

3. Основными операциями являются обновление списка файлов создание новой папки, а также удаление файлов и папок (рисунок 3).

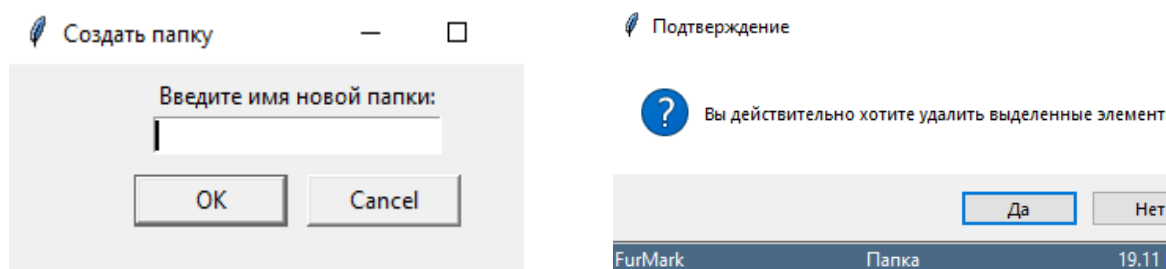


Рис. 3. Основные операции

**Заключение:** в ходе разработки файлового менеджера на языке программирования Python была проведена обширная работа, охватывающая несколько ключевых аспектов. Процесс разработки включал проектирование, реализацию, тестирование и документирование приложения.

В начале работы был проведён анализ требований и определение функциональных возможностей, которые должен предоставлять файловый менеджер. Затем был разработан детальный план проекта, включающий структуру программы, алгоритмы обработки файлов и папок, а также интерфейс пользователя.

После этого была проведена реализация программы, включающая написание кода на языке Python. Были использованы различные библиотеки и модули Python, такие как `os`, `shutil` и `tkinter`, для обеспечения функциональности программы. Важным аспектом разработки было обеспечение безопасности данных пользователя и обработка возможных ошибок.

Список использованных источников:

1. Разумников С.В. Теория алгоритмов : методические указания к выполнению курсовой работы по дисциплине «Теория алгоритмов» для студентов очной формы обучения, обучающихся по направлению 09.03.03 «Прикладная информатика» / С.В. Разумников ; Юргинский технологический институт. – Юрга : Изд-во Юргинского технологического института (филиала) Томского политехнического университета, 2022. – 20 с.
2. Psutil documentation: документация psutil. – URL: [https://psutil.readthedocs.io/en/latest/#psutil.process\\_iter](https://psutil.readthedocs.io/en/latest/#psutil.process_iter) (дата обращения: 10.03.2024). – Текст: электронный.
3. Егоров А. Инди-курс программирования на Python / А.Егоров // Stepik: сайт. – URL: <https://stepik.org/course/63085/syllabus> (дата обращения: 20.12.2023). – Текст: электронный.
4. Руководство по языку программирования Python // METANIT: сайт. – URL: <https://metanit.com/python/tutorial/> (дата обращения: 20.12.2023). – Текст: электронный.

#### «РАЗРАБОТКА ПОГРАММЫ МАТРИЦЫ ПО ПАРНЫМ СРАВНЕНИЯМ» НА ЯЗЫКЕ PYTHON

*М.М. Дехтеренко, студент гр. 17В21*

*Научный руководитель: Захарова А.А.<sup>а</sup>, д.т.н, проф., доц.  
Юргинский технологический институт (филиал)*

*Национального исследовательского Томского политехнического университета  
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26  
E-mail: <sup>а</sup>coffffffffin@gmail.com*

**Аннотация:** В современном информационном обществе, программирование и разработка программных решений играют ключевую роль в решении различных задач. В рамках данной курсовой работы фокусируется внимание на разработке программы на языке программирования Python, которая предназначена для создания матрицы по парным сравнениям. Этот инструмент является важным элементом в анализе предпочтений и принятии решений, позволяя систематизировать и оценивать субъективные предпочтения в различных областях.

**Ключевые слова:** анализ данных, парные сравнения, относительная значимость, предпочтения, программа матрица, управление проектами, принятие решений, маркетинговые исследования, алгоритмы сравнения.

**Abstract:** In the modern information society, programming and development of software solutions play a key role in solving various tasks. Within the framework of this course work, attention is focused on the development of a program in the Python programming language, which is designed to create a matrix based on paired comparisons. This tool is an important element in the analysis of preferences and decision-making, allowing you to systematize and evaluate subjective preferences in various fields.

**Keywords:** data analysis, paired comparisons, relative importance, preferences, matrix program, project management, decision making, marketing research, comparison algorithms.

Программа «Матрица по парным сравнениям» играет существенную роль в облегчении процесса принятия решений, предоставляя систематизированный метод анализа данных. Её влияние проявляется в способности выявления относительной значимости альтернатив и установлении предпочтений на основе четких критериев. Важность программы раскрывается через её способность преобразовывать сложные структуры данных в информацию, способную поддерживать информированные и обоснованные решения в различных сферах жизни.

Программа предназначена для систематизации и оценки предпочтений в различных областях, где необходимо сравнить и ранжировать элементы относительно друг друга. Области применения включают:

1. Принятие решений: Использование матрицы парных сравнений помогает при принятии решений, где необходимо оценить важность различных факторов или альтернатив.
2. Проектное управление: В проектах, где существует несколько вариантов решения или различных параметров, матрица парных сравнений может помочь определить наилучшие варианты.
3. Бизнес-стратегии: В анализе бизнес-стратегий и тактик использование матрицы парных сравнений может помочь выявить сильные и слабые стороны различных подходов.
4. Развитие продукта: при разработке новых продуктов или улучшении существующих матрица парных сравнений может помочь определить, какие характеристики приоритетны.
5. Образование и исследования: В образовательных и исследовательских проектах оценка альтернатив и параметров может быть важным этапом для получения объективных результатов.
6. Сравнение производительности: В области технической или научной деятельности, где необходимо сравнивать производительность различных систем или методов.

Данная программа является фрагментом графического пользовательского интерфейса (GUI) на языке программирования Python с использованием библиотеки tkinter. Входные данные для этого кода могут включать следующее:

1. Взаимодействие с пользователем: Входные данные могут включать в себя ввод пользователя через интерфейс, такой как указание размера матрицы с использованием виджета Spinbox.
2. События пользовательского интерфейса: возможно, код реагирует на события интерфейса, такие как нажатие кнопок, изменение значений виджетов и другие действия пользователя.
3. Данные для генерации матрицы: Входные данные могут использоваться для генерации единичной матрицы определенного размера. Вероятно, размер матрицы может вводиться пользователем.

Программа визуализирует матрицу и предоставляет интерфейс для ее изменения, а также реализует дополнительные функции, такие как сохранение матрицы. Важно отметить, что весь контекст входных данных зависит от интерфейса, с которым эта программа взаимодействует, и описанные данные представляют лишь предполагаемые варианты.

#### Выходные данные

Данный код представляет собой часть приложения с графическим интерфейсом на Python с использованием библиотеки Tkinter. Как программный код для создания интерфейса, он не имеет явных текстовых выходных данных.

#### Руководство пользователя

##### Приложение для сравнения матриц

##### 1. Генерация факторов:

- запустите приложение, чтобы открыть главное окно;
- введите значения факторов в таблицу факторов;
- если факторы уже существуют, они будут обновлены; в противном случае, они будут добавлены.



Рис. 1. Главное окно программы

2. Генерация матриц:

- нажмите кнопку «Генерировать матрицу»;
- укажите количество элементов в матрице в специальном поле;
- новая матрица и соответствующие таблицы будут отображены в окне приложения.

3. Очистка данных:

- для очистки всех данных введите команду «Очистить матрицу»;
- после выполнения этой команды, все таблицы и матрицы будут удалены, и вы сможете создать новые.

4. Общие указания:

- все таблицы представлены в виде древовидных структур и могут быть изменены и обновлены согласно вашим потребностям;
- изменения, внесенные в таблицы факторов, будут отображаться в соответствующих столбцах таблиц матриц.

5. Выход из приложения:

- приложение завершит свою работу при закрытии основного окна.

Список использованных источников:

1. Титов А.Н. Python. Обработка данных : учебно-методическое пособие / А.Н. Титов, Р.Ф. Тазиева. – Казань : Изд-во КНИТУ, 2022. – 104 с. – URL: <https://znanium.com/catalog/product/2069264> (дата обращения: 20.12.2023). – Режим доступа: по подписке. – Текст: электронный.
2. Зыкова Г.В. Основы программирования на языке Python : учебно-методическое пособие / Г.В. Зыкова, А.С. Попов, Т.Н. Сапуглецева ; под. редакцией Г.В. Зыковой. – 2-е изд., стер. – Москва : ФЛИНТА, 2020. – 135 с. – URL: <https://znanium.com/catalog/product/1860057> (дата обращения: 20.12.2023). – Режим доступа: по подписке. – Текст: электронный.
3. Егоров А. Инди-курс программирования на Python / А. Егоров // Stepik: сайт. – URL: <https://stepik.org/course/63085/syllabus>.
4. Разумников С.В. Теория алгоритмов : методические указания к выполнению курсовой работы по дисциплине «Теория алгоритмов» для студентов очной формы обучения, обучающихся по направлению 09.03.03 «Прикладная информатика» / С.В. Разумников ; Юргинский технологический институт. – Юрга : Изд-во Юргинского технологического института (филиала) Томского политехнического университета, 2022. – 20 с.
6. Вестра Э. Разработка геоприложений на языке Python / Э. Вестра ; пер. с англ. А.В. Логунова. – Москва : ДМК Пресс, 2017. – 446 с. – URL: <https://znanium.com/catalog/product/1028127> (дата обращения: 20.12.2023). – Режим доступа: по подписке. – Текст: электронный.
7. Андреева О.В. Основы алгоритмизации и программирования на языке Python : учебник / О.В. Андреева, О.И. Ремизова. – Москва : Издательский Дом НИТУ «МИСиС», 2022. – 149 с. – URL: <https://znanium.com/catalog/product/1914825>.