

практической конференции «Новые информационные технологии в образовании» (Применение технологий «1С» для формирования инновационной среды образования и бизнеса), Москва, 3–4 февраля 2015 г. Москва, 2015. Часть 2.

8. *Аргучинцев, А. В.* Платформа «1С:Предприятие» как основа построения современной корпоративной информационной системы вуза / А. В. Аргучинцев, В. С. Кедрин, Е. С. Чуйко. Текст: непосредственный // Известия Иркутского государственного университета. Серия: Политология. Религиоведение. 2017. Т. 22. С. 121–131.

УДК [378.016:004.4]:[378.146:004]

Карманова Е. В.

**АВТОМАТИЗИРОВАННЫЙ КОНТРОЛЬ ПРИ ОБУЧЕНИИ
ПРОГРАММИРОВАНИЮ НА PYTHON С ИСПОЛЬЗОВАНИЕМ
ПЛАГИНА CODERUNNER LMS MOODLE**

Екатерина Владимировна Карманова

monitor81@mail.ru

*ФГБОУ ВО «Магнитогорский государственный технический университет
им. Г.И. Носова», Россия, г. Магнитогорск*

**AUTOMATED CONTROL FOR TEACHING PROGRAMMING ON THE
PYTHON USING THE PLUGIN CODERUNNER IN LMS MOODLE**

Ekaterina Vladimirovna Karmanova

Nosov Magnitogorsk State Technical University, Russia, Magnitogorsk

Аннотация. В статье раскрываются возможности плагина автоматизированного контроля знаний — CodeRunner, разработанного для платформы электронного обучения MOODLE. Автором описаны виды заданий по программированию на Python, которые реализованы с помощью плагина CodeRunner.

***Abstract.** The article describes the capabilities of plugin “CodeRunner” for automated knowledge control, it’s developed for e-learning platform MOODLE. The author describes the types of programming tasks in Python that are implemented using the plugin “CodeRunner”.*

***Ключевые слова:** программирование на Python, автоматизированный контроль, платформа электронного обучения, плагин CodeRunner.*

***Keywords:** Python programming, automated control, e-learning platform, plugin CodeRunner.*

Современные образовательные учреждения активно используют возможности платформ электронного обучения при реализации смешанного обучения. К примеру, одной из популярных в российском образовательном секторе платформой обучения является LMS MOODLE. Анализ возможностей данной платформы посвящено достаточно много исследований и научных работ [1-4]. Также разработаны и размещены в свободный доступ инструкции и рекомендации по использованию LMS MOODLE в образовательном процессе. Однако стоит отметить, что большинство работ посвящено описанию стандартных элементов и ресурсов, входящих в базовый пакет дистрибутива MOODLE, именно эти компоненты платформы широко известны педагогам и именно их традиционно они используют при построении электронных курсов в поддержку учебных дисциплин. Если же изучить ресурс moodle.org, то в разделе Плагины существует большое разнообразие плагинов для развертывания на платформе новых интерактивных элементов.

Одним из таких плагинов является CodeRunner, особенно он будет интересен тем преподавателям, которые преподают языки программирования. Основным функционалом плагина является:

1. Создание заданий в виде написания исходного кода на следующий языках: JavaScript, Python, C, C++, Java, Pascal, PHP, SQL, Octave.
2. Создание заданий на построение графов.
3. Проверка программного кода неограниченным количеством тестов.

4. Реализация шаблона, импортированного из прототипа или разработанного автором вопроса (подробнее рассмотрено Ю. В. Перязевой) [5].

В рамках данного исследования рассмотрены возможности использования плагина при автоматизированной проверке программ на языке Python. Рассмотрим этапы разработки заданий по программированию на Python.

1 этап — создание категорий в банке вопросов (Рис. 1) — при добавлении любых тестовых заданий в банк вопросов рекомендуется указывать — к какой категории/разделу/теме курса принадлежит данный вопрос. Данный механизм в дальнейшем позволяет создавать различного типа тесты (тест для самоконтроля, тест по теме, тест по нескольким разделам дисциплины и т. д.), выделяя какие категории вопросов, добавить в данный тест.









































































































- По умолчанию для Программирование на Python (9902) (0)
Категория по умолчанию для общих вопросов в контексте «Программирование на Python (9902)».
- 
- Концепция типов данных (0)    
 - Вещественный тип (2)    
 - Кортежи (1)      
 - Логический тип (1)      
 - Приведение типов (1)      
 - Словари (1)      
 - Списки (2)      
 - Строки (2)      
 - Целый тип (1)      
 - Команда print (1)      
 - Форматированный вывод (0)     
- Основные алгоритмические структуры в Python (0)      
 - Составление логических выражений (2)    
 - Оператор ветвления в полной форме (1)      
 - Оператор ветвления в сокращенной форме (1)      
 - Цикл с условием (0)      
 - Цикл с переменной (0)     
- Функции в Python. Основы процедурного подхода (0)     
 - Простые функции (0)    

Рисунок 1 — Категории в банке вопросов курса «Программирование на Python»

2 этап — добавление отдельного вопроса в отдельную категорию банка вопроса — установке плагина CodeRunner, в диалоговом окне выбора типа вопроса появится новый тип (Рис. 2).

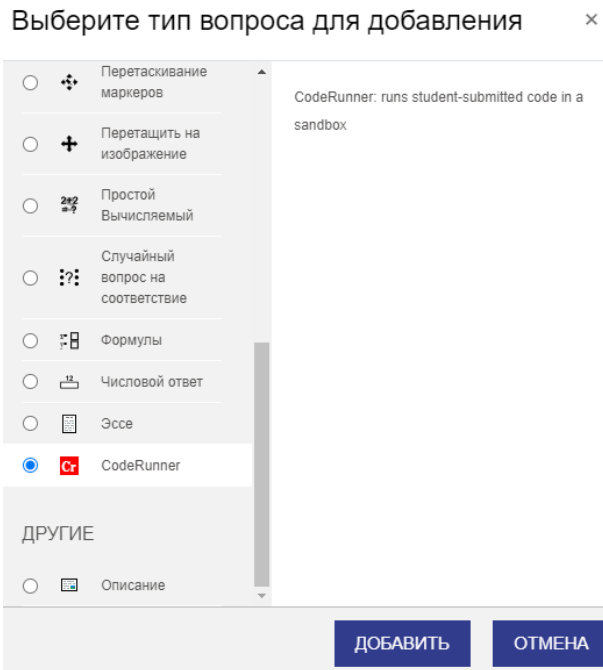
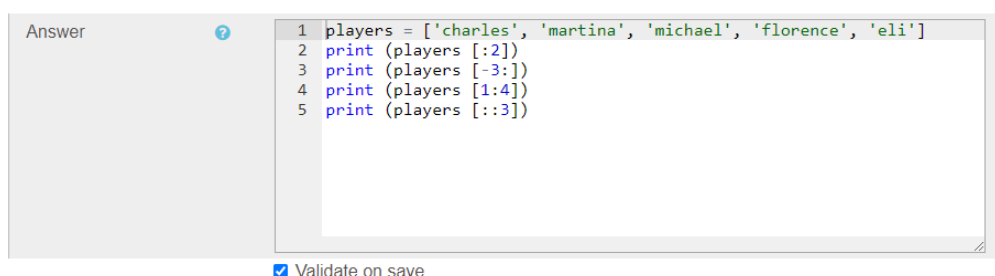


Рисунок 2 — Диалоговое окно с выбором типа вопроса

3 этап —настройка тестового задания — форма добавления нового вопроса рекомендуется заполнить следующие поля:

- Question type (тип вопроса) — на каком языке программирования будет проводиться контроль знаний.
- Название вопроса — краткое название вопроса, которое будет отображаться в банке вопросов.
- Балл по умолчанию — балл, который будет назначен студенту при правильном выполнении данного задания.
- Answer (ответ) — вводится пример программы / скрипта для решения данного задания. Пример ответа используется скриптом теста. На рисунке 3 представлен пример ответа для задачи 1 — «Дан список `players = ['charles', 'martina', 'michael', 'florence', 'eli']` . На первой строке вывести 2 первых значений. На второй строке вывести последние 3 значения. На третьей строке вывести со 2 по 4 элемент. На четвертой строке вывести каждый третий элемент, начиная с первого.»



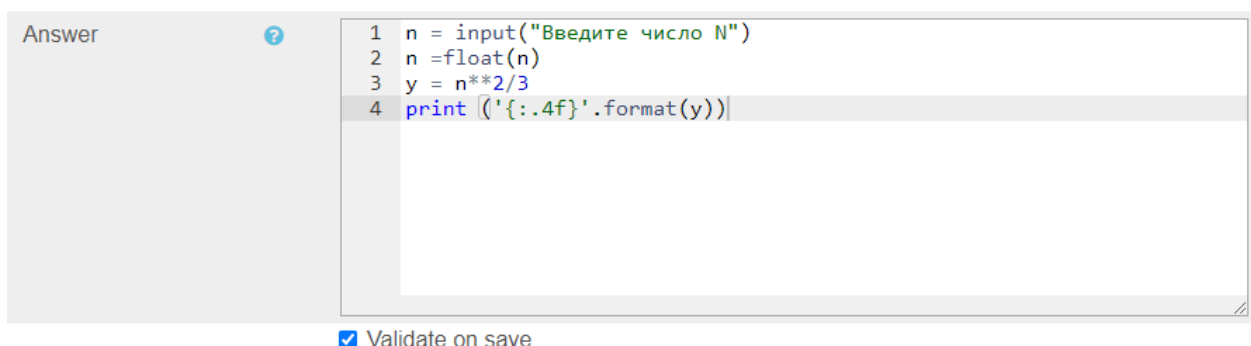
```
Answer ?  
1 players = ['charles', 'martina', 'michael', 'florence', 'eli']  
2 print (players [:2])  
3 print (players [-3:])  
4 print (players [1:4])  
5 print (players [::3])  
 Validate on save
```

Рисунок 3 — Пример заполнения поля «Ответ» для задачи на функцию Print и срезы списков

Стоит отметить, что чаще всего требуется написать программы, которые позволяют запрашивать ввод различных значений. На рисунке 4 представлен пример ответа для задачи 2 — «Напишите скрипт, который запрашивает число N с помощью сообщения "Введите число N", и выводит результат следующего выражения:

$$y = \frac{N^2}{3}$$

в формате вещественного числа с 4 знаками после запятой.»



```
Answer ?  
1 n = input("Введите число N")  
2 n =float(n)  
3 y = n**2/3  
4 print ('{:.4f}'.format(y))  
 Validate on save
```

Рисунок 4 — Пример заполнения поля «Ответ» для задачи на ввод данных и вывод результата выражения в форматированном виде

Test cases (тесты) — реализует проверку введенном пользователем скрипта — вводятся входные данные (если это предусмотрено заданием) в поле Standard Input, поле Expected output должно содержать правильные результаты работы скрипта при указанном вводе. На рисунке 5 представлен пример заполнения Теста 1 для задачи 2.

▼ Test cases

Test case 1

Standard Input: 5.980

Expected output: Введите число N5.980
11.9201

Extra template data

Test properties: Use as example Display Show Hide rest if fail Mark 1.000

Ordering 10

Рисунок 5 — Пример заполнения полей Теста для задачи на ввод данных и вывод результата выражения в форматированном виде

Из рисунка видно, что на выходе необходимо обязательно указать значение сообщения, которое возникает при запросе входного значения, а также само входное значение, которое указано в поле Standard Input, на следующей строке должно быть указано верное значение работы скрипта при заданном вводе. Такие тесты можно добавлять неограниченно для одного задания.

При выполнении заданий CodeRunner испытуемый вводит свой код программы и результатом проверки система выводит отчет, как прошла программа заложенные тесты в задании. На рисунке 6 представлен пример отчета, который видит испытуемый.

	Input	Expected	Got	
✓	5.980	Введите число N5.980 11.9201	Введите число N5.980 11.9201	✓
✓	56	Введите число N56 1045.3333	Введите число N56 1045.3333	✓
✓	-9	Введите число N-9 27.0000	Введите число N-9 27.0000	✓

Passed all tests! ✓

Рисунок 6 — Отчет проверки выполнения программы по 2ой задаче

Таким образом, плагин CodeRunner позволяет облегчить работу педагога в рамках контроля учебных заданий по программированию при реализации смешанного обучения. Кроме того, возможности CodeRunner также позволяют проводить конкурсы, олимпиады по программированию с охватом большого количества участников, при этом потребуется небольшое количество экспертов, а сама проверка будет проводиться быстро и объективно.

Список литературы

1. *Карманова, Е. В.* Особенности реализации смешанного обучения с использованием среды MOODLE / Е. В. Карманова. Текст: непосредственный // Информатика и образование. 2018. № 8. С. 43–50.

2. *Якунина, Е. К.* Преимущества и возможности применения СДО MOODLE в общеобразовательных учебных заведениях / Е. К. Якунина, М. В. Романова. Текст: непосредственный // *Ab ovo... (С самого начала...):* сборник научных трудов / Магнитогор. гос. техн. ун-т им. Г. И. Носова. Магнитогорск, 2019. С. 86–87.

3. *Масленникова, О. Е.* Методика проведения мини-проб «try-a-skills» по компетенции «Программные решения для бизнеса» в среде LMS Moodle / О. Е. Масленникова, О. Б. Назарова, А. А. Олейник. Текст: непосредственный // Дистанционное обучение: актуальные вопросы: сборник материалов Всероссийской научно-практической конференции, Чебоксары, 16 июля 2020 г. / Чуваш. республ. ин-т образования. Чебоксары, 2020. С. 75–79.

4. *Karmanova, E. V.* Modeling knowledge assessment with gamification technology on e-learning platform / E. V. Karmanova, E. V. Chernova, A. S. Dokolin // International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon-2019), Vladivostok, 01–04 October 2019. USA: Institute of Electrical and Electronics Engineers Inc., 2019. С. 1–6.

5. *Перязева, Ю. В.* Возможности автоматической проверки заданий в LMS Moodle / Ю. В. Перязева. Текст: непосредственный // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 876–885. DOI: 10.25559/SITITO.15.201904.876-885.