

Муниципальное бюджетное общеобразовательное учреждение  
«Школа № 102 с углубленным изучением отдельных предметов»  
городского округа Самара

## **ПРОЕКТ**

**ПРЕДМЕТ:** Информатика

**ТЕМА:** Создание приложения в Python

**Выполнил:** обучающийся 10а класса

Котов Михаил Алексеевич

**Руководитель:** учитель информатики высшей категории

Григорьева Алёна Федоровна

Самара, 2023

## Содержание:

Введение.....	3
<b>Глава 1. Предметная область.....</b>	<b>4</b>
п.1.1. Анализ предметной области.....	4
<b>Глава 2. Создание калькулятора.....</b>	<b>.....</b>
п.2.1. Алгоритм написания калькулятора.....	.....
п.2.2. Создание приложения.....	.....
Заключение.....	.....
Список литературы.....	.....

## Введение

**Актуальность:** Прикладные программы являются неотъемлемой частью жизни каждого пользователя персонального компьютера, такие программы отвечают за выполнения множества задач, таких как написание текстовых документов, обработка аудио или видео файлов, а также выполнение математических расчетов. Подобные приложения позволяет решать огромное количество задач быстрее и точнее, чем это смог бы обычный человек и является существенным фактором в развитие различных сфер деятельности человека, начиная с простого ведения учета расходов за месяц и заканчивая запуском ракет в космос. Анализируя «рынок программистов», увидел, что язык Python востребованный и часто используемый. Python — один из самых популярных языков программирования на данный момент, в рейтинге TIOBE он занимает 5 место. У него существуют свои фанаты это языка, но есть и те, кто данный язык не переносит, в любом случае равнодушным он никого не оставляет. На сегодняшний день углубление знаний и расширение навыков по разработке алгоритмов и их реализации на персональном компьютере дает большие возможности в мире программирования.

**Цель:** Реализация навыков по разработке алгоритмов и приложений на персональном компьютере на примере создание калькулятора на языке Python для выполнения простейших математических вычислений.

### **Задачи:**

1. Выбрать программы для создания калькулятора.
2. Создать внешний вид калькулятора.
3. Написать код для работы калькулятора.
4. Проверить исправность работы калькулятора.

## Глава 1. Предметная область

### п.1.1. Анализ предметной области

Python — это скриптовый язык программирования. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС.

#### Преимущества Python:

Это интерпретируемый язык — он не компилируется, то есть до запуска представляет из себя обычный текстовый файл. Программировать можно практически на всех платформах, язык хорошо спроектирован и логичен. Разработка идёт в разы быстрее, потому что кода здесь куда меньше, чем на других языках. И ещё Python отлично подходит новичкам.

#### Как используется Python:

Его можно встретить в вебе и на мобильных устройствах, в приложениях и решениях, связанных с машинным обучением (нейросети и искусственный интеллект), а также в качестве встроенной системы.

#### *Веб-разработка:*

Чаще всего Python используется в веб-разработке. Для работы с ним подключают фреймворки: Pyramid, Pylons, TurboGears, Flask, CherryPy и — самый популярный — Django.

Существуют и движки для создания сайтов на Python:

- Abilian SBE;
- Ella;
- Saleor;
- Wagtail;
- Django-CMS.

#### *Программы:*

Хоть язык не компилируется, с помощью него создают десктопные программы. Вот, к примеру, что было разработано на Python:

- GIMP — визуальный редактор на Linux;

- BitTorrent до 6 версии — менеджер торрент-закачек (позже программу переписали на C++, но сети peer-to-peer всё ещё работают на Python);

- Blender — программа для создания 3D-графики.

#### *Мобильные приложения:*

Мобильная разработка на Python менее популярна. Для Android чаще пишут на Java, C#, C++ или Kotlin, а для iOS — на Swift или Objective-C. На Python обычно программируют серверную часть приложения. Например, клиент Instagram для iOS написан на Objective-C, а сервер — на Python.

#### *Научные исследования:*

В Python есть несколько библиотек, которые пригодятся для проведения исследований и вычислений:

- SciPy — библиотека с научными инструментами;
- NumPy — расширение, которое добавляет поддержку матриц и многомерных массивов, а также математические функции для работы с ними;
- Matplotlib — библиотека для работы с 2D- и 3D-графикой.

Благодаря библиотекам и простоте освоения языка многие учёные выбирают Python — особенно он популярен у математиков и физиков.

#### *Встроенные системы:*

На Python разрабатывают встроенные системы для различных устройств. Например, язык прижился в Raspberry Pi (компьютер размером с карту памяти) и в «Сбербанке» для управления банкоматами.

Еще проекты со встроенной системой на Python:

- The Owl Embedded Python System;
- Python Embedded Tools;
- Embedded Python.

Язык применяется во встроенных системах станков с ЧПУ, средствах автоматического регулирования (температуры, расхода жидкостей, давления и так далее) и в телекоммуникационном оборудовании.

### *Системное администрирование:*

Системным администраторам Python нужен для автоматизации задач. Он простой, мощный и поддерживает специальные пакеты, которые повышают его эффективность. И, самое главное, он по умолчанию установлен на все серверы с ОС Linux. Благодаря лаконичности Python можно быстро прочитать код и найти слабые места. Форматирование в языке — часть синтаксиса.

### *Data Science:*

Python — один из самых используемых в Data Science языков. На нём пишут алгоритмы программ с машинным обучением и аналитические приложения. С помощью него обслуживают хранилища данных и облачные сервисы.

Также он помогает парсить данные из интернета. Например, в Google Python применяют для индексации сайтов.

### *Компании, использующие Python:*

В основном Python используется стартапами и компаниями, которые разрабатывают крупные проекты. Вот лишь часть огромного списка:

- Alphabet использует язык для скраппинга в поисковике Google и реализации сервиса YouTube;
- Pixar, Industrial Light & Magic — для создания анимационных фильмов;
- Intel, Cisco, HP, Seagate, Qualcomm и IBM — для тестирования;
- JPMorgan Chase, UBS, Getco и Citadel — для прогнозирования финансового рынка;
- NASA, Los Alamos, Fermilab, JPL — для научных вычислений;
- iRobot — для разработки коммерческих роботизированных устройств;
- IronPort — для реализации почтового сервера.

Кроме того, его используют в Instagram, Positive Technologies, Houdini, Facebook, Yahoo, Red Hat, Dropbox, Pinterest, Quora, Mail.ru и Яндексe.

## Недостатки языка Python:

*1. Программы на Python считаются одними из самых медленных.*

Приложения для iOS на Swift работают в 8,7 раз быстрее, чем на Python. Реализация PyPy по скорости близка к Java, но в ней есть не все возможности оригинального языка. Python не подходит для задач, требующих большого объёма памяти, — их лучше решать вставками на C или C++.

*2. Сильная зависимость языка от системных библиотек*

Из-за этого затрудняется перенос на другие системы. Для этих целей существует инструмент Virtualenv, но и он с недостатками: избыточность полных методов изоляции, костыли, дублирование системных библиотек.

*Что такое редактор кода?*

IDE (или интегрированная среда разработки) — это программа, предназначенная для разработки программного обеспечения. Как следует из названия, IDE объединяет несколько инструментов, специально предназначенных для разработки. Эти инструменты обычно включают редактор, предназначенный для работы с кодом (например, подсветка синтаксиса и автодополнение); инструменты сборки, выполнения и отладки; и определённую форму системы управления версиями.

Большинство IDE поддерживают множество языков программирования и имеют много функций, из-за чего могут быть большими, занимать много времени для загрузки и установки и требуют глубоких знаний для правильного использования.

## **PyCharm:**

Одной из лучших полнофункциональных IDE, предназначенных именно для Python, является PyCharm. Существует как бесплатный open-source (Community), так и платный (Professional) варианты IDE. PyCharm доступен на Windows, Mac OS X и Linux.

PyCharm «из коробки» поддерживает разработку на Python напрямую — откройте новый файл и начинайте писать код. Вы можете запускать и

отлаживать код прямо из PyCharm. Кроме того, в IDE есть поддержка проектов и системы управления версиями.

**Преимущества:** это среда разработки для Python с поддержкой всего и вся и хорошим комьюнити. В ней «из коробки» можно редактировать, запускать и отлаживать Python-код.

**Недостатки:** PyCharm может медленно загружаться, а настройки по умолчанию, возможно, придётся подкорректировать для существующих проектов. Лично я использую PyCharm, потому что для меня он наиболее удобен. Я выбрал именно его, потому что многие люди, занимающиеся программированием порекомендовали мне этот редактор.



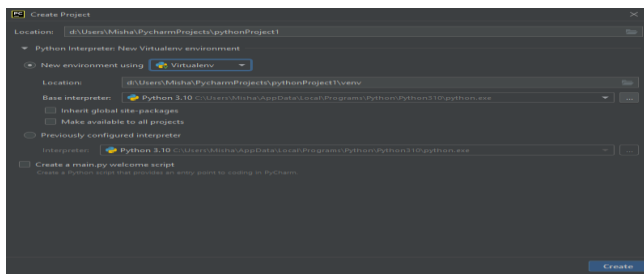
## Глава 2. Создание калькулятора

### п.2.1. Алгоритм написания калькулятора

Для написания кода на языке Python была выбрана программа **PyCharm**.

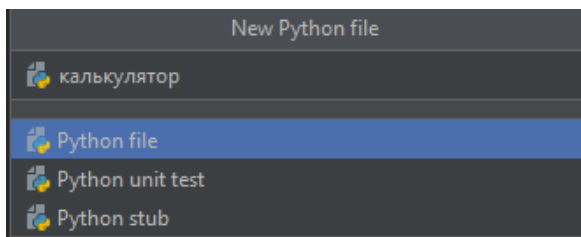
Одной из лучших полнофункциональных IDE, предназначенных именно для Python, является PyCharm. Существует как бесплатный open-source (Community), так и платный (Professional) варианты IDE. PyCharm доступен на Windows, Mac OS X и Linux.

1.Создаем проект, который будет являться основой для работы в PyCharm.



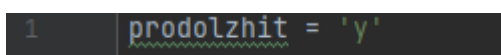
Изображение 1: Окно создания проекта в PyCharm.

2.После создания проекта в папке проекта необходимо создать файл с расширением .py в котором и будет записан весь код. В данном случае это будет файл калькулятор.py.



Изображение 2: Создание файла с расширением .py..

3.Начнём наш код с введения функции (это нужно,чтобы наш калькулятор начал свою работу) .



Изображение 3: Введение функции.

4.Далее вводится цикл while, чтобы делать бесконечно операций в калькуляторе .

```
1 prodolzhit = 'y'
2 while prodolzhit == 'y':
```

Изображение 4: Введение цикла while.

5.Следующим шагом нужно ввести переменные , чтобы дать информацию калькулятору для последующих действий .Дополнительно, я решил написать что делает каждая переменная.

```
1 prodolzhit = 'y'
2 while prodolzhit == 'y':
3     f_sum = float(input("Введите первое число"))
4     oper = input("Введите операцию")
5     s_num = float(input("Введите второе число"))
```

Изображение 5: Введение переменных.

6.Введение условий для калькулятора для того, чтобы калькулятор мог проводить операции .

```
1 prodolzhit = 'y'
2 while prodolzhit == 'y':
3     f_sum = float(input("Введите первое число"))
4     oper = input("Введите операцию")
5     s_num = float(input("Введите второе число"))
6     if oper == '+':
7         print(f_sum + s_num)
8     elif oper == '-':
9         print(f_sum - s_num)
10    elif oper == '*':
11        print(f_sum * s_num)
12    elif oper == '/':
13        print(f_sum / s_num)
14    else:
15        print("Error")
```

Изображение 6:Введение условий.

7.Введение функции prodolzhit , чтобы операции калькулятора имели начало и конец .

```
1 prodolzhit = 'y'
2 while prodolzhit == 'y':
3     f_sum = float(input("Введите первое число"))
4     oper = input("Введите операцию")
5     s_num = float(input("Введите второе число"))
6     if oper == '+':
7         print(f_sum + s_num)
8     elif oper == '-':
9         print(f_sum - s_num)
10    elif oper == '*':
11        print(f_sum * s_num)
12    elif oper == '/':
13        print(f_sum / s_num)
14    else:
15        print("Error")
16 prodolzhit=input("Введите 'y' чтобы продолжить, или любую клавишу, чтобы завершить")
```

Изображение 7:Введение функции prodolzhit.

```
Первое число 10
операция +
Второе число 10
100.0
Введите 'y', чтобы продолжить, или любую клавишу, чтобы завершить
```

Изображение 8:Умножение

```
Первое число 10.3
операция *
Второе число 0.19999999999999993
Введите 'y', чтобы продолжить, или любую клавишу, чтобы завершить
```

### Изображение 9:Выделение остатка

```
Первое число 14.0
операция /
Второе число
7.0
Введите 'y', чтобы продолжить, или любую клавишу, чтобы завершить
```

### Изображение 10:Деление без остатка

На основе написанного кода мне предстоит сделать приложение,внеся некоторые корректировки.Для начала нужно выбрать среду разработки.

## п.2.2. Создание приложения

Средой разработки для внешнего вида будет являться приложение **Qt Designer**.

Qt Designer - это инструмент Qt для проектирования и создания графических пользовательских интерфейсов (GUI) с виджетами Qt. Вы можете создавать и настраивать свои окна или диалоговые окна в стиле "что вы видите, то и получаете" и тестировать их, используя разные стили и разрешения.

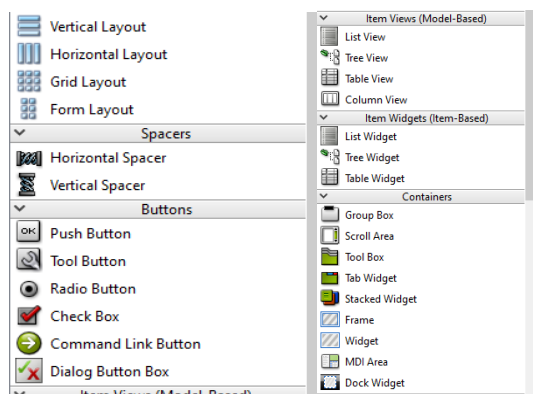
Qt Designer позволяет создавать графические интерфейсы пользователя при помощи ряда инструментов. Существует панель инструментов «Панель виджетов», в которой доступны для использования элементы интерфейса— виджеты, такие как, например, «выпадающий список» ComboBox, «поле ввода» LineEdit, «кнопка» PushButton и многие другие. Каждый виджет имеет свой набор свойств, определяемый соответствующим ему классом библиотеки Qt. Свойства виджета могут быть изменены при помощи «Редактора свойств». Для каждого класса свойств виджета существует свой специализированный редактор. Характерной особенностью Qt Designer является поддержка визуального редактирования сигналов и слотов. Так, например, можно связать сигнал, генерируемый по переключению состояния виджета CheckBox со слотом отвечающим за доступность другого виджета.

Виджеты и формы, созданные с помощью Qt Designer, легко интегрируются с запрограммированным кодом, используя механизм сигналов и слотов Qt, так что вы можете легко назначить поведение графическим элементам. Все свойства, установленные в Qt Designer, могут быть динамически изменены в коде. Кроме того, такие функции, как продвижение виджетов и настраиваемые плагины, позволяют вам использовать свои собственные компоненты с Qt Designer.

Примечание: У вас есть возможность использовать Qt Quick для разработки пользовательского интерфейса, а не для виджетов. Это гораздо более простой способ написания многих видов приложений. Это обеспечивает полностью настраиваемый внешний вид, элементы, реагирующие на прикосновение, и плавные анимированные переходы, подкрепленные мощью графического ускорения OpenGL.

1. Для начала нужно создать внешнюю оболочку приложения. В этом поможет, выше упомянутый QtDesigner.

В этих окнах расположены приспособления для внешнего вида калькулятора (например Push Button нужно для того, чтобы у калькулятора были кнопки, на которых будут цифры для вписания их в строку калькулятора)



Изображение 11;12:окна настроек в QtDesigner.

В этих окнах требуется обозначить кнопки калькулятора (придать им функции).

Object	Class
cPlus	QPushButton
clear	QPushButton
clear_19	QPushButton
div	QPushButton
equals	QPushButton
input	QLabel
min	QPushButton
mul	QPushButton

Изображение 13:обозначение кнопок.

Например, объект div выполняют функцию деления,сPlus-сложения, clear- очистка.

2.После создания внешнего вида нужно из файл.ui (файл QtDesigner) создать файл.py (файл Python).

```

Calculator.setObjectName("Calculator")
Calculator.resize(400, 400)
self.centralwidget = QtWidgets.QWidget(Calculator)
self.centralwidget.setObjectName("centralwidget")
self.num0 = QtWidgets.QPushButton(self.centralwidget)
self.num0.setGeometry(QtCore.QRect(330, 340, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num0.setFont(font)
self.num0.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num3 = QtWidgets.QPushButton(self.centralwidget)
self.num3.setGeometry(QtCore.QRect(330, 260, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num3.setFont(font)
self.num3.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num3.setObjectName("num3")
self.num0 = QtWidgets.QPushButton(self.centralwidget)
self.num0.setGeometry(QtCore.QRect(250, 500, 151, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num0.setFont(font)
self.num0.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.clear.setObjectName("clear")
self.clear.setGeometry(QtCore.QRect(10, 260, 71, 151))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.clear.setFont(font)

```

Изображение 14:конвертированный файл.

```

background-color: rgb(255, 0, 0);")
self.clear.setObjectName("clear")
self.num9 = QtWidgets.QPushButton(self.centralwidget)
self.num9.setGeometry(QtCore.QRect(330, 420, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num9.setFont(font)
self.num9.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num7 = QtWidgets.QPushButton(self.centralwidget)
self.num7.setGeometry(QtCore.QRect(170, 420, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num7.setFont(font)
self.num7.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num8 = QtWidgets.QPushButton(self.centralwidget)
self.num8.setGeometry(QtCore.QRect(250, 420, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num8.setFont(font)
self.num8.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num5 = QtWidgets.QPushButton(self.centralwidget)
self.num5.setGeometry(QtCore.QRect(250, 340, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num5.setFont(font)
self.num5.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(0, 255, 255);")
self.num5.setObjectName("num5")

```

Изображение 15:конвертированный файл.

```

self.num5.setObjectName("num5")
self.num1 = QtWidgets.QPushButton(self.centralwidget)
self.num1.setGeometry(QtCore.QRect(170, 260, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num1.setFont(font)
self.num1.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(189, 169, 255);\n"
";")
self.num1.setObjectName("num1")
self.num2 = QtWidgets.QPushButton(self.centralwidget)
self.num2.setGeometry(QtCore.QRect(250, 260, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num2.setFont(font)
self.num2.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(189, 169, 255);\n"
";")
self.num4 = QtWidgets.QPushButton(self.centralwidget)
self.num4.setGeometry(QtCore.QRect(170, 340, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.num4.setFont(font)
self.num4.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(189, 169, 255);\n"
";")
self.div = QtWidgets.QPushButton(self.centralwidget)
self.div.setGeometry(QtCore.QRect(190, 420, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.div.setFont(font)
self.div.setStyleSheet("border-radius:10px;\n"

```

Изображение 16:конвертированный файл.

```

self.P.setObjectName("P")
self.input = QtWidgets.QLabel(self.centralwidget)
self.input.setGeometry(QtCore.QRect(170, 180, 311, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.input.setFont(font)
self.output = QtWidgets.QLabel(self.centralwidget)
self.output.setGeometry(QtCore.QRect(170, 110, 311, 61))
font = QtGui.QFont()
font.setFamily("Viner Hand ITC")
font.setPointSize(28)
self.output.setFont(font)
self.output.setStyleSheet("color: rgb(195, 195, 195);")
self.output.setObjectName("output")
self.equals = QtWidgets.QPushButton(self.centralwidget)
self.equals.setGeometry(QtCore.QRect(10, 500, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.equals.setFont(font)
self.equals.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(189, 169, 255);")
self.equals.setObjectName("equals")
self.FAC = QtWidgets.QPushButton(self.centralwidget)
self.FAC.setGeometry(QtCore.QRect(170, 500, 71, 71))
font = QtGui.QFont()
font.setFamily("Tempus Sans ITC")
font.setPointSize(28)
self.FAC.setFont(font)
self.FAC.setStyleSheet("border-radius:10px;\n"
"background-color: rgb(189, 169, 255);")
self.FAC.setObjectName("FAC")
Calculator.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(Calculator)

```

Изображение 17:конвертированный файл.

3.Перед тем,как писать код нужно: во-первых нужно ввести команду «import sys»,чтобы работала программа.Во-вторых использовать библиотеку PyQt5 (данная библиотека даст возможность работать с файлом Qtdesigner).В-третьих импортируем библиотеку math as mt для математических вычислений.

```

import sys
from PyQt5 import QtCore, QtGui, QtWidgets
import math as mt;

```

Изображение 18:начало кода и введение библиотеки math as mt.

4. Нужно подключить клавиши,чтобы при нажатии их на внешней оболочке они работали.

```
class Window(QMainWindow):
    def __init__(self):
        super(Window, self).__init__()
        self.ui = Ui_Calculator()
        self.ui.setupUi(self)
        self.ui.num1.clicked.connect(lambda: self.NumBut(1))
        self.ui.num2.clicked.connect(lambda: self.NumBut(2))
        self.ui.num3.clicked.connect(lambda: self.NumBut(3))
        self.ui.num4.clicked.connect(lambda: self.NumBut(4))
        self.ui.num5.clicked.connect(lambda: self.NumBut(5))
        self.ui.num6.clicked.connect(lambda: self.NumBut(6))
        self.ui.num7.clicked.connect(lambda: self.NumBut(7))
        self.ui.num8.clicked.connect(lambda: self.NumBut(8))
        self.ui.num9.clicked.connect(lambda: self.NumBut(9))
        self.ui.num0.clicked.connect(lambda: self.NumBut(0))
        self.ui.P.clicked.connect(lambda: self.NumBut(3.14))
        self.ui.min.clicked.connect(lambda: self.NumBut("-"))
        self.ui.cPlus.clicked.connect(lambda: self.NumBut("+"))
        self.ui.div.clicked.connect(lambda: self.NumBut("/"))
        self.ui.mul.clicked.connect(lambda: self.NumBut("*"))
        self.ui.FAC.clicked.connect(self.Factor)
        self.ui.clear.clicked.connect(self.Clear)
        self.ui.equals.clicked.connect(self.Equals)
```

Изображение 19:подключение клавиш.

5.Следующий шаг добавить функции (сложение, вычитание, деление, умножение).

```
def Equals(self):
    mt =
    cPlus=""
    div=""
    mul=""
    Compare = self.ui.input.text()
    if Compare:
        print("now")
        s = Compare.split("/")
        n = float(s[0])
        ni = float(s[1])
        f = float(n - ni)
        self.ui.input.setText(str(mt.floor(f)))
    elif cPlus in Compare:
        print("now")
        s = Compare.split("+")
        n = float(s[0])
        ni = float(s[1])
        f = float(n + ni)
        self.ui.input.setText(str(mt.floor(f)))
    elif div in Compare:
        print("now")
        s = Compare.split("/")
        n = float(s[0])
        ni = float(s[1])
        f = float(n / ni)
        self.ui.input.setText(str(mt.floor(f)))
    elif mul in Compare:
        print("now")
        s = Compare.split("*")
        n = float(s[0])
        ni = float(s[1])
        f = float(n * ni)
        self.ui.input.setText(str(mt.floor(f)))
    self.ui.input.setText(str(mt.floor(f)))
    self.ui.input.clear()
```

Изображение 20:добавление функций.

6.Также функции очистки строки ввода,а также я решил добавить функцию вычисления факториала.

```
def Clear(self):
    self.ui.input.clear()

def Factor(self):
    N = mt.factorial(int(self.ui.input.text()))
    print(N)
    self.ui.input.setText(str(N))
```

Изображение 21:добавление функций.

7. Эти команды нужны для того, чтобы окно(внешняя оболочка) приложения могла открываться и закрываться.

```
app = QApplication([])
applicatin = Window()
applicatin.show()
sys.exit(app.exec_())
```

Изображение 22:команды для закрытия и открытия приложения.

## Заключение

Прикладные программы являются неотъемлемой частью жизни каждого пользователя персонального компьютера, такие программы отвечают за выполнения множества задач, таких как написание текстовых документов, обработка аудио или видео файлов, а также выполнение математических расчетов. Подобные приложения позволяет решать огромное количество задач быстрее и точнее, чем это смог бы обычный человек и является существенным фактором в развитие различных сфер деятельности человека, начиная с простого ведения учета расходов за месяц и заканчивая запуском ракет в космос. Анализируя «рынок программистов», увидел, что язык Python востребованный и часто используемый. Python — один из самых популярных языков программирования на данный момент.

Python — это скриптовый язык программирования. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС.

Изучив всю теоретическую информацию, я написал код для возможного вычисления простых операций, а также создал саму оболочку калькулятора. Цели и задачи, которые я ставил, считаю, выполнил в своей работе.



## Список литературы:

1. <https://ru.wikipedia.org/wiki/Python>
2. <https://timeweb.cloud/tutorials/python/prostoj-kalkulyator-na-python>
3. <https://it-start.online/articles/gui-kalkuljator-na-python-tkinter>
4. <https://www.8host.com/blog/napisanie-prostejshego-kalkulyatora-v-python-3/>
5. <https://www.geeksforgeeks.org/scientific-gui-calculator-using-tkinter-in-python/>