

2021 年度修士論文
プログラミング学習者のつまずきからの復帰を
リアルタイムに支援するシステム

明星大学大学院情報学研究科情報学専攻 長研究室
20MJ-004：浦上 理

2022 年 1 月 18 日

目次

1	研究概要	1
2	研究目的	1
3	関連研究	1
4	Bit Arrow について	2
5	正解行数について	3
6	リアルタイムな検出に向けた指標	4
7	時間による学習者のつまずきの検出システム	6
7.1	システム概要	6
7.2	bot 配信システム	7
8	評価	11
8.1	DM で解決に導いた事例	12
8.1.1	DM 後に質問があった学生	12
8.1.2	DM 後に返事があった学生	15
8.1.3	DM 後に返事がなかった学生	20
8.2	未解決事例	42
9	アンケート結果	53
10	考察	55
11	まとめ	56

1 研究概要

プログラミング学習において、学習者は何らかの問題（エラーが発生したり、思い通りの実行結果が得られなかったりする場合）が発生したときに、まずは自分で試行錯誤を行った上で、それでも解決できない場合は教員や TA に質問を行うことがある。一方で、そのような質問を行わなければ問題を解決できない状況（これを「つまずき」と呼ぶ）に陥っているにもかかわらず、質問などのアクションをとらないまま、試行錯誤をつづけてしまう学習者もいる。このような学習者に対しては、教員や TA がその学習者のつまずきを検出して、サポートを行う必要がある。

これまで我々は、学習者のつまずきを正解行数 [1] というものを用いて検出する研究をしてきた。BitArrow[2] のログから最後に実行したプログラムを正解と仮定し、その行数からある時点での行数の差分より進捗度を求めるというものであった。

しかし、この方法では学習者が課題が終わっていないとつまずいているかどうかを判断することができないため、授業を行っている最中に即時につまずきを把握することができなかった。このため、つまずいていた場合に指導が遅れることとなった。

2 研究目的

本研究では、試行錯誤を行ったうえで教員や TA に助けを求めることなく、問題解決に至らなかった学生をサポートすることを目指す。そのために、学習者のつまずきを授業中リアルタイムに、かつ自動的に教員に通知することを目的とする。ただし、何らかの試行錯誤をしたうえでつまずいた学生を対象とする。これは、課題に取り組む際に課題解決のために試行錯誤をしてもらい通知を送るほうが学生自身が考えて取り組むと考えたからである。

BitArrow のログには、エラーを起こしたかどうかの情報も含まれており、これに関してはリアルタイムに検出することができるため、即時性を上げるためにエラーの情報も併用することを考えた。また、教員への通知をスムーズに行うため、学習者や教員が使い慣れている Slack を用いて通知を受け取れることが望ましいと考えた。

3 関連研究

プログラム演習のための進捗モニタリングシステム [3] では、全クラスの進捗表示ページとクラス別進捗ページを出力し、評価日時、ファイル更新学生数、評価値の度数、座席番号、学生番号、各課題の評価値とファイル変更日時を表示して進捗状況を確認していた。ここでいう評価値は、事前テストや必須テスト、コンパイル、実行テスト、補足テストがどのくらいできたかを評価し値として出す。

プログラミング入門教育を対象としたリアルタイム授業支援システム [4] では、評価項目としてコンパイル、インデント、クラスの形式的記述、ユニットテストで評価し、進行度の算出として学習者のソースコードが、コンパイル・インデント・クラスの形式的記述、ユニットテストの評価項目に対して正解している問題数を完全正解数と置き、その完全正解数が大きい学習者を上位として算出していた。

小コンテスト形式の初級 C 演習における教師支援 [5] では、複数の予備テストと 1 つの最終テストからなる実行テスト系列を採用しており、予備テストは題意の使用を緩めたもので 3 つ程度までの中間目標を構成し徐々に完成に近付けさせるものである。プログラムの判定結果には、実行テストの種別、および時間調整点と誤答減点を導入した得点ルールを適用している。そのほかに教師支援のモニタリング機能を搭載している。この機能は、提出が滞っている学生に自動通知し、色分けやポップアップなどで教師側の監視を支援していた。

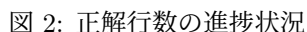
プログラミング演習における補助者の巡回指導のためのタブレット PC 上の支援ツール [6] では、補助者各自がタブレット PC を携帯し、補助者を支援するツール vRoundEd を利用しながら巡回指導を行う。vRoundEd は、受講側と補助側で 4 つの機能があり、1 つ目は、受講者の出席状況の座席表による管理。2 つ目は、受講者の挙手呼び出しと補助者の挙手応対を管理。3 つ目は、受講者と補助者の間での指導チャットとしてのチャット機能。4 つ目は、提出履歴や得点状況などを監視する進捗閲覧機能である。このツールによって巡回指導の最適化を行っていた。

プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案 [7] では、ファイルを開く、自動保存、コンパイル、実行、保存、提出、ファイルを閉じる過程を可視化したシステムであ

これらに対して、本研究では、あらかじめ正解となるプログラムや、テストケースを設定しておく必要がなく、ソースコードだけの内容だけを用いて学習状況の進捗を確認できるシステムとなっている。

2

正解行数とは、あるユーザのあるファイルについてのログを時系列を並べたときに、ある時点での実行のソースコードと、最後の実行のソースコードとの差分を見て、同一であった行数と定義する。このとき、最後の実行のソースコードは正解と仮定しており、正解行数はログの各時点でその正解にどれくらい近いのか、という指標となっている。例えば、正解行数がこれまでの最大値を上回っていなかったとき、その実行は完成に近づいていないと判断することができる。なお、課題となるファイルは Bit Arrow のファイル配布機能によって、全員同じファイル名で配布されている。また、BitArrow のログ閲覧画面で正解行数の進捗状況を確認することができる。BitArrow の正解行数の進捗状況を図 2 に示す。



正解行数の変化例

The graph illustrates the progression of the number of correct rows for three different methods over 31 iterations. The Y-axis represents the number of correct rows (0 to 14), and the X-axis represents the iteration number (1 to 31).

Method A (black squares) shows a steady increase, reaching 10 correct rows by iteration 7. Method B (red triangles) fluctuates between 3 and 6 correct rows. Method C (red diamonds) fluctuates between 3 and 12 correct rows.

Annotations highlight specific points of interest:

- 順調に進んでいる** (Progressing smoothly): Points to the upward trend of Method A.
- 間違いとは違うところを修正** (Correcting differences from errors): Points to the divergence between Method A and Method B.
- 合っているところを修正** (Correcting where they are the same): Points to the convergence of Method B and Method C.

実行回数	A (正解行数)	B (正解行数)	C (正解行数)
1	5	5	5
2	6	3	3
3	7	5	3
4	8	3	6
5	9	5	3
6	10	6	6
7	10	3	3
8	10	6	6
9	10	3	3
10	10	5	5
11	10	3	8
12	10	5	11
13	10	5	8
14	10	5	12
15	10	5	12
16	10	5	12
17	10	5	12
18	10	5	12
19	10	5	12
20	10	5	12
21	10	5	12
22	10	5	12
23	10	5	12
24	10	5	12
25	10	5	12
26	10	5	12
27	10	5	12
28	10	5	12
29	10	5	12
30	10	5	12
31	10	5	12

図 3 で，順調に課題を進めている場合は，A の学生のように順調に右肩上がりになっていると考えられる．間違いとは違ふところの修正をしている場合は，B の学生のように 1 回目の実行から 8 回目の実行まで正解行数に変化がないと考えられる．合っているところを修正している場合は，C の学生のように 2 回目の実行が 1 回目の実行から下がってその後も上がったたり下がったりしていると考えられる．という仮説を立てた．

3

解行数を用いることによって、エラーが発生していない場合でもつまずきを検出できるようになった。一方、ログの最後のプログラムを完成しているものと仮定するため、プログラムが完成しないとつまずいているかわからないという問題が残っていた。

6 リアルタイムな検出に向けた指標

学習者のつまずきのリアルタイムな検出を行うにあたり、授業の指導経験や、ログを閲覧した経験から、「課題着手から時間をかけている人はつまずいている」という仮説を立てた。この仮説が正しいか検証するためには、「時間をかけていてもつまずいていない人がある」というケースがないことを確認する必要がある。

まず、「正解行数が単調増加しているかどうかでつまずきを判定できる」という仮定のもと、「正解行数が単調増加か否か」と、「時間をかけているか否か」という観点で表1のような分類を行った。

表 1: 予備調査の仮定

ケース No.	単調増加	時間かけている	つまずいている？	サポートすべきか？
1	Yes	No	No	No
2	Yes	Yes	No	No
3	No	No	Yes	No
4	No	Yes	Yes	Yes

表1のNo.1のケースは、図3の黒いグラフのように正解行数が順調に増えて、時間もかけていないのでサポートする必要はない。No.2のケースは、時間をかけているものの、順調にプログラムを書いているということになるため、サポートをせずに進捗を見守る必要がある。No.3とNo.4は、正解行数が単調に増えない、つまり減少する箇所や平坦な箇所があり、図3の赤いグラフのように間違いとは違うところを修正している恐れがあるため、サポートをする必要がある。ただし、ある程度の試行錯誤を行わせてからサポートを行ったほうがよいので、No.3のケースでは「つまずいているが、サポートすべきではない」とみなすことができる。

正解行数のグラフをいくつか観察したところ、No.2のようなケースはほとんど見られなかったため、実働時間だけでもサポートすべきかどうかを決定することができるという仮説を立てた。そこで、No.2のケースが実際にほとんどないのかどうかを調査した。調査対象は、今年度の『プログラミング演習3』を対象としている。この講義は、Pythonを用いたオブジェクト指向を学習するもので、受講者は1年次にPythonを用いた入門教育を受けている。演習にはBit Arrowを用い、ログを収集している。ログから正解行数をグラフ化した。グラフが単調増加していて、かつ、課題を完了させるまでの所要時間が15分(900秒)以上であるログを100件ほど調査したものを表2に示す。

ここで、「増加」「平坦」「減少」とは、時系列で隣接する2つのログの組について、正解行数が「時間的に後のほうが値が大きいもの」「等しいもの」「後のほうが値が小さいもの」の数である。これらのログを(増加－減少)/(増加＋平坦＋減少)の値が大きい順に並べ替え、No.を付番している。また、「つまずきの有無」については、ログを目視でみて筆者が判断したものである。なお、課題の対象となっているファイル名とは異なったものや指定された課題と異なるプログラムが書かれているもの、課題が完成していないものを除外している。

また、所要時間の測定については実働時間(actualTime)を使用している。これは、隣り合うログの時刻が5分以上離れている場合、その部分の所要時間をちょうど5分とみなして計測するものであり、他の作業をしている場合や日をまたいで作業している場合に所要時間が過度に長くなるのを防ぐためである。

表2より正解行数が単調増加した学生を一部紹介する。

No.1

No.1の学生の正解行数の変化を図4に示す。

図4の学生は、グラフが単調増加で順調に課題を進めていた。

No.14

No.14の学生の正解行数の変化を図5に示す。

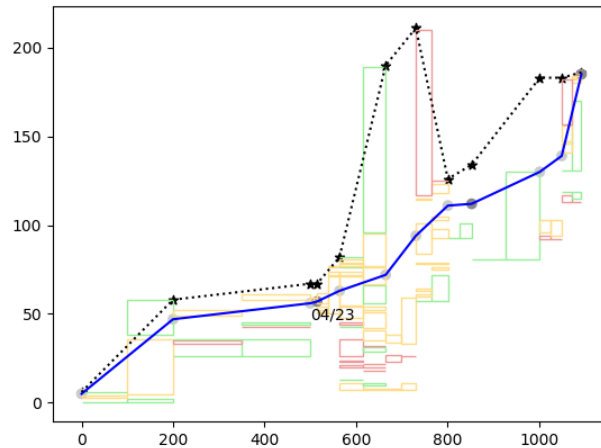


図 4: 単調増加でつまづいていない学生

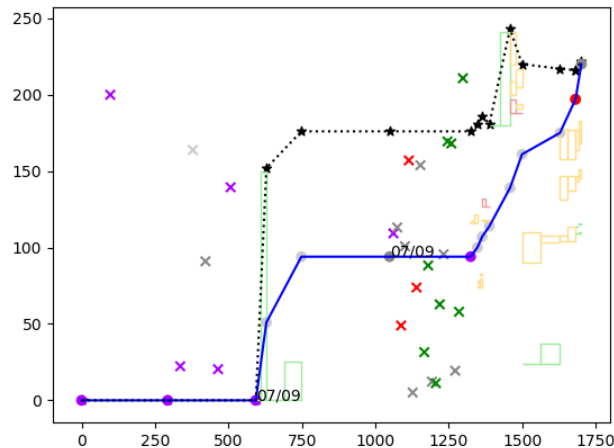


図 5: 単調増加で時間がかかっているが、つまづいていない学生

図 5 のケースは、時間はかかっているが、実際にはつまづいていない学生である。初めグラフが平坦な部分があるが、これは最初のほうはファイルを開いているだけで別の課題に取り組んだので取り組み時間がかかっているようになった。

No.36

No.36 の学生の正解行数の変化を図 6 に示す。

図 6 のケースは、エラーを 2 回出ただけで、文字の部分でダブルクォーテーションで囲っていなかったためエラーが出ただけであった。

なお、参考までに、単調増加していないグラフ（平坦な部分が多い、増減を繰り返している）の例についても 2 件示す。

No.970

図 7 の学生は、最初に入力部分の書き間違いを行ってしまっており、それに気づかずに、原因と無関係なところで試行錯誤を行ってしまっている。最初の 500 秒あたりは他の作業をやっていたとして、その分を引いても 2000 秒近く作業をしていたことになり、サポートが必要になると考えられる。正解行数のグラフ上では平坦な部分が目立つ。

No.1810

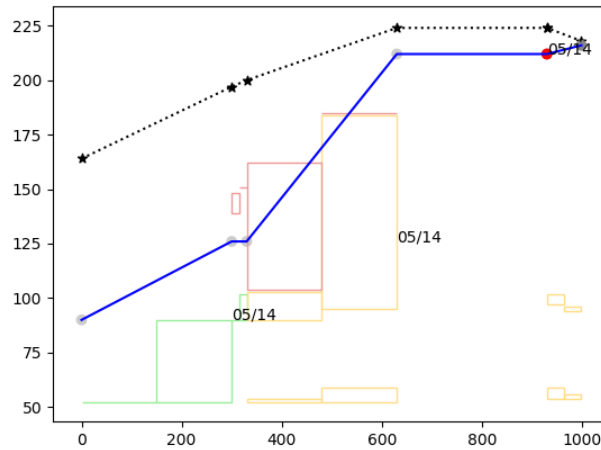


図 6: 単調増加で、つまづいている学生

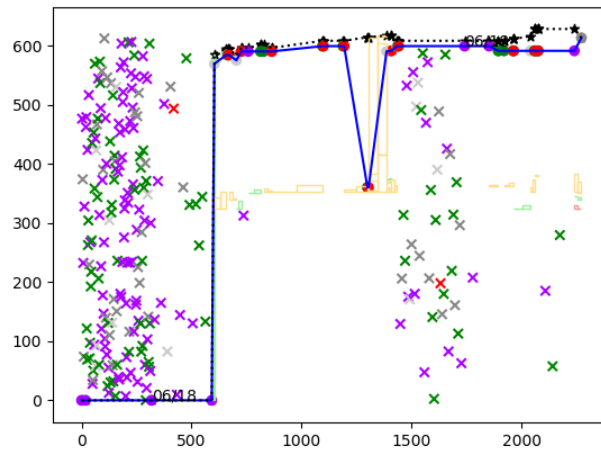


図 7: 時間がかかっている、かつ、サポートが必要な学生 (1)

図 8 の学生は同じところを何回も修正していた。if 文の条件式に苦戦しており、このような場合は正解行数が増減を繰り返しているグラフになる。

これらの予備調査から、正解行数のグラフが単調増加の場合、つまり実際につまづいていない場合には時間をそれほどかけていないことがわかる。1つの課題に時間をかなりかけているケースではつまづいていて、それほど時間をかけていない学生はつまづいていないということが分かった。

7 時間による学習者のつまづきの検出システム

前章の結果を踏まえて、課題着手からの経過時間にもとづき、学習者のつまづきを検出するシステムを開発した。

7.1 システム概要

今回の研究では、エラーが起きている状況であり、かつ時間をかけている学生を見つけ、Bit Arrow で出たエラーを授業時間内に判断して教員や TA に通知する仕組みを設けた。この通知に含まれている学生に対してはダ

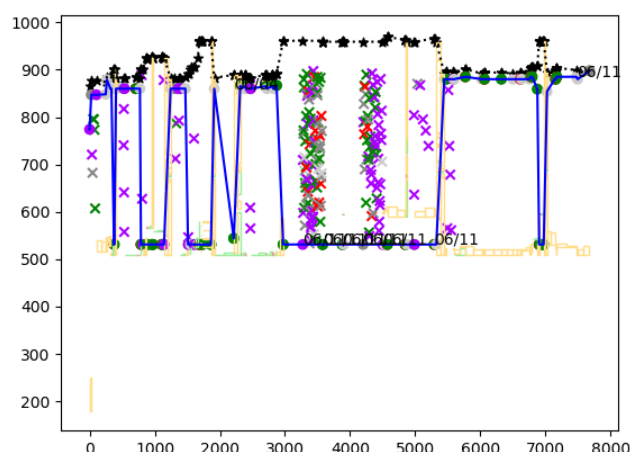


図 8: 時間がかかっている、かつ、サポートが必要な学生 (2)

イレクトメッセージ（以下「DM」と称する）でアドバイスを自動でする。ただし、今回の実装はプロトタイプであるため、通知されたエラー内容を教員や TA が確認し、手動で DM を送ったり、全体に向けて指導をしたりすることを支援するシステムとして実装している。

本研究で使用する『Bit Arrow』は、ログ収集機能が付いており、ユーザ ID、実行時刻、実行時のプログラム、実行結果（エラーの場合はエラーメッセージ）、ファイル名の情報が入っている。これらのログは、実行のたびに収集している。

Bit Arrow のログから流れるエラーのうちエラーを解消できていないものを対象とし、Slack を使用し 5 分間隔で配信を行う。配信されたものの中から目視でつまづいている学習者を取り出す。配信内容には、次の情報が含まれている。

- エラーを検出したときのソースコードが載っている URL
- 学籍番号
- 直近の 5 分間で起きたエラー件数
- ファイル名
- エラーメッセージ (100 文字)
- 課題取り組み時間（実働時間）
- 問題に対するヒント（※）

※この機能に関しては学部生が開発したものを取り入れている。クラスとユーザ名、課題番号を参照して該当の課題に対するヒントを表示するシステムである。[11]

7.2 bot 配信システム

実際に配信されるものを図 9 に示す。

bot 配信システムのソースコードをソースコード 1 に示す。

ソースコード 1: bot 配信システムのソースコード

```
1 <?php
2 function bot() {
```

URL

エラー件数

ファイル名

エラー : [test <http://cho.is.meisei-u.ac.jp/bitarrow/?TeacherLog/view1new&logid=5123965> 20mj004 (1 件) week9/p07write.php

 Notice: Undefined property: stdClass::\$displayname in /var/www/html/20mj004/week9/p07write.php
 Actualtime2= 2277 取り組み時間
 20mj004 ユーザ名 (学籍番号)

week9/p07write.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。
 /home/webpro21/20mj004/week9/p07write.phpに対するヒント
 week9/p07write.php のヒント:
 追加の書き込みをしたいなら、
 week7/p07write.php を参照
 表示名が一致する2つのデータの内容を出力したいなら、
 week9/p06join.php を参照

がんばれ
]

ヒント

エラーメッセージ

図 9: Slack に配信されるもの

```

3   date_default_timezone_set('Asia/Tokyo');
4   $class=Auth::curClass2();
5   if (!$class->getOption("showOtherStudentsLogs")) {
6       Auth::assertTeacher();
7   }
8   req("LogFileToDBCController");
9   LogFileToDBCController::run();
10  $thisURL="a.php?TeacherLog/bot";
11  $now=time();
12  $interval=param('interval',300);
13  if(!param("Y",false)){
14      $min=$now-$interval;
15      $max=$now;
16  }else{
17      $max=strtotime(param('Y')."/".param('m')."/".param('d')." ".param('H').":".param('i').":".param('s'));
18      $min=$max-$interval;
19  }
20  $next=$max+$interval;
21  echo param("Y",false);
22  echo param("s",false);
23  ?>
24  <script>
25  // $interval 秒後にリロードされて、$interval 秒進んだあとの結果が表示される
26  setTimeout( function () {
27      document.forms.dateform.submit();
28  },<?=$interval*1000 ?>);
29  </script>
30  <form name="dateform" action="<?=$_$thisURL$_?" method="POST">
31  <input name="Y" value="<?=date("Y",$next)?>" maxlength="4" size="4">年
32  <input name="m" value="<?=date("m",$next)?>" maxlength="2" size="2">月
33  <input name="d" value="<?=date("d",$next)?>" maxlength="2" size="2">日
34  <input name="H" value="<?=date("H",$next)?>" maxlength="2" size="2">時
35  <input name="i" value="<?=date("i",$next)?>" maxlength="2" size="2">分
36  <input name="s" value="<?=date("s",$next)?>" maxlength="2" size="2">秒<br>
37  <input type="submit" value="Bot で送る"/>
38  </form><?php
39  $logs=$class->getAllLogs($min,$max);
40  $slack_bot_url=$class->getOption("botURL");
41  print "BOT_URL=$slack_bot_url<BR>\n";
42  $errorLogs=[]; //エラーログの取得
43  $prevTime=0;
44  $prevResult=0;
45  $errorSolved=[]; //エラーが解決したかどうか。

```

```

46 // ログの中からエラーのものを抽出する
47 foreach($logs as $log){
48     $raw=json_decode($log["raw"]);
49     $id=$log["id"];
50     $time=$log["time"];
51     $nEtime="";
52     $result=($log["result"]);
53     $name=($log["user"]);
54     $filename=($log["filename"]);
55     $nEcode="";
56     $code=LogUtil::getCode($raw);
57     $detail=json_decode($log["detail"]);
58     if(strpos($result,'Error') !== false){
59         $mesg="";
60         if ($detail && isset($detail->message)) {
61             $mesg=$detail->message;
62         }
63         if ($detail && isset($detail->pos)) {
64             $pos=$detail->pos;
65             $code=substr($code, 0, $pos).":exclamation:".substr($code,$pos);
66         }
67         if (!($log['time']-$prevTime<=1 &&
68             $log['result']==$prevResult &&
69             strpos(mb_strtolower($log['result'],'runtime')!==false)) {
70             $errorLogs[]=array("user"=>$name, "mesg"=>$mesg, "code"=>$code, "time"=>
71                 $time,"filename"=>$filename,"id"=>$id);
72             $prevTime=$log['time'];
73             $prevResult=$log['result'];
74         } else if(strpos($result,'Unsaved') !== false) {
75         } else if(strpos($result,'Build')!==false) {
76         } else if(strpos($result,'Save')!==false) {
77         } else {// エラーがない場合
78             // errorSolvedには各ユーザの「エラーのなかったログ」のうち,
79             // もっとも時刻の新しいものが入る.
80             $errorSolved[$name]=$time;
81         }
82     }
83 //エラーのあったログをユーザごとに別の配列に分ける.
84 /*$errorLogs=[
85     ["user"=>"a", "mesg"=>"Error1"],
86     ["user"=>"a", "mesg"=>"Error2"],
87     ["user"=>"b", "mesg"=>"Error3"],
88     ["user"=>"a", "mesg"=>"Error4"],
89     ["user"=>"b", "mesg"=>"Error5"],
90     ["user"=>"a", "mesg"=>"Error6"],
91 ];
92 */
93 // $stat:
94 // ["a"=> ["Error1","Error2","Error4","Error6"], "b"=>["Error3","Error5"]]
95 $stat=[];
96 foreach ($errorLogs as $log) {
97     $user=$log["user"];
98     $mesg=$log["mesg"];
99     $time=$log["time"];
100     $filename=$log["filename"];
101     $code="";
102     $id=$log["id"];
103
104     if (!isset($stat[$user])) {
105         $stat[$user]=[];
106     }
107     $stat[$user][]=$log;
108 }
109 print_r("-----");
110 uasort($stat, function ($a, $b) { return count($a)-count($b); } );

```

```

111 $buffer="エラー：";
112 foreach ($stat as $s) {
113     print_r("-----");
114     $count=count($s);
115     print_r($s[$count-1]);
116     $mesg=$s[$count-1]["mesg"];
117     $name=$s[$count-1]["user"];
118     $code=$s[$count-1]["code"];
119     $filename=$s[$count-1]["filename"];
120     $id=$s[$count-1]["id"];
121     $lastErrorTime=$s[$count-1]["time"];
122
123     $solved="未解決";
124     // 「エラーが解決した最後の時間」 >= 「最後のエラーが起きた時間」なら解決
125     if (isset($errorSolved[$name]) && $errorSolved[$name]>=$lastErrorTime) {
126         $solved="解決済み";
127     }else{
128         $URL=BA_TOP_URL."?TeacherLog/view1new&logid=$id";
129         $option=[
130             'http' => [
131                 'method' => 'GET',
132                 'timeout' => 600, // タイムアウト時間
133             ]
134         ];
135         if (defined("HINT_URL")){
136             // エラーを起こしたファイルに対するヒントを取得
137             // (学部生の卒業研究で開発したシステムを利用)
138             $Hint=@file_get_contents(HINT_URL."?path=/home/$class->id/$name/$filename",false,stream_context_create($option));
139         } else {
140             $Hint="";
141         }
142
143         $Hint=str_replace('<h1>','',$Hint);
144         $Hint=str_replace('</h1>','',$Hint);
145         $Hint=str_replace('<br/>','',$Hint);
146         $Hint=str_replace('<BR/>','',$Hint);
147         $Hint=str_replace('メインページ','',$Hint);
148         $mesg=substr($mesg,0,100);
149         $user=$class->getUser($name);
150         // 実働時間を取得
151         $a2=self::getActualtime2($user,$filename);
152         $element="test_{$URL}_{$name}_{$count}件_{$filename}_{$mesg}_\n_{$a2}_\n_{$name}_\n_{$filename}_で苦戦中のようなので,ヒントを貼っておきます.それでもわからないなら遠慮なく質問してください._\n_{$Hint}";
153
154         $buffer.="[".$element."]\n";
155     }
156 }
157 if($solved=="未解決"){
158     $data = ['payload' => json_encode( ["text"=>$buffer])];
159     $context = [
160         'http' => [
161             'method' => 'POST',
162             'header' => implode("\r\n", array('Content-Type:_application/x-www-form-urlencoded',)),
163             'content' => http_build_query($data)
164         ]
165     ];
166     $html = file_get_contents($slack_bot_url, false, stream_context_create($context));
167     var_dump($http_response_header);
168     echo $html;
169 }
170 }

```

ソースコード 1 は、bitArrow に実装した bot 機能の部分である。bitArrow から出たエラーを抽出し、5 分間でエラーが未解決だったものを 5 分間隔で Slack に送信するように仕掛けている。

実際に bot を動かしている画面を図 10 に示す。

```
202117
2021 年 12 月 14 日 12 時 00 分 17 秒
Botで送る

BOT URL=https://hooks.slack.com/services/T01RZHT9RQ/B01S02S92JX/tBvpprOZUzoX2QLtEx5OSys
-----Array ( [user] => [mesg] =>
Warning: Use of undefined constant type - assumed 'type' (this will throw an Error in a future version of PHP) in /var/www/html/[redacted]/week12/p07fight.php on line 77

Warning: Use of undefined constant normal - assumed 'normal' (this will throw an Error in a future version of PHP) in /var/www/html/[redacted]/week12/p07fight.php on line 77

Warning: Use of undefined constant type - assumed 'type' (this will throw an Error in a future version of PHP) in /var/www/html/[redacted]/week12/p07fight.php on line 82

Warning: Use of undefined constant skill - assumed 'skill' (this will throw an Error in a future version of PHP) in /var/www/html/[redacted]/week12/p07fight.php on line 82
対戦相手に
のダメージ
対戦相手のHP :22
次へ [code] =>
```

出場させるキャラクターを選択

```
name ?> 選択
?", $userName); ?>
```

対戦相手を選択

```
name ?> 選択
hp; $results=pdo_select("select * from chara where id=?", $enemy_chara_id); $enemy_chara=$results[0]; $_SESSION["enemy_hp"]=$enemy_chara->hp; ?>
```

図 10: bot を動かしている画面

図 10 は、日時・時刻を設定し、Web ブラウザの画面を開いている状態にすると 5 分間隔で自動再読み込みを行い、エラーを含むログから、未解決のものを Slack のほうに送信している。

8 評価

2021 年度に明星大学情報学部 of 2 年次配当で開講された『ウェブプログラミング』の講義を対象とし、現在評価実験を行っている。この講義は、PHP を用いた講義である。この講義には 55 名の学生がいる。情報学部の授業としては PHP を使うのはこの講義が初めてである。

また、この講義は対面とリモートの併用で行っており、連絡・質問などに関しては、Slack を使用している。ウェブプログラミングの講義内容を表 3 に示す。

この講義でははじめは PHP の基礎を学習してその後 SQL を学習したのちに学習した内容を活用して戦闘ゲームを作成を行う形で進められている。

この講義のサポート体制として bot から配信されたエラーの中から、特にサポートが必要と思われる事例を教員や TA が抽出して、bot が配信したチャンネルとは別のチャンネル（つまり管理チャンネル）に転送する。つまり管理チャンネルに転送された項目については、つまりが解消できるまでの推移をスレッド機能で管理する。スレッド機能で管理しているイメージ図を図 11 に示す。

実働時間が 900 秒以上でつまりが解消できない場合は、実際につまづいている学生に DM を送り、修正して正解につなげていくという流れである。サポート体制の流れ図を図 12 に示す。

図 12 よりログからエラーを抽出して bot に送るまでの過程は自動で行うが、bot からつまり管理チャンネルの転送、DM 送信は、教員と TA が行う。

2022/01/11 迄の通知件数：329 件、実際に DM を送った件数：95 件、解決件数：47 件、解決できなかった件数：48 件、解決したのに書き換えてしまった件数：0 件であった。

つまりいた受講者の課題取り組み時間を表 4 に示す。

凡例 S:SA に質問あり、H:返答あり、T:追加質問あり、E:エラー文とソースコードを添付、Q:学生側から質問あり

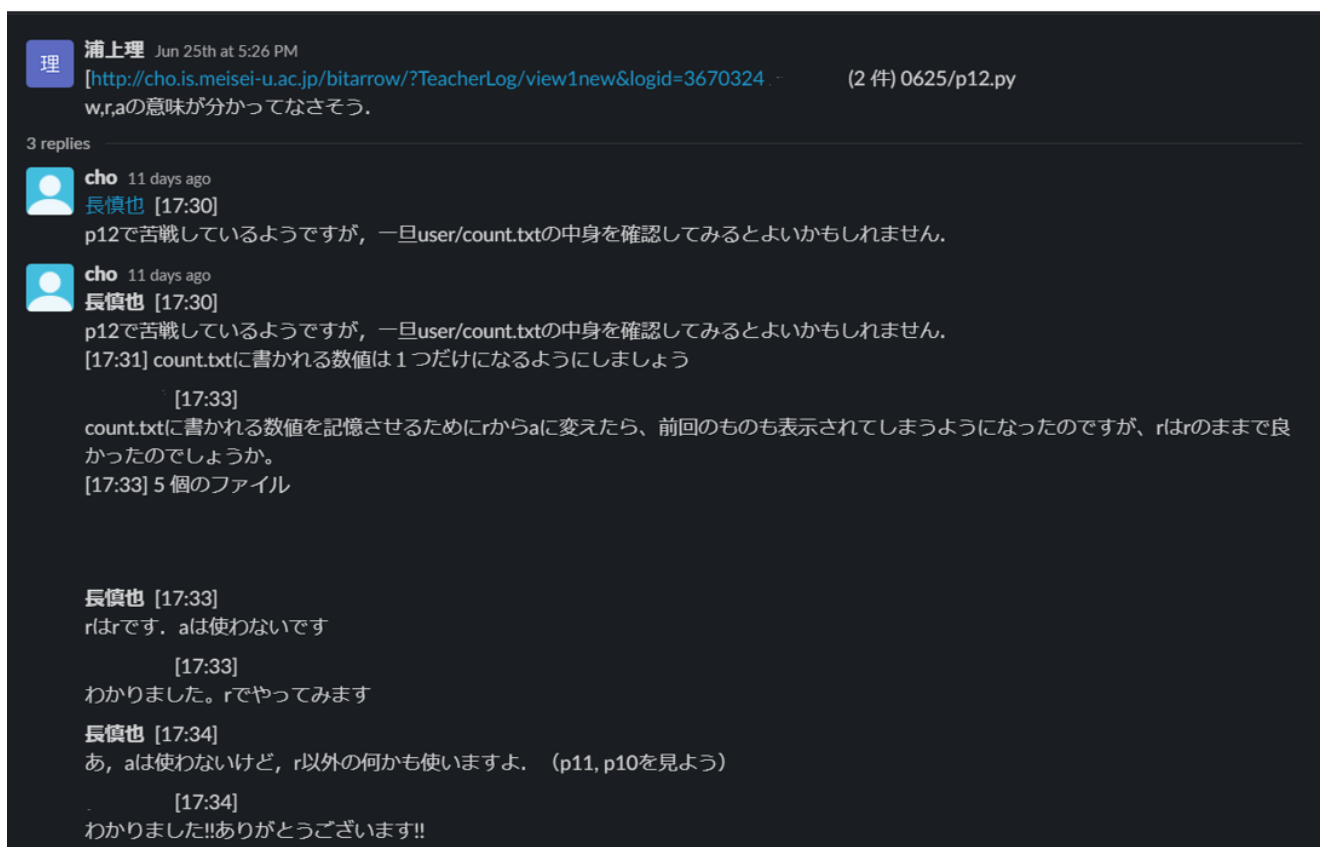


図 11: スレッド機能で管理しているイメージ図

8.1 DM で解決に導いた事例

表 4 より、DM の送信をきっかけに解決に導いた学生の事例を紹介する。

8.1.1 DM 後に質問があった学生

事例 9

week6/p08.php でつまづいていた学生である。week6/p08.php は、name の十の位が 1 のものを表示するプログラムである。bot 送信時のソースコードをソースコード 2 に示す。

ソースコード 2: 事例 9 の week6/p08.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3
4
5 $records=pdo_select("select_*_from_tweet_where_name=preg_match('.....1.', $name)");
6 foreach($records as $record) {
7     print_r($record);
8     print "<BR>\n";
9 }
10
11 ?>

```

ソースコード 2 は、bot 送信時の week6/p08.php である。この学生は、SQL で preg_match を使用しようとしてエラーが出ている。

この学生には、以下のような DM を送信した。学籍番号は、GAKUSEKI と置換している。

week6/p08.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。/home/webpro21/GAKUSEKI/week6/p08.php に対するヒント week6/p08.php

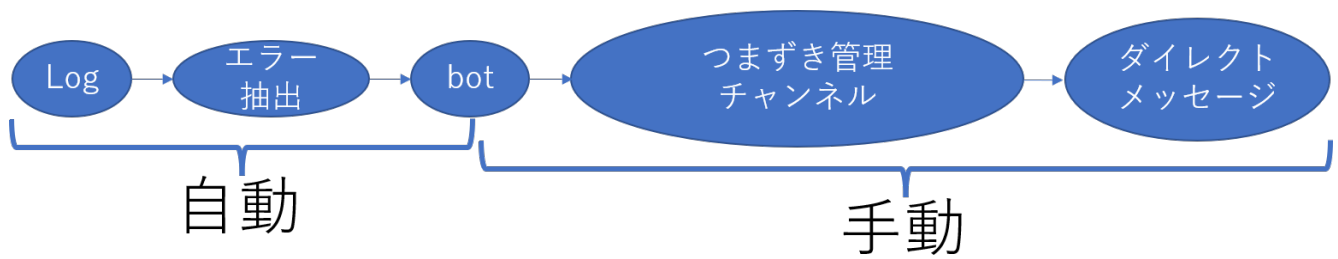


図 12: サポート体制の流れ図

のヒント: 指定した名前からの投稿をすべて出力したいなら、week6/p05.php を参照
指定した文字が含まれる内容の投稿をすべて出力したいなら、week6/p07.php を参照

学生の方から質問が送られた。

十の位が1という条件を指定する方法がわかりません。正規表現かワイルドカードを使うのかなとは思っているのですが、name= ' ' の中に正規表現を用いてもうまく出力されず詰まってしまう。

教員がヒントを送信

p06 の応用です。(↑のヒントから抜けてるけど、p07 ができていればそちらも参考になります)

学生の方からできた旨の連絡があった。

その学生の完成したソースコードをソースコード 3 に示す。

ソースコード 3: 事例 9 の week6/p08.php (完成時)

```

1 <?php
2 require("pdo.php");
3 $records=pdo_select("select * from tweet where name like '19j581%'");
4 foreach($records as $record) {
5     print_r($record);
6     print "<BR>\n";
7 }
8 ?>
  
```

3 行目が変化している。

事例 10

week6/p14.php でつまづいていた学生である。week6/p14.php は、content に「しょっぱかった」が含まれるものを、time が新しい順に 3 件表示するプログラムである。bot 送信時のソースコードをソースコード 4 に示す。

ソースコード 4: 事例 10 の week6/p14.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 $records=pdo_select("select * from tweet where content like '%しょっぱかった%' and order by time desc limit 3");
4 # $records=pdo_select("select * from tweet order by time desc limit 3");
5 foreach($records as $record) {
6     print_r($record);
7     print "<BR>\n";
8 }
9 ?>
  
```

ソースコード 4 は、bot 送信時の week6/p14.php である。3 行目でエラーを起こしている。この学生には、以下のような DM を送信した。学籍番号は、GAKUSEKI と置換している。

week6/p14.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。 /home/webpro21/GAKUSEKI/week6/p14.php に対するヒント week6/p14.php のヒント: 指定した文字が含まれる内容の投稿をすべて出力したいなら、week6/p05.php を参照新しい順に指定した数だけ投稿を出力したいなら、week6/p13.php を参照

学生の方から質問が送られた。

しよっぱかったと limit3 を 1 行で書けますか？

その後教員の方より

まず「しよっぱかった」だけ（件数制限なし）だとどうなりますか？

学生はソースコードを送信

```
$records=pdo_select("select * from tweet where content like '%しよっぱかった%' and order by time desc limit 3");
```

教員が指摘

あ、and が余計ですね

学生の方からできた旨の連絡があった。

その学生の完成したソースコードをソースコード 5 に示す。

ソースコード 5: 事例 10 の week6/p14.php (完成時)

```
1 <?php
2 require("pdo.php");
3 $records=pdo_select("select * from tweet where content like '%しよっぱかった%' order by
    time desc limit 3");
4 #$records=pdo_select("select * from tweet order by time desc limit 3");
5 #and order by time desc limit 3
6 foreach($records as $record) {
7     print_r($record);
8     print "<BR>\n";
9 }
10 ?>
```

3 行目が変化している。

事例 19

week10/p04main.php でつまづいていた学生である。10 週目は、対戦ゲームのログインページとメインページの作成を行うことを目標に授業が行われた。week10/p04main.php は、ログインしているユーザのユーザ名、表示名、所持金が表示されるプログラムである。bot 送信時のソースコードをソースコード 6 に示す。学籍番号の部分は、GAKUSEKI に置換している。

ソースコード 6: 事例 19 の week10/p04main.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4
5 class Main {
6     static function index() {
7         $name="GAKUSEKI";//ここは自分の番号にする
8         pdo_exec("update user set gold=1000 where name=?", $name);
9         $results=pdo_select("select * from user where name=?", $name);
10        foreach ($results as $result) {
11            ?>
12            <h1>メインページ</h1>
13            ユーザ名: <?= $name ?> <br>
14            表示名: <?= $displayname ?> <br>
15            所持金: <?= $gold ?>
```



```

16         <?php
17     }
18 }
19 }
20 run_controller("Main");
21 ?>

```

ソースコード 6 は、bot 送信時の week10/p04main.php である。13～15 行目がデータベースからの呼び出しができていないことからエラーが起きている。

この学生には、以下のような DM を送信した。学籍番号は、GAKUSEKI と置換している。

week10/p04main.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。/home/webpro21/GAKUSEKI/week10/p04main.php に対するヒント
week10/p04main.php のヒント: 変数のまとまりから特定の変数を出力したいなら、week3/p03.php を参照

学生の方から質問が送られた。

データベースから表示名と所持金を呼び出す操作がわかりません

教員がヒントを送信

p01 の select.... と書いてある以降がヒントです

その後の返信はなかったが、ログを閲覧すると少しずつ修正している様子が見られた。その学生の完成したソースコードをソースコード 7 に示す。学籍番号は、GAKUSEKI と置換している。

ソースコード 7: 事例 19 の week10/p04main.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4
5  class Main {
6      static function index() {
7          session_start();
8          $name="GAKUSEKI"; //ここは自分の番号にする
9          $displayname= $_SESSION["displayname"];
10         $records=pdo_select("select * from user where name=?", $name);
11         ?>
12         <h1>メインページ</h1>
13         <?php
14         foreach ($records as $record) {
15             print("ユーザ名: $record->name<br/>");
16             print("表示名: $record->displayname<br/>");
17             print("所持金: $record->gold");
18         }
19     }
20 }
21 run_controller("Main");
22 ?>

```

15～17 行目に変化している。

8.1.2 DM 後に返事があった学生

事例 2

week2/p12.php でつまづいていた学生である。week2/p12.php は、配列の各要素について奇数が赤字、偶数が青字で表示させるプログラムである。bot 送信時のソースコードをソースコード 8 に示す。

ソースコード 8: 事例 2 の week2/p12.php (bot 送信時)

```

1 <html>
2 <meta charset="UTF-8">
3 <body>
4 <?php
5 $array=[12,3,5,70,9,13,8,11,4,2];
6 foreach ($array as $n) {
7     if($n/2 == 0){
8     }
9     ?>
10     <?= $n ?>
11 }
12 </body>
13 </html>

```

ソースコード 8 は, bot 送信時の week2/p12.php である. if 文の分岐ができていない.
この学生には, 以下のような DM を送信した. 学籍番号は, GAKUSEKI と置換している.

p12 を苦戦中のようなので, ヒントを貼っておきます. それでもわからないなら遠慮なく質問してください. week2/p12.php のヒント: 2 の倍数の判定をしたいなら, week1/p20.php を参照配列の要素を出力するには, week2/p11.php を参照

その後以下のような返信があった.

ありがとうございます! 何とか出来ました

その学生の完成したソースコードをソースコード 9 に示す.

ソースコード 9: 事例 2 の week2/p12.php (完成時)

```

1 <html>
2 <meta charset="UTF-8">
3 <body>
4 <?php
5 $array=[12,3,5,70,9,13,8,11,4,2];
6 foreach ($array as $n) {
7     if($n%2==0){
8         print("<font_color=\"blue\">". $n);
9     } else {
10         print("<font_color=\"red\">". $n);
11     }
12 }
13 ?>
14 </body>
15 </html>

```

if 文の分岐と色指定をし, つまづきを解消していた.

事例 14

week5/p04Exam.php でつまづいていた学生である. week5/p04Exam.php は, この課題は, 名前と数学, 英語の点数を入力し, 表示させるプログラムである. bot 送信時のソースコードをソースコード 10 に示す.

ソースコード 10: 事例 14 の week5/p04Exam.php (bot 送信時)

```

1 <?php
2 class Exam {
3     static function index() {
4         ?>
5         <h1>名前入力画面</h1>
6         <form action="p04Exam.php">
7             <input type="hidden" name="action" value="show_math_form"/>
8             名前: <input name="name"/>
9             <input type="submit" value="Next"/>
10        </form>
11    <?php

```

```

12     }
13     static function show_math_form() {
14         session_start();
15         $name=$_GET["name"];
16         $_SESSION["name"]=$name;
17         ?>
18         <h1>成績入力</h1>
19         名前: <?= $name ?>
20         <form action="p04Exam.php">
21         <input type="hidden" name="action" value="show_english_from"/>
22         数学の成績: <input name="math"/>
23         <input type="submit" value="Next"/>
24         </form>
25         <?php
26     }
27     static function show_english_form() {
28         session_start();
29         // $name=$_GET["name"];
30         $_SESSION["name"]=$name;
31         $math=$_GET["math"];
32         $_SESSION["math"]=$math;
33         ?>
34         <h1>成績入力</h1>
35         名前: <?= $name ?>
36         <form action="p04Exam.php">
37         <input type="hidden" name="action" value="show_result"/>
38         英語の成績: <input name="english"/>
39         <input type="submit" value="Next"/>
40         </form>
41         <?php
42     }
43     static function show_result() {
44         session_start();
45         $name=$_SESSION["name"];
46         $math=$_SESSION["math"];
47         $english=$_GET["english"];
48         ?>
49         <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?=$english ?>です。
50         <?php
51     }
52 }
53 require "controller.php";
54 run_controller("Exam");
55 ?>

```

ソースコード 10 は、bot 送信時の week5/p04Exam.php である。21 行目の value が show_english_from にしていたことによるエラーであった。

この学生には、以下のような DM を送信した。学籍番号は、GAKUSEKI と置換している。

week5/p04Exam.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。/home/webpro21/GAKUSEKI/week5/p04Exam.php に対するヒント
week5/p04Exam.php のヒント: 入力した値を変数に代入したいなら、week1/p11.php を参照入力したものを複数のメソッドで使用したいなら、week5/p03Exam.php を参照

その後以下のような返信があった。

今日のスレッドのおかげで無事プログラムが完成しました。ありがとうございます！

今日のスレッドは、10/12 に配信されたヒントのことだと考えられる。配信したヒントは以下のようなものであった。

- show_math_form では、\$_GET["name"] は【使える・使えない】
- show_english_form では、\$_GET["name"] は【使える・使えない】
- show_result では、\$_GET["name"] は【使える・使えない】

- show_english_form では、\$_GET["math"] は【使える・使えない】
 - show_result では、\$_GET["math"] は【使える・使えない】
- 使えないところでは代わりに何をつかうか？（これはヒントなので返信しなくていいです）

このヒントを基に修正したと考えられる。

その学生の完成したソースコードをソースコード 11 に示す。

ソースコード 11: 事例 14 の week5/p04Exam.php（完成時）

```

1  <?php
2  class Exam {
3      static function index() {
4          ?>
5          <h1>名前入力画面</h1>
6          <form action="p04Exam.php">
7              <input type="hidden" name="action" value="show_math_form"/>
8              名前: <input name="name"/>
9              <input type="submit" value="Next"/>
10             </form>
11         <?php
12     }
13     static function show_math_form() {
14         session_start();
15         $name=$_GET["name"];
16         $_SESSION["name"]=$name;
17         ?>
18         <h1>成績入力</h1>
19         名前: <?= $name ?>
20         <form action="p04Exam.php">
21             <input type="hidden" name="action" value="show_english_from"/>
22             数学の成績: <input name="math"/>
23             <input type="submit" value="Next"/>
24             </form>
25         <?php
26     }
27     static function show_english_from() {
28         session_start();
29         // $name=$_GET["name"];
30         // $_SESSION["name"]=$name;
31         $math=$_GET["math"];
32         $_SESSION["math"]=$math;
33         ?>
34         <h1>成績入力</h1>
35         名前: <?= $_SESSION["name"] ?>
36         <form action="p04Exam.php">
37             <input type="hidden" name="action" value="show_result"/>
38             英語の成績: <input name="english"/>
39             <input type="submit" value="Next"/>
40             </form>
41         <?php
42     }
43     static function show_result() {
44         session_start();
45         $name=$_SESSION["name"];
46         $math=$_SESSION["math"];
47         $english=$_GET["english"];
48         ?>
49         <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?=$english ?>です。
50         <?php
51     }
52 }
53 require "controller.php";
54 run_controller("Exam");
55 ?>

```

21 行目が修正されている。

事例 16

第 8.1.1 節の事例 19 と同様 week10/p04main.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 12 に示す。

ソースコード 12: 事例 16 の week10/p04main.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4
5 class Main {
6     static function index() {
7         session_start();
8         $name=$_SESSION["name"];
9         $displayname=$_SESSION["displayname"];
10        $gold=$_SESSION["gold"];
11        $this->name=$name;
12        $this->displayname=$displayname;
13        $this->gold=$gold;
14        ?>
15        <h1>メインページ</h1>
16        ユーザ名 : <?= $this->name ?><BR> 表示名 : <?= $this->displayname ?><BR> 所持金 :
17        <?= $this->displayname ?><BR>
18    }
19 }
20 run_controller("Main");
21 ?>
```

ソースコード 12 は、bot 送信時の week10/p04main.php である。表示名が格納されている displayname は、データベースから取り出さなければならないが、その作業をしていない。セッション変数から取り出そうとしてエラーが出ている。

この学生には、以下のような DM を送信した。学籍番号は、GAKUSEKI と置換している。

week10/p04main.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。/home/webpro21/GAKUSEKI/week10/p04main.php に対するヒント week10/p04main.php のヒント: 変数のまとまりから特定の変数を出力したいなら、week3/p03.php を参照

その後以下のような返信があった。

今やっと完成しました！ありがとうございます！

その学生の完成したソースコードをソースコード 13 に示す。

ソースコード 13: 事例 16 の week10/p04main.php (完成時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4
5 class Main {
6     static function index() {
7         session_start();
8         $name=$_SESSION["name"];
9         $results=pdo_select("select_*_from_user_where_name=?", $name);
10        foreach ($results as $result) {
11            ?>
12            <h1>メインページ</h1>
13            ユーザ名: <?= $result->name ?><BR>表示名: <?= $result->displayname ?><BR>所持
14            金: <?= $result->gold ?><BR>
15        }
16    }
17 }
```

```

16     }
17 }
18 run_controller("Main");
19 ?>

```

13 行目にデータベースより displayname を取り出す記述が加えられた。

8.1.3 DM 後に返事がなかった学生

事例 5

第 8.1.2 節の事例 14 と同様 week5/p04Exam.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 14 に示す。

ソースコード 14: 事例 5 の week5/p04Exam.php (bot 送信時)

```

1  <?php
2  class Exam {
3      static function index() {
4          ?>
5          <h1>名前入力画面</h1>
6          <form action="p03Exam.php">
7              <input type="hidden" name1="action" value="show_math_form"/>
8              名前: <input name1="name1"/>
9              <input type="submit" value="Next"/>
10             </form>
11             <?php
12             ?>
13         }
14         static function show_math_form() {
15             session_start();
16             $name1=$_GET["name1"];
17             $_SESSION["name1"]=$name1;
18             ?>
19             <h1>成績入力 (数学)</h1>
20             名前: <?= $name1 ?>
21             <form action="p03Exam.php">
22                 <input type="hidden" name1="action" value="show_result"/>
23                 数学の成績: <input name1="math"/>
24                 <input type="submit" value="Next"/>
25             </form>
26             <?php
27         }
28
29         static function show_english_form() {
30             session_start();
31             $name=$_GET["name2"];
32             $_SESSION["name2"]=$name2;
33             ?>
34             <h1>成績入力 (英語)</h1>
35             名前: <?= $name2 ?>
36             <form action="p03Exam.php">
37                 <input type="hidden" name2="action" value="show_result"/>
38                 英語の成績: <input name2="english"/>
39                 <input type="submit" value="Next"/>
40             </form>
41             <?php
42         }
43
44         static function show_result() {
45             session_start();
46             $name1=$_SESSION["name1"];
47             $name2=$_SESSION["name2"];
48             $math=$_GET["math"];
49             $english=$_GET["english"]
50             ?>

```

```

51     <?= $name1 ?>の数学の成績は<?= $math ?>です。
52     <?php
53     }
54 }
55 require "controller.php";
56 run_controller("Exam");
57 ?>

```

ソースコード 14 は、bot 送信時の week5/p04Exam.php である。6,21,36 行目の form タグの action が違っていたためつまづいていた。この学生からの質問もなく、week5/p04Exam.php を修正していた。約 1 時間後に自力で完成していた。

その学生の完成したソースコードをソースコード 15 に示す。

ソースコード 15: 事例 5 の week5/p04Exam.php (完成時)

```

1  <?php
2  class Exam {
3      static function index() {
4          ?>
5          <h1>名前入力画面</h1>
6          <form action="p04Exam.php">
7              <input type="hidden" name="action" value="show_math_form"/>
8              名前: <input name="name"/>
9              <input type="submit" value="Next"/>
10             </form>
11             <?php
12         }
13         static function show_math_form() {
14             session_start();
15             $name=$_GET["name"];
16             $_SESSION["name"]=$name;
17             ?>
18             <h1>成績入力(数学) </h1>
19             名前: <?= $name ?>
20             <form action="p04Exam.php">
21                 <input type="hidden" name="action" value="show_english_form"/>
22                 数学の成績: <input name="math"/>
23                 <input type="submit" value="Next"/>
24             </form>
25             <?php
26         }
27         static function show_english_form() {
28             session_start();
29             $math=$_GET["math"];
30             $_SESSION["math"]=$math;
31             ?>
32             <h1>成績入力(英語) </h1>
33             名前: <?= $_SESSION["name"] ?>
34             <form action="p04Exam.php">
35                 <input type="hidden" name="action" value="show_result"/>
36                 英語の成績: <input name="english"/>
37                 <input type="submit" value="Next"/>
38             </form>
39             <?php
40         }
41
42         static function show_result() {
43             session_start();
44
45             $name=$_SESSION["name"];
46             $math=$_SESSION["math"];
47             $english=$_GET["english"];
48             ?>
49             <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?= $english ?>です。
50             <?php
51         }

```

```

52 }
53 require "controller.php";
54 run_controller("Exam");
55 ?>

```

6,20,34 行目が修正されている。

事例 6

第 8.1.2 節の事例 14 と同様 week5/p04Exam.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 16 に示す。

ソースコード 16: 事例 6 の week5/p04Exam.php (bot 送信時)

```

1  <?php
2  class Exam {
3      static function index() {
4          ?>
5          <h1>名前入力画面</h1>
6          <form action="p04Exam.php">
7              <input type="hidden" name="action" value="show_math_form"/>
8              名前: <input name="name"/>
9              <input type="submit" value="Next"/>
10             </form>
11             <?php
12         }
13         static function show_math_form() {
14             session_start();
15             $name=$_GET["name"];
16             $_SESSION["name"]=$name;
17             ?>
18             <h1>成績入力 (数学)</h1>
19             名前: <?= $name ?>
20             <form action="p04Exam.php">
21                 <input type="hidden" name="action" value="show_english_form"/>
22                 数学の成績: <input name="math"/>
23                 <input type="submit" value="Next"/>
24             </form>
25             <?php
26         }
27         static function show_english_form() {
28             /*session_start();
29             $name=$_GET["name"];
30             $_SESSION["name"]=$name;
31             $math=$_GET["math"];
32             $_SESSION["math"]=$math;*/
33             ?>
34             <h1>成績入力 (英語)</h1>
35             名前: <?= $name ?>
36             <form action="p04Exam.php">
37                 <input type="hidden" name="action" value="show_result"/>
38                 英語の成績: <input name="english"/>
39                 <input type="submit" value="Next"/>
40             </form>
41             <?php
42         }
43         static function show_result() {
44             session_start();
45             $name=$_SESSION["name"];
46             $math=$_GET["math"];
47             $english=$_GET["english"]
48             ?>
49             <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?= $english?>です。
50             <?php
51         }
52     }
53     require "controller.php";
54     run_controller("Exam");

```


ソースコード 16 は、bot 送信時の week5/p04Exam.php である。show_english_form() でセッション変数の部分がコメントアウトになっているため数学の点数が出ない状態になっていた。この学生は、DM 送る前に完成している。11:12 に閲覧した際は、ソースコード 16 の状態であった。送信している間に完成したためタイムラグが起きたものとみられる。

事例 8

第 8.1.2 節の事例 14 と同様 week5/p04Exam.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 17 に示す。

ソースコード 17: 事例 8 の week5/p04Exam.php (bot 送信時)

```

1  <?php
2  class Exam {
3      static function index() {
4          ?>
5          <h1>名前入力画面</h1>
6          <form action="p04Exam.php">
7              <input type="hidden" name="action" value="show_math_form"/>
8              名前: <input name="name"/>
9              <input type="submit" value="Next"/>
10             </form>
11             <?php
12         }
13         static function show_math_form() {
14             session_start();
15             $name=$_GET["name"];
16             $_SESSION["name"]=$name;
17             ?>
18             <h1>成績入力</h1>
19             名前: <?= $name ?>
20             <form action="p04Exam.php">
21                 <input type="hidden" name="action" value="show_english_form"/>
22                 数学の成績: <input name="math"/>
23                 <input type="submit" value="Next"/>
24             </form>
25             <?php
26         }
27         static function show_english_form() {
28             session_start();
29             $name=$_GET["name"];
30             $_SESSION["name"]=$name;
31             ?>
32             <h1>成績入力</h1>
33             名前: <?= $name ?>
34             <form action="p04Exam.php">
35                 <input type="hidden" name="action" value="show_result"/>
36                 英語の成績: <input name="english"/>
37                 <input type="submit" value="Next"/>
38             </form>
39             <?php
40         }
41         static function show_result() {
42             session_start();
43             $name=$_SESSION["name"];
44             $math=$_GET["math"];
45             $english=$_GET["english"];
46             ?>
47             <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?= $english ?>です。
48             <?php
49         }
50     }
51     require "controller.php";
52     run_controller("Exam");
53     ?>

```

ソースコード 17 は、bot 送信時の week5/p04Exam.php である。\$math=\$_GET["math"]; を show_result() で記述してしまっているため、数学の点数が取得できない状態になっている。この学生は、誰にも質問せずに翌週課題を完成させていた。

その学生の完成したソースコードをソースコード 18 に示す。

ソースコード 18: 事例 8 の week5/p04Exam.php (完成時)

```
1 <?php
2 class Exam {
3     static function index() {
4         ?>
5         <h1>名前入力画面</h1>
6         <form action="p04Exam.php">
7             <input type="hidden" name="action" value="show_math_form"/>
8             名前: <input name="name"/>
9             <input type="submit" value="Next"/>
10        </form>
11        <?php
12    }
13    static function show_math_form() {
14        session_start();
15        $name=$_GET["name"];
16        $_SESSION["name"]=$name;
17        ?>
18        <h1>成績入力</h1>
19        名前: <?= $name ?>
20        <form action="p04Exam.php">
21            <input type="hidden" name="action" value="show_english_form"/>
22            数学の成績: <input name="math"/>
23            <input type="submit" value="Next"/>
24        </form>
25        <?php
26    }
27    static function show_english_form() {
28        session_start();
29        $name=$_SESSION["name"];
30        $math=$_GET["math"];
31        $_SESSION["math"]=$math;
32        ?>
33        <h1>成績入力</h1>
34        名前: <?= $name ?>
35        <form action="p04Exam.php">
36            <input type="hidden" name="action" value="show_result"/>
37            英語の成績: <input name="english"/>
38            <input type="submit" value="Next"/>
39        </form>
40        <?php
41    }
42    static function show_result() {
43        session_start();
44        $name=$_SESSION["name"];
45        $math=$_SESSION["math"];
46        $english=$_GET["english"];
47        ?>
48        <?= $name ?>の数学の成績は<?= $math ?>、英語の成績は<?= $english ?>です。
49        <?php
50    }
51 }
52 require "controller.php";
53 run_controller("Exam");
54 ?>
```

\$math=\$_GET["math"]; を show_english_form() に移している。

事例 17

第 8.1.1 節の事例 19 と同様 week10/p04main.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 19 に示す。

ソースコード 19: 事例 17 の week10/p04main.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Main {
5     static function index() {
6         session_start();
7         $name = $_GET["name"];
8         $results=pdo_select("select_*_from_user_where_name=?", $name);
9         foreach($results as $result) {
10             ?>
11             <h1>メインページ</h1>
12             ユーザ名: <?= $result->name ?></br>
13             表示名: <?= $result->displayname ?></br>
14             所持金: <?= $result->gold ?></br>
15
16             <?php
17         }
18     }
19 }
20 run_controller("Main");
21 ?>
```

ソースコード 19 は、bot 送信時の week10/p04main.php である。ログインせずに実行したのでエラーが出ている。のちにログインしてつまづきを解消している。

事例 18

week8/p09login.php でつまづいていた学生である。week8/p09login.php は、ログインフォームを作成するプログラムである。bot 送信時のソースコードをソースコード 20 に示す。

ソースコード 20: 事例 18 の week8/p09login.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Login {
5     static function index() {
6         ?>
7         <form action="p09login.php">
8         <input type="hidden" name="action" value="check"/>
9         ユーザ名<input name="name"/><br/>
10
11         パスワード<input type="password" ame="password"/>
12         <input type="submit" value="Login"/>
13         </form>
14         <?php
15     }
16     static function check() {
17         session_start();
18         $name=$_GET["name"];
19         $password=$_GET["password"];
20         $records=pdo_select("select_*_from_user_where_name=?_and_password=?", $name,
21             $password);
22         if(count($records)==0){
23             print "ユーザ名またはパスワードが間違っています。";
24         }else{
25             print "ログインしました。";
26             $_SESSION["name"]= $name;
27         }
28         print "<BR>\n";
29         //ここは、
30         //08とほぼ同じだが、ログインに成功したときに$_SESSION["name"]にユーザ名を書き込む。
31         //最初に session_start()を忘れないように。
```

```

30     }
31     static function status() {
32         session_start();
33         if (isset($_SESSION["name"])) {
34             //ログインしているときの表示
35             $_SESSION["name"];
36             print "でログインしています。";
37         } else {
38             //ログインしていないときの表示
39             print "ログインしていません。";
40         }
41     }
42 }
43 run_controller("Login");
44 ?>

```

ソースコード 20 は、bot 送信時の week8/p09login.php である。11 行目が name="password" となっているためエラーが起きた。

約 2 週間後に完成している。完成したソースコードをソースコード 21 に示す。

ソースコード 21: 事例 18 の week8/p09login.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Login {
5      static function index() {
6          ?>
7          <form action="p09login.php">
8              <input type="hidden" name="action" value="check"/>
9              ユーザ名<input name="name"/><br/>
10
11             パスワード<input type="password" name="password"/>
12             <input type="submit" value="Login"/>
13         </form>
14         <?php
15     }
16     static function check() {
17         session_start();
18         $name=$_GET["name"];
19         $password=$_GET["password"];
20         $records=pdo_select("select * from user where name=? and password=?", $name,
21             $password);
22         if(count($records)==0){
23             print "ユーザ名またはパスワードが間違っています。";
24         }else{
25             print "ログインしました。";
26             $_SESSION["name"]= $name;
27         }
28         print "<BR>\n";
29         //ここは、
30         //08とほぼ同じだが、ログインに成功したときに$_SESSION["name"]にユーザ名を書き込む。
31         //最初に session_start()を忘れないように。
32     }
33     static function status() {
34         session_start();
35         if (isset($_SESSION["name"])) {
36             //ログインしているときの表示
37             $_SESSION["name"];
38             print "でログインしています。";
39         } else {
40             //ログインしていないときの表示
41             print "ログインしていません。";
42         }
43     }
44 }

```

```

43 run_controller("Login");
44 ?>

```

11 行目が変化している。

事例 21

week11/p04makechar.php でつまづいていた学生である。11 週目は、対戦ゲームのキャラクタを作るという目標を基に授業が進められた。week11/p04makechar.php は、対戦ゲームのキャラクタを作成するプログラムである。bot 送信時のソースコードをソースコード 22 に示す。

ソースコード 22: 事例 21 の week11/p04makechar.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8
9          ?>
10         <h1>キャラクタの作成</h1>
11         ユーザ名 <?= $userName ?>
12         <form action="p04makechar.php">
13             <input type="hidden" name="action" value="complete"/>
14             名前<input name="char_name">
15             <input type="submit" value="作成">
16         </form>
17         <?php
18     }
19     static function complete() {
20         session_start();
21         $results=pdo_select("select_*_from_user_where_name=?", $name);
22         $result=$results[0];
23         $userName=$_SESSION["name"];
24         $char_name=$_GET["char_name"];
25         $gold=$_GET["gold"];
26         pdo_exec("update_user_set_gold=?_where_name=?", $gold, $name);
27         ?>
28         ?>
29         <h1>作成完了</h1>
30         ユーザ名 <?= $userName ?><br/>
31         キャラクタ名 : <?= $char_name ?><br/>
32         所持金(作成前) <?= $gold ?><br/>
33         所持金(作成後) <?= $gold-100 ?><br/>
34         <?php
35     }
36 }
37 run_controller("MakeChar");
38 ?>

```

ソースコード 22 は、bot 送信時の week11/p04makechar.php である。22 行目でエラーが起きている。この学生には、以下のような DM を送信した。学籍番号は GAKUSEKI と置換している。

week11/p04makechar.php で苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。/home/webpro21/GAKUSEKI/week11/p04makechar.php に対するヒント week11/p04makechar.php のヒント: データベースにある変数の値を使いたいなら、week9/p05profile.php を参照キャラクタを作成して出力したいなら、week11/p03makechar.php を参照

この DM の後に week9/p05profile.php を閲覧するようになった。

week11/p04makechar.php は、30 分後に完成している。完成したソースコードをソースコード 23 に示す。

ソースコード 23: 事例 21 の week11/p04makechar.php (完成時)

```

1  <?php

```

```

2 require("pdo.php");
3 require("controller.php");
4 class MakeChar {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         ??
9         <h1>キャラクタの作成</h1>
10        ユーザ名 <?= $userName ?>
11        <form action="p04makechar.php">
12        <input type="hidden" name="action" value="complete"/>
13        名前<input name="char_name">
14        <input type="submit" value="作成">
15        </form>
16        <?php
17    }
18    static function complete() {
19        session_start();
20        $userName=$_SESSION["name"];
21        $char_name=$_GET["char_name"];
22        $name=$_SESSION["name"];
23        $results=pdo_select("select * from user where name=?", $name);
24        foreach ($results as $result) {
25            ??
26            <h1>作成完了</h1>
27            ユーザ名 <?= $userName ?><br/>
28            キャラクタ名 : <?= $char_name ?><br/>
29            所持金(作成前) <?= $result->gold ?><br/>
30            所持金(作成後) <?= $result->gold-100 ?><br/>
31
32            <?php
33        }
34    }
35 }
36 run_controller("MakeChar");
37 ??

```

20 行目以降が変化している。

事例 22

week11/p03makechar.php でつまづいていた学生である。week11/p03makechar.php は、対戦ゲームのキャラクタを作成するプログラムである。この時点ではまだデータベースへの書き込みは実装しなくてよいこととしている。bot 送信時のソースコードをソースコード 24 に示す。

ソースコード 24: 事例 22 の week11/p03makechar.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class MakeChar {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         ??
9         <h1>キャラクタの作成</h1>
10        <form action="p03login.php">
11        <input type="hidden" name="action" value="complete"/>
12        ユーザ名 <?= $userName ?><br>
13        名前<input type="name" name="name"/>
14        <input type="submit" value="作成"/>
15        </form>
16        <?php
17    }
18    static function complete() {
19        session_start();
20        $userName=$_SESSION["name"];

```

```

21     $charName=$_SESSION["name"];
22     ?>
23     <h1>作成完了</h1>
24     ユーザ名 <?= $userName ?>
25     キャラクタ名 <?= $charName ?>
26     <?php
27     }
28 }
29 run_controller("MakeChar");
30 ?>

```

ソースコード 24 は、bot 送信時の week11/p03makechar.php である。form タグの action が違っていただけからエラーが起きていた。

この学生は、約 9 時間後に完成している。完成したソースコードをソースコード 25 に示す。

ソースコード 25: 事例 22 の week11/p03makechar.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>
10         <form action="p03makechar.php">
11             <input type="hidden" name="action" value="complete"/>
12             ユーザ名 <?= $userName ?><BR>
13             名前<input type="name" name="name"/>
14             <input type="submit" value="作成"/>
15         </form>
16         <?php
17     }
18     static function complete() {
19         session_start();
20         $userName=$_SESSION["name"];
21         $charName=$_GET["name"];
22         ?>
23         <h1>メインページ</h1>
24         ユーザ名: <?= $userName ?><BR>
25         キャラクタ名: <?= $charName ?>
26         <?php
27     }
28 }
29 run_controller("MakeChar");
30 ?>

```

9 行目が変化している。

事例 23

第 8.1.1 節の事例 19 と同様 week10/p04main.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 26 に示す。

ソースコード 26: 事例 23 の week10/p04main.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  $name=$_POST["name"];
4  pdo_exec("update_user_set_gold=1000_where_name=?", $name);
5  $results=pdo_select("select_*_from_user_where_name=?", $name);
6  foreach ($results as $result) {
7      print_r($result);
8  }
9  ?>

```

ソースコード 26 は、bot 送信時の week10/p04main.php である。3 行目でエラーが起きている。
この学生は、約 1 時間後に完成している。完成したソースコードをソースコード 27 に示す。

ソースコード 27: 事例 23 の week10/p04main.php (完成時)

```
1 <?php
2 session_start();
3 ?>
4 <h1>メインページ</h1>
5 <?php
6 require("pdo.php");
7 $name=$_SESSION["name"];
8 pdo_exec("update_user_set_gold=1000_where_name=?", $name);
9 $results=pdo_select("select_*_from_user_where_name=?", $name);
10 foreach ($results as $result) {
11     print_r("ユーザ名:");
12     print_r($result->name);
13     ?>
14     <br>
15     <?php
16     print_r("表示名:");
17     print_r($result->displayname);
18     ?>
19     <br>
20     <?php
21     print_r("所持金:");
22     print_r($result->gold);
23 }
24 ?>
```

\$name=\$_POST["name"]; が \$name=\$_SESSION["name"]; に変化したうえ、ユーザ名と表示名、所持金が表示できるようになった。

事例 27

事例 21 と同様 week11/p04makechar.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 28 に示す。

ソースコード 28: 事例 27 の week11/p04makechar.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class MakeChar {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         ?>
9         <h1>キャラクタの作成</h1>
10        ユーザ名 <?= $userName ?>
11        <form action="p04makechar.php">
12            <input type="hidden" name="action" value="complete"/>
13            名前<input name="charname">
14            <input type="submit" value="作成">
15        </form>
16        <?php
17    }
18    static function complete() {
19        session_start();
20        $userName=$_SESSION["name"];
21        $charName=$_GET["charname"];
22        $records=pdo_select("select_*_from_user_where_name=?", $userName);
23        ?>
24        <h1>作成完了</h1>
25        ユーザ名 <?= $userName ?><br/>
26        キャラクタ名 <?= $charName ?><br/>
27        <?php
28        foreach($records as $record) {
```



```

29         print "所持金（作成前） ␣$record->gold<br/>";
30         pdo_exec("update␣user␣set␣gold=?␣where␣name=?", $gold, $name);
31         print "所持金（作成後） ␣$record->gold<br/>";
32     }/*
33     pdo_exec("update user set gold=? where name=?", $gold, $name);
34     foreach($records as $record) {
35         print "所持金(作成後) $record->gold<br/>";
36     }*/
37
38     }
39 }
40 run_controller("MakeChar");
41 ?>

```

ソースコード 28 は、bot 送信時の week11/p04makechar.php である。30 行目でエラーが起きている。\$gold が
ないためエラーが起こってしまった。

この学生は、約 1 週間後に完成している。完成したソースコードをソースコード 29 に示す。

ソースコード 29: 事例 27 の week11/p04makechar.php（完成時）

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>
10         ユーザ名 <?= $userName ?>
11         <form action="p04makechar.php">
12             <input type="hidden" name="action" value="complete"/>
13             名前<input name="charname">
14             <input type="submit" value="作成">
15         </form>
16         <?php
17     }
18     static function complete() {
19         session_start();
20         $userName=$_SESSION["name"];
21         $charName=$_GET["charname"];
22         $records=pdo_select("select␣*␣from␣user␣where␣name=?", $userName);
23         ?>
24         <h1>作成完了</h1>
25         ユーザ名 <?= $userName ?><br/>
26         キャラクタ名 <?= $charName ?><br/>
27         <?php
28         foreach($records as $record) {
29             print "所持金（作成前） ␣$record->gold<br/>";
30         }
31
32         //pdo_exec("update user set gold=? where name=?", 1000, $userName);
33         foreach($records as $record) {
34             $gold=$record->gold-100;
35             print "所持金（作成後） ␣$gold<br/>";
36         }
37     }
38 }
39 run_controller("MakeChar");
40 ?>

```

このプログラムでユーザ名、キャラクタ名、所持金（作成前）、所持金（作成後）が表示できるようになった。

事例 30

week12/p07fight.php でつまづいていた学生である。12 週目は、対戦ゲームの戦闘ができるまでを目標に授業
が行われた。week12/p07fight.php は、対戦ゲームを実施した後にプレイヤーが勝った場合に所持金を増やすよう

にするプログラムである。bot 送信時のソースコードをソースコード 30 に示す。

ソースコード 30: 事例 30 の week12/p07fight.php (bot 送信時)

```
1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Fight {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         $results=pdo_select("select_*_from_chara_where_owner=?", $userName);
9         ?>
10        <h1>出場させるキャラクターを選択</h1>
11        <?php
12        foreach ($results as $result) {
13            ?>
14            <?= $result->name ?>
15            <a href="<?=basename(__FILE__)>?>?action=select_enemy&id=<?= $result->id ?>">
16                選択
17            </a><br/>
18
19            体力 <?= $result->hp ?><br/>
20            攻撃力 <?= $result->ap ?><br/>
21            スキルポイント <?= $result->sp ?><br/>
22        <?php
23    }
24 }
25 static function select_enemy() {
26     session_start();
27     $player_chara_id=$_GET["id"];
28     $_SESSION["player_chara_id"]=$player_chara_id;
29     $userName=$_SESSION["name"];
30     $results=pdo_select("select_*_from_chara_where_owner<>?", $userName);
31     ?>
32     <h1>対戦相手を選択</h1>
33     <?php
34     foreach ($results as $result) {
35         ?>
36         <?= $result->name ?>
37         <a href="<?=basename(__FILE__)>?>?action=vs&id=<?= $result->id ?>">
38             選択
39         </a><br/>
40
41         体力 <?= $result->hp ?><br/>
42         攻撃力 <?= $result->ap ?><br/>
43         スキルポイント <?= $result->sp ?><br/>
44     <?php
45 }
46 }
47 static function vs(){
48     session_start();
49     $enemy_chara_id=$_GET["id"];
50     $_SESSION["enemy_chara_id"]=$enemy_chara_id;
51     $player_chara_id=$_SESSION["player_chara_id"];
52     $results=pdo_select("select_*_from_chara_where_id=?", $player_chara_id);
53     $player_chara=$results[0];
54     $_SESSION["player_hp"]=$player_chara->hp;
55     $_SESSION["player_name"]=$player_chara->name;
56     $results=pdo_select("select_*_from_chara_where_id=?", $enemy_chara_id);
57     $enemy_chara=$results[0];
58     $_SESSION["enemy_hp"]=$enemy_chara->hp;
59     $_SESSION["enemy_name"]=$enemy_chara->name;
60     $player_name=$_SESSION["player_name"];
61     $enemy_name=$_SESSION["enemy_name"];
62     ?>
```

```

63 <h1>戦闘開始</h1>
64 <?= $player_name ?> vs <?= $enemy_name ?>
65 <?php
66 foreach ($results as $result) {
67     ?>
68     <?= $result->name ?>
69     <a href="<?=basename(__FILE__)>?>?action=vs&id=<?= $result->id ?>">
70
71     体力 <?= $result->hp ?><br/>
72     攻撃力 <?= $result->ap ?><br/>
73     スキルポイント <?= $result->sp ?><br/>
74     <a href="<?=basename(__FILE__)>?>?action=player_turn">開始</a>
75     <?php
76 }
77 static function player_turn() {
78     session_start();
79     $player_hp=$_SESSION["player_hp"];
80     $enemy_hp=$_SESSION["enemy_hp"];
81     $player_name=$_SESSION["player_name"];
82     $enemy_name=$_SESSION["enemy_name"];
83     ?>
84     <?= $player_name ?>のHP <?= $player_hp ?><BR>
85     <?= $enemy_name ?>のHP <?= $enemy_hp ?><BR>
86     <a href="<?=basename(__FILE__)>?>?action=player_attack&type=normal">通常攻
      撃</a>
87     <?php
88 }
89 static function player_attack() {
90     session_start();
91     $player_hp=$_SESSION["player_hp"];
92     $enemy_hp=$_SESSION["enemy_hp"];
93     $player_name=$_SESSION["player_name"];
94     $enemy_name=$_SESSION["enemy_name"];
95     $player_chara_id=$_SESSION["player_chara_id"];
96     $results=pdo_select("select_*_from_chara_where_id=?", $player_chara_id);
97     $player_chara=$results[0];
98     // ダメージは 1から「プレイヤーの攻撃力」までの乱数
99     $damage = random_int( 1, $player_chara->ap );
100     $enemy_hp -= $damage;
101     $_SESSION["enemy_hp"]=$enemy_hp;
102     ?>
103     <?= $player_name ?>の攻撃: <BR/>
104     <?= $enemy_name ?>に<?= $damage ?>のダメージ<br/>
105     <?= $enemy_name ?>のHP :<?= $enemy_hp ?><br/>
106     <?php if ($enemy_hp<=0) { ?>
107         <a href="<?=basename(__FILE__)>?>?action=win">次へ</a>
108     <?php } else if ($player_hp<=0) { ?>
109         <a href="<?=basename(__FILE__)>?>?action=lose">次へ</a>
110     <?php } else { ?>
111         <a href="<?=basename(__FILE__)>?>?action=enemy_attack">次へ</a>
112     <?php }
113 }
114 static function enemy_attack() {
115     session_start();
116     $player_hp=$_SESSION["player_hp"];
117     $enemy_hp=$_SESSION["enemy_hp"];
118     $player_name=$_SESSION["player_name"];
119     $enemy_name=$_SESSION["enemy_name"];
120     $enemy_chara_id=$_SESSION["enemy_chara_id"];
121     $results=pdo_select("select_*_from_chara_where_id=?", $enemy_chara_id);
122     $enemy_chara=$results[0];
123     // ダメージは 1から「対戦相手の攻撃力」までの乱数
124     $damage = random_int( 1, $enemy_chara->ap );
125     $player_hp-= $damage;
126     $_SESSION["player_hp"]=$player_hp;
127     ?>

```

```

128         <?= $enemy_name ?>の攻撃: <BR/>
129         <?= $player_name ?>に<?= $damage ?>のダメージ<br/>
130         <?= $player_name ?>のHP :<?= $player_hp ?><br/>
131         <a href="<?=basename(__FILE__)>?>?action=player_turn">次へ</a>
132     <?php
133 }
134 //---★↓追加部分↓★---
135 static function win() {
136     session_start();
137     $userName=$_SESSION["name"];
138     $results=pdo_select("select_*_from_user_where_name=?", $userName);
139     $player=$results[0];
140     $gold_after=$player->gold+500;
141     pdo_exec("update_user_set_gold=?_where_name=?", $gold_after, $userName);
142     ?>
143     <h1>勝利</h1>
144     所持金は<?= $gold_after ?> になった。
145     </br>
146     <a href="p04main.php">メイン画面へ</a>
147     <?php
148 }
149 static function lose() {
150     session_start();
151     $userName=$_SESSION["name"];
152     $results=pdo_select("select_*_from_user_where_name=?", $userName);
153     $player=$results[0];
154     ?>
155     <h1>敗北</h1>
156     </br>
157     <a href="p04main.php">メイン画面へ</a>
158     <?php
159 }
160 }
161 run_controller("Fight");
162 ?>

```

ソースコード 30 は、bot 送信時の week12/p07fight.php である。77 行目以降でインデントずれが起きている。この学生は、約 1 週間後に完成している。完成したソースコードをソースコード 31 に示す。

ソースコード 31: 事例 30 の week12/p07fight.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Fight {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          $results=pdo_select("select_*_from_chara_where_owner=?", $userName);
9          ?>
10         <h1>出場させるキャラクターを選択</h1>
11         <?php
12         foreach ($results as $result) {
13             ?>
14             <?= $result->name ?>
15             <a href="<?=basename(__FILE__)>?>?action=select_enemy&id=<?= $result->id_?>
16                 選択
17             </a><br/>
18
19             体力 <?= $result->hp ?><br/>
20             攻撃力 <?= $result->ap ?><br/>
21             スキルポイント <?= $result->sp ?><br/>
22             <?php
23         }
24     }
25     static function select_enemy() {

```

```

26     session_start();
27     $player_chara_id=$_GET["id"];
28     $_SESSION["player_chara_id"]=$player_chara_id;
29     $userName=$_SESSION["name"];
30     $results=pdo_select("select_*_from_chara_where_owner<>?", $userName);
31     ??
32     <h1>対戦相手を選択</h1>
33     <?php
34     foreach ($results as $result) {
35         ??
36         <?= $result->name ??>
37         <a href="<?=basename(__FILE__)_??>?action=vs&id=<?=$_$result->id_?>">
38             選択
39         </a><br/>
40
41         体力 <?= $result->hp ??><br/>
42         攻撃力 <?= $result->ap ??><br/>
43         スキルポイント <?= $result->sp ??><br/>
44     }
45 }
46
47 static function vs(){
48     session_start();
49     $enemy_chara_id=$_GET["id"];
50     $_SESSION["enemy_chara_id"]=$enemy_chara_id;
51     $player_chara_id=$_SESSION["player_chara_id"];
52     $results=pdo_select("select_*_from_chara_where_id=?", $player_chara_id);
53     $player_chara=$results[0];
54     $_SESSION["player_hp"]=$player_chara->hp;
55     $_SESSION["player_name"]=$player_chara->name;
56     $results=pdo_select("select_*_from_chara_where_id=?", $enemy_chara_id);
57     $enemy_chara=$results[0];
58     $_SESSION["enemy_hp"]=$enemy_chara->hp;
59     $_SESSION["enemy_name"]=$enemy_chara->name;
60     $player_name=$_SESSION["player_name"];
61     $enemy_name=$_SESSION["enemy_name"];
62     ??
63     <h1>戦闘開始</h1>
64     <?= $player_name ??> vs <?= $enemy_name ??>
65     <?php
66     foreach ($results as $result) {
67         ??
68
69         <br/>
70         <?= $result->name ??>
71         <a href="<?=basename(__FILE__)_??>?action=vs&id=<?=$_$result->id_?>">
72             </a><br/>
73         体力 <?= $result->hp ??><br/>
74         攻撃力 <?= $result->ap ??><br/>
75         スキルポイント <?= $result->sp ??><br/>
76
77         <br/>
78         <?= $player_chara->name ??>
79         <a href="<?=basename(__FILE__)_??>?action=vs&id=<?=$_$player_chara->id_?>">
80             </a><br/>
81         体力 <?= $player_chara->hp ??><br/>
82         攻撃力 <?= $player_chara->ap ??><br/>
83         スキルポイント <?= $player_chara->sp ??><br/>
84
85         <a href="<?=basename(__FILE__)_??>?action=player_turn">開始</a>
86     }
87 }
88
89 static function player_turn(){
90     session_start();
91     $player_hp=$_SESSION["player_hp"];

```

```

92     $enemy_hp=$_SESSION["enemy_hp"];
93     $player_name=$_SESSION["player_name"];
94     $enemy_name=$_SESSION["enemy_name"];
95     ?>
96     <?= $player_name ?>のHP <?= $player_hp ?><BR>
97     <?= $enemy_name ?>のHP <?= $enemy_hp ?><BR>
98     <a href="<?=$_basename(__FILE__)>?action=player_attack&type=normal">通常攻撃</a>
99     <?php
100 }
101 static function player_attack() {
102     session_start();
103     $player_hp=$_SESSION["player_hp"];
104     $enemy_hp=$_SESSION["enemy_hp"];
105     $player_name=$_SESSION["player_name"];
106     $enemy_name=$_SESSION["enemy_name"];
107     $player_chara_id=$_SESSION["player_chara_id"];
108     $results=pdo_select("select* from chara where id=?", $player_chara_id);
109     $player_chara=$results[0];
110     // ダメージは 1から「プレイヤーの攻撃力」までの乱数
111     $damage = random_int( 1, $player_chara->ap );
112     $enemy_hp -= $damage;
113     $_SESSION["enemy_hp"]=$enemy_hp;
114     ?>
115     <?= $player_name ?>の攻撃: <BR/>
116     <?= $enemy_name ?>に<?= $damage ?>のダメージ<br/>
117     <?= $enemy_name ?>のHP :<?= $enemy_hp ?><br/>
118     <?php if ($enemy_hp<=0) { ?>
119         <a href="<?=$_basename(__FILE__)>?action=win">次へ</a>
120     <?php }else { ?>
121         <a href="<?=$_basename(__FILE__)>?action=enemy_attack">次へ</a>
122     <?php }
123 }
124 static function enemy_attack() {
125     session_start();
126     $player_hp=$_SESSION["player_hp"];
127     $enemy_hp=$_SESSION["enemy_hp"];
128     $player_name=$_SESSION["player_name"];
129     $enemy_name=$_SESSION["enemy_name"];
130     $enemy_chara_id=$_SESSION["enemy_chara_id"];
131     $results=pdo_select("select* from chara where id=?", $enemy_chara_id);
132     $enemy_chara=$results[0];
133     // ダメージは 1から「対戦相手の攻撃力」までの乱数
134     $damage = random_int( 1, $enemy_chara->ap );
135     $player_hp-= $damage;
136     $_SESSION["player_hp"]=$player_hp;
137     ?>
138     <?= $enemy_name ?>の攻撃: <BR/>
139     <?= $player_name ?>に<?= $damage ?>のダメージ<br/>
140     <?= $player_name ?>のHP :<?= $player_hp ?><br/>
141     <?php if ($player_hp<=0) { ?>
142         <a href="<?=$_basename(__FILE__)>?action=lose">次へ</a>
143     <?php }else { ?>
144         <a href="<?=$_basename(__FILE__)>?action=player_turn">次へ</a>
145     <?php }
146 }
147 //---★↓追加部分↓★---
148 static function win() {
149     session_start();
150     $userName=$_SESSION["name"];
151     $results=pdo_select("select* from user where name=?", $userName);
152     $player=$results[0];
153     $gold_after=$player->gold+500;
154     pdo_exec("update user set gold=? where name=?", $gold_after, $userName);
155     ?>
156     <h1>勝利</h1>
157     所持金は<?= $gold_after ?> になった。

```

```

158     </br>
159     <a href="p04main.php">メイン画面へ</a>
160     <?php
161 }
162 static function lose() {
163     session_start();
164     $userName=$_SESSION["name"];
165     $results=pdo_select("select_*_from_user_ where_name=?", $userName);
166     $player=$results[0];
167     ?>
168     <h1>敗北</h1>
169     </br>
170     <a href="p04main.php">メイン画面へ</a>
171     <?php
172 }
173 }
174 run_controller("Fight");
175 ?>

```

インデントずれは修正されている。

事例 33

week12/p01fight.php でつまづいていた学生である。week12/p01fight.php は、自分のアカウントにログインし自分の作成したキャラクタを表示させるプログラムである。bot 送信時のソースコードをソースコード 32 に示す。

ソースコード 32: 事例 33 の week12/p01fight.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Fight {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          $results=pdo_select("select_*_from_chara_ where_chara=?", $userName);
9          ?>
10         <h1>出場させるキャラクターを選択</h1>
11         <?php
12         foreach ($results as $result) {
13             ?>
14             <?= $result->name ?>
15             <a href="<?=basename(__FILE__)_?>?action=select_enemy&id=<?=$_$result->id_?>
16                 ">
17                 選択
18                 </a><br/>
19             <?php
20         }
21     }
22 }
23 run_controller("Fight");
24 ?>

```

ソースコード 32 は、bot 送信時の week12/p01fight.php である。8 行目でエラーが起きている。

この学生は、別課題に取り組んでいたが 11:55 に DM を送信後、11:56 に再びの week12/p01fight.php を取り組むようになった。そして 11:57 に完成している。

完成したソースコードをソースコード 33 に示す。

ソースコード 33: 事例 33 の week12/p01fight.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Fight {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];

```

```

8      $results=pdo_select("select*_from_chara_where_owner=?", $userName);
9      ?>
10     <h1>出場させるキャラクターを選択</h1>
11     <?php
12     foreach ($results as $result) {
13         ?>
14         <?= $result->name ?>
15         <a href="<?=basename(__FILE__)>?>action=select_enemy&id=<?= $result->id ?>
16             ">
17             選択
18         </a><br/>
19     }
20 }
21 }
22 run_controller("Fight");
23 ?>

```

8 行目が変化している。

事例 35

事例 22 と同様 week11/p03makechar.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 34 に示す。

ソースコード 34: 事例 35 の week11/p03makechar.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>
10         ユーザ名 <?= $userName ?>
11         <form action="p03makechar.php">
12         <input type="hidden" name="action" value="complete"/>
13         名前<input name="char_name">
14         <input type="submit" value="作成">
15         <?php
16     }
17     static function complete() {
18         session_start();
19         $userName=$_SESSION["name"];
20         ?>
21         <h1>作成完了</h1>
22         ユーザ名 <?= $userName ?>
23         <?php
24     }
25 }
26 run_controller("MakeChar");
27 ?>

```

ソースコード 34 は、bot 送信時の week11/p03makechar.php である。ログインしていないことによるエラーだった。この学生は、約 15 分後にログインし、終了している。

事例 36

week14/training.php でつまづいていた学生である。14 週目は、対戦ゲームで自分の作成したキャラクタを強化ができるプログラムを作成することを目標に授業が進められた。教員のほうで作成してほしいページが出されてそのページと同様のページを作成する課題である。week14/training.php は、先ほど述べた作成してほしいページを作成する課題である。ソースコードには、あらかじめ教員のほうでヒントとなるコメントが書いており、そのヒントを基にプログラムを完成させる課題である。bot 送信時のソースコードをソースコード 35 に示す。

ソースコード 35: 事例 36 の week14/training.php (bot 送信時)


```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Training {
5     static function index() {
6         session_start();
7         // 「戦闘」のプレイヤーキャラクタ選択の画面を参考に,
8         // 現在ログインしているユーザが所持しているキャラクター一覧を$resultsに入れる
9         $name=$_SESSION["userName"];
10        $results=pdo_select("select_*_from_chara_where_owner=?", $name);
11
12        ?>
13        <h1>トレーニングするキャラクターを選択</h1>
14        <?php
15        foreach ($results as $result) {
16            ?>
17            <?= $result->name ?></br>
18            体力: <?= $result->hp ?></br>
19            攻撃力: <?= $result->ap ?></br>
20            スキルポイント: <?= $result->sp ?></br>
21            <a href="<?=basename(__FILE__)_?>?action=select_enemy&id=<?= $result->id_?>
22                ">
23                選択</br>
24            </a></br>
25            <?php
26        }
27    }
28    static function select_menu() {
29        session_start();
30        // 選択したキャラクタのIDをセッション変数に保存する
31        $_SESSION["id"] = $_GET["id"];
32        $player_chara_id = $_SESSION["id"];
33        // 選択したキャラクタの情報を取得して,$charaに入れる(戦闘の「vs」を参考に)
34        $results=pdo_select("select_*_from_chara_where_id=?", $player_chara_id);
35        $chara=$results[0];
36        //-----
37        ?>
38        <h1>トレーニングメニュー選択</h1>
39
40        名前:<?= $chara->name ?>,
41        HP: <?= $chara->hp ?> ,
42        AP: <?= $chara->ap ?> ,
43        SP: <?= $chara->sp ?>
44        <br/>
45
46        <a href="<?=basename(__FILE__)_?>?action=start&type=hp">体力の最大値を上げる</a><br/>
47        <!-- AP や SP を上げるメニューも作ってみよう -->
48        <a href="<?=basename(__FILE__)_?>?action=start&type=ap">攻撃力を上げる</a><br/>
49        <a href="<?=basename(__FILE__)_?>?action=start&type=sp">スキルポイント最大値を上
50            げる</a><br/>
51        <?php
52    }
53    static function start() {
54        session_start();
55        // 現在ログインしているユーザの所持金を読み出して
56        // $goldに入れる(メインページを参考に)
57        $name=$_SESSION["userName"];
58        $results=pdo_select("select_*_from_user_where_name=?", $name);
59        $player=$results[0];
60        $gold = $player->gold;
61        $chara=$results[0];
62
63        foreach($results as $result) {
64            $gold_after=$gold-100;
65            if ($gold_after<0) {

```

```

64         ?>Gold が足りません<?php
65     } else {
66         //足りていれば、所持金を 100減らす(「キャラクタを作る」などを参考に)
67         pdo_exec("update_user_set_gold=?_where_name=?", $result->gold - 100, $name
68             );
69         // select_menu で保存したキャラクタの ID のセッション変数を $chara_id に読み出す
70         $chara_id = $_SESSION["id"];
71         // 選択したキャラクタの情報を取得して, $chara に入れる
72         $results=pdo_select("select*_from_chara_where_id=?", $chara_id);
73         $chara = $results[0];
74
75         ?><h1>トレーニング完了</h1>
76
77         <?php if($_GET['type']== 'hp' ){?>
78             <?= $chara->name?>の
79             <?php
80                 $hp=$chara->hp+random_int(1,3);
81                 pdo_exec("update_chara_set_hp=?_where_id=?", $hp, $chara_id);
82                 ?> HP が<?= $hp ?> になった.
83                 所持金は<?= $gold ?> になった。
84             <?php
85         } else if($_GET['type']== 'sp'){ ?>
86             <?= $chara->name?>の
87             <?php
88                 $sp=$chara->sp+random_int(1,3);
89                 pdo_exec("update_chara_set_sp=?_where_id=?", $sp, $chara_id);
90                 ?> SP が<?= $sp ?> になった.
91                 所持金は<?= $gold ?> になった。
92             <?php
93         } else if($_GET['type']== 'ap'){ ?>
94             <?= $chara->name?>の
95             <?php
96                 $ap=$chara->ap+random_int(1,3);
97                 pdo_exec("update_chara_set_ap=?_where_id=?", $ap, $chara_id);
98                 ?> AP が<?= $ap ?> になった.
99                 所持金は<?= $gold ?> になった。
100             <?php
101         }
102     }
103 }
104 }
105 }
106 run_controller("Training");
107
108
109 ?>

```

ソースコード 35 は、bot 送信時の week14/training.php である。21 行目の action が select_enemy と記述されているが、select_enemy() が存在しないためエラーが起きている。

この学生は、約 7 分後に完成している。完成したソースコードをソースコード 36 に示す。

ソースコード 36: 事例 36 の week14/training.php (完成時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Training {
5      static function index() {
6          session_start();
7          // 「戦闘」のプレイヤーキャラクタ選択の画面を参考に,
8          // 現在ログインしているユーザが所持しているキャラクター一覧を $results に入れる
9          $name=$_SESSION["userName"];
10         $results=pdo_select("select*_from_chara_where_owner=?", $name);
11
12         ?>

```

```

13 <h1>トレーニングするキャラクターを選択</h1>
14 <?php
15 foreach ($results as $result) {
16     ?>
17     <?= $result->name ?></br>
18     体力:<?= $result->hp ?></br>
19     攻撃力:<?= $result->ap ?></br>
20     スキルポイント:<?= $result->sp ?></br>
21     <a href="<?=basename(__FILE__)>?>?action=select_menu&id=<?= $result->id_?>"
22         >
23     選択<br/>
24     </a><br/>
25     <?php
26 }
27 static function select_menu() {
28     session_start();
29     // 選択したキャラクタのIDをセッション変数に保存する
30     $_SESSION["id"] = $_GET["id"];
31     $player_chara_id = $_SESSION["id"];
32     // 選択したキャラクタの情報を取得して,$charaに入れる(戦闘の「vs」を参考に)
33     $results=pdo_select("select_*_from_chara_where_id=?", $player_chara_id);
34     $chara=$results[0];
35     //-----
36     ?>
37     <h1>トレーニングメニュー選択</h1>
38
39     名前:<?= $chara->name ?>,
40     HP: <?= $chara->hp ?> ,
41     AP: <?= $chara->ap ?> ,
42     SP: <?= $chara->sp ?>
43     <br/>
44
45     <a href="<?=basename(__FILE__)>?>?action=start&type=hp">体力の最大値を上げる</a><br/>
46     <!-- AP や SP を上げるメニューも作ってみよう -->
47     <a href="<?=basename(__FILE__)>?>?action=start&type=ap">攻撃力を上げる</a><br/>
48     <a href="<?=basename(__FILE__)>?>?action=start&type=sp">スキルポイント最大値を上げる</a><br/>
49     <?php
50 }
51 static function start() {
52     session_start();
53     // 現在ログインしているユーザの所持金を読み出して
54     // $goldに入れる(メインページを参考に)
55     $name=$_SESSION["userName"];
56     $results=pdo_select("select_*_from_user_where_name=?", $name);
57     $player=$results[0];
58     $gold = $player->gold;
59     $chara=$results[0];
60
61     foreach($results as $result) {
62         $gold_after=$gold-100;
63         if ($gold_after<0) {
64             ?>Goldが足りません<?php
65         } else {
66             //足りていれば, 所持金を 100減らす(「キャラクタを作る」などを参考に)
67             pdo_exec("update_user_set_gold=?_where_name=?", $result->gold - 100, $name);
68             // select_menu で保存したキャラクタの ID のセッション変数を $chara_id に読み出す
69             $chara_id = $_SESSION["id"];
70             // 選択したキャラクタの情報を取得して,$charaに入れる
71             $results=pdo_select("select_*_from_chara_where_id=?", $chara_id);
72             $chara = $results[0];
73
74             ?><h1>トレーニング完了</h1>

```

```

75
76         <?php if($_GET['type']== 'hp' ){?>
77             <?= $chara->name?>の
78             <?php
79                 $hp=$chara->hp+random_int(1,3);
80                 pdo_exec("update_chara_set_hp=?_where_id=?", $hp, $chara_id);
81                 ?> HP が<?= $hp ?> になった.
82                 所持金は<?= $gold ?> になった.
83             <?php
84         } else if($_GET['type']== 'sp'){ ?>
85             <?= $chara->name?>の
86             <?php
87                 $sp=$chara->sp+random_int(1,3);
88                 pdo_exec("update_chara_set_sp=?_where_id=?", $sp, $chara_id);
89                 ?> SP が<?= $sp ?> になった.
90                 所持金は<?= $gold ?> になった.
91             <?php
92         } else if($_GET['type']== 'ap'){ ?>
93             <?= $chara->name?>の
94             <?php
95                 $ap=$chara->ap+random_int(1,3);
96                 pdo_exec("update_chara_set_ap=?_where_id=?", $ap, $chara_id);
97                 ?> AP が<?= $ap ?> になった.
98                 所持金は<?= $gold ?> になった.
99             <?php
100         }
101     }
102 }
103 }
104 }
105 }
106 run_controller("Training");
107
108
109 ?>

```

21 行目が修正されている。

8.2 未解決事例

表 4 より，解決時刻が未解決となっている学生の事例を紹介する。

事例 3

week2/p09.php でつまづいていた学生である。week2/p09.php は，値を入力したら入力値分の画像を表示させるプログラムである。bot 送信時のソースコードをソースコード 37 に示す。

ソースコード 37: 事例 3 の week2/p09.php (bot 送信時)

```

1  <html>
2  <meta charset="UTF-8">
3  <body>
4  <?php $i=0; ?>
5  <?php $n = $_GET["n"]; ?>
6  <h1>3 Cats</h1>
7  <?php while ($i<$n) { ?>
8      <?php $i+=1; ?>
9      
10 <?php } ?>
11 </body>
12 </html>

```

ソースコード 37 は，bot 送信時の week2/p09.php である。事例 1 と同様 n に数値を入力していないことからエラーは起きていた。この学生は結局数値を入れないまま放置してしまっている。この学生には以下のような DM を送信した。

p09 を苦戦中のようなので、ヒントを貼っておきます。それでもわからないなら遠慮なく質問してください。 week2/p09.php のヒント: while 文の繰り返し処理をしたいなら、week2/p05.php を参照
画像を複数出力するには、week2/p08.php を参照

1 週間後に「ありがとうございます」と返信しているが、数値を入力しないまま終わっている。プログラムは正しいのだが、数値を入れない状態が続いていたので、エラーが出続けたために bot が反応してしまったものと考えられる。課題のほうは通っているので、問題ないと考え放置してしたと考えられる。

事例 4

week4/p05Keisan.php でつまづいていた学生である。 week4/p05Keisan.php は、数値と演算子を入力して計算結果を表示するプログラムである。 bot 送信時のソースコードをソースコード 38 に示す。

ソースコード 38: 事例 4 の week4/p05Keisan.php (bot 送信時)

```
1 <?php
2 class Keisan {
3     static function index() {
4         ?>
5         <h1>入力画面</h1>
6         <form action="p05Keisan.php">
7             <input type="hidden" name="action" value="show_result"/>
8             <input name="a"/>
9             <input type="hidden" name="action" value="show_result"/>
10            <input name="x"/>
11            <input type="hidden" name="action" value="show_result"/>
12            <input name="b"/>
13            <input type="submit" value="Calculate"/>
14        </form>
15        <?php
16    }
17    static function show_result() {
18        $a=$_GET["a"];
19        $b=$_GET["b"];
20        $x=$_GET["x"];
21        ?>
22        <h1>結果表示画面</h1>
23        <?php
24        ?>
25        if ($x==+) {
26            <?= $a ?> + <?= $b ?> = <?= $a+$b ?>
27            <?
28        }
29    }
30    require "controller.php";
31    run_controller("Keisan");
32    ?>
```

ソースコード 38 は、bot 送信時の week4/p05Keisan.php である。演算子別の処理ができていない。なお、この学生は誰にも質問せず課題も完成していない状態である。

事例 11

week1/p21.php でつまづいていた学生である。 week1/p21.php は、age の値が 18 以上なら "*** is an adult", 18 未満なら "*** is a child" と表示させるプログラムである。 ***には name の値が入る。 bot 送信時のソースコードをソースコード 39 に示す。

ソースコード 39: 事例 11 の week1/p21.php (bot 送信時)

```
1 <?php
2 $n = $_GET["name"];
3 $a = $_GET["age"];
4 if ($a>=18) {
5     print "$n_is_an_adult";
6 } else {
7     print "$n_is_a_child";
8 }
```

ソースコード 39 は、bot 送信時の week1/p21.php である。プログラムは完成はしているが、テストパターンで変数の入力をしていないためエラーが出ている。

事例 12

事例 4 と同様 week4/p05Keisan.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 40 に示す。

ソースコード 40: 事例 12 の week4/p05Keisan.php (bot 送信時)

```

1 <?php
2 class Keisan {
3     static function index() {
4         ?>
5         <h1>入力画面</h1>
6         <form action="p05Keisan.php">
7             <input type="hidden" name="action" value="show_result"/>
8             <input name="a"/>
9             <input name="op" value="+"/>
10            <input name="b"/>
11            <input type="submit" value="Calculate"/>
12        </form>
13    <?php
14    }
15    static function show_result() {
16        $a=$_GET["a"];
17        $op=$_GET["op"];
18        $b=$_GET["b"];
19        ?>
20        <h1>結果表示画面</h1>
21        <?php
22        if($op == +){
23            ?>
24            <?= $a ?> + <?= $b ?> = <?= $a+$b ?>;
25            <?php
26        }else($op == *){
27            ?>
28            <?= $a ?> * <?= $b ?> = <?= $a*$b ?>;
29            <?php
30        }
31        ?>
32        <?php
33    }
34 }
35 require "controller.php";
36 run_controller("Keisan");
37 ?>

```

ソースコード 40 は、bot 送信時の week4/p05Keisan.php である。22 行目と 26 行目の演算子をダブルクォーテーションで囲んでいないことからエラーが起きていた。また、else if(条件文) を else(条件文) にしていた。この学生は、教員のほうにエラーが分からないという質問をしている。修正する過程で<?php を<php にしただけだが、エラーを表示している部分とずれていたため起こってしまったものである。のちに修正したが、else if(条件文) を else(条件文) にしたままとなり、課題が未完成となってしまった。

事例 13

第 8.1.2 節の事例 14 と同様 week5/p04Exam.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 41 に示す。

ソースコード 41: 事例 13 の week5/p04Exam.php (bot 送信時)

```

1 <?php
2 class Exam {
3     static function index() {
4         ?>

```

```

5      <h1>名前入力画面</h1>
6      <form action="p04Exam.php">
7      <input type="hidden" name="action" value="show_math_form"/>
8      名前: <input name="name"/>
9      <input type="submit" value="Next"/>
10     </form>
11     <?php
12 }
13 static function show_math_form() {
14     session_start();
15     $name=$_SESSION["name"];
16     $_SESSION["name"]=$name;
17     ?>
18     <h1>成績入力</h1>
19     名前: <?= $name ?>
20     <form action="p04Exam.php">
21     <input type="hidden" name="action" value="show_english_form"/>
22     数学の成績: <input name="math"/>
23     <input type="submit" value="Next"/>
24     </form>
25     <?php
26 }
27
28 static function show_english_form() {
29     session_start();
30     $name=$_SESSION["name"];
31     $_GET["name"]=$name;
32     ?>
33     <h1>成績入力</h1>
34     名前: <?= $name ?>
35     <form action="p04Exam.php">
36     <input type="hidden" name="action" value="show_result"/>
37     英語の成績: <input name="english"/>
38     <input type="submit" value="Next"/>
39     </form>
40     <?php
41 }
42
43 static function show_result() {
44     session_start();
45     $name=$_SESSION["name"];
46     $math=$_GET["math"];
47     $english=$_GET["english"];
48     ?>
49     <?= $name ?>の数学の成績は<?= $math ?>,英語の成績は<?= $english ?>です。
50     <?php
51 }
52 }
53 require "controller.php";
54 run_controller("Exam");
55 ?>

```

ソースコード 41 は、bot 送信時の week5/p04Exam.php である。show_english_form() で数学の点数をセッション変数に入れていなかったためにエラーが出た。

事例 24

第 8.1.3 節の事例 33 と同様 week12/p01fight.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 42 に示す。

ソースコード 42: 事例 24 の week12/p01fight.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Fight {
5     static function index() {

```

```

6      session_start();
7      $userName=$_SESSION["name"];
8      $results=pdo_select("select_*_from_chara_where_owner=?", $userName);
9      ?>
10     <h1>出場させるキャラクターを選択</h1>
11     <?php
12     foreach ($results as $result) {
13         ?>
14         <?= $result->name ?>
15         <a href="<?=basename(__FILE__)>?>?action=select_enemy&id=<?=$result->id_?>
16             ">
17             選択
18         </a><br/>
19     }
20 }
21 }
22 run_controller("Fight");
23 ?>

```

ソースコード 42 は、bot 送信時の week12/p01fight.php である。プログラムはあっているが、ログインせずにいたためエラーが未解決であった。

事例 25

week10/p03login.php でつまづいていた学生である。week10/p03login.php は、ユーザ名とパスワードを入力してログインしたかを表示させるプログラムである。bot 送信時のソースコードをソースコード 43 に示す。学籍番号は GAKUSEKI と置換している。

ソースコード 43: 事例 25 の week10/p03login.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Login {
5      static function index() {
6          ?>
7          <form action="p03login.php">
8              <input type="hidden" name="action" value="check"/>
9              ユーザ名<input name="name"/><br/>
10             パスワード<input type="password" name="password"/>
11             <input type="submit" value="Login"/>
12         </form>
13         <?php
14     }
15     static function index() {
16         $name="GAKUSEKI"; // ここは自分の番号にすること
17         $records=pdo_select("select_*_from_tweet_where_name=?_order_by_time_desc_limit_5", $name);
18         foreach($records as $record) {
19             print_r($record);
20             print "<BR>\n";
21         }
22         ?>
23         <form action="p07write.php">
24             <input type="hidden" name="action" value="write_complete"/>
25             <input name="write"/> <input type="submit" value="Write"/>
26         </form>
27         <?php
28     }
29     static function write_complete() {
30         $name="GAKUSEKI"; // ここは自分の番号にすること
31         $content="ログインしました。";
32         pdo_exec("insert_into_tweet(time,_name,_content)_values(now(),?,?)", $name, $content );
33         ?>
34         <div>メインページ</div>

```



```

35         <a href="p03login.php">トップへ</a>
36     <?php
37     }
38 }
39 run_controller("Login");
40 ?>

```

ソースコード 43 は、bot 送信時の week10/p03login.php である。index() が 2 つあったことからエラーが起きていた。

修正後のソースコードをソースコード 44 に示す。

ソースコード 44: 事例 25 の week10/p03login.php (修正後)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class Login {
5      static function index() {
6          ?>
7          <form action="p03login.php">
8              <input type="hidden" name="action" value="check"/>
9              ユーザ名<input name="name"/><br/>
10             パスワード<input type="password" name="password"/>
11             <input type="submit" value="Login"/>
12         </form>
13
14         <form action="p03login.php">
15             <input type="hidden" name="action" value="write_complete"/>
16             <input name="write"/> <input type="submit" value="Write"/>
17         </form>
18         <?php
19     }
20     static function write_complete() {
21         $name="GAKUSEKI"; // ここは自分の番号にすること
22         $content="ログインしました。";
23         pdo_exec("insert_into_tweet(time, name, content) values(now(),?,?)", $name,
24                 $content );
25         ?>
26         <div>メインページ</div>
27         <a href="p03login.php">トップへ</a>
28         <?php
29     }
30 }
31 run_controller("Login");
32 ?>

```

ソースコード 44 は、修正後の week10/p03login.php である。8 行目の value が check となっているが、check() がないためエラーが出ている。この学生はソースコード 44 以降の書き換えはなく、課題を完成させないまま終わらせている。

事例 28

第 8.1.3 節の事例 21 と同様 week11/p04makechar.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 45 に示す。

ソースコード 45: 事例 28 の week11/p04makechar.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>

```

```

10     ユーザ名 <?= $userName ?>
11     <form action="p04makechar.php">
12     名前<input type="hidden" name="action" value="complete"/>
13     <input name="displayname"/>
14     <input type="submit" value="作成"/>
15     </form>
16     <?php
17
18     }
19     static function complete() {
20         session_start();
21         ?>
22         <h1>作成完了</h1>
23         <?php
24         $userName=$_SESSION["name"];
25         $gold=$SESSION["gold"];
26         $afgold=$gold-100;
27         $results=pdo_select("select * from user where name=?", $userName);
28         foreach ($results as $result) {
29             print "ユーザ名: $userName<br/>";
30             print "キャラクタ名: $result->displayname<br/>";
31             print "所持金: $gold<br/>";
32             print "所持金: $afgold";
33         }
34         $gold=$afgold;
35     }
36 }
37 run_controller("MakeChar");
38 ?>

```

ソースコード 45 は、bot 送信時の week11/p04makechar.php である。25 行目の\$_SESSION を\$SESSION にしてしまったことでエラーが起きていた。

修正後のソースコードをソースコード 46 に示す。

ソースコード 46: 事例 28 の week11/p04makechar.php (修正後)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>
10         ユーザ名 <?= $userName ?>
11         <form action="p04makechar.php">
12         名前<input type="hidden" name="action" value="complete"/>
13         <input name="char_name"/>
14         <input type="submit" value="作成"/>
15         </form>
16         <?php
17
18     }
19     static function complete() {
20         session_start();
21         ?>
22         <h1>作成完了</h1>
23         <?php
24         $userName=$_SESSION["name"];
25         $charname=$_GET["char_name"];
26         $gold = $record->gold;
27         $afgold=$gold-100;
28         $results=pdo_select("select * from user where name=?", $userName);
29         foreach ($results as $result) {
30             print "ユーザ名: $userName<br/>";

```

```

31         print "キャラクタ名:␣$charname<br/>";
32         print "所持金 (作成前):␣$gold<br/>";
33         print "所持金 (作成後):␣$afgold";
34     }
35 }
36 }
37 run_controller("MakeChar");
38 ?>

```

ソースコード 46 は、修正後の week11/p04makechar.php である。complete() の所持金を入れておく \$gold に変数を取できないことからエラーが起きている。\$userName を抽出してから所持金を取得しないといけないが、\$record->gold にしてしまっている。おそらく \$record に userName を抽出して所持金情報を取得しようとしたと考えられる。この学生はソースコード 46 以降の書き換えはなく、課題を完成させないまま終わっている。

事例 29

第 8.1.3 節の事例 21 と同様 week11/p04makechar.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 47 に示す。

ソースコード 47: 事例 29 の week11/p04makechar.php (bot 送信時)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");
4  class MakeChar {
5      static function index() {
6          session_start();
7          $userName=$_SESSION["name"];
8          ?>
9          <h1>キャラクタの作成</h1>
10         <form action="p04makechar.php">
11             ユーザ名 <?= $userName ?>
12             <input type="hidden" name="action" value="complete"/>
13             名前: <input name="name">
14             <input type="submit" value="作成"/>
15         </form>
16         <?php
17     }
18     static function complete() {
19         session_start();
20         $userName=$_SESSION["name"];
21         $name=$_GET["name"];
22         $gold=$_SESSION["gold"];
23         $records=pdo_select("select␣*␣from␣user␣where␣gold=?", $gold);
24         $result=$results[0];
25         ?>
26         <h1>作成完了</h1>
27         ユーザ名 <?= $userName ?>
28         キャラクタ名 <?= $name?>
29         所持金 <?= $result?>
30         <?php
31     }
32 }
33 run_controller("MakeChar");
34 ?>

```

ソースコード 47 は、bot 送信時の week11/p04makechar.php である。20 行目で name を所得したのに 21 行目でも name を取得したためエラーが起きていた。おそらく 21 行目をキャラクタの名前を取得しようとして失敗したものと考えられる。

その後試行錯誤してプログラムを書き換えていた。修正後のプログラムをソースコード 48 に示す。

ソースコード 48: 事例 29 の week11/p04makechar.php (修正後)

```

1  <?php
2  require("pdo.php");
3  require("controller.php");

```

```

4 class MakeChar {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         ?>
9         <h1>キャラクタの作成</h1>
10        <form action="p04makechar.php">
11        ユーザ名 <?= $userName ?>
12        <input type="hidden" name="action" value="complete"/>
13        名前: <input name="name">
14        <input type="submit" value="作成"/>
15        </form>
16        <?php
17    }
18    static function complete() {
19        session_start();
20        $userName=$_SESSION["name"];
21        $name=$_GET["name"];
22        $gold=pdo_exec("update_user_set_gold=?where_name=?", $gold, $name);
23
24        ?>
25        <h1>作成完了</h1>
26        ユーザ名 <?= $userName ?>
27        キャラクタ名 <?= $name?>
28        所持金 <?= $gold ?>
29        <?php
30    }
31 }
32 run_controller("MakeChar");
33 ?>

```

ソースコード 48 は、修正後の week11/p04makechar.php である。22 行目の SQL 文がよくわかっていないように見られる。

事例 31

第 8.1.3 節の事例 36 と同様 week14/training.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 49 に示す。

ソースコード 49: 事例 31 の week14/training.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class Training {
5     static function index() {
6         session_start();
7         // 「戦闘」のプレイヤーキャラクタ選択の画面を参考に、
8         // 現在ログインしているユーザが所持しているキャラクター一覧を$results に入れる
9         $userName=$_SESSION["name"];
10        $results=pdo_select("select_*_from_chara_where_owner=?", $userName);
11        ?>
12        <h1>トレーニングするキャラクターを選択</h1>
13        <?php
14        foreach ($results as $result) {
15            ?>
16            <?= $result->name ?>
17            <a href="<?=basename(__FILE__)>?>?action=select_menu&id=<?= $result->id_?>"
18            >
19            選択
20            </a><br/>
21            <?php
22        }
23    }
24    static function select_menu() {
25        session_start();
26        // 選択したキャラクタのIDをセッション変数に保存する

```

```

26     $chara_id=$_GET["id"];
27     $_SESSION["player_chara_id"]=$chara_id;
28
29     // 選択したキャラクタの情報を取得して,$charaに入れる(戦闘の「vs」を参考に)
30     $userName=$_SESSION["name"];
31     $results=pdo_select("select_*_from_chara_where_id=?", $userName);
32     $chara=$results[0];
33     //-----
34     ?>
35     <h1>トレーニングメニュー選択</h1>
36
37     名前:<?= $chara->name ?>,
38     HP: <?= $chara->hp ?> ,
39     AP: <?= $chara->ap ?> ,
40     SP: <?= $chara->sp ?>
41     <br/>
42
43     <a href="<?=basename(__FILE__)>?>?action=start&type=hp">HP を上げる</a><br/>
44
45     <!-- AP や SP を上げるメニューも作ってみよう -->
46     <?php
47 }
48 static function start() {
49     session_start();
50     // 現在ログインしているユーザの所持金を読み出して
51     // $goldに入れる(メインページを参考に)
52
53     $gold_after=$gold-100;
54     if ($gold_after<0) {
55         ?>Gold が足りません<?php
56     } else {
57         //足りていれば, 所持金を 100減らす(「キャラクタを作る」などを参考に)
58
59         // select_menu で保存したキャラクタの ID のセッション変数を $chara_id に読み出す
60
61         // 選択したキャラクタの情報を取得して,$charaに入れる
62
63         ?><h1>トレーニング完了</h1>
64         <?= $chara->name?>の
65         <?php
66         $hp=$chara->hp+random_int(1,3);
67         pdo_exec("update_chara_set_hp=?_where_id=?", $hp, $chara_id);
68         ?> HP が<?= $hp ?> になった. <?php
69     }
70 }
71 }
72 run_controller("Training");
73
74
75 ?>

```

ソースコード 49 は, bot 送信時の week14/training.php である. コメント付近に何も書かれていないことからどのように記述するかが分からなかったことと推測される.

事例 32

week9/p07write.php でつまづいていた学生である. 9 週目は, ユーザ登録をすることを目標に授業が進められた. ユーザ情報として, ユーザ名, パスワード, 他ユーザから見ると表示される表示名, アイコンの情報が収納されている. week9/p07write.php は, 書き込みを行った際に表示名を表示させるプログラムである. bot 送信時のソースコードをソースコード 50 に示す.

ソースコード 50: 事例 32 の week9/p07write.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class WriteTweet {

```

```

5     static function index() {
6         session_start();
7         $name=$_SESSION["name"];
8         $results=pdo_select("select_*_from_user_where_name=?", $name);
9         $result=$results[0];
10        $records=pdo_select("select_*_from_tweet_where_name=?_order_by_time_desc_limit_5
        ", $name);
11        foreach($records as $record) {
12            print $result->displayname;
13            print "さんの発言：";
14            print $record->content;
15            print "_at_";
16            print $record->time;
17            print "<BR>\n";
18        }
19
20        ??
21        <form action="p07write.php">
22        <input type="hidden" name="action" value="write_complete"/>
23        <input name="content"/>
24        <input type="submit" value="Write"/>
25        </form>
26        <?php
27    }
28 }
29 run_controller("WriteTweet");
30 ??

```

ソースコード 50 は、bot 送信時の week9/p07write.php である。22 行目に write_complete と書いてあるが、write_complete() がないことでエラーが起きていた。DM を送信後、Slack のリアクションを付けていたので、送信内容は一読していると思われる。

事例 34

第 8.1.3 節の事例 21 と同様 week11/p04makechar.php でつまづいていた学生である。bot 送信時のソースコードをソースコード 51 に示す。

ソースコード 51: 事例 34 の week11/p04makechar.php (bot 送信時)

```

1 <?php
2 require("pdo.php");
3 require("controller.php");
4 class MakeChar {
5     static function index() {
6         session_start();
7         $userName=$_SESSION["name"];
8         ??
9         <h1>キャラクタの作成</h1>
10        <form action="p04makechar.php">
11        ユーザ名 <?= $userName ?>
12        <input type="hidden" name="action" value="complete"/>
13        名前: <input name="name">
14        <input type="submit" value="作成"/>
15        </form>
16        <?php
17    }
18    static function complete() {
19        session_start();
20        $userName=$_SESSION["name"];
21        $name=$_SESSION["displayname"];
22        $results=pdo_select("select_*_from_chara_where_owner=?", $name);
23        $result=$results[0];
24        ??
25        <h1>作成完了</h1>
26        ユーザ名 <?= $result->userName?><br>
27        キャラクタ名 <?= $result->name?><br>
28        金額<?= $result->gold?><br>

```

```

29 |         <?php
30 |     }
31 | }
32 | run_controller("MakeChar");
33 | ?>

```

ソースコード 51 は、bot 送信時の week11/p04makechar.php である。キャラクタ名を表示名から取得しようとしてエラーが起きていた。

9 アンケート結果

『ウェブプログラミング』の受講者にアンケートを取った。回答していただいたのは 22 名であった。質問項目は以下のような項目であった。

- 授業内で教員から DM が届きましたか?
- DM についてどのように思いましたか?
- DM をもらったことで解決した問題がありましたか?
- DM をもらったときにはすでに自力で問題が解決していたことがありましたか?
- DM をもらっても問題が解決しなかったことがありましたか?
- DM がどのように参考になったか?
- つまづいている際の対応の仕方や改善すべきところがあれば教えてください。

授業内で教員から DM が届きましたか?

この質問の回答として、「はい」が 60.7%、「いいえ」が 39.3%であった。この結果より回答者の 6 割に届いたことになる。

DM についてどのように思いましたか?

この質問は 4 段階評価で回答していただいた。この質問の結果を表 5 に示す。

表 5 より参考になった人が多いことが分かった。

DM をもらったことで解決した問題がありましたか?

この質問についても 4 段階評価で回答していただいた。この質問の結果を表 6 に示す。

表 6 より解決した人が多いことが分かった。

DM をもらったときにはすでに自力で問題が解決していたことがありましたか?

この質問についても 4 段階評価で回答していただいた。この質問の結果を表 7 に示す。

表 7 より自力解決した人が多いことが分かった。それと同時に DM が遅いということが分かった。

DM をもらっても問題が解決しなかったことがありましたか?

この質問は 4 段階評価で回答していただいた。この質問の結果を表 8 に示す。

表 8 より解決した人が多いことが分かった。

DM がどのように参考になったか?

この質問の回答は以下のようになった。

- どういう風にエラーが出るのか分かった
- 認識が曖昧だったところが少しわかるようになった
- 細かいミスに気づくことができた。
- 60%
- ヒントになるところを教えてくれるところ。質問のきっかけになるところ。

- 既に解決済みだったが、悩んでいた部分について適切なヒントが送られてきたため、きちんと見ているんだなと思った。
- 質問をしなければ分からないのかどうかの区別がついた。
- 「もう少しでできそうだからまだ聞かないでおこう（全くわからない）」と考えているとき、とても参考になった。
- 参考にすべきプログラムがわかった
- 詳しいヒントがなかったためほぼ自力でした
- パスワードの更新は `index()` に書くのではなく、新しく `update_password` という関数を作ってください（`static function update_displayname()` というようにどこがダメでどのように修正すべきかがわかるもの）はとも分かります
- `bit arrow` のヒントに書かれていた内容と殆ど同等の物だったためあまり参考になりませんでした。
- エラーの原因が分かった
- ヒント機能を見ることを思い出す
- ほとんどが `BitArrow` のヒントに記載されていたものと同一であったため、問題解決に直接作用したわけではないですが、考えを整理するのに使わせていただきました。
- いつでも質問できる環境が整っていると安心ができたこと
- 参考にならなかった

回答を見てみるとエラーの原因が分かった学生やどの部分が誤っているかが分かったという回答が得られた。一部ヒントに対しての回答があったが、これは学部生の卒業研究なので今回は無効回答としている。

つまりしている際の対応の仕方や改善すべきところがあれば教えてください。

この質問の回答は以下ようになった。

- 特になし
- `zoom` のブレイクアウトルームで画面共有をすれば文章にしなくていいので楽かなと思います。
- プログラムの全体の設計図があればわかりやすく、エラーの修正もしやすいかも
- ヒントがもう少しわかりやすくしてほしい
- 自分から声をかけることがなかなかできないので、困ってる時に声をかけてくれるのはとてもありがたいです。
- 正解率の低い問題を解説してほしい
- 特にないです。
- つまりいた時は直ぐに質問していました。
- なし
- 答えでなくてももう少し具体的なヒントがほしい。
- 質問にさえ答えていただければ大丈夫です。
- ないです。とても助かってます。
- 全体での指示を増やしたほうがいいと感じました。

- 巡回や採点の頻度が高くなると嬉しいです。例えばある問まで出来たとしてその次の問題で引っかかったときそれは次の問題がわかっていないだけなのか、それともある問の時点でエラー自体は出ていないが問題の趣旨に合っていないなどして修正しなければならないものがあるのかがわかると思います。
- bitarrow に書かれているヒントよりもう一步踏み込んだヒントがあれば良いと思います。
- 悩んでいる時間が表示されるといいと思った。
- 今までの対応の仕方で問題ありません。
- つまづく場面が無かったので分かりません。
- 具体的なヒントがほしかった
- BitArrow に書いてあるヒント以外の助言をいただけると助かります。
- 特にないです
- ヒントに記載されていないヒントが欲しい

回答を見ていると特にないという回答が多いので、このままの方法でよいと考えている。こちらでもヒントに対しての回答があったが、これは学部生の卒業研究なので今回は無効回答としている。

10 考察

今回の実験では、DM を送ることをつまづきを解決することにつなげることができた事例が確認された。また、DM を送ることに対して肯定的な意見を多く得ることができた。今回は過剰な DM が送られることを想定して教員と TA による手動送信を行ったが、DM の送信を完全に自動化しても問題なくつまづきの解決を支援できると考えられる。また、こちらから DM を送信した後に質問するようになった人が増えたことから、積極的に DM を用いた働きかけが有効であり、その内容についてはある程度自動生成した文章でも問題ないことも確認できた。これらの実験を通じて、学生が積極的に質問する機会が得ることができたと考えている。ただ、返答もなく課題もできていない学生も見受けられた。その学生に対して追加課題を出題したり、周りが教えられるような環境を作るなど他の部分で改善が必要と考えた。今回エラーが発生し、実働時間が 900 秒以上の学生に DM を送ったが、課題が終わっていない人のログを見てみると条件には満たしていないがつまづいていそうな学生がいた。ただ、実働時間が短い学生や保存が多い学生が見られたが、実働時間が短い学生に指導したところで考える機会がなくなってしまううえ、保存が多い学生に関しては、実行結果を確認する癖がつかなくなってしまう恐れがあるため現行のシステムのままだと良いと考えている。また、本実験は試行錯誤を行った学生をサポートすることを目的としているため今回は対象外としているが、手が止まっている学生に関しては、例えば BitArrow には最後に捜査した時刻に基づいて、何分間操作がないか、ということを知る機能もあり、これらの機能と組み合わせることにより検出できると思われる。

また、BitArrow のログに課題を取り組んだ形跡があったが、課題が不合格の学生がいた。その学生達のログを見ると実働時間が少ない学生が多く見られた。この学生たちは、先ほど述べたように考える機会がなくなるのでサポートしない方が望ましいと考えている。今回の実験では、授業時間内でのエラーが未解決の学生からつまづきを見つけ出しているが、中には授業時間外にエラーを複数回起こしている学生がいた。当該の学生は、授業実施時に未解決の部分の質問がなかった。授業時間外については bot 機能は動作していないので、つまづきを見つけることは難しいが、今後改善の余地はあると考える。

今回の実験は、第 6 章の指標を検討したときは、エラーに関係なく経過時間が 900 秒以上であればつまづいている可能性があるとして結論付けた。それでもエラーのある学生に絞ったのは、実際に運用を行うにあたって、過度に Bot が動作してしまうのを防ぐためであった。試しに実働時間が 900 秒以上の学生でエラー以外のログまで含めたものを配信を行った。配信間隔は今回の実験と同様 5 分間隔で行った。エラー以外のログも含めた場合、BitArrow にある全ログを参照するため、Slack に配信されるまでに時間がかかってしまった。この状態だとサポートが遅れてしまう恐れがあり、つまづいている学生を救うことができないためこの方法を不適切と考える。また、配信されたもののログを閲覧すると長い期間を開けてファイルを開いただけのものや手順が多い課題だと手順が多い分ロ

グが増え、時間がかかっただけのものが多かった。これらの結果からエラー以外のログまで含めたものを対象にすると配信が遅れるうえ、つまづいた学生を見つけるのに時間がかかるといったデメリットがあったため今回はエラーのみで抽出を行った。今後の課題としてプログラムを編集している実質的な時間をどう正確に測るかを検討する必要がある。現状はファイルを開いたり実行したりするのも含まれているため、正確に測ることが難しい。計測方法は、今後考えていく必要がある課題と考えている。また、課題の内容によって「900 秒」というのは固定でよかったのかどうかも検討する必要があると考えている。

11 まとめ

本研究は、BitArrow のエラーを Slack から教員や TA に自動的に通知する仕組みを実装した。通知されたエラーからつまづきを目視で確認し、学習者に直接指導を行った。実際に行った指導を調査したところ、自分から質問をほとんどしない学生に対して必要な指導を行っていた事例が見つかった。本システムを用いることで、これまで把握が難しかったつまづいている学生に対するサポートができるようになったと考えられる。また、質問をほとんどしない学生にも自発的に質問を行うようになったと考えられる。今後この実験により学生のつまづき傾向を見つけ出し、直接学生につまづいた部分の自動配信が実現できる可能性を見つけ出し、教員が目視でつまづきを確認する負担を軽減することを目指す。

謝辞

本研究に関して終始ご指導ご鞭撻をいただきました本学 長慎也教授、中島由美教授、山中脩也准教授、東京農工大学 長島和平先輩、また、つまづきの検出方法などに助言をいただきました東京農工大学 並木美太郎教授、大阪電気通信大学 兼宗進教授、心より感謝の気持ちとお礼を申し上げます。

参考文献

- [1] 浦上理, 長島和平, 並木美太郎, 兼宗進, 長慎也: プログラミング学習者のつまづきの自動検出, 情報処理学会研究報告コンピュータと教育 (CE), 2020-CE-154, Vol.4, pp.1-8(2020)
- [2] 長島和平他: Web ブラウザを用いたプログラミング学習支援環境 Bit Arrow の設計と評価, 情報処理学会論文誌教育とコンピュータ, Vol. 4, No. 1, pp. 57-69 (2018).
- [3] 内藤広志他: プログラム演習のための進捗モニタリングシステム, 情報処理学会研究報告, 2008-CE-93, pp. 33-40(2008)
- [4] 長谷川伸他: プログラミング入門教育を対象としたリアルタイム授業支援システム, 情報処理学会論文誌, Vol. 52, No.12, pp. 3135-3149(2011)
- [5] 西村智治他: 小コンテスト形式の初級 C 演習における教師支援, 情報処理学会研究報告, Vol.2013-CE-119, pp. 1-8(2013)
- [6] 太田翔也他: プログラミング演習における補助者の巡回指導のためのタブレット PC 上の支援ツール, 情報処理学会研究報告, Vol.2016-CE-137, pp. 1-8(2016)
- [7] 井垣宏他: プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, Vol. 54, No.1, pp. 330-339(2013)
- [8] Cindy Norris, Frank Barry, James B. Fenwick Jr, Kathryn Reid and Josh Rountree: ClockIt: Collecting Quantitative Data on How Beginning Software Developers Really Work, ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education, June 2008, Pages 37-41, <https://doi.org/10.1145/1384271.1384284>

- [9] 市村哲他：プログラミング演習授業における学習状況把握支援の試み, 情報処理学会論文誌, Vol. 54, No.12, pp. 2518-2527(2013)
- [10] 加藤利康他：プログラミング演習授業のための授業支援システムにおける学習状況把握機能の実現, 情報処理学会論文誌, Vol. 55, No.8, pp. 1918-1930(2014)
- [11] 市石舜也, プログラミングの授業中の課題におけるヒント提示システム, 明星大学情報学部 2021 年度卒業論文 (提出予定)

No.	かかった総時間	増加	平坦	減少	つまづき有無	備考
1	1093(s)	11	0	0	なし	
3	1116(s)	9	2	0	なし	
4	1641(s)	13	3	0	なし	300 秒間別課題に取り組んでいる.
5	1055(s)	8	2	0	なし	
6	1084(s)	7	2	0	なし	
7	1038(s)	14	4	0	なし	保存のみで実行なし
11	1011(s)	5	2	0	なし	
12	1512(s)	9	4	0	なし	500 秒間別課題に取り組んでいる.
13	1205(s)	12	6	0	なし	
14	1701(s)	10	0	5	なし	600 秒間別課題に取り組んでいる.
16	1101(s)	9	5	0	なし	
17	1150(s)	16	9	0	なし	300 秒間別課題に取り組んでいる.
19	922(s)	12	7	0	なし	
21	1526(s)	7	5	0	なし	書き換え始めるのは 921 秒
22	1220(s)	9	4	1	なし	
23	989(s)	4	3	0	なし	
24	1218(s)	20	10	2	なし	
26	996(s)	18	6	3	なし	
27	941(s)	5	4	0	なし	
28	1031(s)	10	8	0	なし	
29	1017(s)	6	5	0	なし	
30	1004(s)	6	5	0	なし	
32	1410(s)	7	6	0	なし	
33	1350(s)	10	10	0	なし	
34	971(s)	4	1	1	なし	
35	1065(s)	12	6	2	なし	
36	999(s)	3	3	0	あり	エラーで終了
37	1207(s)	10	7	1	なし	
38	965(s)	3	3	0	なし	
39	1809(s)	10	7	1	なし	はじめファイルを開いただけで実質 1100 秒で完成
40	1081(s)	10	4	2	なし	
55	1231(s)	18	23	0	なし	はじめいろんなファイルを開いていた
59	1262(s)	9	12	0	なし	
63	2717(s)	23	21	3	なし	授業内 (400s) と授業時間外に取り組んでいた
64	1262(s)	9	9	1	なし	
65	1358(s)	9	9	1	なし	
67	1283(s)	11	12	1	なし	794s で完成
77	1329(s)	9	14	0	なし	
78	1441(s)	16	18	2	なし	
80	1513(s)	13	17	1	なし	授業前に取り組んでいた
85	1171(s)	8	13	0	なし	
93	1275(s)	18	16	4	なし	
95	1379(s)	12	17	1	なし	

表 2: 正解行数の変化をグラフ化した結果が単調増加の学習者の傾向

週	内容
1	PHP の基礎
2	PHP の基礎
3	PHP のオブジェクト指向
4	フォームと値の受け渡し
5	セッション
6	データベース
7	データベース 2:追加
8	ユーザ認証
9	ユーザ登録
10	対戦ゲーム：ログインとメインページ
11	対戦ゲーム：キャラクタを作る
12	対戦ゲーム：戦闘する
13	戦闘の改造
14	トレーニングする

表 3: 講義内容

事例	掛かった 総時間 (s)	エラーを 起こした時刻	エラーを 起こした時間 (s)	bot 送信時刻	DM 送信時刻	解決時刻	学生の反応
1	1760	2021/09/21 11:30:09	1111	9/21 11:32	11:38	9/21 11:38 11:36(SA)	S
2	2826	2021/09/21 11:53:45	2298	9/21 11:55	12:00	9/21 12:02:34	H
3	2191	2021/9/21 12:03:22	1434	9/21 12:08	12:18	未解決	H
4	1565	2021/10/5 11:34:33	965	10/05 11:39	11:50	未解決	なし
5	4278	2021/10/12 11:05:04	840	10/12 11:11	11:12	10/12 12:02:22	なし
6	2028	2021/10/12 11:04:48	1702	10/12 11:12	11:17	10/12 11:10:15	なし
7	2336	2021/10/12 11:20:07	1398	10/12 11:29	(送信なし)	10/12 11:35:49	なし
8	1952	2021/10/12 11:28:40	1108	10/12 11:39	11:39	10/19 15:38:41	なし
9	2202	2021/10/19 11:16:33	1148	10/19 11:20	11:23	10/19 11:34:36	T
10	1809	2021/10/19 11:37:35	1076	10/19 11:43	11:44	10/19 11:51:32	T
11	2473	2021/10/19 12:12:16	2370	10/19 12:17	12:19	未解決	なし
12	8568	2021/10/26 11:35:24	2340	10/26 11:38	11:44	未解決	T
13	8179	2021/10/26 11:33:53	2589	10/26 11:41	11:43	未解決	なし
14	4933	2021/10/26 11:59:22	3725	10/26 12:07	12:08	10/26 12:19:30	H
15	7814	2021/11/23 11:30:13	2173	11/23 11:38	11:43	12/7 23:03:58	E
16	2934	2021/11/23 11:27:42	1148	11/23 11:41	11:51	11/23 11:57:03	H
17	3507	2021/11/23 11:40:13	1548	11/23 11:44	11:57	11/23 12:12:51	なし
18	4504	2021/11/23 11:51:28	2106	11/23 12:05	12:07	12/5 2:12:56	なし
19	5927	2021/11/23 11:59:34	1741	11/23 12:11	12:21	11/24 21:08:34	T
20	1890	2021/11/23 12:09:36	2635	11/23 12:18	12:22	12/7 11:47:53	H
21	3306	2021/11/30 11:18:57	1181	11/30 11:22	11:22	11/30 11:58:19	なし
22	5321	2021/11/30 11:13:15	905	11/30 11:22	11:22	11/30 20:05:39	なし
23	4320	2021/12/7 11:20:23	1039	12/07 11:22	11:25	12/7 12:28:47	なし
24	1463	2021/12/7 11:20:10	1389	12/07 11:22	11:25	未解決	なし
25	2430	2021/12/7 11:55:21	1871	12/07 11:58	12:03	未解決	なし
26	3135	2021/12/7 11:54:51	2410	12/07 11:59	(送信なし)	12/7 12:11:00	Q
27	10240	2021/12/7 12:08:30	1320	12/07 12:14	12:14	12/14 1:57:28	なし
28	8688	2021/12/14 11:28:24	2656	12/14 11:33	11:34	未解決	なし
29	3096	2021/12/14 11:45:59	1476	12/14 11:55	11:56	未解決	なし
30	3178	2022/1/11 11:03:50	904	01/11 11:08	11:10	1/11 12:05:53	なし
31	4663	2022/1/11 11:15:44	1343	01/11 11:19	11:19	未解決	なし
32	5879	2022/1/11 11:14:41	2289	01/11 11:20	11:21	未解決	なし
33	3110	2022/1/11 11:50:56	2696	01/11 11:54	11:55	1/11 11:57:50	なし
34	5175	2022/1/11 11:55:48	4035	01/11 11:59	11:59	未解決	なし
35	2934	2022/1/11 11:55:56	1204	01/11 12:00	12:01	1/11 12:16:07	なし
36	3914	2022/1/11 12:01:01	3309	01/11 12:04	12:04	1/11 12:11:07	なし

表 4: つまづいた受講者の課題取り組み時間

あまり参考にならなかった	1	2	3	4	大変参考になった
	11.8%	23.5%	35.3%	29.4%	

表 5: 質問結果

ほとんどなかった	1	2	3	4	かなりあった
	11.8%	23.5%	29.4%	35.3%	

表 6: 質問結果

ほとんどなかった	1	2	3	4	かなりあった
	5.9%	11.8%	64.7%	17.6%	

表 7: 質問結果

ほとんどなかった	1	2	3	4	かなりあった
	17.6%	47.1%	29.4%	5.9%	

表 8: 質問結果