

ПРИМЕНЕНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ В ЛАБОРАТОРНОМ ПРАКТИКУМЕ

П.А. Хорошевич

преподаватель Белорусского государственного педагогического университета имени Максима Танка, г. Минск, Беларусь, khoroshevich.pa@gmail.com

Аннотация: В статье рассматриваются методические и организационные вопросы внедрения системы автоматизированной проверки задач по программированию, разбираемых в лабораторном практикуме. Показаны особенности работы и методика применения Виртуальной лаборатории программирования (VPL), являющейся расширением для распространённой системы дистанционного обучения Moodle. Рассмотрены достоинства и недостатки применения автоматизированной системы проверки задач по программированию.

Ключевые слова: обучение программированию, автоматическая проверка заданий, автоматизированные тесты, moodle.

APPLICATION OF THE SYSTEM OF AUTOMATIC TESTING OF PROGRAMMING TASKS IN THE LABORATORY PRACTICE

Pavel A. Khoroshevich

lecturer, Belarusian State Pedagogical University named after Maxim Tank, Minsk, Belarus, khoroshevich.pa@gmail.com

Abstract: The article describes the methodological and organizational issues of the implementation of the automated testing system for programming tasks, which are considered in laboratory work. The features of the work and the methodology of using the Virtual Programming Laboratory (VPL), which is an extension to the widespread distance learning system Moodle, are shown. The advantages and disadvantages of using an automated system for checking programming tasks are considered.

Keywords: teaching programming, automated program checking, automated tests, moodle.

Решение задач является неотъемлемой частью процесса обучения программированию. Учебная дисциплина может содержать большое количество практических заданий. Проверка решений учащихся, их оценка, управление файлами решений может стать трудоёмкой задачей для

преподавателя. Существует несколько вариантов решения вышеперечисленных проблем: уменьшение количества практических заданий, выборочная оценка решённых учащимися задач или использование систем автоматической оценки. Как показывает практика [1], использование последнего варианта показало свою эффективность и не обладает недостатками, присущими другим методам.

Существующие системы автоматизированной проверки задач по программированию, чаще всего, применяются в олимпиадном программировании [2][3]. Зачастую подобные системы представляют собой программные комплексы, установка и настройка которых происходит независимо от существующих систем обучения, например, таких как СДО Moodle. Плагин Virtual Programming Lab [4] (Виртуальная лаборатория программирования) добавляет в СДО Moodle возможность автоматической проверки задач, управление файлами решений, отправленными учащимися, а также отчёт о существующих заимствованиях решений.

Плагин добавляет элемент курса Виртуальная лаборатория программирования.

Учащийся может ознакомиться с условием задачи, отправить готовое решение на проверку, отредактировать исходный код решения прямо в браузере с помощью облачной IDE (Рисунок 1). После отправки решения учащийся, в зависимости от настроек элемента курса, может либо самостоятельно запустить проверку задания, либо сообщить преподавателю о необходимости запуска автоматических тестов.

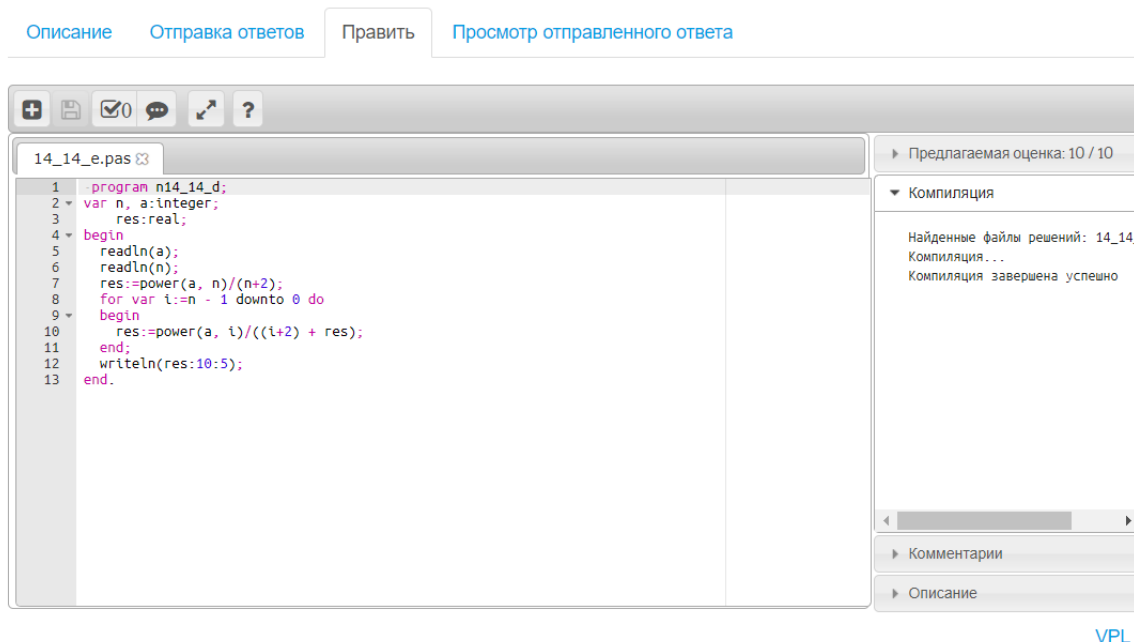


Рисунок 1 - Облачная IDE

Интерфейс элемента курса отличается для преподавателя. Закладка *Список отправленных ответов* содержит информацию обо всех

отправленных файлах решений, позволяет запустить автоматическую проверку всех задач (Рисунок 2). Щелчок, по предлагаемой оценке, выводит результат проверки в отдельном окне.

Описание Списание отправленных ответов Сходство Тест модуля

Изолированные группы Все участники

Submission selection
Все ответы

Протестировать
Выберите...

	Имя / Фамилия	Отправлено в	Отправленные ответы	Оценка	Кто оценил	Протестировано
1	[User Icon]	Среда, 12 Май 2021, 13:14	2	Предлагаемая оценка: 10 / 10		
2	[User Icon]	Понедельник, 10 Май 2021, 19:46	3	Предлагаемая оценка: 10 / 10		
3	[User Icon]	Среда, 12 Май 2021, 13:08	2	Предлагаемая оценка: 10 / 10		

Рисунок 2 - Страница со списком отправленных решений

Вкладка *Сходство* содержит инструменты поиска заимствований решений (Рисунок 3). Отчёт содержит список пар пользователей, чьи решения схожи друг на друга. Степень схожести указывается по трём метрикам.

Описание Списание отправленных ответов Сходство Тест модуля

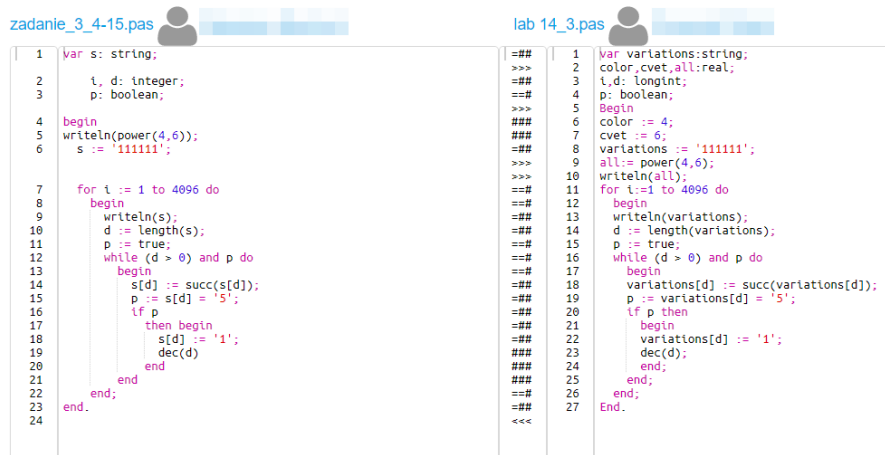
Сходство List of similarities found

#	Имя / Фамилия	Похоже на	Cluster #
1	2.pas	80 91 75*** lab 14_2.pas	1
2	2.pas	73 92 58*** zadan	1
3	2.pas	71 87 38*** Progre	1
4	Progre	60 81 43** zadan	1

Рисунок 3 - Отчёт проверки заимствований решений

Щелчок по метрике сходства открывает новое окно, в котором преподаватель может самостоятельно сравнить степень схожести решений (Рисунок 4). Как видно из рисунка, несмотря на переименование

переменных и расстановку учащимися пробелов и пустых строк, алгоритм определил наличие заимствования.



```
zadanie_3_4-15.pas
1 var s: string;
2   t, d: integer;
3   p: boolean;
4 begin
5   writeln(power(4,6));
6   s := '111111';
7
8   for i := 1 to 4096 do
9     begin
10      writeln(s);
11      d := length(s);
12      p := true;
13      while (d > 0) and p do
14        begin
15          s[d] := succ(s[d]);
16          p := s[d] = '5';
17          if p
18            then begin
19              s[d] := '1';
20              dec(d);
21            end
22          end;
23      end;
24 end.
```

```
lab 14_3.pas
1 var variattons:string;
2   color,cvet:all:real;
3   t,d: longint;
4   p: boolean;
5 begin
6   color := 4;
7   cvet := 6;
8   variattons := '1111111';
9   all:= power(4,6);
10  writeln(all);
11  for i:=1 to 4096 do
12    begin
13      writeln(variattons);
14      d := length(variattons);
15      p := true;
16      while (d > 0) and p do
17        begin
18          variattons[d] := succ(variattons[d]);
19          p := variattons[d] = '5';
20          if p then
21            begin
22              variattons[d] := '1';
23              dec(d);
24            end;
25          end;
26      end;
27  End.
```

Рисунок 4 – Сравнение похожих решений

Для проведения автоматического тестирования решения, преподавателю необходимо подготовить ряд тестовых случаев (кейсов), включающих в себя входные и ожидаемые выходные данных. После непосредственного запуска тестирования, исходный код решения отправляется на сервер выполнения, где в безопасной среде он компилируется или выполняется интерпретатором. По завершению этапа тестирования отчёт о его результатах отображается в браузере.

Элемент курса Виртуальная лаборатория программирования применялся нами при проведении лабораторных работ по учебной дисциплине «Практикум по решению задач по программированию». Исходя из опыта использования автоматизированной системы проверки задач, следует выделить ряд положительных моментов.

Ознакомление с новой системой сдачи лабораторных работ прошло быстро и не вызвало больших затруднений. Процесс проверки решений значительно ускорился. Для формирования тестовых кейсов специально выбирались такие данные, которые, могут вызвать ошибки или исключительные ситуации без должной проработки решения учащимся. Также стоит отметить, что общее количество тестовых кейсов было увеличено по сравнению с ручной проверкой решений. Время, ранее используемое для ввода вручную тестовых данных и проверку результата работы программ, теперь можно использовать для уточняющих вопросов, позволяющих удостовериться в понимании учащимися алгоритма решения поставленной задачи. В случае возникновения сомнений по поводу самостоятельности решения задачи, преподаватель может получить отчёт по заимствованиям в реальном времени. Страница отчёта покажет группу людей со схожим решением и степень схожести. При этом проверка производится не только в пределах одной академической группы, но по

всему курсу, что позволяет определить заимствования не только в рамках одной подгруппы.

Стоит обратить внимание на ряд трудностей, связанных с применением автоматической проверки задач.

Необходимо тщательно информировать учащихся о формате входных и выходных данных. Нередко правильное при ручной проверке решение не проходило автоматизированные тесты из-за наличия лишних данных в выводе программы. Для решения данной проблемы в условии задачи необходимо включать таблицу с несколькими тестовыми случаями, которые послужат образцом формата ввода и вывода данных.

В некоторых случаях учащиеся находили простой способ написания программы, который, в свою очередь позволял пройти все автоматизированные тесты (пример: вывод элементов из старого массива вместо формирования нового и т.д.). В таких случаях стоит акцентировать внимание учащихся на том, что прохождение всех тестов не гарантирует того, что решение будет засчитано, а является одним из этапов защиты лабораторной. Также иногда следует явно указывать в условии задачи, какие типы данных или алгоритмические конструкции обязательно должны присутствовать (или отсутствовать) в исходном коде решения (пример: обязательное наличие процедур и функций пользователя, вывод ответа в файл, запрет на использование определённого типа данных и др.). Стоит учитывать, что нарушение некоторые из вышеперечисленных ограничений можно также проверить автоматически. Например, перед запуском программы на тестовых данных, скрипт компиляции может искать необходимые ключевые слова в исходном коде решения (пример: procedure, function) и выдавать предупреждения в случае их наличия или отсутствия.

Опыт применения плагина «Виртуальная лаборатория программирования» показал, что автоматизированная проверка решений учащихся избавляет преподавателя от этапа монотонного ввода тестовых данных и их сверкой с правильными ответами, позволяя сконцентрировать своё внимание на обсуждении алгоритма решения с учащимся.

Список использованных источников:

1. Попова Е.Н., Трошина Е.Н. Автоматизация контроля знаний // Вестник Моск. гос. открытого ун-та. Сер. Общественно-политические и гуманитарные науки, № 2, 2010. – С. 69-73.

2. Ejudge Contest Management System [Электронный ресурс]. – Режим доступа: <https://ejudge.ru>. – Дата доступа: 29.09.2021.

3. Филинов А.Н. Система автоматического тестирования Ejudge // Информатика и образование, № 9, 2012. – С. 63-64.

4. Juan Carlos Rodriguez-del-Pino. A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features // International

Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government, 2012.