Муницеальное Бюджетное

Общеобразовательное Учреждение

МБОУ Лицей №113

Программа для построения физических графиков на языке Python

Лицей № 113

Автор: Разманов Владислав

МБОУ Лицей №113

9 класс A , Дзержинский район г. Новосибирск

Консультант проекта : Сазыкина

Олеся Николаевна.

Учитель физики

ВВЕДЕНИЕ

Актуальность

Многие ученики и студенты сталкиваются с трудностями в построении физических и математических графиков. Это занимает много времени и может быть источником стресса. Также, учителя тратят много времени на пояснение того, как построить графики, вместо того чтобы сосредоточиться на объяснении нового материала. Поэтому создание программы для построения графиков с интерактивным интерфейсом может значительно облегчить учебный процесс и упростить обучение физике и математике.

Цель

Целью данного проекта является написание программы на языке программирования Python (версии 3.9) для создания различных физических графиков с интерактивным интерфейсом.

Задачи

1. Написание кода программы на Python.

2. Изучение различных подтипов графиков (гиперболы, параболы, прямые и т.д.).

3. Изучение встроенных и загружаемых библиотек Python для работы с графиками (например, matplotlib).

4. Создание интерактивного интерфейса для программы.

Реализация проекта

Программа будет создана с использованием языка программирования Python версии 3.9 и библиотеки matplotlib для построения графиков. Для создания интерактивного интерфейса будет использована библиотека tkinter.

Проект "Построение физических графиков с интерактивным интерфейсом" представляет собой программу на Python. Программа будет использовать библиотеку matplotlib для построения графиков и библиотеку tkinter для создания интерактивного интерфейса. Это позволит пользователям вводить формулы, названия осей и выбирать тип графика через простой и понятный интерфейс. Такой подход делает процесс построения графиков более доступным и удобным, особенно для начинающих пользователей.. Этот проект имеет большую актуальность в образовании и может значительно облегчить процесс обучения физике и математике.

При разработке программы будут использованы принципы объектно-ориентированного программирования (ООП), что позволит создать модульную и расширяемую систему. Это также обеспечит легкость поддержки и дальнейшего развития проекта.

Важным аспектом проекта будет обеспечение возможности сохранения построенных графиков в различных форматах (например, PNG, JPEG, PDF) для последующего использования в учебных материалах или отчётах.

Программа будет предоставлять возможность построения следующих типов графиков:

1. Графики функций: пользователь сможет ввести математическую функцию (например, y = x^2) и получить соответствующий график.

2. Графики траекторий движения: программа позволит строить траектории движения объектов, например, броска мяча под углом к горизонту или движения тела под действием силы тяжести.

3. Графики зависимости физических величин: пользователь сможет построить графики зависимости различных физических величин друг от друга, например, зависимость скорости от времени при равноускоренном движении.

План реализации

Интерактивный интерфейс программы будет включать в себя элементы управления для выбора типа графика, ввода математической формулы, задания названий осей и других параметров. Пользователь сможет масштабировать, перемещать и сохранять построенные графики. Программа будет написана с использованием языка программирования Python, который широко используется в научных и образовательных целях. Это обеспечит легкость в освоении и использовании программы для широкого круга пользователей. После завершения разработки программы, она будет протестирована на различных платформах, чтобы убедиться в её стабильной работоспособности. В случае необходимости будут внесены корректировки для оптимизации производительности и устранения возможных ошибок.

В целом, проект "Построение физических графиков с интерактивным интерфейсом" представляет собой значимый инструмент для обучения физике и математике. Он обеспечит ученикам и студентам возможность быстро и наглядно визуализировать различные физические закономерности, а также учителям - эффективно демонстрировать материал на уроках.

Этапы реализации:

1. Изучение библиотеки matplotlib для построения графиков.

2. Написание кода программы для построения различных типов графиков (гиперболы, параболы, прямые и т.д.) на основе введенной пользователем формулы.

3. Создание интерактивного интерфейса с использованием библиотеки tkinter, включающего поля для ввода формулы, названий осей и выбора типа графика.

4. Объединение кода для построения графиков и интерактивного интерфейса в единый проект.

5. Тестирование программы на различных примерах ввода данных.

План действий:

1. Изучение документации по библиотекам matplotlib и tkinter.

2. Написание кода программы для построения графиков на основе введенной формулы.

3. Создание интерактивного интерфейса с использованием библиотеки tkinter.

Тестирование программы на различных примерах.

Ожидаемые результаты

Ожидается получение программы на языке Python, которая позволит пользователям строить различные физические графики (например, гиперболы, параболы, прямые) с помощью ввода формулы, названий осей и выбора типа графика через интерактивный интерфейс.

Теоретическая часть

Сейчас вам будет представлена краткая информация о языке программирования и библиотеках которые мы будем использовать

Python - это высокоуровневый интерпретируемый язык программирования, который известен своей простотой и читаемостью кода. Он широко используется для веб-разработки, научных вычислений, анализа данных, искусственного интеллекта, автоматизации задач и многих других областей.

Python имеет простой и понятный синтаксис, что делает его отличным выбором для начинающих программистов. Он поддерживает различные парадигмы программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Python также обладает обширной стандартной библиотекой, которая предоставляет множество полезных инструментов для разработки.

Благодаря своей популярности и активному сообществу разработчиков, Python имеет множество сторонних библиотек и фреймворков, таких как Django, Flask, NumPy, Pandas, TensorFlow и многие другие, которые делают его универсальным инструментом для решения разнообразных задач. Python также доступен на различных платформах, что делает его удобным для разработки кроссплатформенных приложений.

PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Она предоставляет разработчикам широкий набор инструментов для удобной и эффективной работы над проектами на Python. PyCharm обладает множеством функций, которые делают разработку на Python более удобной.

Это включает в себя подсветку синтаксиса, автодополнение кода, интеграцию с системами контроля версий (например, Git), отладчик, автоматическое форматирование кода, анализ кода на наличие ошибок и многое другое. Также PyCharm поддерживает различные фреймворки и библиотеки Python, такие как Django, Flask, NumPy, Pandas, и многие другие. Одной из ключевых особенностей PyCharm является его удобная интеграция с другими продуктами JetBrains, такими как WebStorm, IntelliJ IDEA и PhpStorm. Это позволяет разработчикам работать над проектами, включающими различные языки программирования, используя единую среду разработки.

PyCharm доступен в двух версиях: Community Edition (бесплатная) и Professional Edition (платная). Professional Edition включает дополнительные возможности, такие как поддержка разработки веб-приложений, базы данных, JavaScript и многие другие. PyCharm также доступен на различных операционных системах, включая Windows, macOS и Linux.

Для создания нашего проекта мы будем использовать данное IDLE в категории Communiti Edition

Tkinter - это стандартная библиотека для создания графических пользовательских интерфейсов (GUI) в Python. Она предоставляет разработчикам инструменты для создания окон, кнопок, полей ввода, меню и других элементов интерфейса.

Tkinter основан на библиотеке Tk, которая является одной из самых популярных библиотек для создания GUI. Tkinter обеспечивает простой способ создания интерфейсов с использованием виджетов (widgets) и размещения их на окне приложения. Одной из основных особенностей Tkinter является ее простота в использовании.

С ее помощью можно легко создавать простые интерфейсы для приложений, а также более сложные GUI с использованием различных виджетов и макетов. Tkinter также поддерживает событийную модель программирования, что позволяет реагировать на действия пользователя, такие как нажатие кнопок или ввод текста. Благодаря своей стандартной поставке с Python, Tkinter является широко используемой библиотекой для разработки GUI-приложений на этом языке. Она предоставляет удобный способ создания кроссплатформенных приложений с использованием Python.

NumPy (Numerical Python) - это библиотека для языка программирования Python, предоставляющая поддержку многомерных массивов (включая матрицы) и большое количество математических функций для работы с этими массивами. NumPy также предоставляет эффективные инструменты для работы с линейной алгеброй, преобразованиями Фурье, генерации случайных чисел и другими задачами. Основным объектом в NumPy является многомерный массив (ndarray), который представляет собой таблицу элементов одного типа данных (обычно чисел). Массивы NumPy могут быть созданы из списков или других массивов, а также с использованием специальных функций для генерации массивов с определенной структурой. NumPy обеспечивает мощные операции над массивами, такие как индексирование, слайсинг, транспонирование, арифметические операции, логические операции и многое другое. Благодаря использованию оптимизированных алгоритмов на языке C, операции с массивами NumPy выполняются очень быстро, что делает эту библиотеку популярным выбором для вычислительно интенсивных задач. В дополнение к этому, NumPy также предоставляет функции для работы с линейной алгеброй, такие как умножение матриц, нахождение определителя, решение систем линейных уравнений и многое другое. NumPy часто используется вместе с другими библиотеками для научных вычислений в Python, такими как SciPy, Matplotlib, Pandas и другими. Вместе они образуют мощный инструментарий для анализа данных, визуализации и выполнения сложных вычислений.

Matplotlib.pyplot - это модуль библиотеки Matplotlib, который предоставляет возможность создания различных типов графиков и визуализации данных в языке программирования Python. Matplotlib является одной из самых популярных библиотек для визуализации данных и научной графики в Python. Matplotlib.pyplot позволяет создавать графики различных типов, такие как линейные графики, столбчатые диаграммы, круговые диаграммы, точечные графики, контурные графики, гистограммы и многое другое. Этот модуль обладает широким набором функций для настройки внешнего вида графиков, включая настройку осей, меток, заголовков, цветов, стилей линий и маркеров. Matplotlib.pyplot также поддерживает создание множества графиков на одном изображении, а также возможность сохранения графиков в различных форматах файлов, таких как PNG, PDF, SVG и другие. Matplotlib.pyplot часто используется вместе с другими библиотеками для научных вычислений в Python, такими как NumPy и Pandas, для визуализации данных из массивов или таблиц.

Конец формы

Практическая часть

В практической части проекта "Построение физических графиков с интерактивным интерфейсом" можно выделить несколько ключевых аспектов:

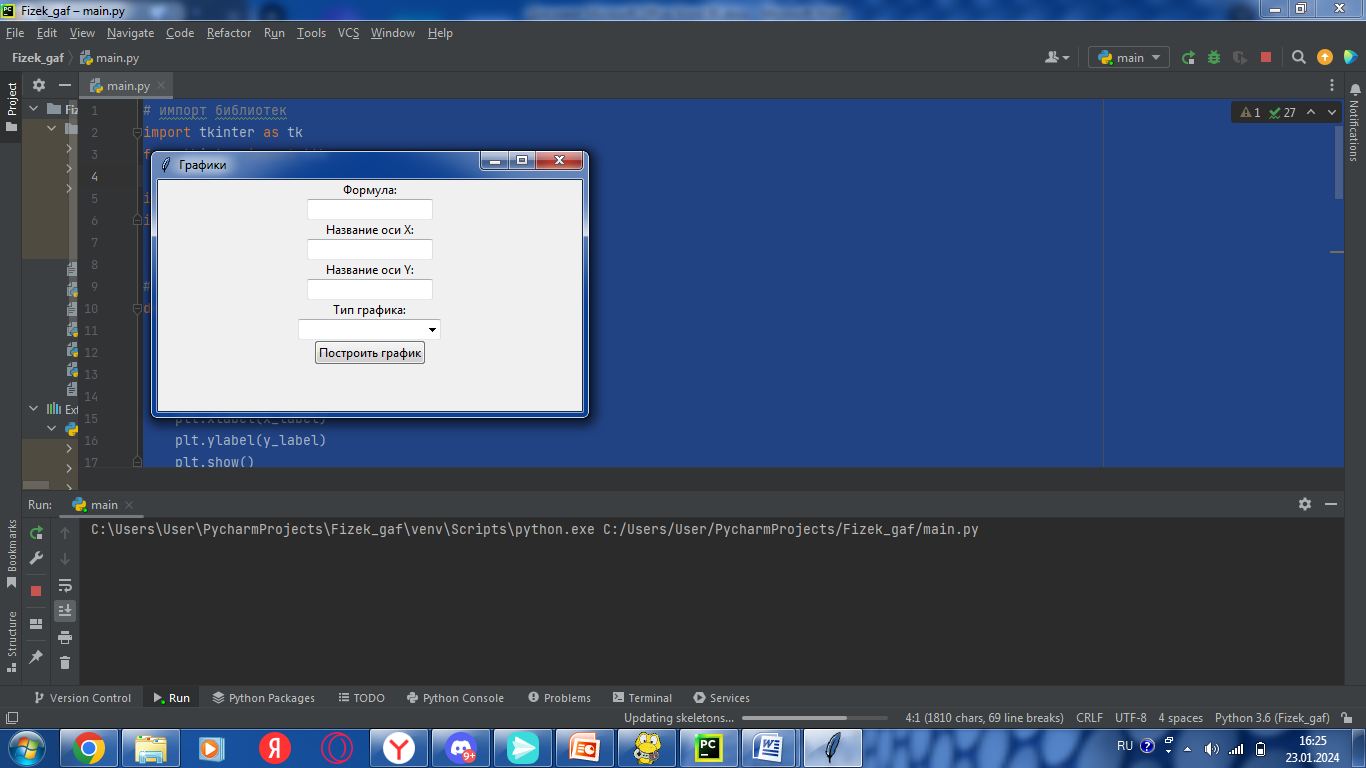
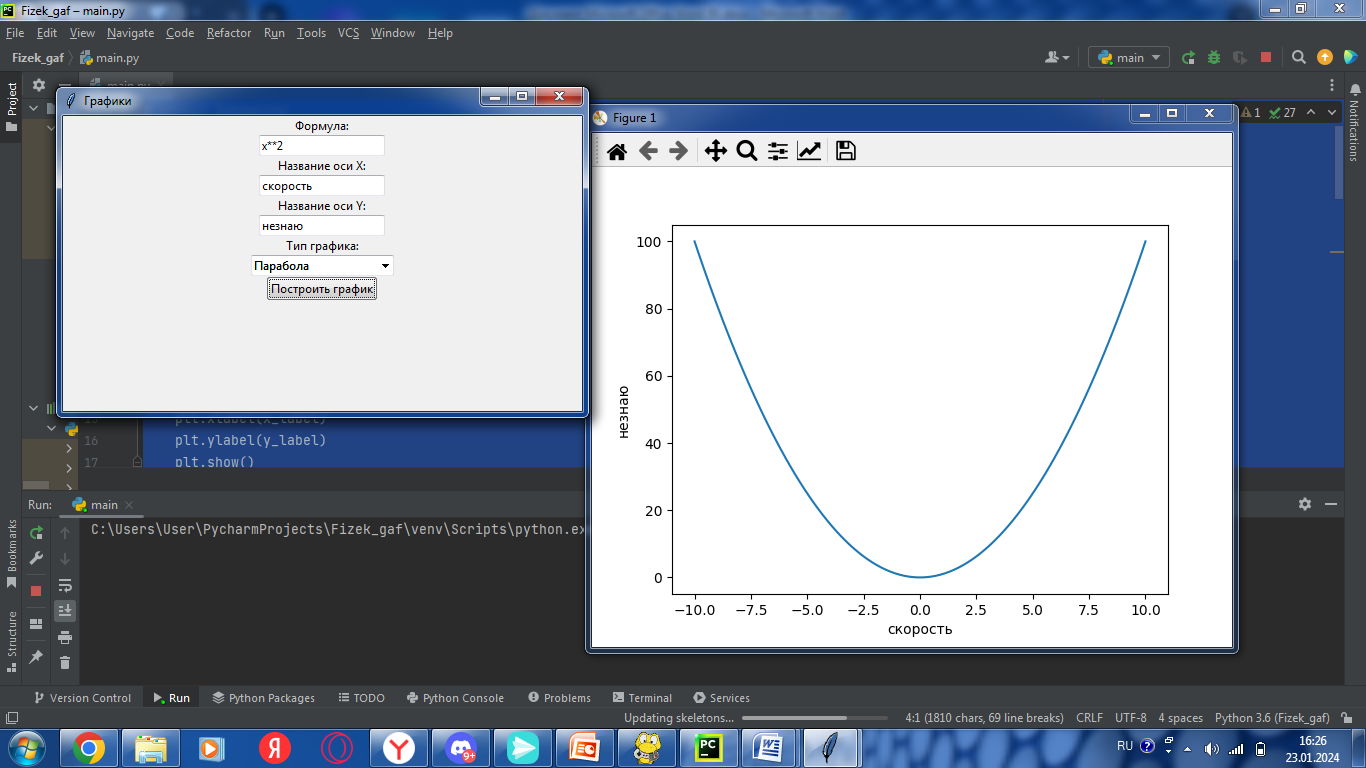
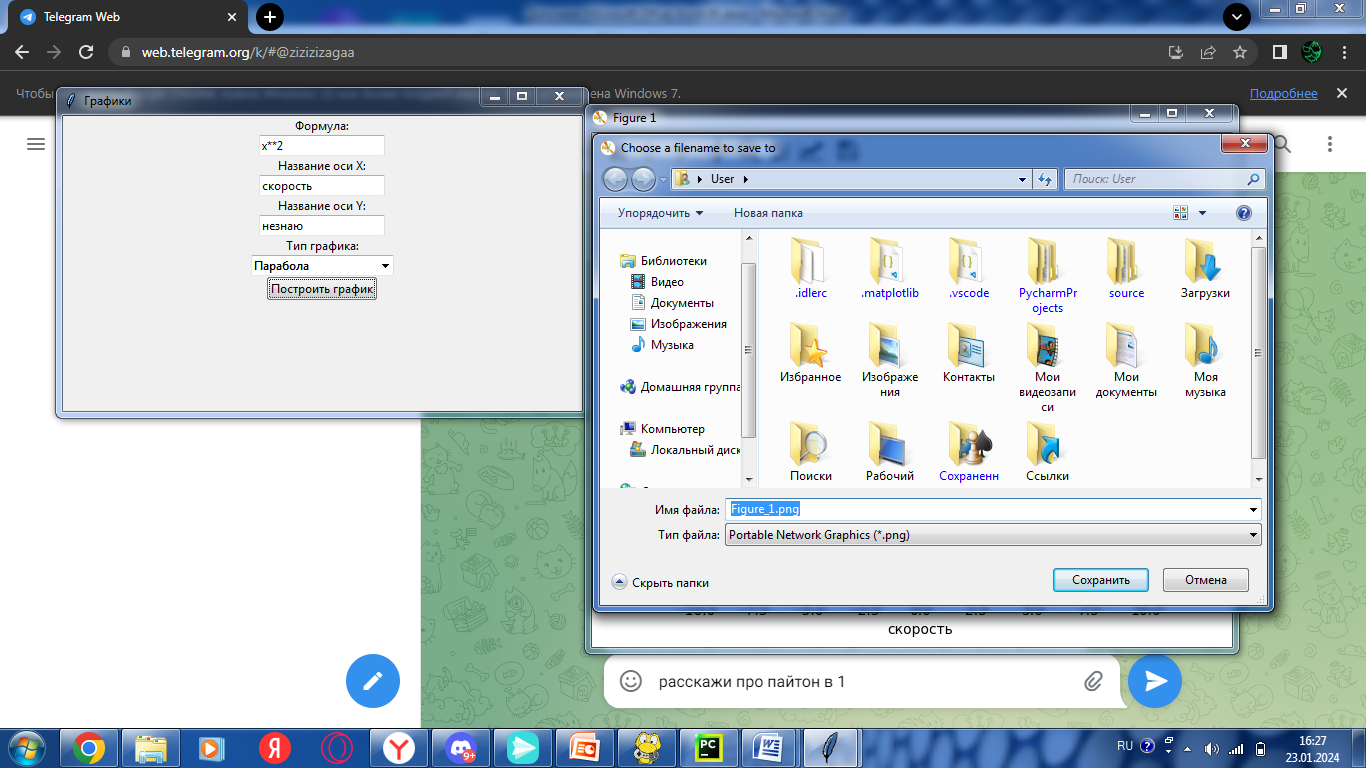
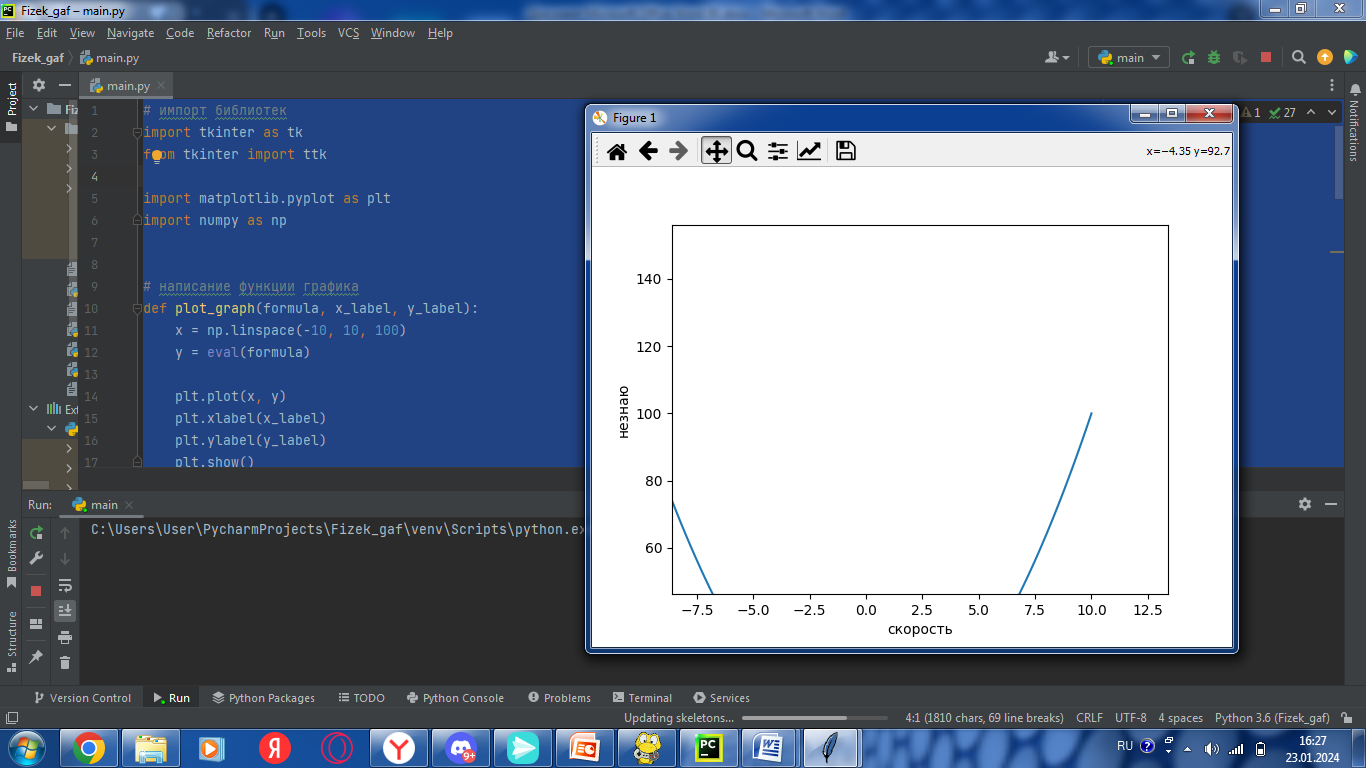
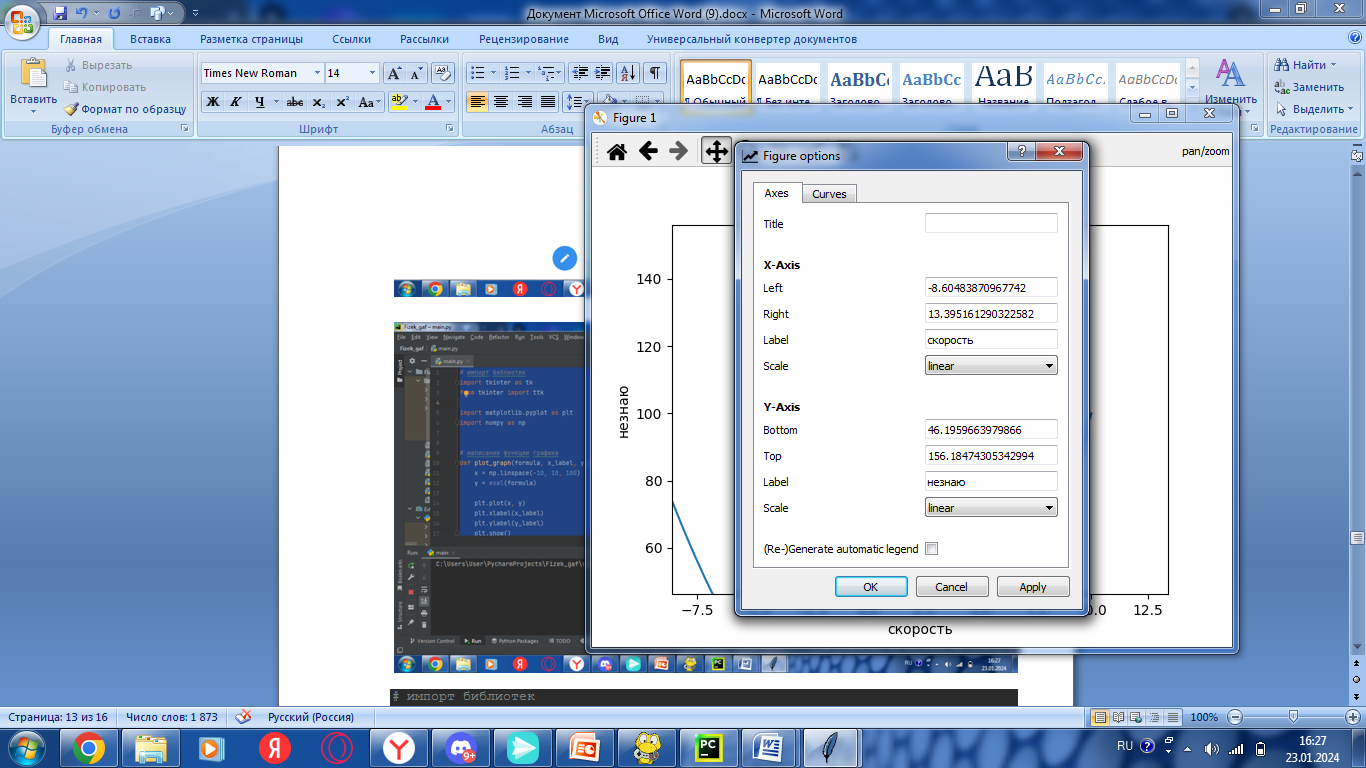
1. Разработка пользовательского интерфейса: важной частью проекта является создание интуитивно понятного и удобного интерфейса для построения физических графиков. Это включает в себя выбор методов ввода данных, настройку параметров графика, возможность масштабирования и перемещения осей координат, а также предоставление пользователю информации о построенных графиках.

2. Использование современных технологий программирования: разработка проекта требует использования современных технологий программирования, таких как JavaScript, HTML5, CSS, библиотеки для визуализации данных (например, D3.js) и другие инструменты для создания интерактивных элементов.

3. Работа с физическими моделями: практическая часть проекта также включает в себя реализацию физических моделей и алгоритмов для построения графиков, учитывая различные законы и зависимости в физике. Это может включать в себя моделирование движения тел, изменение физических параметров во времени, а также визуализацию различных физических явлений.

4. Тестирование и оптимизация: важной частью практической работы над проектом является тестирование созданного интерфейса и функционала на различных устройствах и браузерах, а также оптимизация производительности для обеспечения плавной работы при построении и взаимодействии с графиками.

Эти аспекты практической работы над проектом "Построение физических графиков с интерактивным интерфейсом" помогут обеспечить его эффективное функционирование и удовлетворение потребностей пользователей в области образования и научного познания.

# импорт библиотек  
import tkinter as tk  
from tkinter import ttk  
  
import matplotlib.pyplot as plt  
import numpy as np  
  
  
# написание функции графика  
def plot\_graph(formula, x\_label, y\_label):  
 x = np.linspace(-10, 10, 100)  
 y = eval(formula)  
  
 plt.plot(x, y)  
 plt.xlabel(x\_label)  
 plt.ylabel(y\_label)  
 plt.show()  
  
  
# создание события создания кнопки  
def on\_button\_click():  
 formula = formula\_entry.get()  
 x\_label = x\_label\_entry.get()  
 y\_label = y\_label\_entry.get()  
 plot\_graph(formula, x\_label, y\_label)  
  
  
# выбор типа графика  
def on\_graph\_type\_selected(event):  
 selected\_graph = graph\_type\_combobox.get()  
 if selected\_graph == "Гипербола":  
 formula\_entry.delete(0, tk.END)  
 formula\_entry.insert(0, "1 / x")  
 elif selected\_graph == "Парабола":  
 formula\_entry.delete(0, tk.END)  
 formula\_entry.insert(0, "x\*\*2")  
 elif selected\_graph == "Прямая":  
 formula\_entry.delete(0, tk.END)  
 formula\_entry.insert(0, "x")  
  
  
# вызов окна  
root = tk.Tk()  
root.title("Графики")  
  
formula\_label = ttk.Label(root, text="Формула:")  
formula\_label.pack()  
formula\_entry = ttk.Entry(root)  
formula\_entry.pack()  
  
x\_label\_label = ttk.Label(root, text="Название оси X:")  
x\_label\_label.pack()  
x\_label\_entry = ttk.Entry(root)  
x\_label\_entry.pack()  
  
y\_label\_label = ttk.Label(root, text="Название оси Y:")  
y\_label\_label.pack()  
y\_label\_entry = ttk.Entry(root)  
y\_label\_entry.pack()  
  
graph\_type\_label = ttk.Label(root, text="Тип графика:")  
graph\_type\_label.pack()  
graph\_type\_combobox = ttk.Combobox(root, values=["Гипербола", "Парабола", "Прямая"])  
graph\_type\_combobox.bind("<<ComboboxSelected>>", on\_graph\_type\_selected)  
graph\_type\_combobox.pack()  
  
plot\_button = ttk.Button(root, text="Построить график", command=on\_button\_click)  
plot\_button.pack()  
  
root.mainloop()

Заключение

В заключение, проект представляет собой инновационный инструмент, который объединяет современные технологии программирования с потребностями образования. Создание удобного и интуитивно понятного интерфейса для построения физических графиков поможет учащимся и преподавателям в изучении и преподавании физики и математики. Этот проект не только облегчит процесс создания графиков, но и способствует более глубокому пониманию физических явлений, благодаря визуализации различных законов и зависимостей. В результате, ученики смогут лучше усваивать материал, а преподаватели смогут более эффективно демонстрировать и объяснять сложные концепции. Благодаря использованию современных технологий программирования, проект обладает высокой гибкостью и расширяемостью, что позволит внедрять новые функции и поддерживать его актуальность в будущем. Таким образом, проект "Построение физических графиков с интерактивным интерфейсом" не только представляет собой важный инструмент для образования, но и демонстрирует потенциал технологий для улучшения процесса обучения и понимания науки.