ページであるが，同時に

図3　課題（評価）一覧ページの例

ユーザの評価一覧としての機能を持つ(図3)。受講者が自身の進捗を把握しやすいよう，授業回ごとの行に締切日時と出題課題の評価の列を持たせ，表にしてある。各評価は記号で表し，

- ×　　未提出
- ▲　　不合格
- 確認中 手動での採点が未完了
- ●　　合格（遅刻提出）
- ○　　合格
- ◎　　合格（加点対象）

としてある。クリックするとそれぞれの課題内容ページへ遷移する。なお，各授業回の最後の課題は授業の感想を任意で書かせる形である。

評価の達成度は概算評点という形で数値的にも示してある。これは，このページ内での各評価記号の数を単純に重みづけ和した値であり，実際の評点とは異なるが，目安となる。

4.2.2　show_p.php

課題の詳細な内容が表示されるとともに，答案の提出を受け付けるページである。その一例を図4に示す。課題番号はクラスと授業回（2桁）と課題の順番からなるが，EclipseでJavaプロジェクトを作成する際のプロジェクト名も兼ねている。そのため，課題の説明文中のプロジェクト名が課題番号になるよう，表示時に置換している。タイトルに続き，説明，コード，図，配布物，その他，提出，判定，コメント，正答例が並ぶが，コード以下はそれぞれ不要な場合には表示されない。

「説明」には前述のdirection.txtの中身を修飾済みhtmlとして表示し，「コード」にはcode.txtの中身を標準テキストとして読み込んだ後にhighlight.js[注6)]を用いて装飾表示する。そのため，「説明」にはhtmlの機能を利用でき，「コード」にはソースコードファイルをほぼそのまま流用できる。なお，「コピー」のボタンを押すとソースコードがクリップボードにコピーされる。「図」にはdisp_png.phpを用いてfigure.pngを表示し，「配

注6)　Webページ上に表示したソースコードに色を付けるJavaScriptライブラリ。
https://highlightjs.org

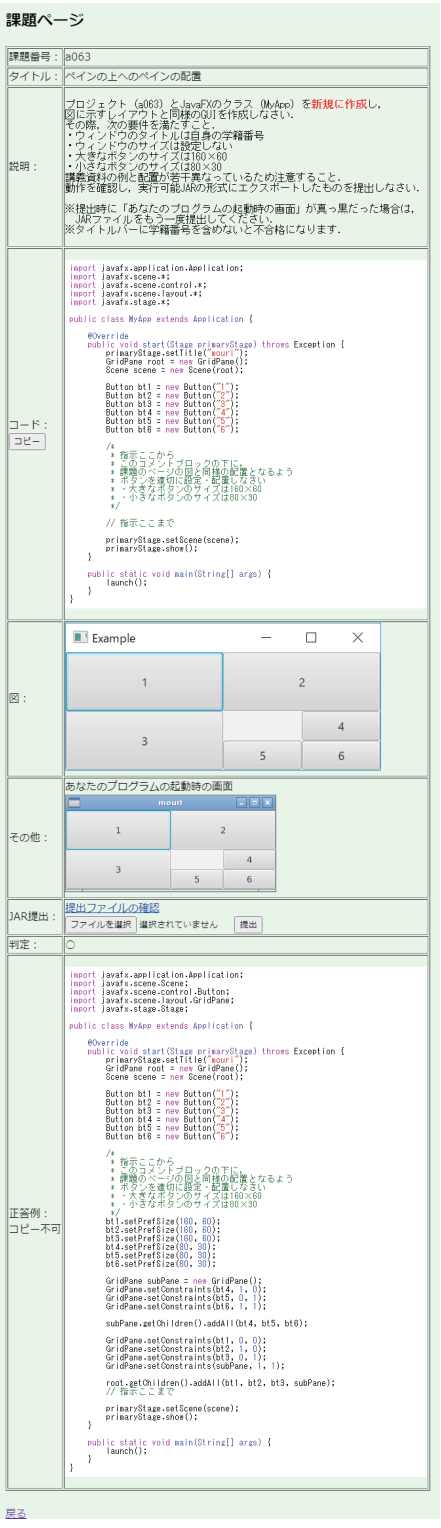


図4　課題ページの例

布物」にはdistに記載されたファイルをダウンロードできるリンクを表示する。「その他」には，現在はJavaFXを用いる課題について，提出されたJARファイルをサーバで自動実行した際のスクリーンショットを表示する。「図」の近くに配置することで受講者自身が比較しやすいようにしてある。

「提出」は課題タイプによって表示が変わる。タイプ1〜3ではJARファイルの提出用inputと提出ファイルのダウンロードリンクを，タイプ4〜5では解答欄のテキストエリアを，タイプ6ではPNGファイルの提出用inputと提出済みの画像を，タイプ7〜8ではそれぞれドキュメントファイルやZIPファイルの提出用inputと提出済みファイルのダウンロードリンクを表示する。

「判定」には答案の評価結果を表示する。なお，合格（加点対象）の際はページ背景全体に花火が上がり続けるアニメーションが表示されるようになっている。「コメント」には答案に対する教員からのコメント（出題課題ごとの受講者のフォルダにcomment.txtとして保存）を表示する。

「正答例」は課題バンクフォルダ内に該当ファイルが存在する場合にそれを表示する。answer.txtに関しては，その中身を標準テキストとして読み込んだ後にhighlight.jsを用いて装飾表示する。ただし，タイプ5ではanswer.txtの各行の

第1タブ前まで（または改行前まで）を標準テキストとして表示する。これは，タイプ5が別解をサポートしているためである。answer.pngに関しては，disp_png.phpを用いて表示する。なお，「正答例」は課題の提出期限後にのみ表示し，その部分のテキストはコピーできないようにしてある。期限後にも提出を受け付けることと併せて，受講者が復習する機会を設けるためである。

4.3　課題の提出および採点関連

該当するファイルは以下の通りである。

- ul_jar.php
  JARファイルのアップロード，自動採点スクリプトの呼び出し，結果表示
- ul_txt.php
  テキストエリアのデータの投稿，自動採点スクリプトの呼び出し，結果表示
- ul_file.php
  その他の指定ファイルのアップロード
- submit_p9.php
  授業への感想・意見を提出

この中のul_jar.phpとul_txt.phpについて，以降の項で説明する。

4.3.1　ul_jar.php

最初に，POSTされたJARファイルを既定の名前「a.jar」にリネームし，出題課題のユーザ別フォルダに保存する。続いて，実行フォルダを同フォルダに移し，「timeout[注7)]15」コマンドとともに自動採点スクリプトを実行する。これにより，無限ループなど実行時のトラブルが発生してもシステムが機能不全に陥らない。なお，JavaFXの課題の場合はGUIが含まれるため，「xvfb-run[注8)] -a」コマンドを「timeout」の前に付け加えて仮想ディスプレイ上で自動採点スクリプトを実行する。

自動採点スクリプトは，リダイレクトを用いて途中の標準出力や標準エラー出力を全てそれぞれ指定のファイルに吐き出し，最終的な評価値を表す1文字（不合格："F"，確認中："K"，合格："G"，加点対象の合格："S"）のみを標準出力する形をとる。この評価値をPHP側で受け取り，それに基づいてデータベースへの評価の登録と，ユーザへ示す評価結果ページを作成・表示する。

評価結果ページには，正解プログラムの実行結果と提出されたプログラムの実行結果を併記し，diff_match_patch.js[注9)]を

注7)　引数で指定した時間で強制的に停止する形で実行するためのコマンド。

注8)　仮想のディスプレイフレームバッファを生成して実行するためのコマンド。

注9)　2つのテキストを比較し，異なる部分を色付け表示するJavaScriptライブラリ。
https://github.com/google/diff-match-patch

用いて両者の差異を色付けする。出題課題のユーザ別フォルダ内にa.pngが存在していれば，それを「起動時の画面」として表示する。これらの他，同フォルダ内に存在する「*.log」ファイルの中身を列記するが，これは出題者側のデバッグ兼，評価結果に対する受講者への説明用である。なお，合格（加点対象）の際は，課題内容ページと同様に花火のアニメーションが表示されるようになっている。

4.3.2　ul_txt.php

ul_jar.phpと大部分が共通しているため，異なる部分についてのみ説明する。最初に，POSTされたテキストエリアのデータを既定の名前「a.txt」にリネームし，出題課題のユーザ別フォルダに保存する。評価結果ページには，テキストデータを受け取るのみの課題については，提出された解答を表示する。行ごとに採点する課題については，提出された解答とともに，誤りが存在する箇所を行ごとに表示する。

4.4　管理関連

該当するファイルは以下の通りである。

- admin.php
  管理用の機能へのアクセスページ
- show_ls.php
  授業情報の一覧ページ
- edit_l.php
  授業ごとの設定ページ
- update_l.php
  授業情報の更新
- edit_el.php
  授業ごとの評価の管理ページ
- show_ap.php
  課題ごとの答案の一覧・編集ページ
- read_p9.php
  授業ごとの感想・意見の確認ページ
- update_es.php
  評価の更新
- update_ps.php
  課題バンク情報をデータベースに登録
- show_es.php
  数値化した評価の一覧ページ
- edit_us.php
  ユーザ情報の管理ページ
- update_us.php
  ユーザ情報の更新

この中のedit_el.phpとedit_ap.phpについて，以降の項で説明する。

4.4.1　edit_el.php

同一授業における同一クラスの全受講者の全評価を表示することに加え，各評価の編集も可能なページである。その一例を図5に示す。出題した課題について，どれだけの受講者が答案を提出したか，合格したかが分かるため，授業中の進度調整に利用できる。評価の編集については，誤りを避けるため，編集対象の受講者を選択してから課題ごとのドロップダウンリストで評価を編集する形にした。

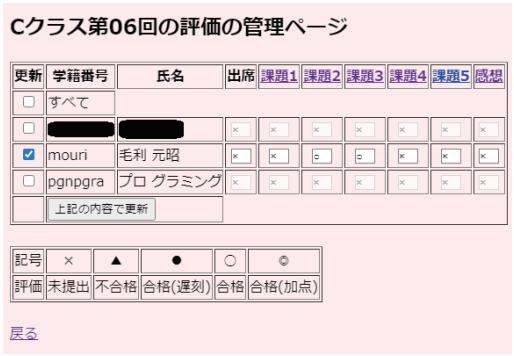


図5　edit_el.php

図6　show_ap.php

ただ，ほとんどの場合，評価の編集の際は答案を確認する必要がある。項目のリンクから後述の「課題ごとの答案の一覧・編集ページ」に遷移できるため，筆者はそこで評価を編集することが多い。

4.4.2　show_ap.php

同一課題における同一クラスの全受講者の答案・評価を表示することに加え，各評価の編集も可能なページである。その一例を図6に示す。提出されたファイルへのリンクやテキストは原則全員分を，ユーザ別フォルダ内の画像ファイル「a.png」に関しては「確認中」のものを表示する。これにより，手動採点が必要な対象を優先的に確認できる。オプションを指定することで，全員分の画像や，ユーザ別フォルダの任意のテキストファイルの中身を表示できる。コメント欄には，受講者個人へのコメントを書きこむことができる。筆者は主に，手動採点で評価を「不合格」にした場合に，その理由を伝えるために利用している。特にオンライン学習の際に本機能が活躍した。評価およびコメントの編集については，誤りを避けるため，対象の受講者を選択してから課題ごとのドロップダウンリストで評価を編集する形にした。

5. 自動採点スクリプト

5.1　要求

自動採点を実装するにあたって要求としたことは次の通りである。

1. Linuxで利用できる無償のソフトウェア群・コマンド群で構成・構築
2. PHPから呼び出し可能かつ採点結果をPHPから参照可能
3. JARファイルを解析し採点に利用可能
4. JARファイルの実行中に標準入力からデータ入力する形に対応し，標準出力は指定のファイルに保存
5. JavaFXの課題の場合は起動時のスクリーンショットを撮影可能

6．テキストの自動採点は別解に対応

それぞれの理由を説明する。まず1については，筆者がシステム用の予算を有していないためである。また，サーバとなるマシンのOSがLinuxであることは前提のため，その上で利用可能な形で全てを構成・構築する必要がある。2については，表層部をPHPで構成することが前提のためである。PHPからの呼び出し方法や参照方法の自由度は高いため，容易な要求と言える。3については，第2章で述べたように，プログラムの構造も採点対象に含めるためである。

4については，入出力に着目した採点のためである。ソースコードから実行する形であればコンパイル前にソースコードの一部を置換することでデータを差し替えられるが，本システムの場合はそれができない。実行時引数かファイル，標準入力で与える必要がある。ここで，実行時引数やファイルを用いる場合は，初心者にとってハードルが高い。例えばEclipseの場合，起動構成の適切な設定やファイルの適切な配置・編集などが必要となり，受講者が対応しきれずトラブルが多発する恐れがある。一方，標準入力で与える場合，その部分を提示するソースコードにあらかじめ含めることができる。また，実行時の標準出力を指定の名前のファイルへ一旦保存することで，実行と採点を切り離して考えることができる。

5については，GUIを表示するプログラムの採点を高効率化するためである。本来，GUIを表示するプログラムを採点するには，提出されたプログラムを採点者の手元で実行する必要がある。その負担は大きい。しかし例えば，GUIのレイアウトのみを採点対象とする場合，起動時のスクリーンショットが表示されていれば，それで採点できる。また，提出先の誤りなどを受講者側でも採点者側でも採点前に気付きやすくなる。

6については，答案に自由度を持たせておきたいためである。例えば数式や論理式を記述させる課題には，同じ結果を生むが表現の異なる答案を想定できる。現実的には全てを網羅できない場合も多いが，別解はできるだけ正解としたい。

5.2　テキストベースでの自動実行・採点

タイプ2の課題，つまり，実行可能JARファイルをテキストベースで自動実行および採点する場合の標準のスクリプトを図7に示す。1～4行目はフォルダの把握と移動，6行目は以前のファイルの削除である。7～12行目では，JARファイル内に含まれるクラスファイルをそれぞれ「javap」コマンドで解析し，その結果を一旦output_jp.txtに保存している。環境によっては元のファイル名のエンコードがShift-JISであるため，それに対応してある。

14～22行目では，自動実行および実

行結果と正答例による出力結果との比較を行っている。まず，課題バンクフォルダ内にファイル「input*.dat」（*はワイルドカード）が存在する個数分，それぞれを標準入力に流し込みながら実行し，それぞれの標準出力を対応するファイ

```
 1 d_sh=$(pwd)
 2 p_id=$(basename ${d_sh})
 3 flag_k=0
 4 cd ../../${1}/${2}
 5
 6 rm -f diff* output* error* 2>> error.log
 7 zipinfo -1 a.jar *.class | sed 's/.class$//g' ¥
 8  | xargs javap -classpath a.jar -verbose 1> output_jp.txt 2>> error.log
 9 if [ ! -s output_jp.txt ]; then
10     zipinfo -1 -O sjis a.jar *.class | sed 's/.class$//g' ¥
11  | xargs javap -classpath a.jar -verbose 1> output_jp.txt 2>> error.log
12 fi
13
14 for f_i in ${d_sh}/input*.dat
15 do
16     f_o=$(basename ${f_i/input/output})
17     cat ${f_i} | java -jar a.jar | sed -r 's/¥x1b¥[[0-9;]*m//g' > ${f_o}
18     if [ -s ${d_sh}/${f_o} ]; then
19         flag_k=1
20         diff -dBw --strip-trailing-cr ${f_o} ${d_sh}/${f_o} >> diff_res.log
21     fi
22 done
23
24 pjnm=`sed -n 1p output_jp.txt | sed -e 's/.*¥!¥///' -e 's/¥/.*//'`
25 cat output_jp.txt | grep "Classfile" | sed -e 's/.*¥///g' | sort > output_jp2.txt
26 cat output_jp.txt | grep "static .*);$" | sed -e 's/^.*static/static/g' -e "s/${pjnm}//g" ¥
27  | sort >> output_jp2.txt
28 cat output_jp.txt | grep "extends" | sed -e "s/${pjnm}//g" | sort >> output_jp2.txt
29 if [ -s ${d_sh}/output_jp2.txt ]; then
30     flag_k=1
31     diff -di --strip-trailing-cr output_jp2.txt ${d_sh}/output_jp2.txt >> diff_jp2.log
32 fi
33
34 if [ ${flag_k} -eq 1 ];then
35     if [ -s diff_jp2.log ]; then
36         echo -n "F"
37     else
38         if [ -s diff_res.log ]; then
39             echo -n "F"
40         else
41             echo -n "G"
42         fi
43     fi
44 else
45     echo -n "K"
46 fi
```

フォルダの把握と移動

javap による解析

自動実行および
実行結果の比較

※1

※2

※1　javap の結果から情報を抽出

※2　正答例のものと比較

評価値を決定
比較無し：　"K"（確認中）
差分がある：　"F"（不合格）
差分がない：　"G"（合格）

図7　JARファイルをテキストベースで自動実行・採点するスクリプト
（タイプ2標準）

ル「output*.dat」に装飾用の制御コードを省いて保存する。それと同時に，「diff[注10)]」コマンドで正答例による出力結果と比較し，差分をdiff_res.logに追記していく。こうすることで，入出力の関係を採点の対象とできる。

24～32行目では，まずoutput_jp.txtからクラスファイル名，クラスメソッド名，子クラス名をそれぞれ抽出しoutput_jp2.txtに追記保存している。この際，パッケージ名(プロジェクト名)を消去している。課題バンクフォルダ内にoutput_jp2.txtが存在する場合は「diff」コマンドで比較し，その結果をdiff_jp2.txtに保存する。こうすることで，プログラムの構造の大枠部分を確認し，採点の対象とできる。

34～46行目では，これまでに得られた結果から評価値を決定している。何らかの比較が行われた場合，それらの差分が存在すれば（差分ファイルが空でないならば）不合格（F），存在しなければ合格（G）とし，比較が行われていない場合は確認中（K）としている。

この標準の自動採点スクリプトの利点は，課題バンクフォルダ内に比較用のファイルが存在するかどうかによって，自動的に適切な挙動をとることにある。そのため，入出力の関係やプログラムの構造の大枠のみ確認したい課題のほぼ全てが，比較用のファイルを用意するだけで自動採点できる。

5.3　JavaFXの課題の自動起動・採点

タイプ3の課題，つまりJavaFXを利用しており，自動起動時のスクリーンショット撮影を行う場合の自動採点スクリプトを抜粋し図8に示す。なお，割愛された部分はタイプ3の課題のものとほぼ同一である。

```
11 timeout 14 openbox 2>> error.log 1>> error1.log &
12 timeout 12 java --module-path /usr/share/openjfx/lib ¥
13  --add-modules javafx.controls,javafx.fxml,javafx.media,javafx.swing ¥
14  -jar a.jar 2>> error.log 1>> error1.log & sleep 5 2>> error.log 1>> error1.log
15 w_id=$(xdotool search --onlyvisible --name ${2}) 2>> error.log 1>> error1.log
16 xdotool windowactivate ${w_id} 2>> error.log 1>> error1.log ;¥
17  import -window ${w_id} -frame a.png 2>> error.log 1>> error1.log
27 if [ -s diff_jp2.log ]; then
28     echo -n "F"
29 else
30     echo -n "K"
31 fi
```

※1　仮想ディスプレイ上で起動
※2　スクリーンショットを撮影

評価値を決定
差分あり：　"F"（不合格）
差分なし：　"K"（確認中）

図8　JavaFXの課題を自動起動・採点するスクリプト（タイプ3標準）の抜粋

注10) Linuxに標準で搭載されている，2つのテキストファイルを比較し，差分を出力するコマンド。空白や改行を無視した比較も可能。

11～16行目は，タイプ2の場合の14～22行目を置き換える部分である。ここでは，まずウィンドウマネージャである「openbox[注11)]」を時限的に立ち上げた後に，JARファイルを実行している。その後，5秒待ってから「xdotool[注12)]」コマンドを用いてウィンドウを検索，最前面にし，「import[注13)]」コマンドでウィンドウ指定して撮影し「a.png」として保存している。ここで，検索に用いるのは学内アカウント名であり，受講者に提示するソースコード内に事前に含まれるよう工夫してある。こうすることで，採点のページにて起動時のスクリーンショットを表示できる。なお，第4.2節で述べたように，スクリプトの実行前に仮想ディスプレイを生成しておく必要がある。

27～31行目は，タイプ2の標準の場合の34～46行目を置き換える部分である。タイプ3の標準では，javapでの解析結果との比較のみで評価値を決定し，不備がなければ「確認中」としている。

5.4　テキストデータの行ごとの自動採点

タイプ5の課題，つまりテキストデータを行ごとに自動採点する場合は，例えば図9に示すような説明文と正答例を用意しておく。行ごとに採点するため，小設問の番号と回答の行・正答例の行とを対応させておく必要がある。正答例については，タブ文字で区切りながら別解を併記することができる。タブ文字はブラウザ上のテキストエリアへ入力することが困難であるため，区切り文字として適している。

direction.txt （課題の説明）より 回答させる部分を抜粋	1	変数 a の値が 56 である
	2	変数 a の値が 78 でない
	3	変数 a の値が正である
	4	変数 a の値が 2 以下である
	5	変数 a の値が 9 の倍数である
	6	変数 a の値が奇数である
	7	変数 a の値を 4 で割った余りが 3 である
answer.txt （正答例） タブ文字を網掛けで表現	1	a == 56　56 == a
	2	a != 78　78 != a　!(a == 78)　!(78 == a)
	3	a > 0　0 < a
	4	a <= 2　2 >= a
	5	(a % 9) == 0　0 == (a % 9)　a % 9 == 0　0 == a % 9
	6	(a % 2) == 1　1 == (a % 2)　a % 2 == 1　1 == a % 2
	7	(a % 4) == 3　3 == (a % 4)　a % 4 == 3　3 == a % 4

図9　タイプ5の課題の例（日本語の条件式をJava風に表記）

注11) Linux用のウィンドウマネージャ。シンプルかつ軽量で立ち上がりが早い。

注12) コマンドラインからウィンドウやマウス・キーボードを操作するツール。

注13) ImageMagickに付属する，スクリーンショットを撮影するコマンド。

タイプ5の課題の自動採点スクリプトを図10に示す。1～2行目はフォルダの把握と移動，4行目は以前のファイルの削除である。

6～21行目では，正答例との比較を行っている。answer.txtから1行ずつ読み取り，さらにタブ文字で分割した文字列と，答案であるa.txtの同じ行とを比較していく。もし文字列が一致した場合は，その行を正解したと見做してcount.txtに記録し，次の行に移る。タブ文字で分割された文字列群に一致するものが存在しなかった場合はそのまま次の行に移る。全ての比較を終えたら不要なファイルを削除する。この形であれば，タブ文字で区切ることで行ごとの別解を複数用

```
 1  d_sh=$(pwd)
 2  cd ../../${1}/${2}
 3
 4  rm -f diff* output* res.log count.txt
 5
 6  count=0
 7  cat ${d_sh}/answer.txt | while read line1
 8  do
 9      count=$((count+1))
10      echo ${count} >> count.txt
11      echo "${line1}" | tr '¥t' '¥n' | while read line2
12      do
13          diff -dBw --strip-trailing-cr <(sed -n ${count}p a.txt) <(echo "${line2}") > diff_res.log
14          if [ ! -s diff_res.log ]; then
15              echo ${count} >> res.log
16              break
17          fi
18      done
19  done
20
21  rm -f diff_res.log
22
23  count=1
24  cat count.txt | while read i
25  do
26      j=`sed -n ${count}p res.log`
27      if [ ${i} -eq ${j} ]; then
28          echo "" >> output.dat
29          count=$((count+1))
30      else
31          echo ">>" >> output.dat
32          echo ${i} >> diff_res.log
33      fi
34  done
35
36  if [ -s diff_res.log ]; then
37      echo -n "F"
38  else
39      echo -n "G"
40  fi
```

正答例から1行ずつ読み取り
タブ文字で区切る（=line2）

line2と回答の差分がない物があった場合
現在の行番号をres.logに記録して次の行へ

全行番号とres.logに記録した行番号を比較し
res.logに含まれる場合は空行
含まれない場合は誤り箇所を指摘する">>"を出力
（併せて，差分ファイルに記録）

評価値を決定
差分なし： "F"（不合格）
差分あり： "G"（合格）

図10　テキストデータを行ごとに自動採点するスクリプト（タイプ5標準）

意することができる。

23～34行目では，誤りのあった行を受講者に指摘するためのテキストファイル「output.dat」をcount.txtから作成している。それと同時に，誤りのあった行の番号をdiff_res.logに追記保存している。

36～40行目はdiff_res.logをもとに評価を行っている。ファイルが空でなければ不合格（F）とし，そうでなければ合格（G）としている。

5.5　特殊なケースの自動採点

ここまで，タイプ別の標準的な自動採点スクリプトについて説明してきたが，課題バンクフォルダ内にtest_cmd.shを用意することで，特殊なケースの自動採点も可能である。ただし，特殊なケースの自動採点を行うには，課題作成者による課題への深い理解が必要になるとともに，現状では「javap」コマンドの結果に対する読解力やLinuxシェルスクリプトでの記述力が必要になる。筆者自身，幾度となく試行錯誤を繰り返して課題を作成しており，かかった労力は大きい。しかし，学習効果を高めるような工夫のある課題には，このような機能が必要であると考える。

5.5.1　含まれる条件式の個数で評価値を変更する場合

例えば，条件式の個数で評価値を変更したい課題を作成した際は，図7の34行目以降を図11のように置き換えたものを用意した。図の34行目では「javap」コマンドによる解析結果output_jp.txtから条件式の個数を数え上げ，output_jp3.txtに出力している。40行以降にて，条件式の数が6以下であれば加点対象の合格（S）とし，10以下であったら合格（G），いずれでもなければ不合格（F）としている。

```
34  cat output_jp.txt | grep -c "if_" > output_jp3.txt
35
36  if [ -s diff_res.log ]
37  then
38      echo -n "F"
39  else
40      if [ `cat output_jp3.txt` -le 6 ]; then
41          echo -n "S"
42      elif [ `cat output_jp3.txt` -le 10 ]; then
43          echo -n "G"
44      else
45          echo -n "F"
46      fi
47  fi
```

→ javapの結果から条件式の個数（"if_"を含む行数）を数え上げてoutput_jp3.txtに記録

評価値を決定
実行結果に差分あり：　"F"（不合格）
条件式の個数が6以下：　"S"（加点対象）
条件式の個数が10以下：　"G"（合格）
それ以外：　"F"（不合格）

図11　特殊な自動採点スクリプトの例
（含まれる条件式の個数で評価値を変更する場合の抜粋）

5.5.2　大量のデータ入力を必要とする場合

例えば，大量のデータ入力を必要とするアプリケーションを想定した場合，ファイルからデータを入力する方が効率的であるその実装方法としては，プログラム内で特定のファイルを参照する固定的な形，実行時の引数でファイル名を指定する形，実行中に標準入力でファイル名を指定する形が挙げられる。しかし，これらのような課題は標準的な自動採点スクリプトでは対応できない。

ファイルでデータを指定する場合の自動採点スクリプトの例をまとめて図12

特殊な自動採点スクリプトの例（ファイルでデータを指定する場合の抜粋）図12

（a）プログラム内で特定のファイル（「data.txt」）を参照する固定的な形

```
14  for f_i in ${d_sh}/input*.dat
15  do
16          cp ${f_i} data.txt
17          f_o=$(basename ${f_i/input/output})
18          java -jar a.jar | sed -r 's/¥x1b¥[[0-9;]*m//g' > ${f_o}
19          if [ -s ${d_sh}/${f_o} ]; then
20                  flag_k=1
21                  diff -dBw --strip-trailing-cr ${f_o} ${d_sh}/${f_o} >> diff_res.log
22          fi
23  done
```

（b）実行時の引数でファイル名（「input*.dat」内にそれぞれ記載）を指定する形

```
14  for f_i in ${d_sh}/input*.dat
15  do
16          f_ii=`cat ${f_i}`
17          cp ${d_sh}/${f_ii} ${f_ii}
18          f_o=$(basename ${f_i/input/output})
19          java -jar a.jar ${f_ii} | sed -r 's/¥x1b¥[[0-9;]*m//g' > ${f_o}
20          if [ -s ${d_sh}/${f_o} ]; then
21                  flag_k=1
22                  diff -dBw --strip-trailing-cr ${f_o} ${d_sh}/${f_o} >> diff_res.log
23          fi
24  done
```

（c）実行中に標準入力でファイル名（「input*.dat」内にそれぞれ記載）を指定する形

```
14  for f_i in ${d_sh}/input*.dat
15  do
16          f_ii=`cat ${f_i}`
17          cp ${d_sh}/${f_ii} ${f_ii}
18          f_o=$(basename ${f_i/input/output})
19          cat ${f_i} | java -jar a.jar | sed -r 's/¥x1b¥[[0-9;]*m//g' > ${f_o}
20          if [ -s ${d_sh}/${f_o} ]; then
21                  flag_k=1
22                  diff -dBw --strip-trailing-cr ${f_o} ${d_sh}/${f_o} >> diff_res.log
23          fi
24  done
```

図12　特殊な自動採点スクリプトの例（ファイルでデータを指定する場合の抜粋）

に示す。それぞれ，(a) プログラム内で特定のファイルを参照する固定的な形，(b) 実行時の引数でファイル名を指定する形，(c) 実行中に標準入力でファイル名を指定する形であり，図7に示したスクリプトの14〜22行目を置き換えるものである。

(a) では，タイプ2標準でも用いる入力用ファイル「input*.txt」を，JARの実行前にプログラム内で固定的に参照するファイル名（図の例では「data.txt」）に書き換えつつ実行フォルダにコピーしている。その代わり，実行時の標準入力にファイルの中身を流し込まないようにしている。

(b) では，「input*.txt」内に書かれたファイル名をあらかじめ取り出しておき，それをもとに課題バンクフォルダ内のファイルをJARの実行前に実行フォルダにコピーしている。その代わり，実行時の標準入力にファイルの中身を流し込まないようにするとともに，実行時引数にファイル名を指定している。

(c) では，「input*.txt」内に書かれたファイル名をあらかじめ取り出しておき，それをもとに課題バンクフォルダ内のファイルをJARの実行前に実行フォルダにコピーしている。(b) (c) と異なり，実行時の標準入力にファイルの中身を流し込んでいる。

注意点としては，セキュリティ上の制約から，実行フォルダの下にファイルを配置する必要があり，相対パスでのみファイル名を指定する必要がある。また，ファイルを正しく受け取ることができるよう，課題のプログラムを作成する必要がある。

6. システムの運用状況と受講者の反応

本システムの運用開始が2017年であり，以降，受講者やアシスタントへのインタビューの他，毎回の授業における任意課題の一環として授業への意見・感想を募ってきた。寄せられたものの大半は授業内容や進度，課題の難易度に関するものであったが，システムへの意見や要望もいくつか見られた。統計処理をはじめ分析を行っていなかったため，数値的に示すことができないが，受講者の反応を中心に本システムの運用の状況を述べる。

まず，実行可能JARファイルの作成については，授業回を重ねるごとに，筆者やアシスタントへ質問することが減っていった。時折見られたのが，「ローカルでは正しく実行できるのに自動採点で不合格になる」というもので，大半が提出ファイルの間違い（エクスポート時の起動構成の選択ミスを含む）であったが，特殊な課題などプログラム構造も採点対象としていることが原因の物もあった。その旨と課題の説明文を改めて伝えると，大半は納得し，後に合格を得ていた。いずれにしても，受講者によるロー

カルでの実行が習慣づいており，本システムの目的が達せていることが示唆される。また，受講者がプログラム構造について考えるきっかけにもなっていることが分かる。

受講者が考えるきっかけ，あるいはモチベーション維持のきっかけとしては，加点対象の合格の存在も大きい。加点対象の合格のある課題を出題した際は，「自力で加点がもらえて嬉しかった」「加点がもらえなくて悔しい」「どうすれば加点になるのかヒントが欲しい」という意見が，特に成績上位の受講者から多く寄せられた。また，第4.2節で述べた，加点対象の合格時の花火の演出を喜ぶ受講者も多かった。成績上位者は通常の課題の合格が早く，そのままでは面白みを感じないことが多い。かといって，課題数を単純に増やしたり難易度を高めたりでは，対応しきれない通常レベルの受講者のモチベーションの低下が懸念される。加点対象の合格がそれらを緩和できていると言える。

受講者の理解につながる機能としては，課題の提出締切後に正答例を表示する機能とコメント機能が挙げられる。「締切に間に合いそうにないため，正答例を見て学ぶつもり」という感想が時折見られ，遅刻提出の合格も度々見られた。また，コメント機能はコロナ禍の影響で授業がオンデマンド型学習になった際，双方向に情報量の大きな対面で対応できなくなることから導入した。これらの機能なしでは受講者・課題ごとの細かな対応が難しかったはずである。いずれも，理解しようとする受講者の一助になっていたと言える。

運用中のシステム面の改善としては，課題内容ページにて表示されるソースコードの装飾とクリップボードへのコピー機能の導入も挙げられる。当初は等幅フォントで表示するのみであったが，「Eclipseやインターネット上にある解説記事のように色付けしてほしい」という要望があり，また課題の際にコピー＆ペーストの範囲を間違える受講者が多かったためである。機能の導入により，ソースコードを把握・利用しやすくなったためか，同様のトラブルは大幅に減少した。

評価結果ページに表示する実行結果について，正答例のものと異なる部分を装飾表示する機能も途中から導入した。「実行結果の違いが分かりにくい」という意見に加え，採点対象が入出力のみでないという本システムの特殊性から，実行結果の違いを明確に伝える必要が生じたためである。違いがなければプログラム構造など別の要因を検討しやすくなるのである。

廃止した機能には，出欠の受講者への表示が挙げられる。筆者はグループワークを除いて出席を重視していないが，連絡のあった欠席者のフォローをしつつ，

無断欠席者のフォローを後回しにしたいため，出欠を記録していた。ところが，授業回が進むと出欠に関する成績の問い合わせのような意見が増えてしまった。出欠の受講者への表示はプレッシャーを与え，その解消のための説明機会を増やす要因となるとともに，有用な意見・感想を得る機会を減らす要因にもなる。よって，出欠は授業担当者のみが把握できる形とした。

これまでに述べた様々な状況から，本システム自体は目的を十分に達成していると言える。あとは課題・資料の作成と授業の実施といった，授業担当者のマターであるが，もう少しの機能拡張を考えると，課題の作成を補助する機能が挙げられる。現行では，メンテナンスの容易さから，サーバのフォルダ・ファイルを直接編集あるいはそれに準じる形をとっている。ブラウザ上から安全に課題を作成・編集する機能など，検討してみたい。ただし，近年はJavaやJavaFXの導入やメンテナンスに関して難が生じていることから，来年度は使用プログラミング言語を変更する予定である。まずは同様の目的を果たせるよう，新たなシステムを構築していきたい。

7. まとめ

本稿では，ローカルの開発環境にて作成された実行可能JARファイルを提出させることを前提とした，自動採点機能を有するプログラミング実習用システムの設計・構成や運用の状況を説明した。実行可能JARファイルを提出させる意義に始まり，外部･内部のシステム設計，自動採点スクリプトについて，抜粋しながら詳細に示した。また，受講者からの反応を中心に，システムの運用状況についても示し，本システム自体は目的を十分に達成していることも示した。

今後の課題として，来年度は使用プログラミング言語を変更する予定であるため，まずは同様の目的を果たせるような新たなシステムを構築していきたい。また，受講者の学習効率を向上させるようなさらなる工夫を検討していきたい。

参考文献

1）中島秀樹，宮地恵佑，髙橋直久：プログラミング演習支援システムCAPESのための答案評価機構の実現，情報処理学会研究報告，2006（16），pp.127-134，2006。

2）中村慎司，筧捷彦：プログラミング授業支援システムWOJの開発，情報処理学会第77回全国大会講演論文集2015（1），pp.939-940，2015。

3）Aizu Online Judge, https://judge.u-aizu.ac.jp

4）Progate, https://prog-8.com

5）paiza, https://paiza.jp

6）AtCoder, https://atcoder.jp

（URL閲覧日は全て2021年11月23日）

Moodleワークショップを利用した学生による相互評価の実践

岩田　員典（愛知大学経営学部）

要旨

2020年度は新型コロナウイルスの蔓延から，本学の春学期（前期）の開講が遅くなっただけでなく，オンラインですべての授業を実施することになった。そのような状況において，本学ではMoodleを活用してこの事態に対処することになったが，どのような機能を活かせば学習効果を高められるのか分からないという問題が生じた。そこで，学生同士の相互評価をおこなえるワークショップモジュールについて着目した。これは学生が課題を相互評価することで，単に課題を提出するよりも授業の理解度を深めることができると考えたからである。本稿ではこのワークショップモジュールについて解説し，その実践結果についても述べる。

キーワード：Moodle，e-Learning，ワークショップ，相互評価，ルーブリック

1. はじめに

本学で利用しているMoodle[1]はLearning Management System（LMS）の一つであり，数多くの大学で利用されている。また，Moodleはオープンソースソフトウェアであり GNU GPLv3+[2]というライセンス形態に基づいて配付されている。したがって，このライセンスを遵守していれば再配布や改変も可能である。これにより，各ユーザは必要な機能を追加し，それを配付することもできる。このことがMoodleの安定性と多彩な機能に繋がっている。

その一方で，その機能の豊富さと操作の煩雑性から，利用されていない機能や知られていない機能が多数ある。本学でも独自のマニュアルを作成するとともにサポートデスクも用意されているが，それでも機能を活かし切れているとは言い切れない。

また，昨年度の新型コロナウイルスの蔓延によるオンライン授業の導入において，豊富な機能からどの機能を活用すれば学習効果を高められるのか分からないという問題が生じた。

そこで本稿ではMoodleが持つ学生同士の相互評価を可能にするワークショップモジュール[3), 4)]について着目し，その機能と利用方法について解説し，実践結果についても報告する。

2. ワークショップモジュール設置と設定

ワークショップモジュールはMoodleが標準で備えている機能であり，このモジュールを使うことで学生同士の相互評価が可能になる。

学生はレポートを提出し，レポートの締め切り後に他の学生[注1)]のレポートを評価することになる。これを行えるようにするために，教員はレポート期限の設定だけでなく，評価基準の設定，学生間の課題の割り当てをする必要がある。さらに，相互評価の結果に基づいた採点結果の返却もおこなわなければならない。

これら教員側がおこなうべき内容は大きく「基本設定」と「各種フェーズ」の二つに分けられる。本章ではこれらの設定について説明する。

ワークショップの設置と基本設定は以下の手順でおこなう。

①そのコースの編集権限のあるアカウントで，編集モードに切り替える

②ワークショップを追加したいトピックで，「活動を追加する」から「ワークショップ」を選択する（図1）

③「新しいワークショップを追加する」が表示されるので，基本設定をおこなう。基本設定には次の項目がある。これらの項目の詳細については2.1節～2.9節でそれぞれ説明する。

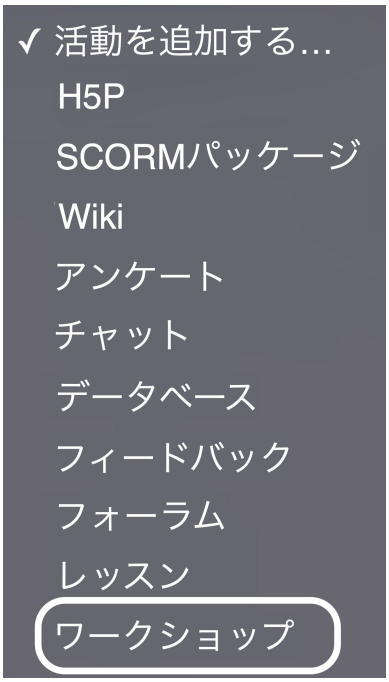


図1　活動の追加

- ●一般
- ●評定設定
- ●提出設定
- ●評価設定
- ●フィードバック
- ●提出例
- ●利用
- ●モジュール共通設定
- ●利用制限

2.1. 一般

一般では，図2で示すようにその他の活動と同じく名称と説明を記載すること

[注1)] ただし，自己評価を行えるように設定した場合は自分自身の課題の評価もできるようになる。

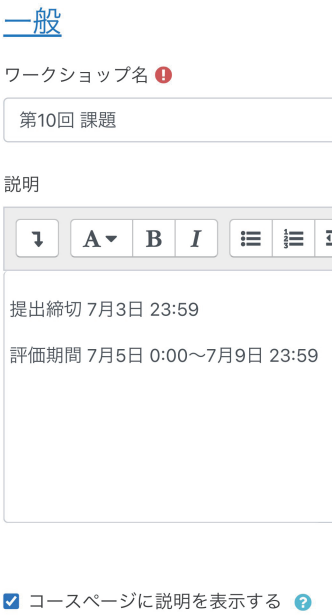


図2　一般設定

ができる。なお，説明をコースに表示したい場合は，「コースページに説明を表示する」にチェックを入れる必要がある。

2.2. 評定設定

評定設定（図3）では，学生同士が互いに評価する際の方法と配点を設定する。まず評価の際の方法は「評定方法」から，「累積評価」，「コメント」，「エラー数」，「ルーブリック」のいずれかを選択する。これは学生が他の学生を評価する際にどのような方法を用いるかを表している。各方法の評価の概要は以下の通りである。なお，詳細はワークショップの評定方法[5]に記載されている。

図3　評定設定

- 累積評価：各問題に得点を割り振り，満点の場合に100％になるように正規化する方法。評価基準は説明として記載するのみである。
- コメント：コメントを返すだけの方法で評価を与えることはできない。また，評価された提出物の合計は常に100％になる。
- エラー数：各評価項目に対して要件を満たしているかどうかをyes／no（すなわち○／×）で評価する。また，noの割合に合わせて得点率を設定することが出来る。例えば，yesの割合が40％を下回った場合に評価は20％のように設定することが可能である。
- ルーブリック：解答に対する基準を設け，学生は最も適合する基準を選ぶことで採点をおこなう。教員は基準ごとに得点率を設定することが出来る。

これらのうち，本稿では相互評価で効果的であると考えられるルーブリックによる評定設定について説明をする。

さらに，「提出に対する評点」と「評価に対する評点」も設定することが出来る。これは，提出された課題自体の得点率と，他の学生の課題を評価したことで得られる得点率を表す。また，それぞれに合格点を設定することもできる。

カテゴリは評定カテゴリ[6]の設定をしている場合のみ利用でき，設定しない状態では「カテゴリなし」のみ利用できる。

評点の小数位は最終的な評点の小数点以下の桁数を指定する。この値が0の場合には評点は整数値となる。

2.3. 提出設定

提出設定では，学生にどのような課題を提出するかなどの説明（提出のインストラクション）を記載したり，提出方法や，ファイル提出の場合は提出するファイルの形式を限定したりすることができる。説明に関しては2.1節の一般で設定しておいてもよい（図4）。

ここで「提出期限後の提出」にチェックを入れると締め切り後にも学生は課題の提出が可能になる。これを可能にした場合に後述のフェーズにおいて，評価フェーズになるまで学生は課題の提出が可能になる。その結果として，評価対象の自動割り当てをおこなっていると進行

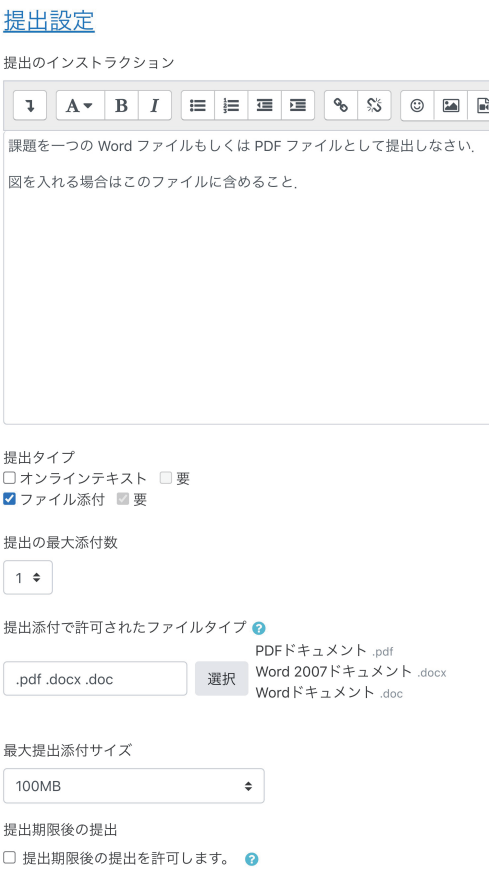


図4　提出設定

に問題が起きる事があるので，こちらは設定しない方が望ましい。

2.4. 評価設定

ここでの設定内容は，課題の締め切り後に学生がワークショップにアクセスした場合に表示される（図5）。どのように評価するかや締め切りなどについて記載するとよい。

また，「自己評価を使用する」にチェックを入れると，学生は自分自身の課題に

図5　評価設定

図6　フィードバック

対しても評価をおこなうことができてしまうので，前述の「提出期限後の提出」同様にチェックを入れない方が望ましい。

2.5. フィードバック

「全体フィードバックモード」を設定することで，採点する学生がコメントを記載できるようにしたり，コメントの記載を必須にしたりすることができる（図6）。また，これを有効にした場合に，添付ファイルとしてフィードバックを返すことも可能である。その際に，添付できるファイル数，ファイルタイプ，ファイルサイズも指定できる。

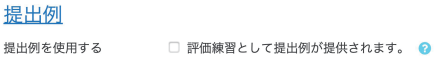


図7　提出例

図8　利用

2.6. 提出例

図7に示した提出例では，「提出例を使用する」にチェックを入れることで，教員が用意した解答例などに対して学生が評価の練習をおこなうことができる。

この提出例を評価するタイミングは，課題の提出前か，提出後で他者の課題の評価前のいずれかである。

2.7. 利用

利用では，図8のように提出開始日時，提出終了日時，評価開始日時，評価期限と提出後の自動フェーズ移行の設定がおこなえる。それぞれの日時を設定するに

はチェックボックスにチェックを入れ，日付と時刻を設定する。

「提出終了日時後、次のフェーズに移行する」にチェックを入れておくと課題の提出締め切り後に自動で評価フェーズにフェーズが切り替わる。チェックを入れていない場合は手動でフェーズを変更する必要がある。

この自動フェーズ移行の設定において，提出終了日時と評価開始日時を同日同時刻にすることもできるが，きちんとフェーズの移行がおこなわれない恐れがあるため，ある程度の間隔を空けた方がよい[注2)]。

2.8. モジュール共通設定

この項目には「利用」，「IDナンバー」，「グループモード」が存在しており，それぞれ以下の設定がおこなえる。

利用：このワークショップをコースに表示するかどうかの設定する。

IDナンバー：評定計算をおこなう際のIDとなる。

グループモード：グループで取り組むかどうかを設定することができる。

この項目と次の利用制限はワークショップだけではなく他の活動でも利用することができるため，詳細[7), 8)]は省略する。

2.9. 利用制限

利用制限は日時や，ユーザID等を使った制限を設定することができる。

3. 各種フェーズの設定

ワークショップを設置・設定すれば完了ではなく，フェーズと呼ばれる課題の状況に合わせた設定が必要である。ワークショップの設定をして「保存してコースに戻る」を選択した後，コースからそのワークショップを選択することでフェーズの設定画面（ワークショッププランナー：図9）に移動できる。

フェーズは「セットアップフェーズ」，「提出フェーズ」，「評価フェーズ」，「成績評価フェーズ」「終了」の5つに分けられる。本章ではこれらのフェーズについて説明をする。

なお，2.7節「利用」で「提出終了日時後、次のフェーズに移行する」を設定している場合だけ「提出フェーズ」から「評価フェーズ」に自動で移行されるが，それ以外の場合にフェーズの移行は原則として教員が手動でおこなう。

注2) 設定次第ではあるが10分程度間隔があればよいはずである。

図9　ワークショップランナー

3.1. セットアップフェーズ

セットアップフェーズでは，ワークショップの設置で実施した「一般」から「提出設定」と，課題を評価するための評価フォームの編集をおこなう。

ここで新たにおこなうべき事は評価フォームの編集であるため，本節では評価フォームの編集についてのみ説明をする。なお，この編集では評定設定で設定した評定方法が「ルーブリック」の場合について取り上げる。

ワークショップランナー（図9）のセットアップフェーズにある「評価フォームを編集する」をクリックすることでルーブリックを作成することができる。

ルーブリックでは，各課題に対して評価基準を設定することで，評価者による評価の偏りを少なくなるようにする。どのような観点で評価するかをクライテリアといい，このクライテリア毎に評価基準を設ける。

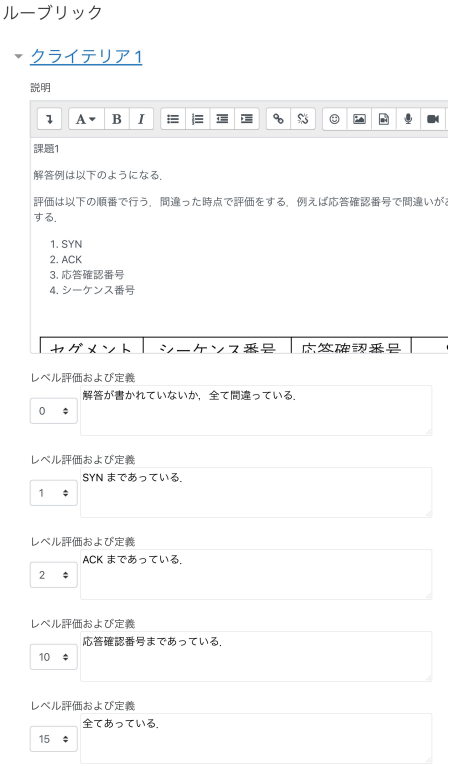


図10　クライテリアの設定

図10ではコンピュータネットワークにおけるパケットのヘッダ情報に関する設問に対して評価基準を設定した例を示している。ここで，「レベル評価およ

び定義」で各基準に対しての点数を指定している。ここで，点数は最高点と最低点を基準として正規化されるため0は0％で，15は100％となる。同様に1は7％（1 ÷ 15=0.0666…），2は13％（2 ÷ 15=0.1333…），10は67％（10 ÷ 15=0.6666…）となる。

また，クライテリアが複数ある場合は，2.2節の評定設定で設定した「提出に対する評点」をクライテリアの数に分けて点数を付ける。例えば，提出に対する評点が80でクライテリアが2つある場合には，各クライテリアには評点として40が割り当てられる。この評点に前述の基準で設定されたパーセントを掛けた値が，そのクライテリアにおける評点となる。つまり，前述の基準で10（67％）と評価された場合には，40 × 67％ =27が評点となる。

クライテリアは必要に応じて追加することができるので，課題の内容に合わせて設定するとよい。

クライテリアの設定が完了したら「保存してプレビューする」をクリックすることで，学生にどのように表示されるかを確認することができる（図11）。

これらの設定が完了したら次の提出フェーズに移行する。

図11　クライテリアのプレビュー

3.2. 提出フェーズ

提出フェーズは，学生からの課題提出を受け付けるフェーズである（図12）。また，教員は締め切り後に，評価対象とな

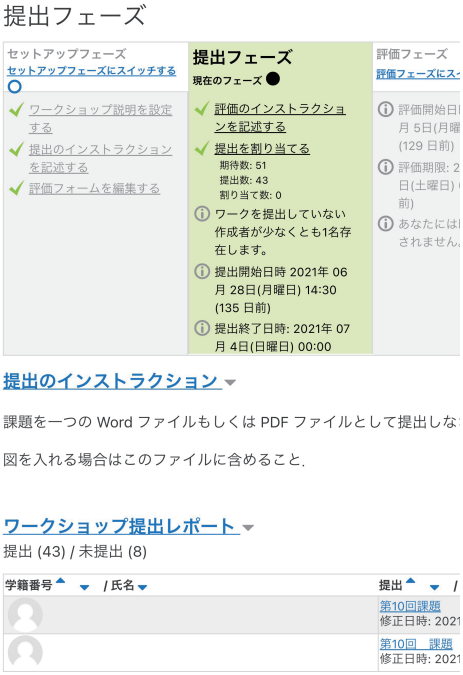


図12　提出フェーズ

る課題を割り当てるためのフェーズでもある。提出可能期間に学生が課題を提出するとこのフェーズで提出状況も確認ができる。

そして，ワークショップランナーの提出フェーズにある「提出を割り当てる」をクリックすることで，学生同士で評価対象の割り当てをおこなう。

割り当て方法には「手動で割り当て」，「ランダム割り当て」，「スケジュール割り当て」の3種類ある。それぞれ以下のように動作する。

- 手動割り当て：教員が特定の学生の課題に対して，評価する学生を選択し手動で割り当てる。
- ランダム割り当て：学生の課題に対して評価する学生をランダムに割り当てる（図13）。この割り当て設定において「レビュー数」を選択することができる。また，レビュー数は「提出あたり」と「評価者あたり」に対して数を決める。「提出あたり」とは，1つの課題提出に対して何名の学生を割り当てるかを定める。一方で，「評価者あたり」では，学生一人あたりいくつの課題を評価するかを定める。

  ある課題を評価する人数を同じにしたければ「提出あたり」を選び，ある人が評価しなければならない課題の数を同じにしたければ「評価者あたり」を使用する。

  また，「参加者は何も提出せずに評価できる」にチェックを入れると，課題は未提出でも他の人の課題を評価する権利が与えられる。

  2.4節の評価設定で「自己評価を使用する」にチェックを入れている場合は，自己評価をおこなえるようになる。
- スケジュール割り当て：ランダム割り当てを自動でおこなうことができる。これは提出フェーズの終わりに実施される。この割り当てにおける「割り当て設定」の内容はランダム割り当てと同じである。2.7節の利用で「提出終了日時後、次のフェーズに移行する」にチェックを入れておき，こちらの設定をしておくと，課題の締め切り後に自動で割り当てがおこなわれ，評価フェーズに移行できる。

図13　ランダム割り当て

3.3. 評価フェーズ

評価フェーズでは，学生が割り当てられた課題に対して各自評価をおこなう。

教員は特におこなうことはない。

3.4. 成績評価フェーズ

学生が評価をおこなう評価フェーズが終了したら「成績評価フェーズ」に移行する。このフェーズでは，「評価の比較」を適用することで，複数の学生からの評価を元に点数を決定する。「評価の比較」には，「非常に甘い」，「甘い」，「公平」，「厳しい」，「非常に厳しい」の5段階ある。この比較は「評価の比較の加重」の値として利用される。詳細は文献[4]に記載されている。

また，他の学生を評価した結果に対しても学生は評価を受ける。これが「評価に対する評点」である。この評点は同一のレポートに対する他の学生の採点結果と比較して決定される。言い換えれば，他の学生とかけ離れた採点をおこなった場合に，この評点が低くなるように設定されている。例えば，ある学生がきちんと評価をせずに0点と評価した課題に対して，その他の学生が60点と評価をしている場合に，この適当な評価をおこなった学生の「評価に対する評点」は低くなる。

これら評点の例を図14に示す。数値は括弧の外が評点で，括弧の中が評価に対する評点となる。この図では上の学生は80（20），45（12），80（20）と評価されたため評点は68となっている。この場合に45と評価した学生は他の二人の学生と評価が離れているため評価に対する評点が12（格好内の数値）となっている。同様にこの学生が他の学生を評価した事による評点は80（20），67（17），80（20）となっているため，20，17，20をもとに19となる。このように他の学生から得た評点を基に評点が計算され，他の学生に与えた評点の適性度から評価に対する評点が決定する。

3.5. 終了フェーズ

終了フェーズは，ワークショップを終了し評点を評定表に公開する。これにより学生は提出した課題に対する評点と評価に対する評点を見られるようになる。

4. パーミッションの設定

ワークショップ利用する上で重要な設

学籍番号 / 氏名	提出 / 最終更新日時	与えられた評点	提出に対する評点 (最大 80)	与えた評点	評価に対する評点 (最大 20)
	network10 修正日時: 2021年 07月 3日(土曜日) 19:19	80 (20) 45 (12) 80 (20)	68	80 (20)> 67 (17)> 80 (20)>	19
	課題10 修正日時: 2021年 07月 2日(金曜日) 13:38	80 (20) 80 (20) 80 (20)	80	80 (20)> 45 (20)> 80 (20)>	20

図14　成績評価フェーズにおける評点

定にパーミッションがある。相互評価を実施する際に，誰の課題を評価しているか分かってしまうと公平な評価が行えない可能性がある。そのため学生間で課題の提出者が誰であるか分からないようにパーミッションの設定をおこなう必要がある。また，逆に誰が評価したかという評価者名も表示されないようにしなければならない。

なお，このパーミッションの設定はMoodleの管理者によりすべてのコースに適用済みである可能性もある。その場合は特に設定をしなくてもよい。

パーミッションを設定するには「ワークショップ管理」（図15）にある「パーミッション」からおこなう。

このパーミッションの設定で重要なのは「作成者名を表示する」，「評価者名を表示する」の2つである。本学のMoodleの初期設定では図16のようになっている。そのため，少なくとも「作成者名を表示する」のパーミッションから「学生」を除かなければならない。パーミッションを削除するには図16で囲ってある学生の右側にあるゴミ箱マークをクリックする。また，図17に実際に使用したパー

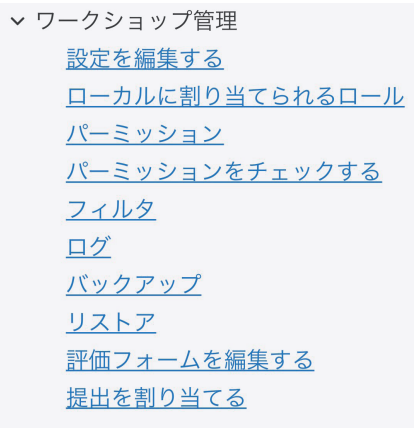


図15　ワークショップ管理

図16　パーミッションの初期設定

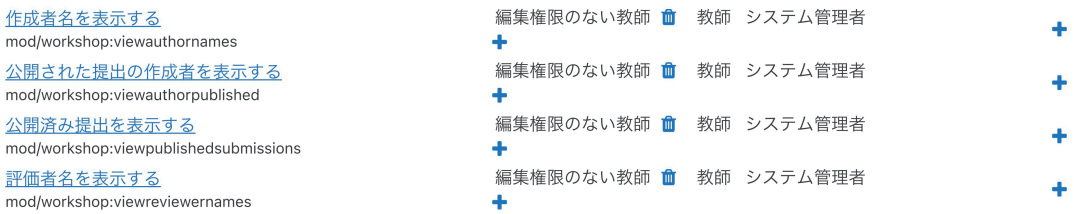


図17　変更後のパーミッション

ミッション例を示す。ただし，課題の公開を利用する場合には「公開済み提出を表示する」にはパーミッションがあってもよいはずである[注3)]。

5. 評定の設定

4章までの設定でワークショップ自体は問題なく動作するはずである。しかし，評価者の厳しさによって評価が分かれ，評点が下がってしまうことがある。

図14の上の学生は，おそらく80が妥当な評点のように思われるが，実際には68となってしまう。そこで，本稿のワークショップを用いていた授業では，評定の設定からこの点数をそのまま採用するのではなく，点数から段階評価に変更した。

ここでは，以下のように，課題の評点と評価に対する評点の合計（元の総評点）を計算し，その値により最終的な評点を計算する。

評点0：元の総評点が0
評点1：元の総評点が0以上40未満
評点2：元の総評点が40以上60未満
評点3：元の総評点が60以上80未満
評点4：元の総評点が80以上

この式を用いることで，図14の上の学生の評点は，元の総評点が87のため，4となり満点と学生と同じ評点となる。ただし，この設定はあくまでも一例であり，課題の内容や難易度によって適宜設定する方が望ましい。

6. 活用実績

2020年度，2021年度のコンピュータネットワーク論，データベース論において，それぞれ授業で各6回の課題に対してワークショップを活用して評価を実施した。

これらの授業の受講生の総数は表1の通りである。これら8コマの授業を通じ

表1　受講者数情報

年度	講義名	受講者数
2020	コンピュータネットワーク論A	50
	コンピュータネットワーク論B	60
	データベース論A	60
	データベース論B	60
2021	コンピュータネットワーク論A	44
	コンピュータネットワーク論B	51
	データベース論A	51
	データベース論B	33

[注3)] この設定では念のために学生のパーミッションを削除している。

て，学生からの異議は1件であり，質問はほとんどなかった。

異議は評点がおかしい（低い）というものであったが，5章で説明した評定の設定により最終的な評点では問題ないことを説明したところ納得してもらえた。

質問は主にルーブリックにおける評価でどれに当てはまるかという内容だったため，その都度指示をした。その他の質問で1件だけ誰にも評価がされていなく評点が入っていないようにみえるというのがあったため，教員が評価して評点をつけた。

また，きちんとしたアンケートではないが学生に口頭で聞いたところ以下のような肯定的な意見をもらった。

- 解答例を見てポイントを考えながら評価するようにルーブリックで指定されているので，教員がどのような観点から評価しているかがわかる。
- 他の人の表現を見られる（文章で書く課題の場合）ので，自分がレポートを書くときの参考になる。
- 自分の間違っていた点が明らかになるので，復習になる。

一方で，以下のような欠点も指摘された。

- 課題を提出しただけで終わりでないので大変である。
- 評価をどちらにすればよいか迷うことがあった。

このように学習効果を高める役割を果たしていることがわかるが，多少の課題があることも明らかである。ただし，欠点の一つ目はオンライン授業になったことで，ほとんどの授業がテストではなく課題による評価を採用していたため大変だと感じた学生からの意見であるとも言える。

7. まとめと今後の課題

本稿ではMoodleの標準機能であるワークショップモジュールを使い学生同士の相互評価を実施する方法について解説をした。また，実際に授業で用いた結果についても簡単ではあるが述べた。

この，ワークショップモジュールは設定しなければいけない項目や，注意しなければならない点がいくつかあるため，利用しづらいといえる。また，情報も豊富ではないため，本稿が今後ワークショップモジュールを利用したい方々への一助となることを願う。

今後は，学生が評価をおこないやすいようにルーブリックの内容を見直す予定である。また，活用実績を積み重ねることで，今は明らかになっていない問題点やさらなる活用方法について検討する。

参考文献

1）Moodle，2021年9月13日更新。
https://moodle.org,［アクセス日：2021年11月10日］。

2）GNU一般公衆ライセンス，2007年6月29日更新。
https://www.gnu.org/licenses/gpl-3.0.html［アクセス日：2021年11月10日］。

3）ワークショップ活動，2020年11月17日更新。
https://docs.moodle.org/3x/ja/ワークショップ活動，アクセス日：2021年11月10日］。

4）吉岡一志，宇田川暢，“Moodleワークショップモジュールのマニュアル作成と課題”，山口県立大学学術情報［山口県立大学学術情報センター］，Vol.10，pp.161-167，(2017)。

5）ワークショップの評定方法，2020年10月19日更新。
https://docs.moodle.org/3x/ja/ワークショップの評定方法［アクセス日：2021年11月10日］。

6）評定カテゴリ，2021年1月27日更新。
https://docs.moodle.org/3x/ja/評定カテゴリ［アクセス日：2021年11月10日］。

7）モジュール共通設定，2020年7月27日更新。
https://docs.moodle.org/3x/ja/モジュール共通設定［アクセス日：2021年11月10日］。

8）活動へのアクセスを制限する，2020年8月3日更新。
https://docs.moodle.org/3x/ja/アクセス制限の設定#.E6.B4.BB.E5.8B.95.E3.81.B8.E3.81.AE.E3.82.A2.E3.82.AF.E3.82.BB.E3.82.B9.E3.82.92.E5.88.B6.E9.99.90.E3.81.99.E3.82.8B［アクセス日：2021年11月10日］。

2019年度と2020年度におけるMoodleアクセス状況調査

森野　誠之（非常勤）

要旨

新型コロナウィルスの影響により2020年度は全学でMoodleを利用することとなった。通期での利用コース数は1,000コースほどから一気に8,500コースほどと8倍以上になり，アクセス数も月によってばらつきはあるが10～20倍となっている。全体のアクセス数，利用デバイスや時間帯などの状況を調査した結果を報告する。

キーワード：Moodle，新型コロナウィルス，Googleアナリティクス

1．はじめに

本学で運用しているMoodleは2019年度以前では利用希望教員のみの利用であったが，2020年度は新型コロナウィルスの影響により全学で利用することとなった。教職員，学生はユーザーとして全員登録していたため問題はなかった。コース登録は全コースを情報システム課にて一括登録し，履修者も同様に一括登録をしている。授業での利用は，2019年度まではリアルタイムでの利用はPC教室がほとんどであり，同時アクセス数はPCの台数が上限に近かったものの，2020年は教員と学生のPCに加えスマートフォンからのアクセスも含まれているために大きく変化している。コース数は通期での利用コース数は1,000コースほどから一気に8,500コースほどと8倍以上になり，アクセス数も月によってばらつきはあるが10～20倍となっている。

2．計測について

ログイン数と活動数についてはMoodleのサーバーログからデータを取得している。活動とはコース内に設置した課題や小テストなどへのリンクをクリックした場合にカウントされ，Moodleにログイン後にどれだけのページを閲覧したのかが分かる。ただし，ワープロファイルやPDFなどの電子ファイルをダウンロード閲覧した場合は含まれない。学内・学外の判断はIPアドレスで行っているために，学内からのアクセスでもWi-Fiに接続していない場合は学外となる。

Googleアナリティクスでの計測はJavaScriptの計測タグをMoodle内に設置し，そのタグが読み込まれた場合に計測される。運用時の参考データを取ることを目的と

しているため，100％正確なデータではないことをあらかじめご理解いただきたい。Googleアナリティクスではアクセス数をセッションでカウントしている。セッションは何も操作されずに30分を超えた場合か日付をまたいだ際に終了する。終了した時点で1セッションとなるが，30分以内にMoodle上で何らかの操作をすれば，その間に他のWebサイトを閲覧していてもセッションは継続される。また，30分以内であれば何度ログインしても1セッションとカウントされることには注意が必要である。日付をまたいだ場合は，例えば小テストのように継続的にアクセスしている場合でも終了し，新たなセッションが始まるために2セッションとカウントされる。

3．利用状況

3-1．ログイン数と活動数推移

表1に19/04～20/02までの学内・学外からのログイン数・活動数推移と合計・平均を示す。3月は授業などの利用がないために割愛している。

表1．2019年度のログインと活動数

		4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	合計	平均
19年度	総ログイン数(学内)	23,767	34,861	32,118	37,200	990	17,824	34,205	28,272	27,356	14,593	347	251,533	22,867
	総ログイン数(学外)	38,789	58,230	53,939	74,926	7,418	26,245	48,612	44,312	44,681	49,042	2,216	448,410	40,765
	総活動数(学内)	461,465	574,161	533,844	700,879	19,288	276,896	471,716	357,383	375,973	239,322	7,007	4,017,934	365,267
	総活動数(学外)	505,463	746,053	766,831	1,164,778	105,650	306,291	544,806	483,347	528,962	650,874	42,804	5,845,859	531,442
	ログインあたり活動数(学内)	19.42	16.47	16.62	18.84	19.48	15.54	13.79	12.64	13.74	16.40	20.19	183.13	16.65
	ログインあたり活動数(学外)	13.03	12.81	14.22	15.55	14.24	11.67	11.21	10.91	11.84	13.27	19.32	148.07	13.46

最もログイン数と活動数が多かったのは7月で学内ログイン数は37,200，学外ログイン数は74,926，学内活動数は700,879，学外活動数は1,164,778であった。7月はテスト前の期間でもあり休日もないのでアクセスが増えやすい。なお，この数値は2018年度以前の中でも最大となっている。図1に19/04～20/02までの学内・学外からのログイン数・活動数推移を示す。

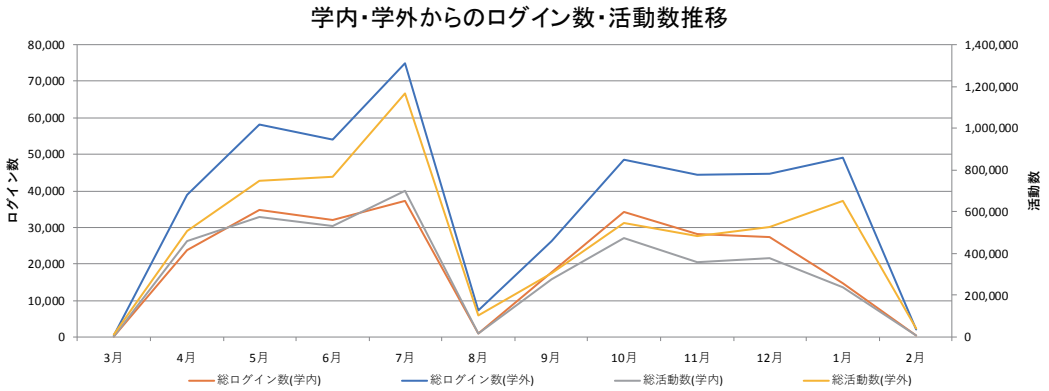


図1. 2019年度のログインと活動の推移

春学期は7月にピークが来て，秋学期はピークがなく均等にアクセスされていることがわかる。表2に20/04～21/02までの学内・学外からのログイン数・活動数推移と合計・平均を示す。

表2. 2020年度のログインと活動数

		4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	合計	平均
19年度	総ログイン数(学内)	2,713	8,487	6,698	5,394	2,123	7,793	32,979	27,522	26,722	9,697	892	131,020	11,911
	総ログイン数(学外)	20,006	648,209	661,396	685,698	139,800	357,397	465,944	363,780	353,502	272,696	16,835	3,985,263	362,297
	総活動数(学内)	136,249	523,016	254,279	201,620	163,036	427,351	707,450	577,629	560,515	236,015	51,181	3,838,341	348,940
	総活動数(学外)	289,722	13,123,453	13,463,238	14,148,403	2,907,150	6,517,112	8,293,424	6,204,644	6,213,003	4,806,766	357,769	76,324,684	6,938,608
	ログインあたり活動数(学内)	50.22	61.63	37.96	37.38	76.80	54.84	21.45	20.99	20.98	24.34	57.38	463.97	42.18
	ログインあたり活動数(学外)	14.48	20.25	20.36	20.63	20.80	18.23	17.80	17.06	17.58	17.63	21.25	206.07	18.73

2020年度も7月がピークとなっており，学内ログイン数は5,394，学外ログイン数は685,698，学内活動数は201,620，学外活動数は14,148,403であった。学内からのログイン数と活動数はオンライン授業のために前年度を下回っているが，学外からのログインと活動数は前年度の10倍ほどと大きく増加している。7月以外でも学外からの活動数が増えており8,9月に関しては20倍以上となっている。図2に20/04～21/02までの学内・学外からのログイン数・活動数推移を示す。

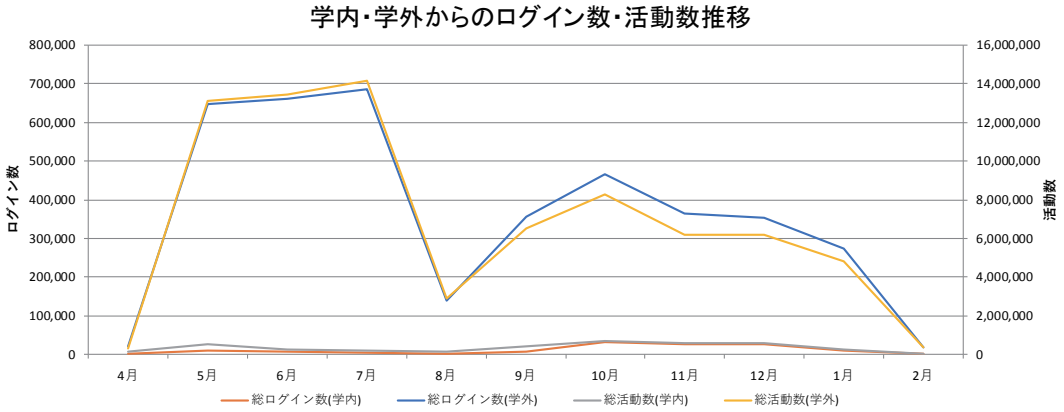


図2. 2020年度のログインと活動の推移

学外からのログインに偏っており，5〜7月は活動数が12,000,000を超えている。新型コロナウィルスの影響でMoodleを用いたオンライン授業になったため，今まででは想定できないようなアクセスがあったことがわかる。

3-2. ブラウザ別のセッション数

図3に2019年度のブラウザ別のセッション数を示す。

		901,012 全体に対する割合: 100.00% (901,012)	901,012 全体に対する割合: 100.00% (901,012)
1.	Safari	467,065	51.84%
2.	Internet Explorer	210,878	23.40%
3.	Chrome	112,303	12.46%
4.	Edge	83,051	9.22%
5.	Firefox	10,695	1.19%
6.	Safari (in-app)	9,277	1.03%
7.	Samsung Internet	4,250	0.47%
8.	Android Webview	2,599	0.29%
9.	Opera	305	0.03%
10.	UC Browser	153	0.02%

図3. 2019年度のブラウザ別のセッション数

SafariはほとんどがiPhoneからのアクセスである。学生の大半がiPhoneユーザーであること，移動中などのちょっとした時間に頻繁にアクセスができることから全体の半数以上を占めるセッション数となっている。Internet ExplorerはPC教室とWindowsの標準ブラウザであったことで全体の23％を締めている。ChromeはAndroidとWindowsユーザーからのセッションだと考えられる。また，Edgeユーザーは1割に満たず2019年度はほとんど使われていないこともわかる。

		5,446,374 全体に対する割合: 100.00% (5,446,374)	5,446,374 全体に対する割合: 100.00% (5,446,374)
1.	Edge	2,165,087	39.75%
2.	Safari	1,596,230	29.31%
3.	Chrome	1,390,854	25.54%
4.	Internet Explorer	191,262	3.51%
5.	Firefox	46,772	0.86%
6.	Safari (in-app)	32,134	0.59%
7.	Samsung Internet	12,296	0.23%
8.	Android Webview	5,826	0.11%
9.	Opera	3,107	0.06%
10.	YaBrowser	747	0.01%

図4．2020年度のブラウザ別のセッション数

図4は2020年度のブラウザ別のセッション数である。2019年度と大きく変化し，Edgeが40％ほど，Safariが30％ほど，Chromeが25％ほどとなっており，Internet Explorer利用は3.5％と激減している。Internet ExplorerはMoodleなどのWebサービスから動作対象外となっており，実際にうまく表示できないなどのトラブルが多発したために利用者がEdge，もしくはChromeに乗り換えたことが要因である。

3-3. OS別のセッション数

		901,012 全体に対する割合: 100.00% (901,012)	901,012 全体に対する割合: 100.00% (901,012)
1.	iOS	475,207	52.74%
2.	Windows	348,148	38.64%
3.	Android	62,596	6.95%
4.	Macintosh	14,562	1.62%
5.	Linux	356	0.04%
6.	(not set)	132	0.01%
7.	Chrome OS	5	0.00%
8.	Playstation 4	5	0.00%
9.	Playstation Vita	1	0.00%

図5. 2019年度のOS別のセッション数

図5に2019年度のOS別のセッション数を示す。iOSは53％ほどとなっており図3のSafariと近しい値となっている。またiOSとWindowsで全体の9割を占めている。AndroidとMacはほとんど使われていないこともわかる。

		5,446,374 全体に対する割合: 100.00% (5,446,374)	5,446,374 全体に対する割合: 100.00% (5,446,374)
1.	Windows	3,555,395	65.28%
2.	iOS	1,378,516	25.31%
3.	Macintosh	351,076	6.45%
4.	Android	159,738	2.93%
5.	Linux	744	0.01%
6.	(not set)	485	0.01%
7.	Chrome OS	404	0.01%
8.	Playstation 4	15	0.00%
9.	Nintendo 3DS	1	0.00%

図6. 2020年度のOS別のセッション数

図6の2020年度のOS別のセッション数ではWindowsからセッション数が38.64％

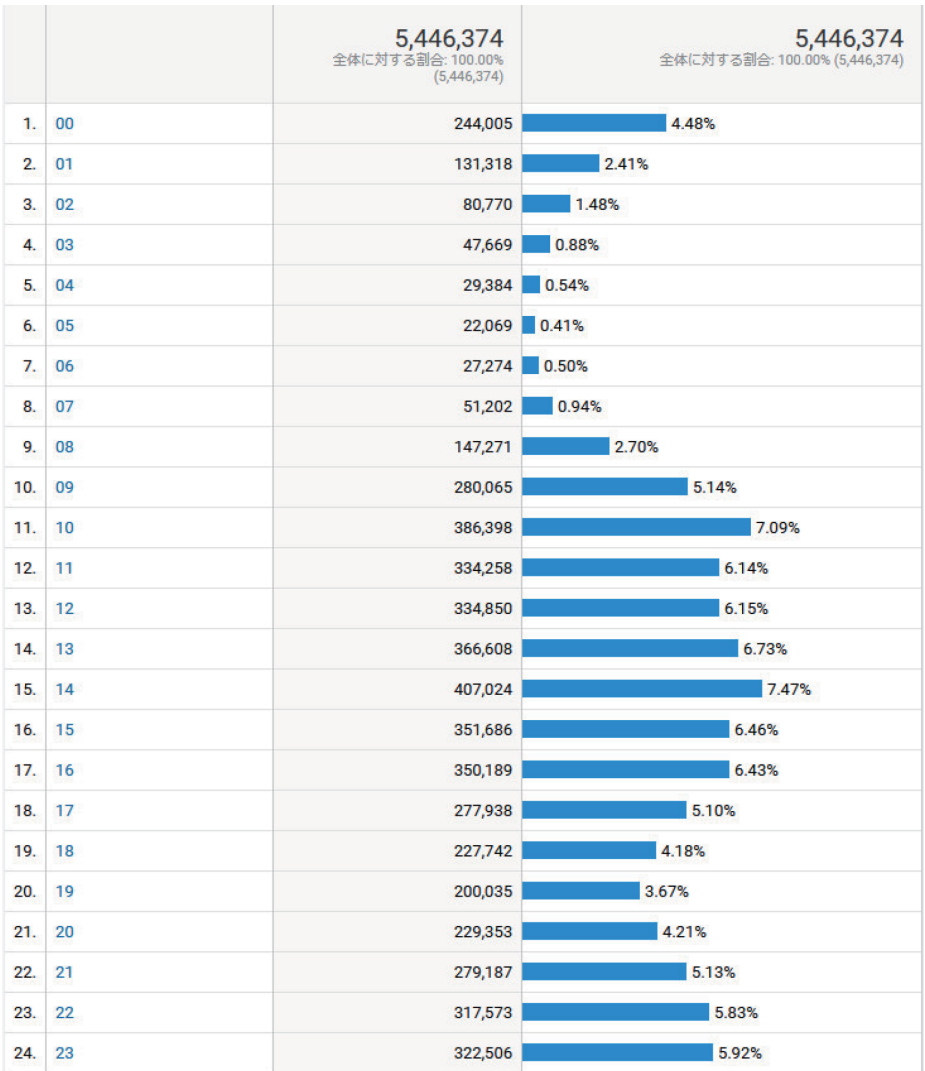
から65.28％と急増している。19年度はMoodleの利用は学内でPC，学外ではスマートフォンとなっており，学外では閲覧を主にした利用だったことが想定されるが，20年度は学外でもPCからアクセスされ，学外の学習でもPCがメインデバイスとして使われるようになったことがわかる。授業の動画自体はスマートフォンでも閲覧可能だが，課題などはPCからの提出がやりやすいことが影響していると考えられる。MacはiPhoneとの相性も良いことから，遠隔授業を機に新しく購入した学生と教員が増えていると考えられ，Androidとセッション数比率が逆転している。

3-4. 時間帯別のセッション数

		901,012 全体に対する割合: 100.00% (901,012)	901,012 全体に対する割合: 100.00% (901,012)
1.	00	31,422	3.49%
2.	01	17,046	1.89%
3.	02	9,221	1.02%
4.	03	4,890	0.54%
5.	04	3,049	0.34%
6.	05	2,969	0.33%
7.	06	6,097	0.68%
8.	07	13,407	1.49%
9.	08	29,266	3.25%
10.	09	52,834	5.86%
11.	10	81,992	9.10%
12.	11	62,350	6.92%
13.	12	76,006	8.44%
14.	13	78,349	8.70%
15.	14	91,095	10.11%
16.	15	56,635	6.29%
17.	16	62,383	6.92%
18.	17	36,008	4.00%
19.	18	28,342	3.15%
20.	19	23,235	2.58%
21.	20	26,175	2.91%
22.	21	31,522	3.50%
23.	22	37,201	4.13%
24.	23	39,518	4.39%

図7. 2019年度の時間帯別のセッション数

2019年度の時間帯別セッション数を図7に示す。授業時間内のセッション数が多く夜間のセッションは5％以下となっている。前述のように学内のPC教室でのMoodle利用が主であることからこのような傾向になっており夜間の利用は少ない。



		5,446,374 全体に対する割合: 100.00% (5,446,374)	5,446,374 全体に対する割合: 100.00% (5,446,374)
1.	00	244,005	4.48%
2.	01	131,318	2.41%
3.	02	80,770	1.48%
4.	03	47,669	0.88%
5.	04	29,384	0.54%
6.	05	22,069	0.41%
7.	06	27,274	0.50%
8.	07	51,202	0.94%
9.	08	147,271	2.70%
10.	09	280,065	5.14%
11.	10	386,398	7.09%
12.	11	334,258	6.14%
13.	12	334,850	6.15%
14.	13	366,608	6.73%
15.	14	407,024	7.47%
16.	15	351,686	6.46%
17.	16	350,189	6.43%
18.	17	277,938	5.10%
19.	18	227,742	4.18%
20.	19	200,035	3.67%
21.	20	229,353	4.21%
22.	21	279,187	5.13%
23.	22	317,573	5.83%
24.	23	322,506	5.92%

図8.　2020年度の時間帯別のセッション数

図8の2020年度の時間帯別のセッション数では21時以降のセッション数が増えており日中とほぼ同数となっている。夜間に学習に取り組む学生が増加したことがわかる。課題の締め切りが23：59に設定されることも多いことも要因の1つであろう。

3-5. 曜日別のセッション数の比較

図9に曜日別のセッション数を比較したものを示す。Googleアナリティクスの仕様により0が日曜で7が土曜の表示になっているが曜日の順に並んでいる。

		504.47% ▲ 5,446,374 と 901,012	504.47% ▲ 5,446,374 と 901,012
1.	0		
	2020/04/01 - 2021/02/28	542,074	9.95%
	2019/04/01 - 2020/02/28	63,499	7.05%
2.	1		
	2020/04/01 - 2021/02/28	992,494	18.22%
	2019/04/01 - 2020/02/28	189,819	21.07%
3.	2		
	2020/04/01 - 2021/02/28	999,072	18.34%
	2019/04/01 - 2020/02/28	182,397	20.24%
4.	3		
	2020/04/01 - 2021/02/28	931,152	17.10%
	2019/04/01 - 2020/02/28	176,492	19.59%
5.	4		
	2020/04/01 - 2021/02/28	696,569	12.79%
	2019/04/01 - 2020/02/28	101,405	11.25%
6.	5		
	2020/04/01 - 2021/02/28	838,650	15.40%
	2019/04/01 - 2020/02/28	138,995	15.43%
7.	6		
	2020/04/01 - 2021/02/28	446,363	8.20%
	2019/04/01 - 2020/02/28	48,405	5.37%

図9. 2019年度と2020年度の曜日別セッション数の比較

授業がない土日，半日授業の木曜でセッション数が増加しており，反対に月・火・水曜日のセッション数は減っている。オンライン授業になり課題などが増えたことから授業時間内の学習では時間が足りなくなり，土日などに学習や課題の作成をしている学生が多いと考えられる。

4. まとめ

2019年度と新型コロナウィルスの影響を受けた2020年度は以下のような変化があった。

・学内PC教室の利用から自宅でのPC利用にシフト
・スマートフォンからのアクセスからPCからのアクセスにシフト
・夜間，土日の授業時間外のアクセスが増加

オンライン授業になり教材と課題や小テストなどに常時アクセスできることから，学生自身の都合に合わせて学習をすることができるようになったメリットあるが，直前まで着手していない可能性もあり，より学生の自主性が求められるようになっている。2022年度の授業形態がどのようになるかは現段階では分からないものの，Moodle利用教員の増加と学生のオンライン授業慣れの影響からこの傾向は続くと考えられるため，継続的な調査をしていく予定である。

1. ICT委員会　会議報告

愛知大学情報メディアセンターの事業および運営は，ICT企画会議のもと，三校舎合同のITC委員会を設置し，豊橋および名古屋(車道メディアゾーン含む)情報メディアセンターの事業を推進する。　（2020年10月から2021年9月まで）

2020年度
◇第3回11月20日
議題：
1. 2021年度予算申請について
2. 2021年度情報メディアセンター開館カレンダーについて
3. 車道校舎実習室のパソコンについて

協議・報告：
1. 2020年度システム更新について
2. 推奨PCについて
3. 標的型攻撃メール訓練について

2021年度
◇第1回5月13日
議題：
1. COM編集委員選出について
2. 情報セキュリティ部会委員選出について

◇第2回9月30日
議題：
1. 補正予算申請について
2. 2022年度実習室環境について

協議・報告：
1. 2021年度秋学期の教育用ソフトウェア利用申請について
2. 2021年度COMについて
3. SSLVPNサービスについて
4. 学生の推奨化ノートパソコンの環境について
5. 2022年度予算申請について

2. 情報メディアセンター主催行事

（2020年10月～2021年9月）

◆名古屋校舎

開　講　日	講　習　会　名	教室	参加人数
2020年12月17日(木)	Word講習会・レポート編（Word2016）	オンライン	2 人
2021年 4 月 8 日(木)	はじめてのTeams講習会	オンライン	9 人
2021年 4 月 8 日(木)	はじめてのTeams講習会	オンライン	7 人
2021年 5 月13日(木)	はじめてのPowerPoint講習会(PowerPoint2016)	オンライン	3 人
2021年 5 月13日(木)	はじめてのPowerPoint講習会(PowerPoint2016)	オンライン	3 人
2021年 5 月27日(木)	Word講習会・レポート編（Word2016）	オンライン	14人
2021年 6 月 3 日(木)	PowerPoint講習会（PowerPoint2016）	オンライン	13人
2021年 6 月11日(金)	Excel講習会・グラフ編（Excel2016）	L707	3 人
2021年 8 月11日(水)	Word講習会・レポート編（Word2016）	オンライン	1 人
2021年 9 月30日(木)	Word講習会・卒論編（Word2016）	オンライン	13人

◆豊橋校舎：主催行事なし

◆車道校舎：主催行事なし

Moodle（LMS）運営業務報告（2019・2020年度）

1. 2019年度　業務報告

1.1 Moodle講習会

Moodleの利用促進のため、Moodle講習会を以下の通り実施した。

①第30回Moodle講習会

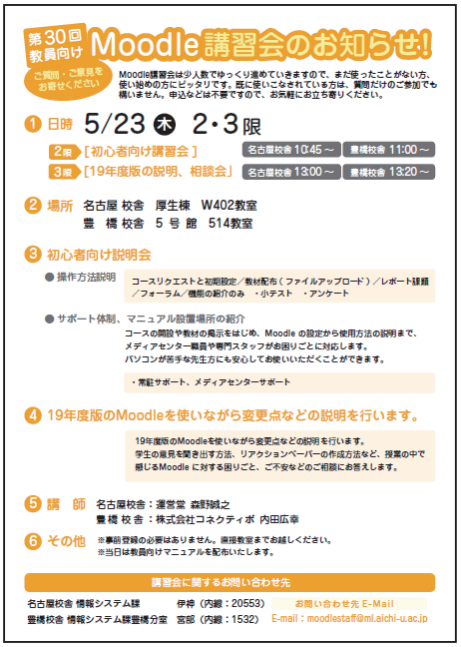
校舎	開催日時	場所
豊橋	2019年5月23日	5号館　514教室
名古屋	2・3限	厚生棟W402教室

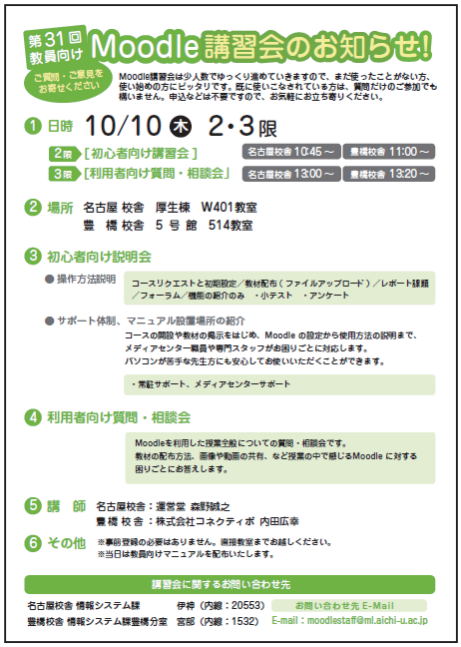
②第31回Moodle講習会

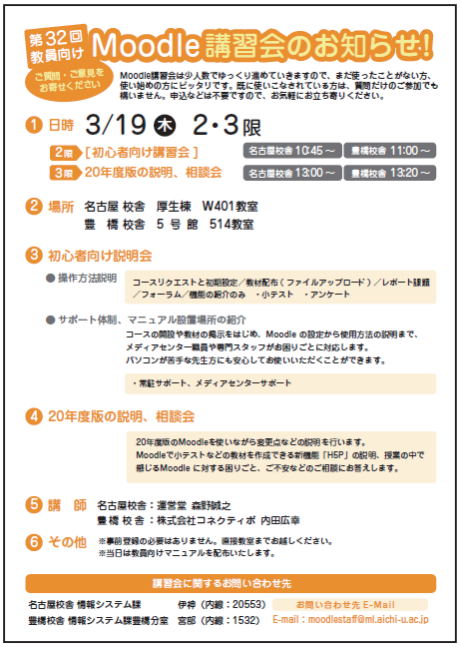
校舎	開催日時	場所
豊橋	2019年10月10日	5号館　514教室
名古屋	2・3限	厚生棟W401教室

③第32回Moodle講習会

校舎	開催日時	場所
豊橋	2020年3月19日	5号館　514教室
名古屋	2・3限	厚生棟W401教室







1.2 Moodle利用状況

（A）コース利用状況

運用開始11年目の2019年度は，731コース，延べ363名の教員の利用があった。合計コース数および利用教員数ともに，前年度と比較して4～7％程度増加した。

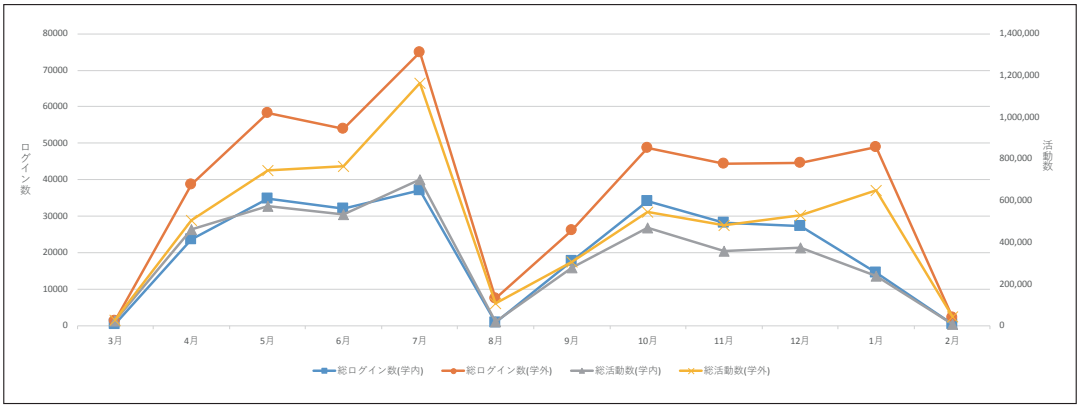
2018・2019年度コース登録数及び利用教員数（利用人数は，延べ人数）

カテゴリ	19年度春学期		19年度秋学期		合計		前年比率	
	コース数	教員数	コース数	教員数	コース数	教員数	コース数	教員数
共通教育科目〈豊橋〉	36	29	28	15	64	44	74％	77％
共通教育科目〈名古屋〉	78	36	29	18	107	54	96％	100％
法	30	26	10	8	40	34	100％	136％
現中	22	12	9	5	31	17	115％	100％
経営	67	23	14	9	81	32	123％	110％
経済	56	24	18	12	74	36	119％	106％
文	67	25	22	11	89	36	119％	113％
国コミ	39	14	15	7	54	21	90％	84％
地域	53	21	28	15	81	36	129％	133％
短大	16	8	6	5	22	13	169％	163％
法科	13	6	2	2	15	8	94％	100％
大学院	5	2	1	1	6	3	60％	43％
資格課程	43	15	18	11	61	26	133％	137％
その他	5	2	1	1	6	3	86％	50％
合計	530	243	201	120	731	363	107％	104％

カテゴリ	18年度春学期		18年度秋学期		合計		前年比率	
	コース数	教員数	コース数	教員数	コース数	教員数	コース数	教員数
共通教育科目〈豊橋〉	48	30	38	27	86	57	246％	228％
共通教育科目〈名古屋〉	90	42	21	12	111	54	78％	77％
法	34	21	6	4	40	25	105％	104％
現中	17	10	10	7	27	17	123％	155％
経営	56	24	10	5	66	29	106％	107％
経済	47	23	15	11	62	34	129％	117％
文	55	25	20	7	75	32	142％	128％
国コミ	47	19	13	6	60	25	82％	96％
地域	45	18	18	9	63	27	117％	108％
短大	9	6	4	2	13	8	130％	100％
法科	15	7	1	1	16	8	123％	114％
大学院	8	5	2	2	10	7	200％	140％
資格課程	38	14	8	5	46	19	131％	146％
その他	4	3	3	3	7	6	100％	120％
合計	513	247	169	101	682	348	114％	116％

（B）サイトアクセス状況

2019年度　学内・学外からのログイン数・活動数推移（月別）



		3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	合計	平均
2017年度	総ログイン数（学内）	252	22,285	31,648	29,730	27,430	748	14,941	26,339	19,795	18,381	12,241	301	204,091	17,008
	総ログイン数（学外）	807	26,443	34,528	32,313	42,790	4,679	18,719	35,109	26,105	27,550	33,932	1,963	284,938	23,745
	総活動数（学内）	3,321	502,910	484,384	540,335	582,451	13,854	219,535	461,564	298,655	244,556	223,295	7,013	3,581,873	298,489
	総活動数（学外）	12,434	354,446	486,314	485,676	787,910	75,479	243,875	458,915	419,385	358,040	521,772	41,407	4,245,653	353,804
	ログインあたり活動数（学内）	13.18	22.57	15.31	18.17	21.23	18.52	14.69	17.52	15.09	13.3	18.24	23.3	17.55	17.59
	ログインあたり活動数（学外）	15.41	13.4	14.08	15.03	18.41	16.13	13.03	13.07	16.07	13	15.38	21.09	14.9	15.34
2018年度	総ログイン数（学内）	366	24,810	38,348	36,359	33,528	986	14,090	32,035	24,770	21,096	11,290	324	238,002	19,834
	総ログイン数（学外）	1,077	36,696	49,167	48,262	57,294	6,114	19,064	38,293	32,208	31,549	35,778	1,986	357,488	29,791
	総活動数（学内）	26,564	677,532	644,977	658,176	630,869	17,445	211,906	423,267	311,240	265,314	183,726	4,931	4,055,947	337,996
	総活動数（学外）	26,299	555,986	669,514	857,416	903,313	84,874	223,651	431,908	379,364	370,352	523,280	40,219	5,066,176	422,181
	ログインあたり活動数（学内）	72.58	27.31	16.82	18.1	18.82	17.69	15.04	13.21	12.57	12.58	16.27	15.22	17.04	21.35
	ログインあたり活動数（学外）	24.42	15.15	13.62	17.77	15.77	13.88	11.73	11.28	11.78	11.74	14.63	20.25	14.17	15.17
2019年度	総ログイン数（学内）	421	23,767	34,861	32,118	37,200	990	17,824	34,205	28,272	27,356	14,593	347	251,954	20,996
	総ログイン数（学外）	1,276	38,789	58,230	53,939	74,926	7,418	26,245	48,612	44,312	44,681	49,042	2,216	449,686	37,474
	総活動数（学内）	22,626	461,465	574,161	533,844	700,879	19,288	276,896	471,716	357,383	375,973	239,322	7,007	4,040,560	336,713
	総活動数（学外）	26,069	505,463	746,053	766,831	1,164,778	105,650	306,291	544,806	483,347	528,962	650,874	42,804	5,871,928	489,327
	ログインあたり活動数（学内）	53.74	19.42	16.47	16.62	18.84	19.48	15.54	13.79	12.64	13.74	16.40	20.19	16.04	19.74
	ログインあたり活動数（学外）	20.43	13.03	12.81	14.22	15.55	14.24	11.67	11.21	10.91	11.84	13.27	19.32	13.06	14.04
前年同月比	総ログイン数（学内）	115.0%	95.8%	90.9%	88.3%	111.0%	100.4%	126.5%	106.8%	114.1%	129.7%	129.3%	107.1%	105.9%	105.9%
	総ログイン数（学外）	118.5%	105.7%	118.4%	111.8%	130.8%	121.3%	137.7%	126.9%	137.6%	141.6%	137.1%	111.6%	125.8%	125.8%
	総活動数（学内）	85.2%	68.1%	89.0%	81.1%	111.1%	110.6%	130.7%	111.4%	114.8%	141.7%	130.3%	142.1%	99.6%	99.6%
	総活動数（学外）	99.1%	90.9%	111.4%	89.4%	128.9%	124.5%	137.0%	126.1%	127.4%	142.8%	124.4%	106.4%	115.9%	115.9%

※2018年度4月は日本語ｅラーニングのデータ削除が総活動数（学内）に含まれています

2. 2020年度　業務報告

2.1　Moodle講習会

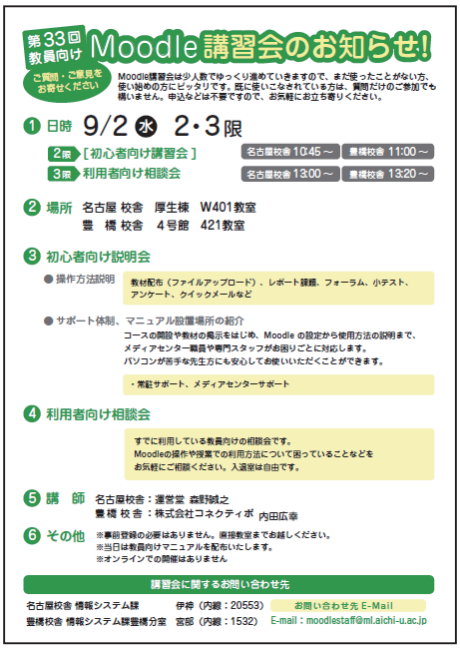
Moodleの利用促進のため、Moodle講習会を以下の通り実施した。

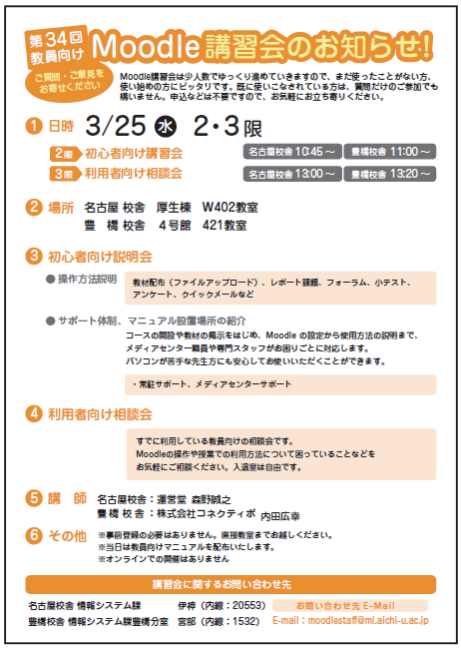
①第33回Moodle講習会

校舎	開催日時	場所
豊橋	2020年9月2日	4号館　421教室
名古屋	2・3限	厚生棟W401教室

②第34回Moodle講習会

校舎	開催日時	場所
豊橋	2021年3月25日	4号館　421教室
名古屋	2・3限	厚生棟W402教室





2.2 Moodle利用状況

（A）コース利用状況

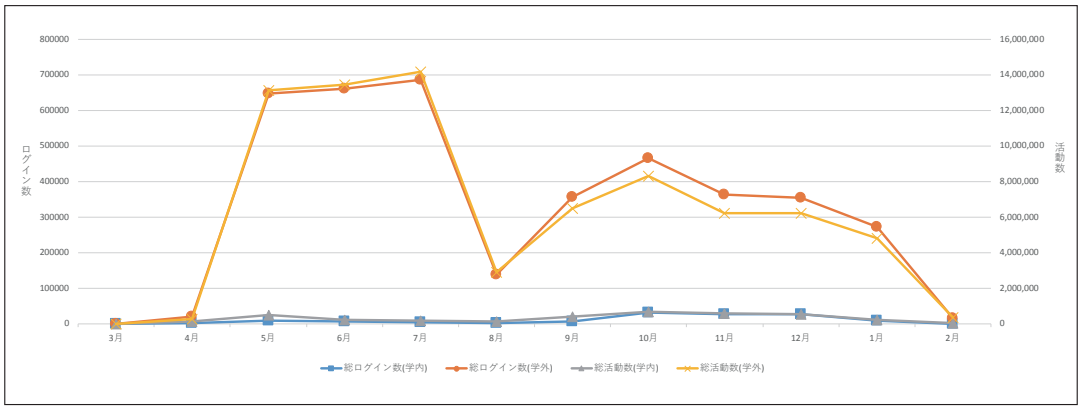
運用開始12年目の2020年度は，新型コロナウイルス感染症の拡大に伴い大学の全授業を遠隔授業で実施した。そのため，すべての授業でコースを作成することとなったため，前年度に比べてコース数が大きく増え，約6倍となった。（新型コロナウイルス感染症の蔓延状況により，一部の授業は，対面で実施している）

カテゴリ	2020年度　利用コース数			
	春学期	秋学期	通年	合計
共通教育科目（名古屋）〈法・経済・経営・現中・国際〉	412	424	6	842
共通教育科目（豊橋）〈文・地域・短大〉	260	239	2	501
法学部	48	77	47	172
経済学部	79	79	66	224
経営学部	109	144	58	311
現代中国学部	175	143	22	340
国際コミュニケーション学部	172	190	7	369
文学部	212	205	27	444
地域政策学部	120	163	12	295
短期大学部	75	86	2	163
大学院	94	81	56	231
法科大学院	44	47	0	91
資格課程	74	87	13	174
協定留学生日本語コース	11	0	0	11
自習用教材	0	0	36	36
ヘルプ	0	0	9	9
その他	0	0	14	14
合計	1885	1965	377	4227

（B）サイトアクセス状況

2020年度は，新型コロナウイルス感染症の拡大に伴い，授業が遠隔授業となり，授業開始日を4月から5月に変更したため，5月からログイン回数が急激に増大している。また，5月の総ログイン数（学外）は，昨年と比べて約11倍に増加している。

2020年度　学内・学外からのログイン数・活動数推移（月別）



		3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	合計	平均
2018年度	総ログイン数（学内）	366	24,810	38,348	36,359	33,528	986	14,090	32,035	24,770	21,096	11,290	324	238,002	19,834
	総ログイン数（学外）	1,077	36,696	49,167	48,262	57,294	6,114	19,064	38,293	32,208	31,549	35,778	1,986	357,488	29,791
	総活動数（学内）	26,564	677,532	644,977	658,176	630,869	17,445	211,906	423,267	311,240	265,314	183,726	4,931	4,055,947	337,996
	総活動数（学外）	26,299	555,986	669,514	857,416	903,313	84,874	223,651	431,908	379,364	370,352	523,280	40,219	5,066,176	422,181
	ログインあたり活動数（学内）	72.58	27.31	16.82	18.1	18.82	17.69	15.04	13.21	12.57	12.58	16.27	15.22	17.04	21.35
	ログインあたり活動数（学外）	24.42	15.15	13.62	17.77	15.77	13.88	11.73	11.28	11.78	11.74	14.63	20.25	14.17	15.17
2019年度	総ログイン数（学内）	421	23,767	34,861	32,118	37,200	990	17,824	34,205	28,272	27,356	14,593	347	251,954	20,996
	総ログイン数（学外）	1,276	38,789	58,230	53,939	74,926	7,418	26,245	48,612	44,312	44,681	49,042	2,216	449,686	37,474
	総活動数（学内）	22,626	461,465	574,161	533,844	700,879	19,288	276,896	471,716	357,383	375,973	239,322	7,007	4,040,560	336,713
	総活動数（学外）	26,069	505,463	746,053	766,831	1,164,778	105,650	306,291	544,806	483,347	528,962	650,874	42,804	5,871,928	489,327
	ログインあたり活動数（学内）	53.74	19.42	16.47	16.62	18.84	19.48	15.54	13.79	12.64	13.74	16.40	20.19	16.04	19.74
	ログインあたり活動数（学外）	20.43	13.03	12.81	14.22	15.55	14.24	11.67	11.21	10.91	11.84	13.27	19.32	13.06	14.04
2020年度	総ログイン数（学内）	122	2,713	8,487	6,698	5,394	2,123	7,793	32,979	27,522	26,722	9,697	892	131,142	10,929
	総ログイン数（学外）	1,272	20,006	648,209	661,396	685,698	139,800	357,397	465,944	363,780	353,502	272,696	16,835	3,986,535	332,211
	総活動数（学内）	1,704	136,249	523,016	254,279	201,620	163,036	427,351	707,450	577,629	560,515	236,015	51,181	3,840,045	320,004
	総活動数（学外）	25,766	289,722	13,123,453	13,463,238	14,148,403	2,907,150	6,517,112	8,293,424	6,204,644	6,213,003	4,806,766	357,769	76,350,450	6,362,538
	ログインあたり活動数（学内）	13.97	50.22	61.63	37.96	37.38	76.80	54.84	21.45	20.99	20.98	24.34	57.38	29.28	39.83
	ログインあたり活動数（学外）	20.26	14.48	20.25	20.36	20.63	20.80	18.23	17.80	17.06	17.58	17.63	21.25	19.15	18.86
前年同月比	総ログイン数（学内）	29.0%	11.4%	24.3%	20.9%	14.5%	214.4%	43.7%	96.4%	97.3%	97.7%	66.4%	257.1%	52.0%	52.0%
	総ログイン数（学外）	99.7%	51.6%	1113.2%	1226.2%	915.2%	1884.6%	1361.8%	958.5%	821.0%	791.2%	556.0%	759.7%	886.5%	886.5%
	総活動数（学内）	7.5%	29.5%	91.1%	47.6%	28.8%	845.3%	154.3%	150.0%	161.6%	149.1%	98.6%	730.4%	95.0%	95.0%
	総活動数（学外）	98.8%	57.3%	1759.1%	1755.7%	1214.7%	2751.7%	2127.8%	1522.3%	1283.7%	1174.6%	738.5%	835.8%	1300.3%	1300.3%

※2018年度4月は日本語ｅラーニングのデータ削除が総活動数（学内）に含まれています

3. ICT委員会構成員

◆ICT委員（2021年10月1日現在）

役職名	所　　属	氏　　名
情報メディアセンター所長	経　営　学　部	岩田　員典
委　　員	文　　学　　部	近藤　暁夫
	地域政策学部	蒋　　湧
	短期大学部	迫田　耕作
	法　　学　　部	松井　吉光
	経　営　学　部	毛利　元昭
	現代中国学部	吉川　　剛
	経　済　学　部	池森　　均
	国際コミュニケーション学部	梅垣　敦紀
	法科大学院	春日　　修

◆情報メディアセンター事務室

情報システム課	課　　長	三浦　文博
	係　　長	石原有希子
		水谷　伸司
	課　　員	伊神　真悟
		太田　裕介
情報システム課　豊橋分室	係　　長	宮部　浩之

4. 愛知大学　情報メディアセンター沿革・歴代所長

年度	組織		所長（任期）		システム沿革
			豊橋	名古屋	
1978					IBM製ホストコンピュータ4331　導入
1979					
1980	電子計算機センター	電子計算機センター委員会	津村　善郎 （1980. 4. 1～1982. 4. 30）		
1981					
1982			福田　治郎 （1982. 5. 1～1985. 3. 31）		
1983					
1984					
1985			高橋　正 （1985. 4. 1～1989. 3. 31）		
1986					
1987					
1988					第1期教育研究情報システム稼動　1988.4-1991.3
1989	情報処理センター	情報処理センター委員会 豊橋情報処理センター委員会 名古屋情報処理センター委員会	藤田　佳久 （1989. 4. 1～1994. 9. 30）	坂東　昌子 （1989. 4. 1～1990. 9. 30）	日立製ホストコンピュータ（HITAC M-640/20）導入
1990					
1991				浅野　俊夫 （1990. 10. 1～1992. 9. 30）	第2期教育研究情報システム稼動　1991.4-1994.3
1992					
1993				有澤　健治 （1992. 10. 1～1994. 9. 30）	
1994					第3期教育研究情報システム稼動　1994.10-1997.3 （全校舎学内LAN敷設）
1995			樋口　義治 （1994. 10. 1～1998. 9. 30）	長谷部　勝也 （1994. 10. 1～1998. 9. 30）	
1996					
1997					第4期教育研究情報システム稼動　1997.4-2000.9 （延長6ヶ月）
1998					
1999			宮沢　哲男 （1998. 10. 1～2000. 3. 31）	有澤　健治 （1998. 10. 1～2000. 9. 30）	
2000			小津　秀晴 （2000. 4. 1～2002. 9. 30）		10月　第5期教育研究情報システム稼動
2001				田川　光照 （2000. 10. 1～2002. 9. 30）	
2002			龍　昌治 （2002. 10. 1～2008. 9. 30）	坂東　昌子 （2002. 10. 1～2006. 9. 30）	
2003					
2004	情報メディアセンター	情報メディアセンター委員会 豊橋情報メディアセンター委員会 名古屋情報メディアセンター委員会			4月　第6期教育研究情報システム稼動
2005					
2006		情報メディアセンター運営会議 豊橋情報メディアセンター運営会議 名古屋情報メディアセンター運営会議			
2007				中尾　浩 （2006. 10. 1～2008. 9. 30）	
2008		ICT企画会議 豊橋ICT委員会 名古屋ICT委員会			4月　第7期教育研究情報システム稼動
2009			蒋　湧 （2008. 10. 1～2010. 9. 30）	伊藤　博文 （2008. 10. 1～2012. 9. 30）	
2010					
2011			沓掛　俊夫 （2010. 10. 1～2012. 9. 30）		
2012		ICT委員会	中尾　浩 （2012. 10. 1～2014. 9. 30）		4月　新名古屋校舎システム稼働
2013					
2014					
2015			松井　吉光 （2014. 10. 1～2018. 9. 30）		
2016					
2017					
2018			岩田　員典 （2018. 10. 1～）		
2019					
2020					
2021					

編集後記

今回のCOM46号は，論文3本と研究ノート1本を掲載し，読者の皆さんに届けることになりました。大学のコロナ感染対策が実施されてから，オンライン形式の講義が大幅に増えました。多くの学生がPCを持参しはじめ，教員から配信した電子教材による自主学習に取り込んできました。現状として，まだ多くの課題が残っているものの，学生のITスキルの向上や自主的な学習習慣の形成などの効果は，事実として多くの学生と教員が実感しています。大学教育のIT推進に関わる改善と改革は，皮肉なことに，コロナ感染対策とともに進めてきたように見えますが，今回の寄稿はいずれも教育システムの開発と管理運用に関する実践的な研究成果であり，こうした長期にわたる研究の積み重ねこそ，コロナ危機の対応に大きく貢献してきました。寄稿いただいた著者の方々には厚く御礼申し上げます。

この編集後記を執筆している間に，2022年を迎えることになりました。新年の初日に，日経新聞のトップ記事から「資本主義創り直す」との衝撃的なタイトルが目に飛び込んできました。北欧諸国と比べ，日本はフレキシキュリティに欠け，「成長の鈍化が格差を広げ，人々の不満の高まりが民主主義の土台まで揺さぶり始めた」と書かれています。フレキシキュリティ（flexicurity）とは，柔軟性のflexibilityと安全性のsecurityを組み合わせてできたものであり，雇用の柔軟性と安定性，そのバランスを重視する政策です。

記事によると，高い実質GDP成長率と実質賃金伸び率を持つ北欧諸国は，日本と違って，解雇制度が緩やかで人員削除しやすい一方，学びなおし，つまり，リスキリング（reskilling）や再就職の支援などの保障を手厚くする政策を遂行し，その結果，競争→失敗→解雇→再挑戦→成長，こうした「新陳代謝」の好循環が生み出すことで，企業イノベーション力の向上と個人収入のアップを同時に収めることができました。

教育に携わる者として，リスキリングの言葉に共鳴が湧いてきました。大学の高質な教育水準こそ，社会的な新陳代謝を生み出す1つの原動力に違いないと思います。愛知大学においては，2021年度に全学の学生に対しパソコン持参の勧告が出され，2022年度の4月からはデータサイエンス科目が開講することになりました。開講までの時間的な余裕がないため，初年度のデータサイエンスは外部の共通教材を利用することになりました。筆者の知る限りで，学内において，ロボットプログラミング，AI関連のゼミと教育，3Dプリンター実習室，ドローン勉強会，GISカリキュラムなど，斬新な知識と技術を駆使した教育活動が多数芽生えてきており，こうした学内の知恵と工夫で創出した愛知大学のデータサイエンス科目の到来を心から願っています。そのために，COMは皆さんの情報交流の場として，その役割を果たすことができます。これからも引き続き，皆さんの寄稿と閲覧をお待ちしております。

（Y.J, Jan. 2, 2022）

2018年6月1日改訂

愛知大学情報メディアセンター紀要〈COM〉　原稿募集要項

情報メディアセンター紀要〈COM〉は，下記の要領で原稿を募集しています。詳細につきましては，情報メディアセンターまでお問い合わせください。

1. 著者の資格
   （1）本学教職員および本学教職員との共著者
   （2）本学非常勤教員
   （3）本学学生（教員と共著とする。）
   （4）編集委員会が認めたもの

2. 投稿原稿の内容
   投稿原稿は未発表のもので，下記に関係する内容とする。
   （1）情報教育に関する理論と実践
   （2）情報科学や情報工学に関する理論とその応用
   （3）情報システムに関する調査，分析，理論
   （4）コンピュータを活用した研究，教育，および業務等の実践報告
   （5）本学のコンピュータ利用に関して必要と思われる情報メディアセンターの報告
   （6）その他（編集委員会が認めたもの）

3. 投稿原稿の区分
   投稿された原稿は編集委員会の審査に従って，下記のように区分して掲載する。ただし，法令等に抵触する，内容に著しい不備がある，執筆要項に従わないなどの問題があるものは，原稿の修正を依頼することや，掲載を見あわせることがある。
   （1）論文
   （2）研究ノート
   （3）情報教育実践報告
   （4）書評（新刊・古典・ソフトウェア）
   （5）学会動向
   ※原稿の体裁と見本については別紙を参照のこと。

4. 原稿の提出要領
   （1）原稿は，電子ファイルで提出すること。
   （2）完成された投稿原稿のみを受理する。
   （3）提出する電子ファイル名は，投稿原稿のタイトルとすること。
   （4）図版等がある場合は，その電子ファイルもあわせて提出すること。
   図版等のファイル形式はjpegとする。
   （5）提出ファイルは，原則Microsoft Wordまたはテキスト形式とする。

ただし，その他の形式であっても編集委員会が認めた場合は受理する。

(6) 裏表紙（目次用）として，タイトル，著者名の欧文を添えること。

(7) 著者は連絡先（ゲラ等の送付先）の住所，電話番号を申し込み先の担当者まで連絡すること。

5. 投稿原稿の体裁

投稿原稿は横書きとし，図・表などは適切な場所に分かりやすく挿入すること。なお，投稿原稿はCOM編集委員会にて共通したフォーマットに統一する。

6. 校正

(1) 校正は著者校正を2回とする。

(2) 校正段階での内容の変更は，編集作業に支障をきたさない範囲で行なうこと。

7. 著作権

(1) 提出された論文の著作権は，原則として愛知大学情報メディアセンターに属し，無断で複製あるいは転載することを禁じる。

(2) 論文作成に際して用いたコンピュータソフトや映像ソフト等の著作権に関する問題は，著者の責任において処理済みであること。他人の著作権の侵害，名誉毀損，その他の問題が生じないよう十分に配慮すること。

(3) 万一，執筆内容が第三者の著作権を侵害するなどの指摘がなされ，第三者に損害を与えた場合，著者がその責を負う。

(4) 著作人格権は著者に属する。

(5) 本誌に掲載された原稿は，学内においては，愛知大学情報メディアセンターホームページおよび愛知大学リポジトリにてデジタル公開するものとする。

(6) 本誌に掲載された原稿は，学外においては国立情報学研究所等へ登録される。

8. その他

(1) 別刷りは著者に対して希望を調査し，原則として30部以内で無料進呈する。

(2) 著者には紀要を2部進呈する。ただし希望があれば10部を限度として進呈する。

以上

申し込み・問い合わせ：愛知大学情報メディアセンター
担当：情報システム課　太田　裕介
E-mail：johosystem@ml.aichi-u.ac.jp
TEL：052-564-6117（内線20554）
FAX：052-564-6217（内線20569）

2014年8月8日改訂

愛知大学情報メディアセンター紀要〈COM〉　執筆要項

1. 執筆言語

和文もしくは英文とする。

2. 原稿

(1) 論文……和文の場合は30,000文字程度，英文の場合は15,000 words程度を上限とする。ただし，図版等の数量に応じて調節すること。

(2) 研究ノート……和文の場合は20,000文字程度，英文の場合は10,000 words程度を上限とする。ただし，図版等の数量に応じて調節すること。

(3) 情報教育実践報告……和文の場合は20,000文字程度，英文の場合は10,000 words程度を上限とする。ただし，図版等の数量に応じて調節すること。

(4) 書評（新刊・古典・ソフトウェア）……和文の場合は5,000文字程度，英文の場合は3,000 words程度を上限とする。書評（新刊・古典）には図版等を挿入することはできないが，ソフトウェアレビューについては若干の図版を添えることが出来る。

(5) 学会動向……COMのフォーマットに従う。
長文の場合は分裁や再提出等の措置を求めることがある。

3. 著者と所属

著者名と所属を記載し，著者名のあとにカッコ（　）に入れて所属を記載する。

4. セクションタイトルとセクション記号

本文中の章，節，項，目などの立て方は，原則として以下のとおりとする。

（例）

1. 章タイトル
1.1　節タイトル
1.1.1　項タイトル
(1) 目タイトル

5. 図・表・写真

図・表・写真は，本文中の適当な箇所に挿入すること。または，挿入箇所を明確にすること。

ただし，COM編集委員会にて挿入位置，サイズを変更する場合があるが，変更不可の場合は明記のこと。

(1) 表について
表の上部に「表○　表名」（○は表の一連番号）を記載すること。

(2) 図・写真について
図・写真の下部に「図○　図名」(○は図の一連番号) または「写真○　写真名」(○は写真の一連番号) を記載すること。

6. 要旨とキーワード

論文と研究ノートには要旨とキーワードをつける。要旨は400字以内(200words以内)で執筆し，本文と同じ言語でもよいし，異なった言語でもよい。キーワードは国立情報学研究所のCiNii等への正確な登録のために，5〜7語程度のキーワードをつける。

7. 謝辞

謝辞を記載する場合は，本文の最後に謝辞と小見出しを使い記載する。

8. 注

注を記載する場合は，以下のいずれかの方法による。

(1) 該当ページの下部または見開きの前後2ページ分の後のページの本文の下部に脚注として記載する。
(2) 本文の末尾に後注として一括して記載する。本文の後に1行空けてから「注」という見出しを立て，その次の行から，注を一括して記載する。

上記のいずれの場合も本文中の該当箇所には，番号と右丸括弧を使い[注1)]のように上付きで記すこと。

9. 参考文献

参考文献の記載は，本文の後（注がある場合は注の後）に1行空けてから「参考文献」という見出しを立て，その次の行から，参考文献を一括して記載すること。本文中の該当箇所には，番号と右丸括弧を使い[1)]のように上付きで記すこと。

参考文献は原則として，雑誌の場合には，著者，標題，雑誌名，巻，号，ページ，発行年を，単行本の場合には，著者，書名，ページ数，発行所，発行年を，この順に記す。引用番号の記し方は本文上に出現した順番とし，次の例を参照にされたい。

(例)

1) 山田太郎：偏微分方程式の数値解法，情報処理，Vol.1, No.1, pp.6-10 (1960).
2) Feldman, J.and Gries, D.: Translater Writing System, Comm. ACM, Vol.11, No.2, pp.77-113 (1968).
3) 大山一夫：電子計算機，p.300，情報出版，東京 (1991).
4) Wilkes, M. V: Time Sharing Computer Systems, p.200, McDonald, New York (1990).

以上

愛知大学情報メディアセンター紀要 COM〔コム〕

Vol.31 No.1 第46号

2022年３月１日 印刷
2022年３月５日 発行

編集 愛知大学情報メディアセンター

「COM」編集委員会

発行 愛知大学情報メディアセンター

（名古屋）名古屋市中村区平池町四丁目60-6
〒453-8777 TEL（052）564-6117（直通）
FAX（052）564-6217
（豊 橋）豊橋市町畑町1-1
〒441-8522 TEL（0532）47-4124（直通）
FAX（0532）47-4125
（車 道）名古屋市東区筒井二丁目10-31
〒461-8461 TEL（052）937-8120（直通）
FAX（052）937-8121

印刷 株式会社荒川印刷

情報メディアセンター教育用パソコン　機種および設置台数

○豊橋校舎

設　置　場　所		機　　種	台数
情報メディアセンター（4号館）	420教室	富士通 ESPRIMO D587/R	35
		富士通 ESPRIMO D587/S	30
	421教室	富士通 ESPRIMO D587/S	52
	423教室	富士通 ESPRIMO D587/S	60
	424教室	富士通 LIFEBOOK A574-M	40
	413教室	富士通 ESPRIMO D587/R	25
5号館	514教室	富士通 LIFEBOOK A574-M	20
	523教室	富士通 ESPRIMO D587/R	50
図書館棟1F	メディアゾーン	富士通 ESPRIMO D587/S	40
豊橋　計			352

○名古屋校舎

設　置　場　所		機　　種	台数
厚生棟4F	W401教室	富士通 ESPRIMO D587	60
	W402教室	富士通 ESPRIMO D587	60
	W403教室	富士通 ESPRIMO D587	60
	W404教室	富士通 ESPRIMO D587	60
	メディアゾーン	富士通 ESPRIMO D587	120
講義棟7F	L707教室	富士通 LIFEBOOK A576/P	80
	L708教室	富士通 LIFEBOOK A576/P	80
	L709教室	富士通 LIFEBOOK A576/P	80
	L710教室	富士通 LIFEBOOK A576/P	24
	L711教室	富士通 LIFEBOOK A576/P	24
	L712教室	富士通 LIFEBOOK A576/P	24
	L713教室	富士通 LIFEBOOK A576/P	24
名古屋　計			696

○車道校舎

設　置　場　所	機　　種	台数
K802	富士通 LIFEBOOK A574/M	35
K804	HP ProBook 4540s	50
メディアゾーン	HP Pro 4300 SFF	6
車道　計		91

Journal of Aichi University Media Center

vol.31 No.1

CONTENTS

Preface ..... Director: Kazunori Iwata

Articles

Implementation and Operation of Integrated System for Information Literacy:
Automatic Scoring Systems for Microsoft Excel and Word ..... Yoshimitsu Matsui, Kazunori Iwata 1

Construction and operation of Java programming training system with
automatic scoring function for JAR files ..... Motoaki Mouri 11

Practice Report of Information Education
Mutual Evaluation Among Students by Using the Moodle
Workshop Module ..... Kazunori Iwata 37

Notes

Survey of Moodle access in FY 2019 and FY 2020 ..... Seiji Morino 51

Miscellaneous 61

Editorial 71

Aichi University Media Center