



УДК 378.16

THE EXPERIENCE OF USING PYTHON AND LATEX FOR AUTOMATIC GENERATION OF MATHEMATICS TESTS**ОБ ОПЫТЕ ИСПОЛЬЗОВАНИЯ PYTHON И LATEX ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ КОНТРОЛЬНЫХ РАБОТ ПО МАТЕМАТИКЕ****Byzov V.A. / Бызов В.А.***senior lecturer / старший преподаватель.*

ORCID: 0000-0002-3613-5949

SPIN: 4783-3844

*Vyatka State University, Russia, Kirov, Moskovskaya, 36, 610000**Вятский государственный университет, Россия, Киров, Московская, 36, 610000*

Аннотация. В статье рассматриваются практические аспекты автоматической генерации контрольных работ по математике. Для создания шаблона контрольной работы используется система компьютерной вёрстки LaTeX. На основе разработанного шаблона скрипт, написанный на языке Python, выполняет генерацию файлов PDF, содержащих разные варианты работы. Кроме того, используемый подход позволяет автоматически генерировать ответы к задачам, что значительно упрощает проверку работ обучающихся.

Ключевые слова: контрольная работа, автоматическая генерация, шаблон работы, LaTeX, Python.

Вступление.

Неотъемлемой составляющей образовательного процесса в университете является проверка результатов усвоения учебного материала студентами. Устоявшимся способом такой проверки считается проведение контрольных и проверочных работ. При этом часто используются работы, содержащие несколько вариантов заданий, поскольку они сводят к минимуму вероятность несамостоятельного выполнения работ студентами.

Облегчением работы преподавателя могла бы стать автоматизация процесса создания разных вариантов контрольной работы. В некоторых разделах математики практические задания отличаются только набором числовых параметров. Так, например, для проверки у студентов навыков нахождения определителей матриц можно составить комплект задач, содержащих одинаковую текстовую часть («найдите определитель матрицы, используя метод разложения по строке или столбцу») и разные числовые данные (например, матрицы четвёртого порядка).

Стоит отметить, что рассмотрение данного аспекта преподавательской деятельности ранее уже встречалось в литературе; например, в работах [1-4]. При этом в некоторых статьях приведены схемы генерации «хороших», «решаемых» задач, но упущен аспект практической реализации подхода. В других случаях авторы рассматривают вопрос разработки сложного комплекса, состоящего из веб-интерфейса, СУБД, графического интерфейса и других компонентов.

Целью настоящей работы является рассмотрение именно вопроса практической реализации модуля для автоматической генерации комплекта заданий. Положительной особенностью описываемого подхода является простота его реализации. Кроме того, разработанный модуль несложно



модифицировать для проведения работ по другим темам.

Шаблон контрольной работы.

В описываемом нами подходе для генерации контрольных работ используется шаблон документа, разработанный при помощи системы компьютерной верстки LaTeX. Данный выбор обусловлен рядом причин.

Во-первых, система LaTeX позволяет создавать документы, содержащие математические формулы любой сложности. Даже если преподавателю потребуется использовать в контрольной работе специфические математические обозначения, то почти наверняка он сможет найти необходимые для этого команды.

Во-вторых, при использовании системы LaTeX достаточно просто отредактировать TeX-файл внешней программой или скриптом. Программное редактирование, например, документа Microsoft Word – более сложная задача.

В-третьих, для системы верстки LaTeX характерен высокий уровень совместимости. Практически всегда исходный код документа корректно преобразуется в PDF-файл вне зависимости от версии LaTeX.

В рассматриваемом способе генерации TeX-файл шаблона должен содержать строки специального формата, которые в дальнейшем будут заменены конкретными числовыми значениями в разных вариантах контрольной работы. Нами для этой цели использовались строки, расположенные между символами «решёток»: #value#.

Ниже в листинге 1 приведён пример фрагмента шаблона контрольной работы по теме «Матрицы».

Листинг 1. Фрагмент шаблона контрольной работы

```
\begin{center}
\texttt{Вариант #varno#}
\end{center}

\textbf{Задача 1.} Найдите определитель матрицы:
\begin{equation*}
\begin{pmatrix}
#m1_11# & #m1_12# & #m1_13# & #m1_14# \\
#m1_21# & #m1_22# & #m1_23# & #m1_24# \\
#m1_31# & #m1_32# & #m1_33# & #m1_34# \\
#m1_41# & #m1_42# & #m1_43# & #m1_44#
\end{pmatrix}
\end{equation*}
% #answer1#

\textbf{Задача 2.} Решите матричное уравнение:
\begin{equation*}
X \cdot
\begin{pmatrix}
#m2_11# & #m2_12# & #m2_13# \\
#m2_21# & #m2_22# & #m2_23#
\end{pmatrix}
```



```

#m2_31# & #m2_32# & #m2_33#
\end{pmatrix}
=
\begin{pmatrix}
#m3_11# & #m3_12# & #m3_13# \\
#m3_21# & #m3_22# & #m3_23# \\
#m3_31# & #m3_32# & #m3_33#
\end{pmatrix}
\end{equation*}
% #answer2#

```

Обратим внимание на строки `#answer1#` и `#answer2#`. Они используются для обозначения позиций расположения ответов на задачи. Ответы находятся в закомментированных строках и будут видны только преподавателю при открытии TeX-файлов сгенерированных вариантов работ.

Скрипт для генерации вариантов работы.

Для генерации разных вариантов контрольной работы на основе созданного шаблона использовался скрипт, написанный на языке программирования Python. Основная причина выбора этого языка заключается в том, что для Python доступен большой набор готовых библиотек решения различных математических задач. Это библиотеки SymPy, NumPy, SciPy и другие.

Скрипт генерации контрольных работ должен выполнять следующие операции:

1. генерация исходных данных для задач на основе генератора псевдослучайных чисел;
2. нахождение ответов к полученным задачам;
3. замена строк TeX-файла, которые заключены в «решётки», числовыми данными;
4. компиляция TeX-файла в файл формата PDF.

Ниже в листинге 2 приведён фрагмент скрипта, генерирующего варианты контрольных работ по теме «Матрицы». В этом листинге вызов функции `generate_matrix(4, 4, 1, 5)` используется для генерации случайно целочисленной матрицы 4×4 , модули элементов которой находятся в интервале от 1 до 5.

Листинг 2. Фрагмент скрипта для генерации вариантов работы

```

import numpy as np
import os

# Настройки
input_name = 'template1.tex' # файл шаблона
output_path = 'Work1'       # директория для результатов
variants_count = 6           # количество вариантов

with open(input_name, 'r', encoding = 'UTF-8') as input_file:
    input_data = input_file.read()

```



```

for ind in range(variants_count):
    cur_data = input_data
    cur_data = cur_data.replace('#varno#', str(ind + 1))

    # Первое задание
    # Генерация матрицы 4 на 4 для расчёта определителя
    matrix1 = generate_matrix(4, 4, 1, 5)
    for i in range(matrix1.shape[0]):
        for j in range(matrix1.shape[1]):
            cur_data = cur_data.replace('#m1_' + str(i + 1) + str(j + 1) + '#',
str(int(matrix1[i, j])))
        determinant1 = np.linalg.det(matrix1)
        cur_data = cur_data.replace('#answer1#', 'Ответ: ' + str(determinant1))

    #...
    # Сохраняем контрольную
    result_name = os.path.join(output_path, 'variant_' + str(ind + 1) + '.tex')
    with open(result_name, 'w', encoding = 'UTF-8') as output_file:
        output_file.write(cur_data)

    # Компилируем TeX-файл в PDF
    os.system('pdflatex -output-directory ' + output_path + ' ' + result_name)

```

На рис. 1 изображён фрагмент итогового варианта работы, полученного при использовании скрипта. В файле `variant_1.tex` закомментированные строки содержат ответы к задачам.

Математика, 2020-2021 учебный год, 1 курс, 1 семестр

Контрольная работа

Вариант 1

Задача 1. Найдите определитель матрицы:

$$\begin{pmatrix} 2 & -4 & -1 & 3 \\ 5 & 1 & 1 & -2 \\ 1 & 1 & -2 & -2 \\ 3 & -2 & -4 & 3 \end{pmatrix}$$

Задача 2. Решите матричное уравнение:

$$X \cdot \begin{pmatrix} 1 & -1 & 1 \\ -4 & 3 & -3 \\ 2 & 2 & -4 \end{pmatrix} = \begin{pmatrix} -5 & 6 & -2 \\ -3 & -1 & 3 \\ 2 & 1 & 1 \end{pmatrix}$$

Рис. 1 Пример сгенерированного варианта контрольной работы



Заключение и выводы.

Описанный подход, позволяет в достаточно короткие сроки создать комплект заданий контрольной работы по различным разделам математики. Важно отметить, что были описаны именно основные идеи метода. Развивая эти идеи, можно создать полноценную программную систему генерации контрольных работ.

Литература:

1. Посов И.А. Автоматическая генерация задач // Компьютерные инструменты в школе. – 2007. – № 1. – с. 54–62.
2. Юдин С.В. Генератор контрольных (расчетно-графических) работ по математике для студентов первых курсов университетов // Научно-методический электронный журнал «Концепт». – 2016. – № 8. – с. 9–14. – URL: <http://e-koncept.ru/2016/16158.htm>
3. Коновалов Я.Ю., Соболев С.К. Методические аспекты компьютерного генерирования заданий по математике // Наука и образование: научное издание. – 2016. – № 7. – с. 285–295.
4. Окишев С.В. Проблема создания и использования генераторов и решателей математических задач // Интернет-журнал «Мир науки». – 2018. – Т. 6. – № 3. – URL: <https://mir-nauki.com/PDF/56PDMN318.pdf>

References:

1. Posov I.A. Automatic task generation // Computer tools in school. – 2007. – No. 1. – pp. 54–62.
2. Yudin S.V. Generator of control (computational and graphic) works in mathematics for first-year university students // Scientific and methodological electronic journal «Concept». – 2016. – No. 8. – pp. 9–14. – URL: <http://e-koncept.ru/2016/16158.htm>
3. Konovalov Y.Y., Sobolev S.K. Methodological aspects of computer generation of tasks in mathematics // Science and Education: Scientific Publication. – 2016. – No. 7. – pp. 285–295.
4. Okishev S.V. The problem of creating and using generators and solvers of mathematical exercises // World of Science. Pedagogy and psychology. – 2018. – Vol. 6. – No. 3. – URL: <https://mir-nauki.com/PDF/56PDMN318.pdf>

***Abstract.** In the article, practical aspects of automatic generation of Mathematics tests are considered. For the creation of a test template, the LaTeX typesetting system is used. On the basis of the pre-developed template, the script written in the Python programming language generates the PDF documents containing different variants of the test. Besides, the given approach allows to automatically generate the answers for the tasks, which significantly simplifies checking of students' works.*

***Keywords:** test, automatic generation, test template, LaTeX, Python.*

Статья отправлена: 13.11.2020 г.

© Бызов В.А.