Муниципальное бюджетное образовательное учреждением

«Гатчинский лицей № 3 имени Героя Советского Союза А.И.Перегудова»

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

“Упрощение работы с системами счисления с помощью электронного приложения”

Выполнил:
ученик 10-1 класса
Ручьев Григорий Викторович

Руководитель:
Татьяна Михайловна Черникова

Работа допущена к защите “__” _____ 2023 г

Руководитель: _____ /Т. М. Черникова/

г. Гатчина
2023

СОДЕРЖАНИЕ

Введение

Все мы когда-то учились на уроках информатики работе с системами счисления: переводу из одной системы счисления в другую и выполнению различных математических операций с ними, но не всегда это бывает просто и легко, ведь эта тема имеет прямое отношение к математической теории чисел. Однако в школьном курсе математики она, как правило, не изучается.

Необходимость изучения этой темы связана с тем фактом, что в повседневной жизни мы используем числа, которые являются объектами различных систем счисления. Они используются всегда, когда появляется потребность в числовых расчётах, начиная с вычислений младшего школьника, выполняемых карандашом на бумаге, кончая вычислениями, выполняемыми на суперкомпьютерах.

Существует несколько способов упростить работу с системами счисления. Например, выбор наиболее простого и удобного способа перевода или алгоритма для выполнения математических действий. Но, просмотрев множество ресурсов, я нашел однотипные алгоритмы перевода, которые никак не упрощали работу. Поэтому я решил создать многофункциональное приложение, которое сильно упростит и ускорит работу с системами счисления.

Целью моего проекта является – создание удобного приложения для работы с системами счисления.

Для выполнения данной цели я поставил перед собой следующие задачи:

1.Изучить теоретический материал по системам счисления;

2. Изучить основы создания и дизайна приложений;

3. Написать код приложения.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Что такое системы счисления и их виды

Система счисления (англ. numeral system или system of numeration) — символический метод записи чисел, представление чисел с помощью письменных знаков.

Система счисления:

1. Даёт представления множества чисел (целых и/или вещественных);
2. Даёт каждому числу уникальное представление (или, по крайней мере, стандартное представление);
3. Отражает алгебраическую и арифметическую структуру чисел.

Системы счисления подразделяются на:

1. Позиционные;
2. Непозиционные;
3. Смешанные.

1.1.1. Позиционные системы счисления

В позиционных системах счисления один и тот же числовой знак (цифра) в записи числа имеет различные значения в зависимости от того места, где он расположен. Изобретение позиционной нумерации, основанной на поместном значении цифр, приписывается шумерам и вавилонянам; развита была такая нумерация индусами и имела неоценимые последствия в истории человеческой цивилизации. К числу таких систем относится современная десятичная система счисления, возникновение которой связано со счётом на пальцах. В средневековой Европе она появилась через итальянских купцов, в свою очередь заимствовавших её у арабов.

Наиболее часто употребляемыми в настоящее время позиционными системами являются:

2 — двоичная (в дискретной математике, информатике, программировании);

3 — троичная;

8 — восьмеричная;

10 — десятичная (используется повсеместно);

12 — двенадцатеричная (счёт дюжинами);

16 — шестнадцатеричная (используется в программировании, информатике);

20 — двадцатеричная;

60 - шестидесятеричная (единицы измерения времени, измерение углов и, в частности, координат, долготы и широты).

В позиционных системах чем больше основание системы счисления, тем меньшее количество разрядов (то есть записываемых цифр) требуется при записи числа.

1.1.2. Непозиционные системы счисления

В непозиционной системе счисления изменение положения символа в числе не влияет на значение самого числа. Отличие позиционных и непозиционных систем хорошо видно при сравнении арабских и римских чисел. Числа, записанные арабскими цифрами, составляются в позиционной системе. И здесь важно учитывать понятие разрядности. Одна и та же цифра, в зависимости от того, в каком разряде числа она записывается, обозначает разную числовую величину. Например, в числе 234 цифра 2 обозначает величину двести, а в числе 324 — соответствует двадцати.

В римской системе счисления, цифра, в какое положение ее не помещай, всегда означает одно и то же. Например, с помощью римских цифр V и I, эквивалентных арабским 5 и 1, можно составить числа VI и IV, что соответствует 6 и 4. В непозиционной системе расположение цифры никак не влияет на её значение.

История возникновения непозиционных систем счисления уходит корнями в глубокую древность. Жители древних государств: Вавилона, Майя, Древнего Египта, Греции и Рима, пользовались непозиционным принципом в составлении чисел. Некоторые из таких систем, например, римские цифры, используются и по сей день.

1.1.3. Смешанные системы счисления

Смешанные системы счисления - это такие системы, в которых числа, заданные в системе счисления с основанием P изображают с помощью цифр другой системы с основанием Q, где Q < P. Такая система называется (Q-P)-ичной со старшим основанием P и младшим основанием Q.

Наиболее известным примером смешанной системы счисления является представление времени в виде количества суток, часов, минут и секунд. День (86400 секунд.) = 24 (часа) * 60 (минут) * 60 (секунд). Так можно представить минуты и секунды.

1.2 Язык программирования Python

Python — популярнейший язык, используемый в разных ИТ-сферах: и в машинном обучении, и при разработке программных приложений, и для написания скриптов автоматизации, и во многих других случаях. Python является высокоуровневым языком, который доминирует во многих областях. Его можно смело назвать универсальным языком общего назначения. Благодаря своей универсальности, “Пайтон” выделяется среди других языков (languages), вызывая большой интерес. Поклонники Python нередко называют его языком номер 1 (Python — is a language № One). А современные рейтинги и сервисы статистики PYPL, TIOBE и statista.com отчасти это подтверждают, т. к. там “Питон” стабильно находится в топ-3, причем уже не первый год подряд. Благодаря своему синтаксису, язык программирования Python всегда выделялся на фоне остальных.

Причины:

— отсутствие избыточности;

— схожесть с обычным английским, облегчающая понимание кода программы на Python;

— минимизация объема кода, который приходится писать;

— не надо применять такие символы, как “;”, “{“, “}”;

— для обозначения вложенности блоков в Python используются отступы, что тоже повышает читаемость кода, заодно приучая и к правильному оформлению.

Во многом такая простота обусловлена за счет того, что Python создавали на основе ABC — языка, который применялся в целях обучения программированию (programming), а также для повседневной работы тех людей, которые программистами не являются. То есть при желании “Питон” поймет любой, поэтому его нередко рекомендуют к

изучению в качестве первого языка. Таким образом, Python сочетает и простоту, и мощный инструментарий, причем эта простота не мешает использовать его для решения сложных задач. По сути, вы сможете применять его в целях создания прототипа почти любой программы. Также в целях ускорения разработки, к примеру, ту часть программы, которая принципиально не влияет на скорость работы всей программы, нередко пишут на “Питоне”. В “Питон” встроено настолько много возможностей, что кажется, что легче перечислить то, что нельзя сделать с помощью этого языка программирования, чем то, что можно. Python подходит для решения широкого круга задач и применяется на всех популярных платформах.

Во всей ИТ-сфере больше всего вакансий (более 500 тыс) на языке программирования Python. На нём написаны такие приложения, как Blender, Battlefield и т.д. Так что, изучив этот язык программирования, можно не беспокоиться о том, что в будущем можно остаться без работы. Поэтому реализация этого проекта будет для меня очень полезна. Это позволит мне понять и попробовать, что такое “настоящее” программирование на языке Python.

И именно благодаря своей простоте язык хорошо прижился и занял лидирующее место в области Machine Learning. Ведь люди, которые связаны с наукой, очень не любят тратить много времени на такие вспомогательные вещи, как написание кода, поэтому для них “Питон” подошел просто идеально, позволяя успешно реализовывать поставленные задачи.

1.2.1. Библиотека TKinter языка Python

Tkinter (от англ. tk interface) - это графическая библиотека, позволяющая создавать программы с оконным интерфейсом. Эта библиотека является интерфейсом к популярному языку программирования и инструментом создания графических приложений

tcl/tk. Tkinter, как и tcl/tk, является кроссплатформенной библиотекой и может быть использована в большинстве распространённых операционных систем (Windows, Linux, Mac OS X и др.).

ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Среда разработки и язык программирования

С языком программирования Python я довольно давно уже знаком, поэтому и выбрал его для реализации своего проекта. Также в Python есть графическая библиотека TKinter, с помощью которой я буду создавать свое приложение с оконным графическим интерфейсом. Чтобы работать с данным языком, существует множество сред программирования, но я остановился на одной - PyCharm, так как у данной среды понятный, удобный и многофункциональный интерфейс. Еще у данной среды есть автодополнение кода и вывод информации о встроенных методах и функциях языка Python, что сильно помогло при написании кода.

2.2. План действий

После выбора необходимого ПО, языка программирования и необходимых библиотек, я решил составить план действий по созданию приложения, который представлен ниже:

1) Изучить основы создания и дизайна приложений с помощью библиотеки TKinter;

2) Подобрать и визуализировать начальный дизайн приложения;

3) Создать первоначальный интерфейс и функционал приложения;

4) Протестировать первую версию приложения;

5) Доработать недостатки и завершить дизайн приложения.

2.3. Начало создания приложения

После изучения пособий по созданию приложений и их дизайну, я начал создавать первоначальный дизайн на бумаге(рис. 1), а затем воспроизвел тоже самое, но уже в среде программирования(рис. 2, 3, 4). По моей задумке, приложение будет состоять из 3 окон: главного меню, переводчика и калькулятора.

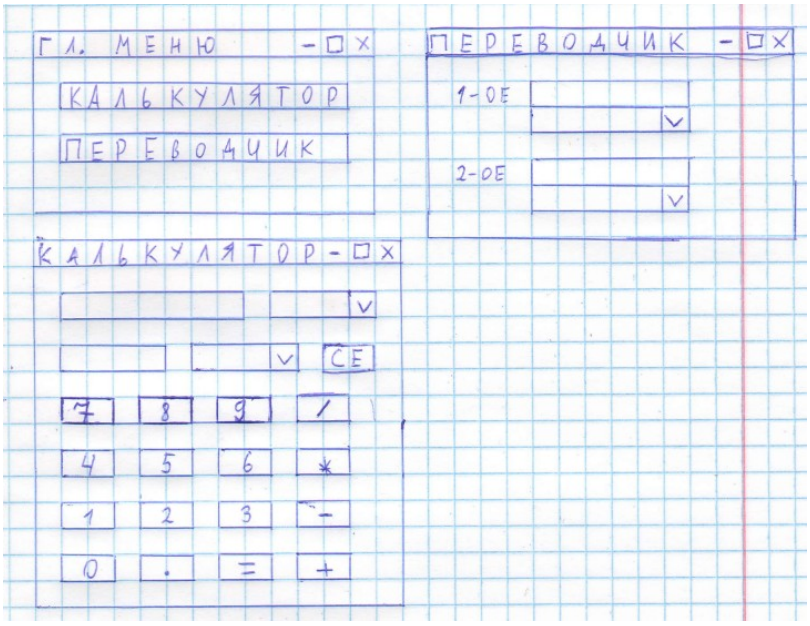

Рис. 1 - Первый дизайн на бумаге

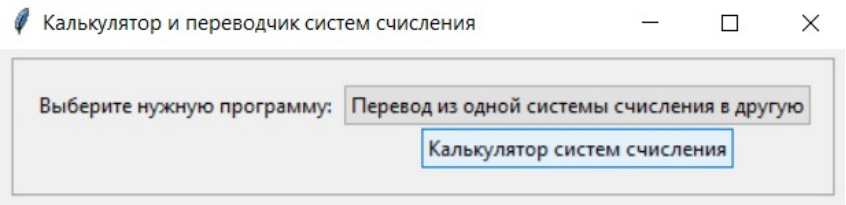

Рис. 2 - Главное меню

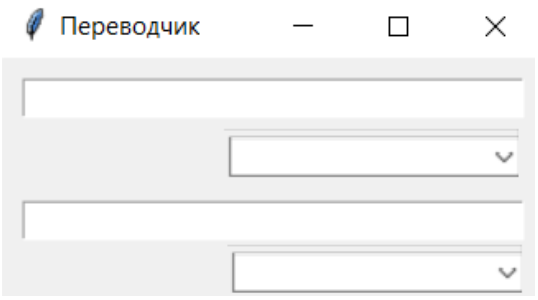

Рис. 3 - Переводчик

Рис. 4 - Калькулятор

Следующим этапом стало написание функционала и логики для каждого элемента(виджета) приложения, что заняло больше всего времени и большую часть кода. Первым делом я написал алгоритм перевода чисел из одной системы счисления в другую в диапазоне от 2-ичной до 36-ричной(рис. 5). Этот алгоритм будет в дальнейшем использован как в переводчике систем счисления, так и в калькуляторе.

```
def conv(self1, m, n, j):
    digs = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    a1 = 0
    z = str(m).upper()

    for a in z:
        k = digs.find(a)
        a1 = a1 * n + k
    r = ""

    while (a1 > 0):
        k = a1 % j
        r = digs[k] + r
        a1 = a1 // j

    return r
```

Рис. 5 - Алгоритм перевода

Вторым этапом стала логика приложения, которую я строил на самописных функциях и событиях(ивентах), после которых эти функции выполнялись.

2.4. Тестирование и доработка приложения

В ходе тестов первой версии приложения я выявил несколько недоработок и багов, поэтому было решено доработать приложение и исправить все ошибки. Когда приложение полностью работало без ошибок, я решил обновить дизайн, сделать его более приятным, а также немного подкорректировать интерфейс. В новом дизайне я добавил больше красок, логотип, а также возможность поменять светлую и темную тему(рис. 6, 7, 8, 9).

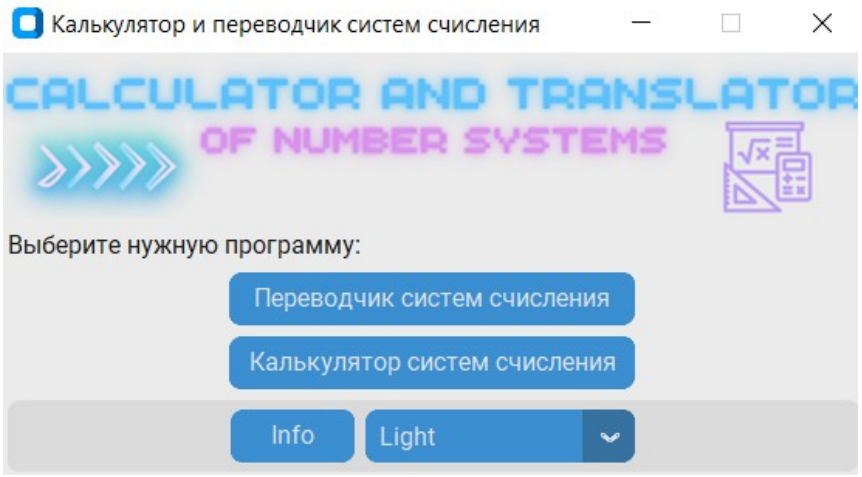

Рис. 6 - Главное меню(Светлая тема)

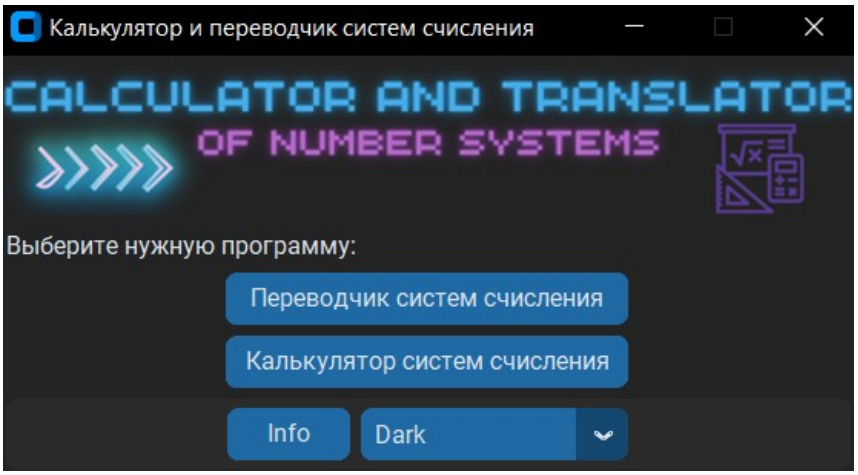

Рис. 7 - Главное меню(Тёмная тема)

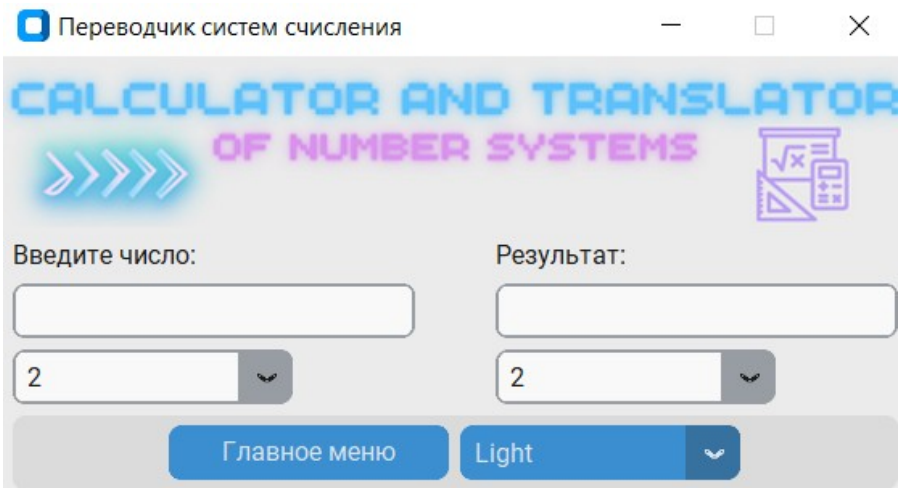

Рис. 8 - Переводчик

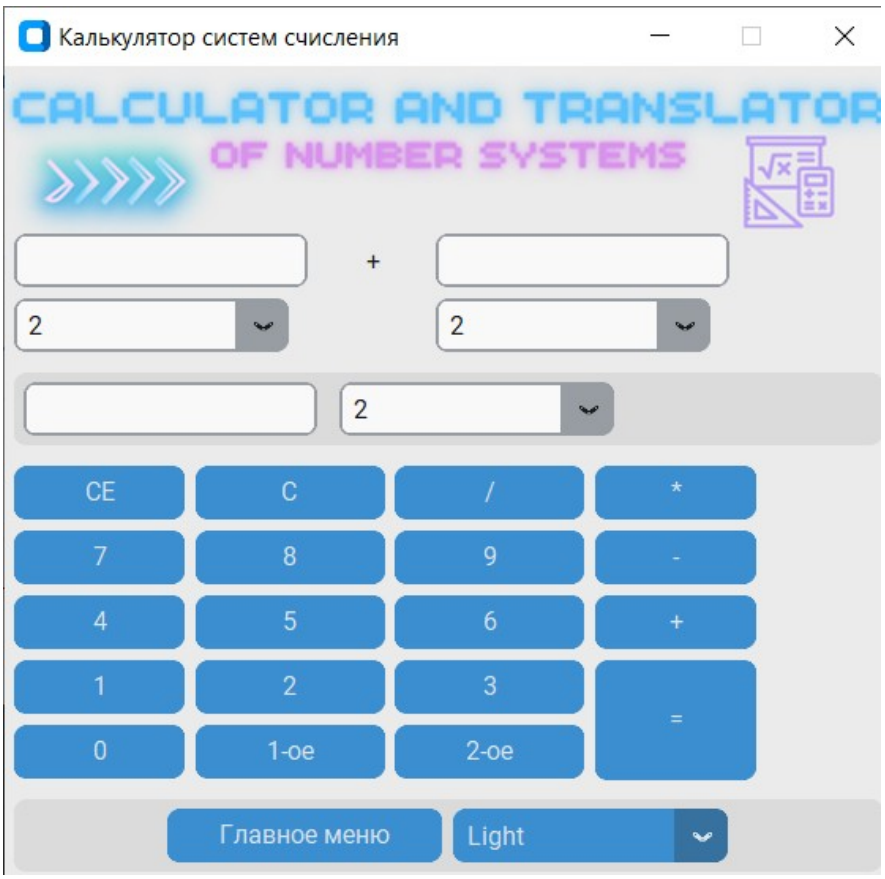

Рис. 9 - Калькулятор

Приложение для ОС Windows уже готово(Запускать файл Project1.exe) и скачать его можно по QR-коду (рис. 10)

Рис. 10 - QR-код

Заключение

В ходе работы над проектом я достиг свою цель, которая заключалась в создании приложения.

Для достижения этой цели я выполнил все поставленные задачи, а именно:

-Изучил теоретический материал;

-Научился создавать приложения с оконным графическим интерфейсом, изучил основы дизайна;

-Написал код для приложения.

В процессе создания приложения я получил много новых знаний и навыков, которые смогу применять в будущих работах.

Интернет-источники

1) https://python.org (документация Python)
2) https://ru.wikipedia.org (википедия)
3) https://customtkinter.tomschimansky.com (Документация библиотеки Tkinter)
4) https://www.onlinegdb.com/online_python_interpreter (Онлайн компилятор Python)
5) https://habr.com/ru/all/ (Примеры работы с библиотекой Tkinter)

ПАСПОРТ ПРОЕКТА

Название проекта: Упрощение работы с системами счисления с помощью электронного приложения

Руководитель проекта: Черникова Татьяна Михайловна

Автор проекта: Ручьев Григорий

Учебная дисциплина: Информатика (программирование)

Тип проекта: Практико-ориентированный

Цель работы: Создание удобного приложения для работы с системами счисления

Задачи работы:

1.Изучить теоретический материал по системам счисления;
2. Изучить основы создания и дизайна приложений;
3. Написать код приложения

Проблема проекта: Сложность в выполнении перевода и математических операций с числами в различных системах счисления

Краткое содержание проекта: Разработка приложения, которое поможет в работе с системами счисления, а код приложения будет написан в приложении PyCharm

Результат проекта (продукт): Электронное приложение

Реализация проекта: Поэтапное выполнение всех задач, получение новых знаний и их использование в проекте