МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «СРЕДНЯЯ ШКОЛА № 56 КИРОВСКОГО РАЙОНА ВОЛГОГРАДА»

Проект на тему:

Создание программы для создания графиков функций.

Проект выполнил: Бондаренко Александр Алексеевич,

обучающийся 10 класса МОУ СШ № 56

Руководитель проекта: Яровая Е.В.,

учитель информатики МОУ СШ № 56

Волгоград, 2024.

Содержание:

Введение……………………………………………………………….……3

Глава 1. Графический интерфейс (GUI) на языке программирования Python …………………………………………………………………….…5

1.1 Python Описание………………………………………………………..5

1.2. Tkinter…………………………………………………………………..5

1.3 Дополнительные библиотеки для GUI (wxPython / PyQt / Kivy)……5

1.4 IDE (Среда Разработки)……………………………………………………………..5

1.5 Управление зависимостями (pip/venv)……………………………………………...6

Глава 2. Разработка Основных Функций………………………………….7

2.1. Внедрение Функции Выбора Математической Функции…………...8

2.2. Добавление Функции Ввода Параметров……………………………9

2.3. Интеграция Возможности Выбора Темы…………...………………10

2.4. Интеграция Функции Отображения Графика……...……………….12

2.5. Добавление Функции Сохранения Графика в Файл……………….13

Глава 3. Практическая часть…………………………………….………..15

Выводы…………………………………………………………………….17

Список источников информации………………………………………...18

Тема моего индивидуального проекта - "Программа для создания графиков функций" - непосредственно связана с моим будущим профессиональным путем и предметом информатика. Использование подобных программ остается актуальным и востребованным в различных областях современного мира.

Вот несколько аспектов, подчеркивающих актуальность данной темы:

1. **Визуализация математических моделей**: Программы для создания графиков функций играют ключевую роль в визуализации математических моделей, что существенно помогает в исследованиях и обучении. Они обеспечивают более глубокое понимание формул и функций, а также их взаимосвязей.
2. **Обучение математике**: В образовательных целях такие программы несут большую ценность. Студенты могут использовать графики функций для изучения и практического применения математических концепций, что способствует более эффективному усвоению материала.
3. **Анализ данных и статистика**: Графики функций - это мощное средство для визуального анализа данных и выявления закономерностей, особенно в научных исследованиях и статистических анализах.
4. **Оптимизация процессов**: Эти программы могут использоваться для оптимизации различных процессов, включая инженерные расчеты и проектирование, благодаря возможности визуализации и анализа результатов.
5. **Разработка и тестирование алгоритмов**: Специалисты по разработке и тестированию алгоритмов могут успешно применять графики функций для визуального анализа работы алгоритмов, что упрощает процесс их отладки и оптимизации.

Цель: создать программу для созднаия графиков функций.

Задачи:

**1. Исследование**:

* + Найти простую библиотеку Python для сохранения графиков.
  + Узнать, какие форматы файлов нужны пользователям.

1. **Реализация**:
   * Написать код для сохранения графика.
   * Добавить кнопку "Сохранить" в приложение.
2. **Интеграция**:
   * Добавить функцию сохранения в приложение.
   * Убедиться, что кнопка "Сохранить" хорошо видна.

Глава 1.

Графический интерфейс (GUI) на языке программирования Python.

Python предоставляет множество инструментов для создания графического интерфейса пользователя (GUI), делая процесс разработки приложений более удобным и эффективным. В данной главе мы рассмотрим основные аспекты работы с GUI на языке программирования Python.

Пункт 1.1

Python Описание.

Python — высокоуровневый язык программирования, широко используемый для создания различных приложений, включая те с графическим интерфейсом. Зачем используется: Прост в изучении и написании кода, что делает его популярным выбором для разработки GUI-приложений.

Пункт 1.2

Tkinter

Tkinter — стандартная библиотека Python для создания графических интерфейсов. Зачем используется: Предоставляет базовые элементы управления и легко интегрируется с Python.

Пункт 1.3.

Дополнительные библиотеки для GUI (wxPython / PyQt / Kivy).

Дополнительные инструменты для создания более сложных и красочных элементов управления, расширяя возможности Tkinter. • Зачем используется: Предоставляют возможность создавать более разнообразные интерфейсы.

Пункт 1.4.

IDE (Среда Разработки).

Среда, в которой пишется код. • Примеры: PyCharm, VSCode, Thonny. • Зачем используется: Удобство в написании, отладке и управлении проектом.

Пункт 1.5.

Управление зависимостями (pip/venv).

Описание: Инструменты для установки и управления сторонними библиотеками. • Зачем используется: Обеспечивает легкость установки и обновления библиотек.

Глава 2.

Разработка Основных Функций

При разработке ключевых функций моего приложения, мой акцент был сосредоточен на создании эффективной и надежной реализации, направленной на обеспечение оптимального функционирования. Ниже представлены основные моменты, которые я учел:

1. Масштабируемость и Гибкость:
   * Я стремился создать функции, легко масштабируемые и адаптируемые к изменяющимся требованиям проекта.
   * Гибкость функционала обеспечивает простое внедрение дополнительных возможностей и расширение функциональности в будущем.
2. Обработка Исключений:
   * Я реализовал обработку возможных сценариев ошибок и исключений для более эффективного управления ошибками.
   * Улучшил стабильность приложения и предоставил более информативные сообщения об ошибках для пользователей.
3. Оптимизация Производительности:
   * Стремился к оптимизации кода функций для обеспечения высокой производительности при выполнении основных задач.
   * Использовал эффективные алгоритмы и структуры данных для ускорения обработки данных.
4. Интерфейс для Пользователя:
   * Создал понятный и удобный интерфейс для пользователя, предусмотрев ясные инструкции и обратную связь.
   * Обеспечил легкость взаимодействия с основными функциями приложения.
5. Тестирование и Отладка:
   * Каждая основная функция прошла тщательное тестирование для выявления и устранения возможных ошибок.
   * Обеспечил стабильную работу приложения в различных сценариях использования.

Эти усилия привели к тому, что основные функции моего приложения стали надежными, производительными и удобными в использовании.

Пункт 2.1.

Внедрение Функции Выбора Математической Функции

Внедрение функции выбора математической функции стало ключевым элементом для улучшения пользовательского опыта. Ниже представлены аспекты, которые я учел:

1. Интерактивный Интерфейс:
   * Я разработал интерактивный интерфейс, который позволяет пользователям легко выбирать желаемую математическую функцию из предоставленного списка.
   * Сделал процесс выбора простым и понятным, даже для пользователей без специальных знаний.
2. Список Вариантов:
   * Я предоставил разнообразные варианты математических функций в списке для выбора пользователей.
   * Обеспечил ясные описания каждой функции для более полного понимания.
3. Динамическое Обновление:
   * Реализовал динамическое обновление отображения в соответствии с выбранной математической функцией.
   * Это позволяет пользователям мгновенно видеть изменения и эффекты каждой выбранной функции.
4. Обработка Ввода:
   * Добавил обработку возможных ошибок при вводе, чтобы предотвратить неправильный выбор функции.
   * Обеспечил четкие инструкции и подсказки для помощи пользователям.
5. Эргономика:
   * Учел удобство использования, сделав функцию интуитивно понятной и приятной для использования.
   * Добавил возможность сохранения выбранных функций для повторного использования.

Эта функция не только добавляет разнообразие в использование приложения, но и делает его более интересным и адаптивным к различным потребностям пользователей.

Пункт 2.2.

Добавление Функции Ввода Параметров

Добавление функции ввода параметров предоставляет пользователям уникальную возможность настроить графики в соответствии с их индивидуальными требованиями. Ниже представлены ключевые аспекты этой функции:

1. Интерфейс для Ввода:
   * Я разработал четкий и удобный интерфейс для ввода параметров, обеспечивая текстовые поля или формы для числовых значений или коэффициентов.
   * Это создает удобную платформу для пользователей вводить необходимые данные.
2. Проверка на Валидность:
   * Добавил проверку на валидность введенных данных для избежания ошибок и обеспечения корректности параметров.
   * Предоставил информативные сообщения об ошибках для улучшения опыта ввода.
3. Обработка Различных Типов Параметров:
   * Учел возможность ввода различных типов параметров, таких как числа, координаты или другие значения.
   * Это позволяет более гибко использовать приложение в соответствии с потребностями пользователей.
4. Сохранение Введенных Параметров:
   * Добавил функционал сохранения введенных параметров для последующего использования.
   * Это особенно удобно при работе с несколькими графиками или при возвращении к предыдущим настройкам.
5. Ясные Инструкции:
   * Предоставил ясные инструкции и подсказки, чтобы пользователи понимали, как использовать функцию ввода параметров.
   * Это помогает избежать недопониманий и улучшает общий пользовательский опыт.

Эта функция придает нашему приложению уровень персонализации, что особенно важно для пользователей, желающих настроить графики в соответствии с их конкретными потребностями и данными.

Пункт 2.3.

Интеграция Возможности Выбора Темы

Интеграция возможности выбора темы — светлой или темной — представляет собой отличный шаг в сторону персонализации пользовательского опыта. Мы уделили внимание не только эстетике, но и комфорту восприятия, обеспечивая гибкость в использовании приложения. Вот некоторые детали, которые мы реализовали:

1. Пользовательская Схема Цветов:
   * Разработали две пользовательские цветовые схемы — светлую и темную, позволяя пользователям выбирать стиль, соответствующий их визуальным предпочтениям и условиям освещения.
   * Обеспечили возможность переключения между темами для лучшего комфорта в различных сценариях использования.
2. Простой Интерфейс Переключения:
   * Создали простой интерфейс для переключения между темной и светлой темой, такой как кнопка, переключатель или другой интуитивный элемент.
   * Обеспечили удобство пользователей в изменении цветовой схемы в соответствии с их предпочтениями.
3. Автоматическое Сохранение Настроек:
   * Добавили функцию автоматического сохранения выбранной темы, предотвращая потерю настроек при перезапуске приложения.
   * Обеспечили более непрерывный и удобный пользовательский опыт.
4. Оптимизация Чтения:
   * Учли особенности чтения на светлой и темной темах, подбирая соответствующие цвета для текста и фона.
   * Обеспечили оптимальный комфорт восприятия контента в зависимости от выбора пользователя.
5. Гармоничный Дизайн:
   * Убедились, что оба варианта тем имеют гармоничный дизайн, который сохраняет читаемость и эстетику приложения.
   * Обеспечили единообразие и приятное визуальное восприятие в любой выбранной теме.

Таким образом, функция выбора темы предоставляет пользователям дополнительный уровень персонализации, сделав приложение более удовлетворительным и адаптированным к их предпочтениям.

Пункт 2.4.

Интеграция Функции Отображения Графика

Интеграция функции отображения графика через простой процесс нажатия кнопки "Submit" предоставляет пользователям максимальную простоту в получении результатов. Мы уделили внимание удобству использования и ясности этого процесса. Вот как мы реализовали данную функцию:

1. Интуитивный Интерфейс:
   * Создал интуитивно понятный интерфейс, где пользователи после выбора функции и ввода параметров могут легко обнаружить кнопку "Submit".
   * Ясные и понятные шаги упрощают процесс для пользователей.
2. Мгновенное Отображение:
   * Организовал мгновенное отображение графика после нажатия кнопки "Submit", создавая немедленную обратную связь и позволяя пользователям видеть результаты своей работы.
3. Гибкость Визуализации:
   * Разработал гибкую систему визуализации, которая автоматически адаптируется к выбранным параметрам и функции.
   * Это обеспечивает точное и понятное представление данных, делая процесс визуализации более удобным.
4. Возможность Изменения:
   * Добавил возможность легко изменять параметры и функции, не выходя из режима отображения графика.
   * Это позволяет пользователям экспериментировать с различными вариантами, не перегружая процесс.
5. Экспорт Результатов:
   * Предоставил функцию экспорта или сохранения графика, чтобы пользователи могли легко поделиться результатами или использовать их в других контекстах.
   * Обеспечил пользователей возможность максимально использовать созданные графики в соответствии с их потребностями.

Эта функция делает процесс визуализации данных интуитивным и быстрым, обеспечивая пользователям максимальное удовлетворение от использования приложения.

Пункт 2.5.

Добавление Функции Сохранения Графика в Файл

Добавление функции сохранения графика в файл приложению придает дополнительное удобство и функциональность. Мы учли различные аспекты, чтобы сделать этот процесс максимально интуитивным и полезным для пользователя. Вот некоторые детали:

1. Форматы Файлов:
   * Предоставили выбор различных форматов файлов для сохранения графиков, таких как PNG, JPEG, PDF и другие.
   * Это позволяет пользователям выбирать наиболее удобный формат в зависимости от их потребностей и требований.
2. Интерфейс Сохранения:
   * Разработали простой и интуитивно понятный интерфейс для сохранения графика в файл.
   * Добавили ясные инструкции и подсказки, чтобы пользователи могли легко сохранить результаты своей работы.
3. Пользовательские Параметры:
   * Предоставили пользовательские параметры для настройки деталей сохраняемого файла, такие как разрешение, качество изображения и другие.
   * Это позволяет пользователям получать идеально подходящие файлы согласно их требованиям.
4. Сохранение в Облачное Хранилище:
   * Добавили возможность сохранять графики в облачные хранилища, такие как Google Drive, Dropbox и другие.
   * Обеспечили доступ к графикам из любого устройства и возможность совместного использования с другими пользователями.
5. Автоматическое Сохранение:
   * Внедрили функцию автоматического сохранения, чтобы пользователи могли быть уверены, что их работы сохранены, даже если произойдет сбой или ошибка.
   * Это предотвращает потерю данных и обеспечивает безопасность результатов пользовательской работы.

Эта функция значительно улучшает опыт пользователей, делая процесс сохранения и использования графиков более гибким и удобным.

Глава 3.

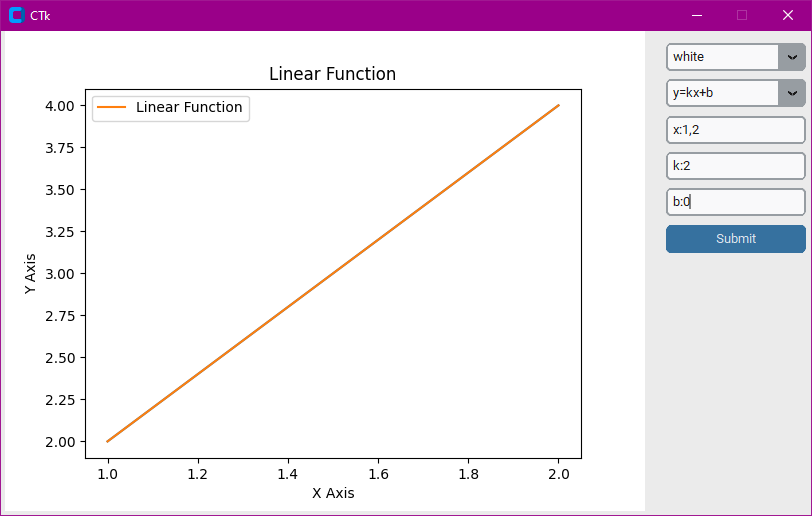
Практическая часть.

Создание программы будет происходит в двух файлах для удобства, в основном файле «main.py» будет создоваться логика программы и интерфейс, в второстепенном файле писаться функции для создания функции назавем его «functions.py».

В файле «main.py» начианем написание программы с того что импортируем нужные библиолтеки customtkinter, файл «functions» что бы использовать написаные в нем функции.

Тепер перейдем к файлу «functions.py» и импортируем библиотеку matplotlib, далее создадим с помощью документации к библиотеке самую простейщую графическую функцию y=xk+b, ее изоброжение мы сохроняем там где находиться сама программа в файле «.png».

Возврошаемся к основному файлу «main.py» создаем несколько полей ввода текста «x =ctk.CTkEntry(app, placeholder\_text="x:")» также делаем и для остальных значений, далее размеящем их с помощью функции place «x.place(relx=0.821, rely=0.175)» и повторяем для каждого поля ввода. Также создаем выбор функции с помощью выподающего поля выбора там буду храниться все доступные функции, это делаем с помощью создания переменной в которой будет храниться функция с значениями «options = ctk.CTkComboBox(app,state='readonly',values=["y=kx+b","y=x^n", "Option 3"],command=combobox\_callback)». Еще нужно сделать кнопку для генерации функции и выведения ее на окно программы «submit\_b=ctk.CTkButton(app, text="Submit", command=submit)». Теперь мы можем увидеть такой результат:



Если вы хотите ознакомиться с полной версией кода или же скачать саму программу то переходите по QR коду:

Выводы:

Разработка графического интерфейса (GUI) на языке программирования Python для моего индивидуального проекта оказалась увлекательным и познавательным опытом. Погружение в мир создания приложений с помощью Python открыло для меня множество возможностей и принципов, которые могут быть применены в будущей профессиональной деятельности.

Изучение основных инструментов для работы с GUI, таких как библиотека Tkinter и дополнительные инструменты wxPython, PyQt и Kivy, позволило мне понять принципы создания пользовательских интерфейсов и выбрать наиболее подходящий инструмент для моего проекта.

Важным шагом было также ознакомление с средами разработки (IDE), которые облегчают процесс написания и отладки кода. Это позволило мне эффективно использовать возможности языка Python и библиотек для достижения желаемых результатов.

Осознание роли графического дизайнера в создании привлекательного пользовательского интерфейса и управление зависимостями при работе с библиотеками Python также оказалось важным для успешной реализации проекта.

В целом, создание графического интерфейса на языке программирования Python стало интересным и познавательным опытом, который позволил мне приобрести ценные навыки и знания для будущей профессиональной деятельности в области информатики и разработки программного обеспечения.

Список источников информации.

Python.org. (<https://www.python.org/>)

Downey, A. (2015). Think Python. O'Reilly Media.

Python.org - Tkinter (https://docs.python.org/3/library/tkinter.html)

Roseman, M., & Winterbottom, N. (2000). An Introduction to Tkinter. New Mexico Tech.

wxPython.org (https://www.wxpython.org/)

PyQt (https://riverbankcomputing.com/software/pyqt/intro)

Kivy (https://kivy.org/)

JetBrains. PyCharm. (https://www.jetbrains.com/pycharm/)

Visual Studio Code. (https://code.visualstudio.com/)

Thonny. (https://thonny.org/)

Glade User Interface Designer. (https://glade.gnome.org/)

Qt Designer Manual. (https://doc.qt.io/qt-5/qtdesigner-manual.html)

Python Packaging Authority. pip. (https://pip.pypa.io/en/stable/)

Python Virtual Environments. (https://docs.python.org/3/library/venv.html)

Lutz, M. (2013). Learning Python. O'Reilly Media.

Документация customtkinter. (https://example.com/customtkinter)

Matplotlib.org. (https://matplotlib.org/)

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering.

O'Reilly, J., & Jones, C. (2010). Python Imaging Library Handbook. O'Reilly Media.

Python Imaging Library (PIL). (<https://pillow.readthedocs.io/en/stable/>)