Государственное бюджетное общеобразовательное учреждение
«Школа №1502» города Москвы

Индивидуальный проект

«Расширенный Калькулятор»

Работу выполнил

Ученик 8 Н класса ГБОУ школы 1502

Лопатин Антон Юрьевич

Руководитель проекта

учитель Информатики, ГБОУ г. Москвы Школа №1502,

Шаров Иван Юрьевич

Москва, 2026

Оглавление

Введение 3

Теоретическая часть 4

Анализ и подготовка материалов 4

Подготовка среды разработки 5

Практическая часть 9

Разработка приложения 9

Тестирование приложения 11

Написать сайт для распространения 11

Выбор фреймворка 11

Написание сайта 13

Выводы 13

Заключение 13

Список литературы 15

Введение

Актуальность проекта

В наше время существует множество калькуляторов с совершенно разным функционалом. Есть чисто алгебраический калькулятор, есть калькулятор с возможностью решать линейные уравнение, есть статистический калькулятор. Но совместного калькулятора, где есть одновременно все функции нет.

Цель работы

Разработать и распространить калькулятор с нужным функционалом для школьников/студентов

Задачи

1. Анализ и подготовка материалов
    a. Исследовать текущие калькуляторы их функциональность
2. Подготовка среды разработки
    a. Проанализировать доступные языки программирования и выбрать наилучший.
    b. Выбрать необходимые модули для расчетов
3. Разработка приложения
    a. Разработать UI/UX для данного приложения
    b. Продумать логику вычислений
4. Написать сайт для распространения
    a. Найти наилучший фреймворк
    b. Написать данный сайт

Теоретическая часть

Анализ и подготовка материалов

Чтобы разработать с функционалом, который удовлетворит школьникам и студентам, нужно для начала изучить калькуляторы, которые очень часто используют школьники, студенты и люди. Были выбраны для анализа такие калькуляторы Windows (Приложение) и веб калькулятор Desmos

Давайте посмотрим их плюсы и минусы:

1. Калькулятор для Windows:
    - Плюсы
        - Присутствие истории
        - Возможность запоминать числа
        - Возможность построить график по любому примеру
        - Возможность перевода и работы с другими системами счисления
        - Возможность перевода единиц
    - Минусы
        - Отсутствие справки по функциям
2. Desmos – научный калькулятор (WEB)
    - Плюсы
        - Возможность решать уравнения
        - Возможность нескольких действий в строку
        - Возможность вычисления среднего арифметического
    - Минусы
        - Отсутствие обратных тригонометрических функций

Так как эти калькуляторы являются самыми популярными на пк. Самое необходимое для школьников/студентов это: возможность построения графика, работа с разными системами счисления, решение уравнений, возможность выполнять несколько действий за один пример, возможность вычислять тригонометрию и статистика

Подготовка среды разработки

На данный момент разработано много языков программирования: Python, C++, Java, Kotlin. И давайте разберем его плюсы и минусы для разработки графического приложения

1. Python
   - Плюсы
     - Встроенная библиотека для работы с графическими приложений
     - Множество других библиотек для вычислений
     - Возможность быстро установить нужную библиотеку с помощью pip
     - Простота и читаемость
   - Минусы
     - Относительная медлительность.
2. C++
   - Плюсы
     - Высокая производительность
     - Кроссплатформенность
   - Минусы
     - Сложность изучения и использования
     - Необходимость ручного управления памятью
     - Долгое время компиляции
3. Java
   - Плюсы
     - Платформенная независимость
     - Высокая производительность.
     - Масштабируемость и надёжность.
   - Минусы
     - Сложность синтаксиса.
     - Большой объём кода.
     - Потребление памяти.
4. Kotlin
   - Плюсы
     - Чистота

- Безопасность
- Простота синтаксиса.

- Минусы
  - Низкая скорость компиляции.
  - Мало дополнений.

Исходя из всего этого мы можем выделить такие языки программирования: Python, Java, Kotlin

А теперь давайте посмотрим, что нам важно

- Легкость в программировании
- Множество библиотек
- Быстрота компиляции

Под все это из наших выделенных языков подходит Python.

Выбор библиотек для создания приложения

Для написания графического приложения калькулятора нужно использовать графические библиотеки, и математические – для написания логики вычислений

Изначально проект делался на Tkinter, но после многих добавлений было принято решение поменять на PyQt. Потому что:

1. Функционал
   a. Tkinter
      Tkinter предоставляет базовые виджеты (кнопки, метки, текстовые поля и т.д.) И простое управление макетом. Она подходит для небольших проектов и простых приложений, но может быть ограничена для более сложных проектов.
   b. PyQt
      - PyQt предлагает широкий спектр дополнительных функций, таких как пользовательские виджеты, интеграция с OpenGL, поддержка многопоточности и обширное управление макетом. Она подходит для сложных приложений и профессиональных проектов, где требуется расширенная функциональность.

2. Внешний вид
    a. TkinterВнешний вид Tkinter по умолчанию устарел и не всегда хорошо сочетается с современными операционными системами. Тем не менее, можно настроить внешний вид с помощью тем и стилей.
    b. PyQt

       - Приложения PyQt имеют собственный внешний вид на всех поддерживаемых платформах, что делает их визуально привлекательными и совместимыми с пользовательской ОС. PyQt также предоставляет расширенные параметры стиля, позволяющие осуществлять обширную настройку.
3. Производительность
    a. Tkinter

       Tkinter обладает достойной производительностью для приложений малого и среднего размера. Однако с большими проектами или приложениями, требующими быстрого обновления пользовательского интерфейса, могут возникнуть проблемы.
    b. PyQt

       - PyQt обладает лучшей производительностью по сравнению с Tkinter, что делает его подходящим для более крупных проектов и приложений, требующих быстрого обновления пользовательского интерфейса.

Для создания логики вычислений следует взять sympy за его преимущества, однако рассматривается использование библиотеки math в противовес

1. SymPy
    - Плюсы:
        - Работа с символьными вычислениями. SymPy позволяет сформулировать результаты вычислений в аналитическом виде (с помощью формул, системы выражений).
        - Поддержка различных разделов математики: геометрии, тригонометрии, логики и других. В SymPy есть функции для

алгебраического упрощения выражений, решения уравнений, дифференцирования, интегрирования, манипуляции с полиномами и матрицами.
    - Лёгкая интеграция с другими библиотеками Python, такими как NumPy и Matplotlib. Можно комбинировать символьные вычисления с численными и визуализировать их без предварительной настройки.
- Минусы:
    - Явное объявление символьных переменных. В SymPy символьные переменные и функции нужно объявить, как таковые, в то время как в модуле math они рассматриваются как числовые.
    - Ограничения на вычисления. Например, в SymPy квадратные корни чисел, которые не являются идеальными квадратами, не вычисляются.

2. Math
    - Плюсы:
        - Предоставление широкого набора математических функций и констант. Модуль math полезен, когда нужно выполнять точные вычисления с числами — от элементарной арифметики до более сложных операций, включая работу с тригонометрией и логарифмами.
        - Возможность импортировать только нужные функции. Иногда может потребоваться всего одна или несколько функций из модуля, что делает код более читаемым и может незначительно ускорить выполнение.
    - Минусы:
        - Ограничения на работу с большими объёмами данных или сложными структурами — такими как векторы, матрицы или таблицы чисел — возможностей модуля math может быть недостаточно.

Практическая часть

Разработка приложения

Для этого надо сначала сделать UI, для отображения программы. На версии с Tkinter я это делал в формате все в одном окне

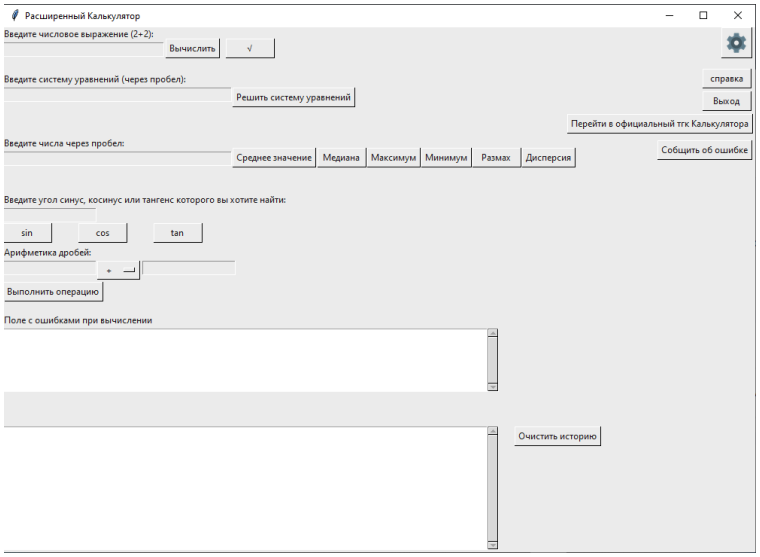

Рис. 1. Интерфейс Tkinter

Но после стало ясно, что если будет расширение мне стоит разделить интерфейс, и использовать другую библиотеку – PyQt6 самый простой способ - использовать функцию вкладок, которая из каждого класса забирает коробку (Layout),

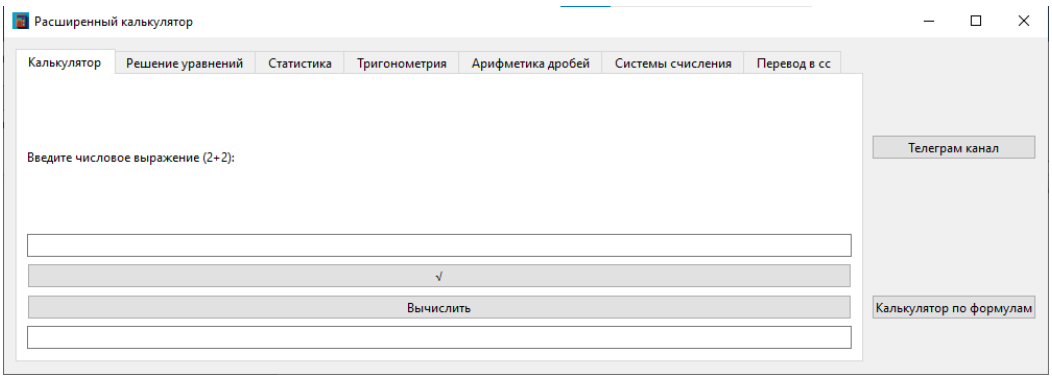

Рис. 2. Интерфейс PyQt6

После всего этого нужно реализовать логику вычислений. Для математики достаточно было eval, но он такой способ небезопасен

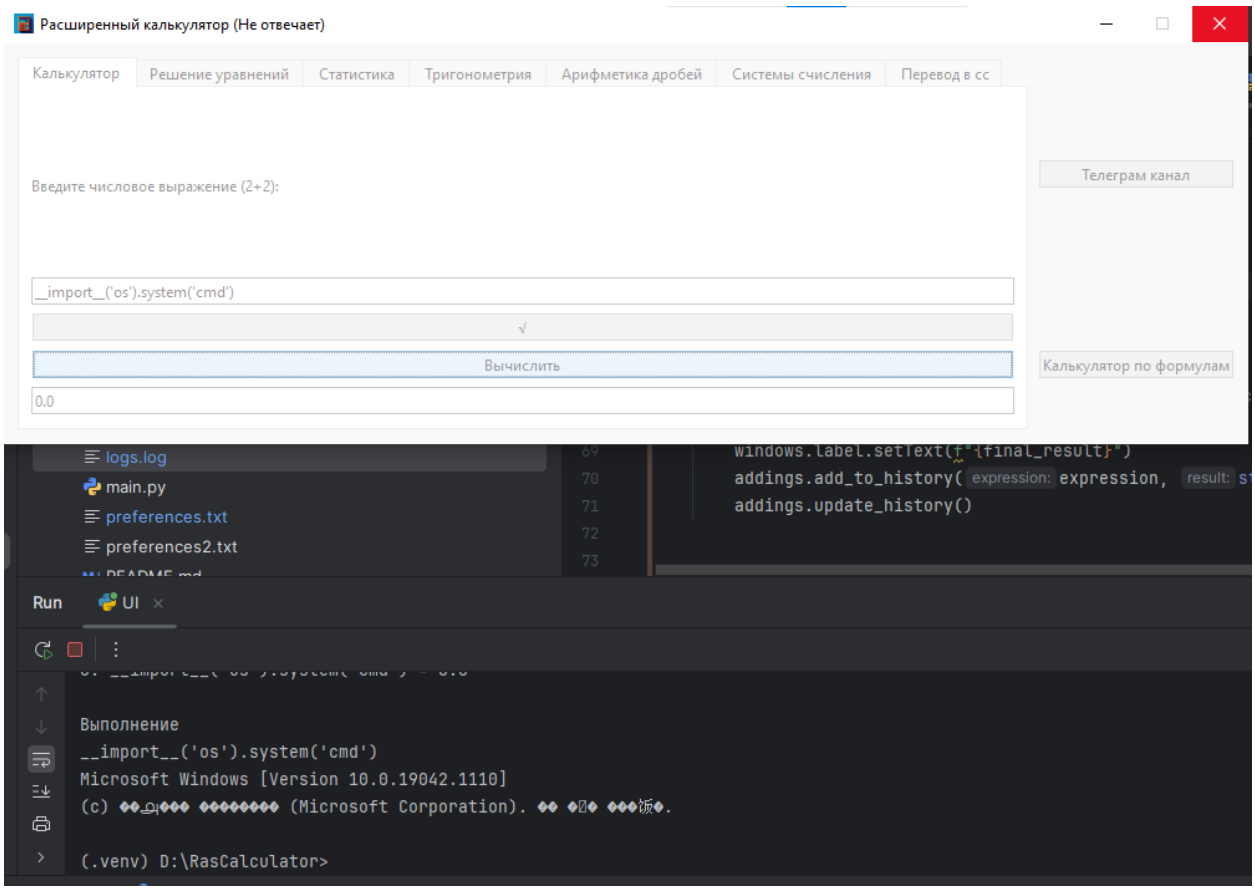

Рис. 3. Опасность использования eval (). Вызов командной строки через библиотеку os

Поэтому наилучшим решением было использовать evalf из библиотеки Sympy что делает безопасным использование приложения. Потому что эта функция использует классы Sympy. И SymPy имеет много математических функций такие как log, arc sin, ln. Также имеет возможность решать уравнения, что нам полезно для следующего блока, который я считаю самым сложным т. к. в нем есть множество видов выводов, если система не доопределена. Статистику я решаю через встроенные функции python: max min sum. Тригонометрия я считаю одним из сложных блоков, потому что sin cos и tan ожидают и выводят в радианах и поэтому надо дополнительно переводить результат в выражение. Следующий блок: Работа с другими системами счисления: сложение, вычитание, деление и умножение. Здесь самое сложное – перевод из 10 системы счисления в другие начиная с 2 до 36 (26 букв английского языка + 10 цифр) это делается через ascii_uppercase из встроенной библиотеки string.

Тестирование приложения

Тестирование приложения — критически важный этап разработки, обеспечивающий качество, надёжность и безопасность продукта. Его значимость проявляется в нескольких ключевых аспектах:

- Выявление и устранение ошибок
- Тестирование позволяет обнаружить баги, логические недочёты и сбои до релиза. Раннее нахождение проблем существенно снижает затраты на их исправление (по статистике, устранение дефекта на стадии тестирования в 5–10 раз дешевле, чем после выпуска).
- Повышение надёжности

Тестируются:

- устойчивость к нагрузкам (стресс-тестирование);
- обработка некорректных входных данных;
- восстановление после сбоев.
- Оптимизация производительности, выявляются:
- «узкие места» в коде;
- задержки отклика;
- избыточное потребление ресурсов (память, CPU).
- Ошибки в релизной версии могут привести к:
- негативным отзывам и потере аудитории;
- убыткам из-за простоя сервиса;

В итоге разработчик протестировал приложение и убедился, что все хорошо работает и сервис может спокойно выпускаться в интернет

Написать сайт для распространения

Выбор фреймворка

Так как в проекте был использован Python, то наилучшим решением будет использовать python и дальше. Он дает возможность написать сайт

самому используя библиотеки-помощники или использовать фреймворк для обработки, а отображение писать самому. Для python разработано 3 наиболее известных фреймворка – это Django, Flask, FastAPI. Django самый распространённый фреймворк, однако имеет один недостаток: замедленная работа. В противовес он имеет много плюсов

Плюсы Django

- Принцип «Всё включено»: Фраза «всё включено» означает, что большинство инструментов для создания приложения — часть фреймворка, а не поставляются в виде отдельных библиотек.
- Django содержит огромное количество функциональности для решения большинства задач веб-разработки. Вот некоторые из высокоуровневых возможностей Django, которые вам придётся искать отдельно, если вы предпочтёте микро-фреймворк:
- ORM: Миграции базы данных, Аутентификация пользователя, Панель администратора, Формы
- Стандартизированная структура: Django как фреймворк задаёт структуру проекта. Она помогает разработчикам понимать, где и как добавлять новую функциональность. Благодаря одинаковой для всех проектов структуре гораздо проще найти уже готовые решения или получить помощь от сообщества. Огромное количество увлеченных разработчиков поможет справиться с любой задачей гораздо быстрее.
- Приложения Django: Приложения в Django позволяют разделить проект на несколько частей. Приложения устанавливаются путём добавления в settings.INSTALLED_APPS. Этот подход позволяет легко интегрировать готовые решения. Сотни универсальных модулей и приложений очень сильно ускорят разработку. Взгляните на их список на сайте djangopackages.org.
- Безопасный по умолчанию: Django безопасен изначально и включает механизмы предотвращения распространенных атак вроде SQL-инъекций (XSS) и подделки межсайтовых запросов (CSRF). Подробнее об этом можно почитать в официальном руководстве по безопасности.
- Graphene-Django позволит легко добавить соответствующую функциональность в ваш проект. Модели, формы, аутентификация,

политики разрешений и другие функциональные возможности. Библиотека так же поставляется с утилитой для тестирования результата.

Flask для легких проект и имеет только один плюс из уже сказанных. FastAPI вообще для создания API (получил – отправил) и не желателен для массивных сайтов громадные сайты следует писать на Django

Написание сайта

На сайты должна быть представлена информация о приложении, которое пользователь хочет скачать, несколько скриншотов и сама страница для скачивания, разработчику необходимо иметь доступ к административной панели чтобы оттуда можно было добавлять версии для скачивания, так же разработчик хотел, чтобы можно было извещать пользователей о нововведениях

Также в Django есть возможность написать шаблон для всех сайтов и туда добавлять необходимое содержимое, что является удобным улучшением

Так как арендовать сервер слишком сложно разработчику для старта продукта следует использовать бесплатный хостинг такие как https://render.com

Так же можно внедрить Яндекс метрику для слежения посещений сайта

Выводы

Итог, было написано приложение для компьютера и сайт-распространитель. Лично я его активно использую в повседневной жизни. Во время разработки я научился использовать PyQt для создания графических приложений и библиотеку SymPy для создания вычислений. Научился создавать сайты на основе фреймворка Django.

Калькулятор активно скачивают и другие люди, что является показателем успешной разработки приложения.

Заключение

Наше приложения было разработано и успешно распространено. Во время работы было много сложных моментов и очень интересных. В будущем планируется:

- Сделать android версию

- Расширить функционал
- Оптимизировать сайт
- Сделать платную версию

Список литературы

1. Выбор правильной библиотеки GUI для ваших проектов на Python: Tkinter vs PyQt // Dzen.ru [Электронный ресурс]. Режим доступа: https://dzen.ru/a/ZGh-TREqSzYcWNT1
2. Плюсы и минусы Django // Habr.com [Электронный ресурс]. Режим доступа: https://habr.com/ru/articles/473042
3. Важность тестирования мобильных приложений // App72.ru [Электронный ресурс]. Режим доступа: https://app72.ru/blog/680-vazhnost-testirovaniya-mobilnykh-prilozhenij
4. Тестирование и 7 основных этапов его проведения // Neiros.ru [Электронный ресурс]. Режим доступа: https://neiros.ru/blog/code/testirovanie-i-7-osnovnykh-etapov-ego-provedeniya
5. Роль тестирования в разработке ПО // Edu.Eureca.ru [Электронный ресурс]. Режим доступа: https://edu.eureca.ru/about/articles/37
6. SymPy vs Mathematica // GitHub Wiki [Электронный ресурс]. Архивировано с оригинала 17 сентября 2021 г., режим доступа: https://web.archive.org/web/20210917031245/https://github.com/sympy/sympy/wiki/SymPy-vs.-Mathematica
7. Математическая библиотека Python SymPy // PythonRu.com [Электронный ресурс]. Режим доступа: https://pythonru.com/biblioteki/sympy-v-python
8. Как работать с библиотекой SymPy в Python // SkillBox Media [Электронный ресурс]. Режим доступа: https://skillbox.ru/media/code/kak-rabotat-s-bibliotekoy-sympy-v-python
9. Библиотека SymPy: символьные вычисления в Python // Pythonist.Ru [Электронный ресурс]. Режим доступа: https://pythonist.ru/biblioteka-sympy-simvolnye-vychisleniya-v-python
10. Титов А.Н., Тазиева Р.Ф. Символьные вычисления в Python. Основы работы с библиотекой SymPy // KSU.RU [Электронный ресурс]. Режим доступа: https://moodle.kstu.ru/pluginfile.php/605576/mod_resource/content/1/Titov-Simvolnye_vychisleniya_Python_2023.pdf

11. Символьные вычисления средствами Python. Часть 1. Основы // Habr.com [Электронный ресурс]. Режим доступа: https://habr.com/ru/articles/423731
12. Математические функции и модуль math // Metanit.com [Электронный ресурс]. Режим доступа: https://metanit.com/python/tutorial/6.2.php
13. Python Math Module // GeeksforGeeks.org [Электронный ресурс]. Режим доступа: https://www.geeksforgeeks.org/python/python-math-module
14. Модуль math в Python: функции для решения математических задач // Sky.Pro [Электронный ресурс]. Режим доступа: https://sky.pro/wiki/media/kak-rabotat-s-modulem-math-v-python