



## АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕСТОВЫХ ЗАДАНИЙ В EXCEL С ИСПОЛЬЗОВАНИЕМ PYTHON

**Маншуров Шерзод Туйчибойевич**

Алмалыкский государственный технический институт,  
кафедра «Математика и естественные науки» старший  
преподаватель

E-mail: [manshurov\\_sh@mail.ru](mailto:manshurov_sh@mail.ru)

**Абдуганиева Юлдузой Шахабидиновна**

Алмалыкский государственный технический институт,  
кафедра «Математика и естественные науки» старший  
преподаватель

E-mail: [yulduzabduganieva@mail.ru](mailto:yulduzabduganieva@mail.ru)

***Аннотация** В данной статье представлена разработка программы на языке Python для обработки тестовых вопросов, хранящихся в файле Excel, и автоматической генерации нескольких случайных вариантов. Программа считывает вопросы из Excel, разделяет текст и варианты ответов, перемешивает ответы и заново назначает правильный вариант. Затем создаётся заданное количество тестовых вариантов (например, 40), каждый из которых содержит определённое число вопросов, и сохраняется в новом Excel-файле на отдельных листах. Такой подход обеспечивает автоматизацию, гибкость и эффективность при подготовке разнообразных версий тестов, способствует справедливой оценке и упрощает работу преподавателей.*

***Ключевые слова:** Программирование на Python, Файл Excel, Тестовые вопросы, Генерация вариантов, Перемешивание ответов, Автоматизация.*

**ABSTRACT** This article presents the development of a Python-based program for processing test questions stored in Excel and generating multiple randomized variants automatically. The program reads questions from an Excel file, separates the text and answer options, shuffles the answers, and reassigns the correct choice. It then creates a specified number of test variants (e.g., 40), each containing a set number of questions, and saves them into a new Excel file with

*separate sheets. The approach ensures automation, flexibility, and efficiency in preparing diverse test versions, supporting fair assessment and simplifying teachers' work.*

**Keywords:** *Python programming, Excel file, Test questions, Variant generation, Answer shuffling, Automation*

## **ВВЕДЕНИЕ**

В образовательном процессе тестовые вопросы являются одним из наиболее часто используемых инструментов для оценки уровня знаний учащихся. С помощью тестов определяется степень усвоения материала, понимание темы и способность применять знания на практике. Кроме того, тестовые задания имеют важное значение для преподавателей, так как позволяют анализировать учебный процесс и оценивать качество образования. Однако повторное использование одних и тех же вопросов может привести к тому, что учащиеся просто заучивают ответы. Это негативно влияет на основную цель тестирования — объективную оценку знаний. Поэтому представление вопросов в различных вариантах играет важную роль в повышении эффективности обучения и справедливости оценки. В данной статье рассматривается разработка программы на языке Python в среде Excel, которая позволяет перерабатывать тестовые вопросы и автоматически создавать несколько вариантов. Программа считывает вопросы из Excel-файла, перемешивает варианты ответов и заново отмечает правильный ответ. В результате формируется заданное количество тестовых вариантов, которые сохраняются в новом Excel-файле на отдельных листах.

## **СТРУКТУРА ПРОГРАММЫ И ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ**

### **Необходимость создания вариантов**

В учебном процессе составление тестовых заданий в нескольких вариантах имеет важное значение для справедливой оценки знаний учащихся. Повторное использование одинаковых вопросов может привести лишь к заучиванию ответов. Разные варианты помогают выявить более глубокое понимание материала. Кроме того, наличие нескольких вариантов:

- обеспечивает академическую честность, так как вероятность списывания снижается;
- стимулирует учащихся к более широкой подготовке;
- предоставляет преподавателю возможность гибко организовать процесс оценки.

### **Проблемы при создании вариантов вручную**

Однако подготовка вариантов вручную вызывает ряд трудностей:

- требует много времени и усилий;
- возможны ошибки при перемешивании ответов;
- нарушается порядок вопросов, что может привести к потере логической последовательности;
- при работе с большим объемом данных возникают технические ограничения.

### **Преимущества автоматизации**

Автоматизация процесса создания вариантов позволяет эффективно решить эти проблемы. Автоматизация:

- экономит время — за несколько минут можно получить десятки вариантов;
- снижает количество ошибок — перемешивание ответов и определение правильного ответа выполняется алгоритмом;
- обеспечивает объективность — вопросы выбираются случайным образом, что способствует справедливой оценке;
- создает гибкость — количество вариантов и вопросов в каждом из них преподаватель задает самостоятельно;

- повышает удобство — результаты сохраняются в Excel-файле в упорядоченном виде, что облегчает использование для преподавателей.

Конфигурация: исходный и выходной файлы, количество вариантов

На этапе конфигурации программе задаются параметры: исходный файл, выходной файл и количество вариантов.

- **Исходный файл (source file)** — Excel-файл, содержащий тестовые вопросы. Программа считывает из него данные. Пример: TEST-2025.xlsx
- **Выходной файл (output file)** — новый Excel-файл, в котором сохраняются созданные варианты. Каждый вариант записывается на отдельный лист. Пример: 40\_variant\_test2025.xlsx
- **Количество вариантов (number of variants)** — задается число тестовых вариантов, которые необходимо сформировать. Пример: num\_variants = 40
- **Количество вопросов в одном варианте (per variant)** — указывается число вопросов в каждом варианте. Пример: per\_variant = 20

| Вопрос        | A         | B         | C         | D         |
|---------------|-----------|-----------|-----------|-----------|
| Текст вопроса | Вариант A | Вариант B | Вариант C | Вариант D |

### Функция парсинга вопроса

function parse\_question(line):

if "a)" нет:

return (вопрос=line, A="", B="", C="", D="")

else:

вопрос, хвост = line.split("a)")

A = хвост

если "b)" есть: A, B = A.split("b)")

если "c)" есть: B, C = B.split("c)")

если "d)" есть: C, D = C.split("d)")

return (вопрос, A, B, C, D)

### Перемешивание ответов и определение правильного

Один из важных этапов при формировании вариантов тестов — это случайное перемешивание ответов и повторное определение правильного. Этот процесс предотвращает повторяемость вопросов в одинаковом виде и исключает возможность простого заучивания порядка.

| Этап                                     | Описание                                                                                                                                                                                                  |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Сбор вариантов ответов</b>            | Для каждого вопроса ответы А, В, С, D помещаются в отдельный массив (список). Для каждого варианта сохраняется исходная правильная буква ( <b>CorrectLetter_org</b> ).                                    |
| <b>Случайное перемешивание (shuffle)</b> | В Python используется функция <code>random.shuffle()</code> , которая изменяет порядок ответов. Для каждого варианта задаётся отдельный seed (ключ случайности), что обеспечивает уникальность вариантов. |
| <b>Назначение нового порядка</b>         | Перемешанные ответы заново располагаются в порядке А, В, С, D. Каждый ответ связывается с новой буквой.                                                                                                   |
| <b>Определение правильного ответа</b>    | Исходный правильный текст ( <b>CorrectText</b> ) ищется среди перемешанных вариантов. Буква, под которой оказался правильный ответ, фиксируется как новый правильный символ ( <b>CorrectLetter_new</b> ). |

### СТРУКТУРИРОВАНИЕ ВОПРОСОВ НА ОСНОВЕ DATAFRAME

После считывания тестовых вопросов из Excel-файла важным этапом является их структурирование с помощью **pandas DataFrame** (приведение к табличному виду). Этот процесс позволяет хранить вопросы в упорядоченном, модульном формате и удобно использовать их на следующих этапах (создание вариантов, перемешивание ответов).

- **Чтение вопросов**

- Все строки считываются из Excel-файла.

- Каждая строка разделяется на вопрос и варианты ответов с помощью функции `parse_question()`.
- **Формирование таблицы** Для каждого вопроса создаются следующие столбцы:
  - **ID вопроса** – порядковый номер вопроса
  - **Вопрос** – текст вопроса
  - **A\_org, B\_org, C\_org, D\_org** – варианты ответов
  - **CorrectLetter\_org** – правильный ответ (например, A, B, C или D)
- **Приведение к DataFrame**
  - Каждый вопрос и ответы сохраняются в виде словаря (dictionary).
  - Список словарей преобразуется в таблицу с помощью функции `pd.DataFrame()`.

**Пример в табличной форме**

| ID вопроса | Вопрос                 | A_org                 | B_org                | C_org         | D_org            | CorrectLetter_org |
|------------|------------------------|-----------------------|----------------------|---------------|------------------|-------------------|
| 1          | Что такое информатика? | Наука о данных        | Наука о жизни        | Наука о языке | Наука о культуре | A                 |
| 2          | Python — это ...       | Язык программирования | Операционная система | База данных   | Браузер          | A                 |

**Заключение**

Программа по переработке тестовых вопросов на основе Excel и автоматическому созданию вариантов является важным инструментом в образовательном процессе для эффективного управления тестами и справедливой оценки знаний учащихся. Программа считывает вопросы из Excel-файла, разделяет ответы, случайным образом перемешивает их и заново определяет правильный вариант. В результате формируется несколько

тестовых вариантов в быстром и точном виде, которые сохраняются в Excel-файле на отдельных листах.

Данный подход:

экономит время,

снижает количество ошибок,

обеспечивает объективность оценки,

создаёт удобство для преподавателей.

Таким образом, программа, автоматизируя работу с тестовыми вопросами, способствует повышению качества образования, стимулирует учащихся к более глубокому освоению знаний и делает процесс оценки более эффективным.

### **Использованная литература**

1. McKinney, W. Python for Data Analysis. O'Reilly Media, 2017.
2. Van Rossum, G., & Drake, F. L. Python Tutorial. Python Software Foundation.
3. Manshurov, Sh. T. (2025). Programma sokhraneniya temperatury reaktsii ammoniya dikromata v fayly na Python. Oriental Renaissance: Innovative, Educational, Natural and Social Sciences, 5(5), 710–714. ООО “Oriental renessans”.
4. Manshurov Sh. T., Utabov U. A., & Abduganieva Yu. Sh. (2023). Otrabotka dannykh temperatury na C++ i sokhranenie v fayle. Oriental Renaissance: Innovative, Educational, Natural and Social Sciences, 3(2), 730–740. ООО “Oriental renessans”.
5. Manshurov Sh. T. (2025). Kompyuternaya vizualizatsiya temperaturnykh izmeneniy v proizvodstvennykh i laboratornykh usloviyakh. Oriental Renaissance: Innovative, Educational, Natural and Social Sciences, 5(5), 691–703. ООО “Oriental renessans”.

6. Manshurov Sh. T., & Jonqobilov J. T. (2024). C++ dasturlash tilida n-xonali palindromik sonlarni topish. *Oriental Renaissance: Innovative, Educational, Natural and Social Sciences*, 4(6), 173–178. OOO “Oriental renessans”.
7. Manshurov Sh. T., & Jonqobilov J. T. (2024). C++ dasturlash tilida n-xonali palindromik sonlarni topish. *Oriental Renaissance: Innovative, Educational, Natural and Social Sciences*, 4(6), 173–178. OOO “Oriental renessans”.
8. Manshurov Sh. T., & Abduganieva Yu. Sh. (2024). Noveyshie tekhnologii interneta svyazi: ot 5G k budushchim standartam peredachi dannykh. *Oriental Renaissance: Innovative, Educational, Natural and Social Sciences*, 4(3), 315–328. OOO “Oriental renessans”.
9. Rahmonkulov R., & Manshurov Sh. T. (2023, May 6). Visualization of technological processes for management purposes. *The Peerian Journal*, 18, 8–14.
10. Sulaymanov D. B., Manshurov Sh. T., Abduganieva Yu. Sh., & Khakimova G. A. (2023). Development and improvement of media technologies in the educational process. *Estudios Italianos*, 10(2), 487–501.
11. Samandarov I. R., Manshurov Sh. T., Dushatov N. T., Miratoyev Z. M., & Mustafin R. R. (2023). Obrabotka izobrazheniy v C++ s pomoshch'yu biblioteki OpenCV. *Universum: Tekhnicheskie Nauki*, (5-2)(110), 19–28. Mezhdunarodnyy tsentr nauki i obrazovaniya.
12. Samandarov I. R., Manshurov Sh. T., Miratoyev Z. M., Jonqobilov J. T., & Abdukhshimov J. U. (2023). Algoritmy raspoznavaniya otdel'nykh i nalozhennykh real'nykh ploskikh ob'ektov. *Universum: Tekhnicheskie Nauki*, (5-2)(110), 11–18. Mezhdunarodnyy tsentr nauki i obrazovaniya.
13. Manshurov Sh. T. (2023). Programmirovaniye na C++ dlya izucheniya VAK i primesnogo sostava pri pyatikratnoy pereplavke metallurgicheskogo kremniya na solnechnoy pechi. *Oriental Renaissance: Innovative, Educational, Natural and Social Sciences*, 3(2), 517–529. OOO “Oriental renessans”.

14. Manshurov Sh. T. (2022). Spektral'naya voltovaya chuvstvitel'nost' polikristallichesкого кремния, poluchennogo pyatokratnoy pereplavkoy polukristallichesкого кремния na solnechnoy pechi. *Oriental Renaissance: Innovative, Educational, Natural and Social Sciences*, 2(12), 1079–1082. ООО "Oriental renessans".