МИНОБРНАУКИ РОССИИ

ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ
(ФИЛИАЛ ОГУ))

Механико-технологический факультет

Кафедра программного обеспечения

ОТЧЕТ

по производственной практике

*(*технологическая*)*

Орский гуманитарно-технологический институт (филиал) ОГУ

09.03.01. 7021. 047 ПЗ

Руководитель от кафедры
ст. преподаватель _____ В. С. Богданова
подпись, дата

Руководитель от предприятия
Начальник ИКЦ _____ М. В. Сапрыкин
подпись, дата

Исполнитель
студент группы
18ИВТ(ба)ПОВТ _____ Д.В. Брагин
подпись, дата

Орск 2021

Аннотация

Отчет по производственной практике (технологической) содержит 19 страниц, в том числе 14 рисунков, 5 источников, 1 приложение.

В первой части отчета по производственной практике приводится структурная схема предприятия и описание отдела прохождения практики. Также прописаны основные задачи сотрудников отдела.

Во второй части приводится программная реализация мобильного приложения. Также описывается среда разработки приложения, инструмент для создания интерфейса приложения и язык программирования, выбранный для создания приложения.

Содержание

Введение

Производственная практика (технологическая) после шестого семестра для студентов направления академического бакалавриата «Информатика и вычислительная техника» проходила в Орском гуманитарно-технологическом институте, располагающегося по адресу : пр. Мира, 15А, Орск, Оренбургская обл., 462419. Практика проходила с 28.06.2021 г. по 10.07.2021 г.

1 Теоретическая часть

1.1 Структура предприятия, основные виды деятельности

Организатором производственной практики является Орский гуманитарно-технологический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет». Данное предприятие имеет сложную структуру, которая предполагает деление на факультеты. Факультет – это объединение нескольких кафедр, каждая из которых отвечает за подготовку студентов различных направлений в одной сфере деятельности. Помимо факультетов предприятие имеет несколько подразделений, обеспечивающих полноценное функционирование. Одним из таких является Информационно-коммуникационный центр, где и проходила производственная практика.

Основным видом деятельности предприятия является подготовка специалистов к профессиональной деятельности. Такая подготовка ведётся по многим направлениям гуманитарного и технического характера. Присутствуют различные формы обучения, среди которых есть и основные бакалавриат и магистратура.

1.2 Структура отдела

Информационно-коммуникационный центр (ИКЦ) входит в состав Орского гуманитарно-технологического института (филиала) федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» в качестве структурного подразделения.

Структура и численность определяется штатным расписанием института.

Организационная диаграмма представлена на рисунке 1:

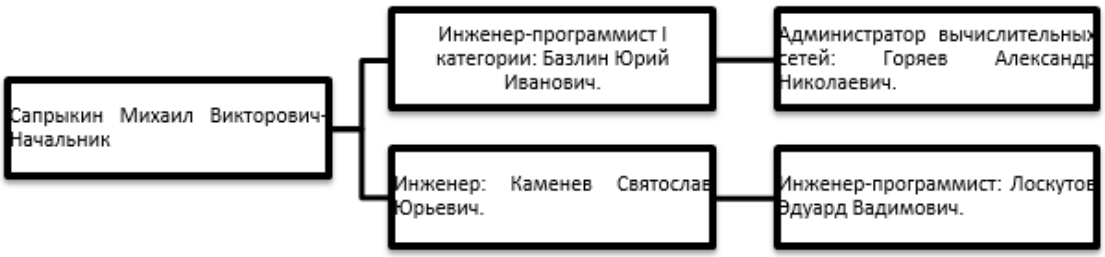

Рисунок 1 – Организационная диаграмма ИКЦ

На данный момент в состав подразделения входят: начальник, инженеры, программисты, инженеры-программисты и администратор вычислительных сетей.

Его деятельность направлена на осуществление комплекса работ по обеспечению функционирования и развитию информационно-технического инфраструктуры института. Целями ИКЦ является повышение эффективности деятельности института посредством использования современных информационных систем и оптимизация доступа к сетевым информационным ресурсам.

1.3 Обязанности программиста

Обязанности программиста разнятся в зависимости от специализации и решаемых задач. Основными же являются:

Проектирование и разработка ПО.

Внедрение ПО и организация корректного взаимодействия с другими программами.

Поддержка ПО и помощь специалистам техподдержки в устранении ошибок.

Техник-программист исполняет следующие должностные обязанности: методы проектирования механизированной и автоматизированной обработки информации; средства вычислительной техники, сбора, передачи и обработки информации и правила их эксплуатации; технологию механизированной и автоматизированной обработки информации; рабочие программы, инструкции, макеты и другие руководящие материалы, определяющие последовательность и технику выполнения расчетных операций; виды технических носителей информации, правила их хранения и эксплуатации; действующие системы счислений, шифров и кодов; основные формализованные языки программирования; основы программирования; методы проведения расчетов и вычислительных работ; методы расчета выполненных работ; основы экономики, организации труда и производства; правила и нормы охраны труда.

Инженер-программист выполняет следующие должностные обязанности: специалист должен разработать и внедрить программы, которые обеспечат выполнение алгоритма поставленных целей с помощью вычислительной техники, протестировать и наладить бесперебойную работу, на любом этапе подборки и обработки информации должен суметь наладить процесс и разработать технологию процесса, подбирает оптимальный язык программирования для поставленных целей, проводит отладку в случае сбоев в программах, запускает программы, вводит необходимую исходную информацию для правильного функционирования, в случае некорректного результата работы проводит корректировку программы, разрабатывает инструкции по работе с программами, обучает в случае необходимости коллег и других заинтересованных пользователей, контролирует внедрение программ на всех

этапах работы, доводит систему до автоматического функционирования, оформляет техническую документацию, подбирает информацию, которую нужно обработать с помощью ЭВМ, определяет объемы и ее структуру, вводит дополнительную защиту данных и действий, отчитывается перед руководством о выполнении поставленных задач, своевременно изучает необходимый материал для улучшения рабочего процесса.

1.4 План мероприятий по разработке приложения

Разработка приложения разделена на несколько этапов, каждый из которых важен и незаменим.

Первым и самым важным этапом является анализ требований. Он включает в себя постановку задачи и выбор инструментальных средства разработки и реализации ПО.

Следующим этапом является проектирование программного обеспечения.

После чего наступают следующие этап: программирование, компиляция и отладка программы.

1.5 Техническое задание на разработку приложения

Необходимо разработать мобильное приложение на основе PyCharm.

1.6 Технорабочий проект

Приложение написано на языке программирования Python. Разработка велась в среде разработки PyCharm, который располагает большим количеством встроенных инструментов для различных задач.

2 Практическая часть

2.1 Постановка задачи

Необходимо разработать мобильное приложение на языке Pythonв среде разработки PyCharm

2.2 Инструментальные средства разработки и реализации ПО

Для разработки приложений используется интегрированная среда разработки – PyCharm. Язык на котором будет написано приложение – Python.

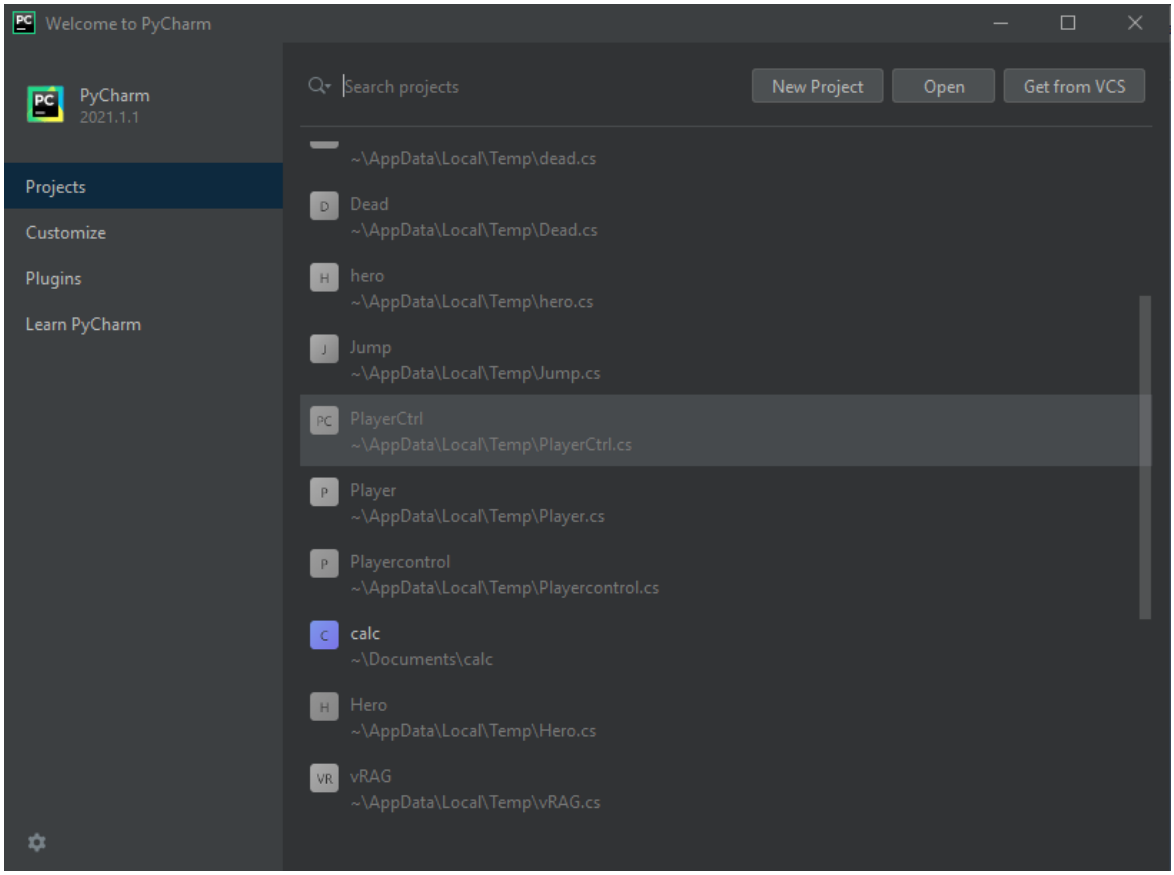

Рисунок 1 – Начальный экран PyCharm

Данная среда разработки универсальна, так как позволяет оптимизировать работу будущих приложения для работы не только на смартфонах, но и для планшета, портативных ПК, которые работают на основе рассматриваемой операционной системы.

2.3 Проектирование мобильного приложения

Создание программного кода для работаспобности калькулятора

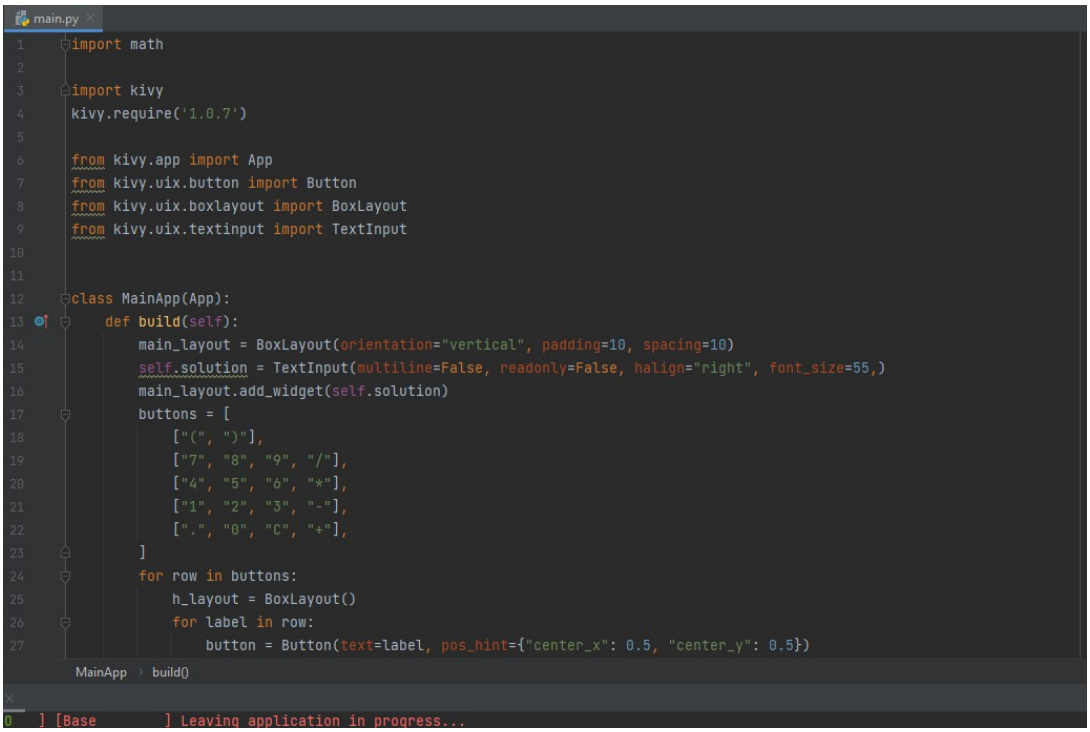

Рисунок 2 – Реализация программного кода

2.4 Основные функции кулятора

Сложение – одна из главных функций калькулятора, которые он может выполнять

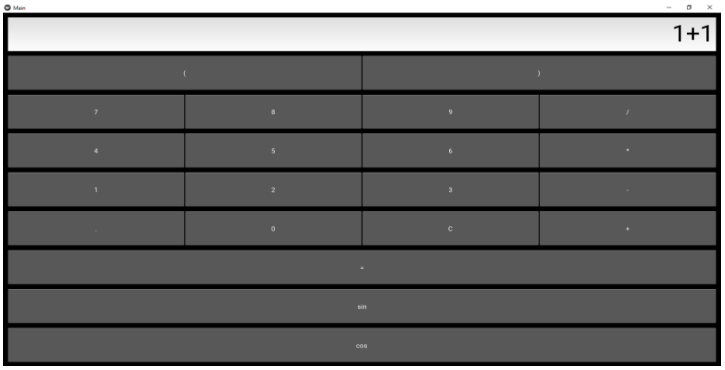

Рисунок 3 – Вписание функции сложения

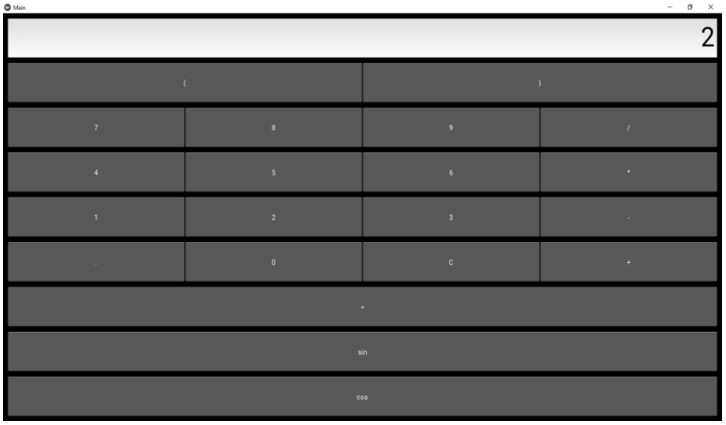

Рисунок 4 – Результат сложения

Вычитание – вторая функция без которой ни один не кулькудятор не может обойтись

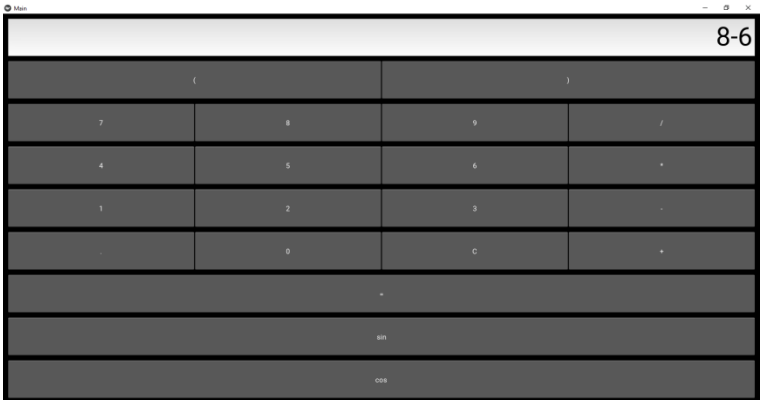

Рисунок 5 – Вписание функции вычитания

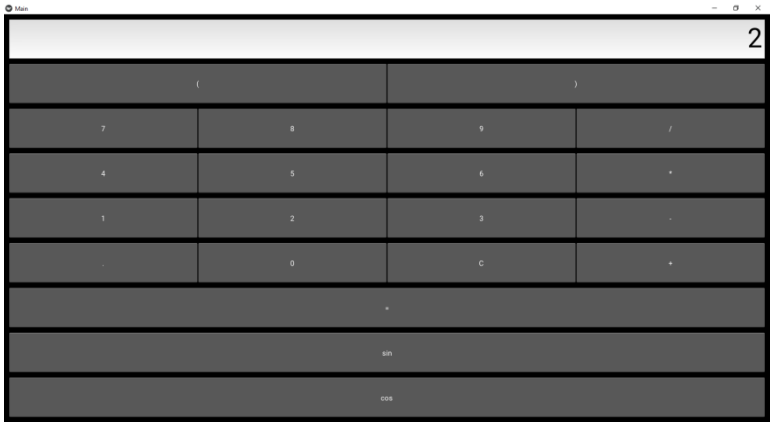

Рисунок 6 – Результат вычитания

Умножение – 3ья функция без которой нельзя обойтись

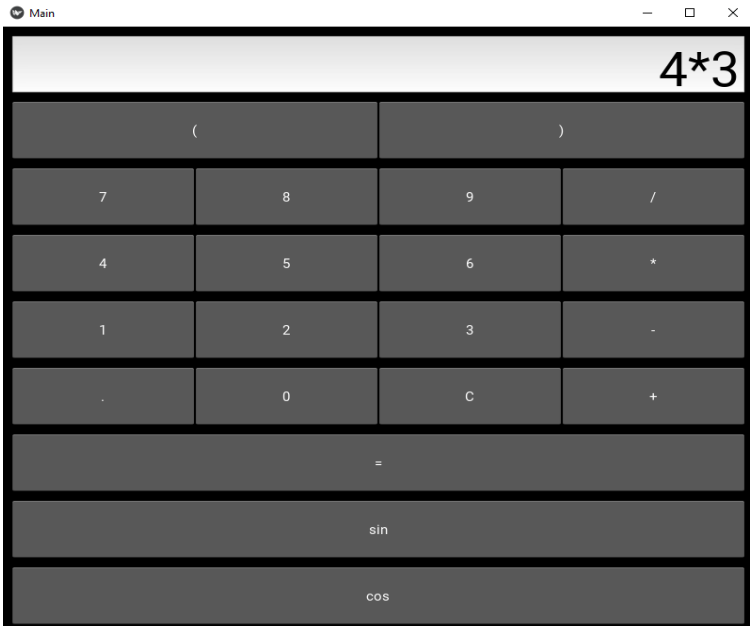

Рисунок 7 – Вписание функции умножения

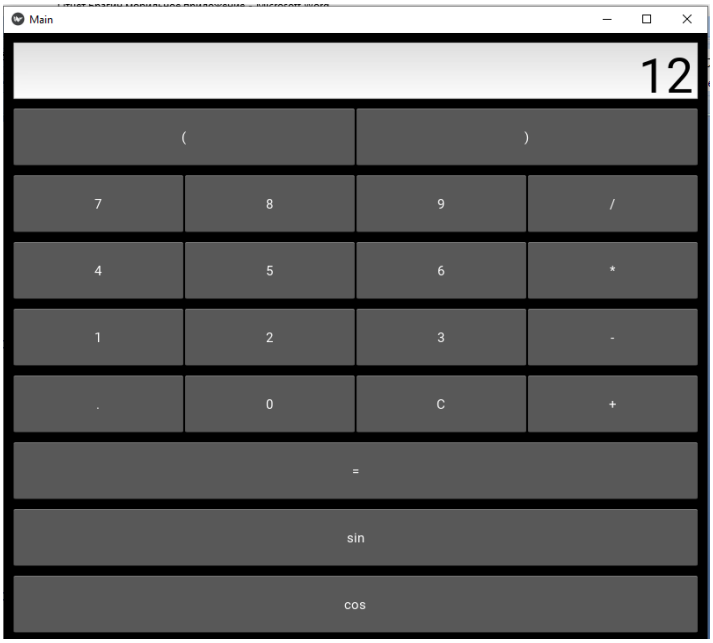

Рисунок 8 – Результат умножения

Деление – Последняя функция без который не может сущетсовать ни один калькулятор

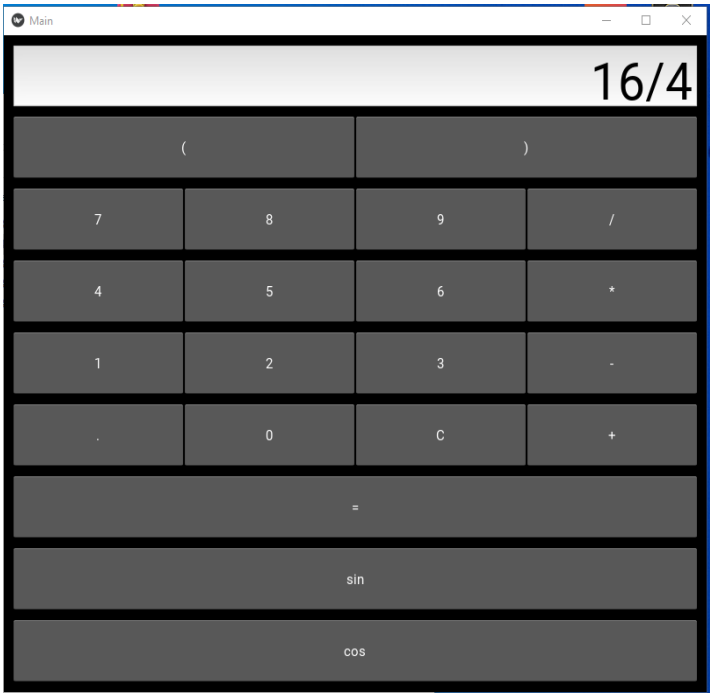

Рисунок 9 – Вписание функции деления

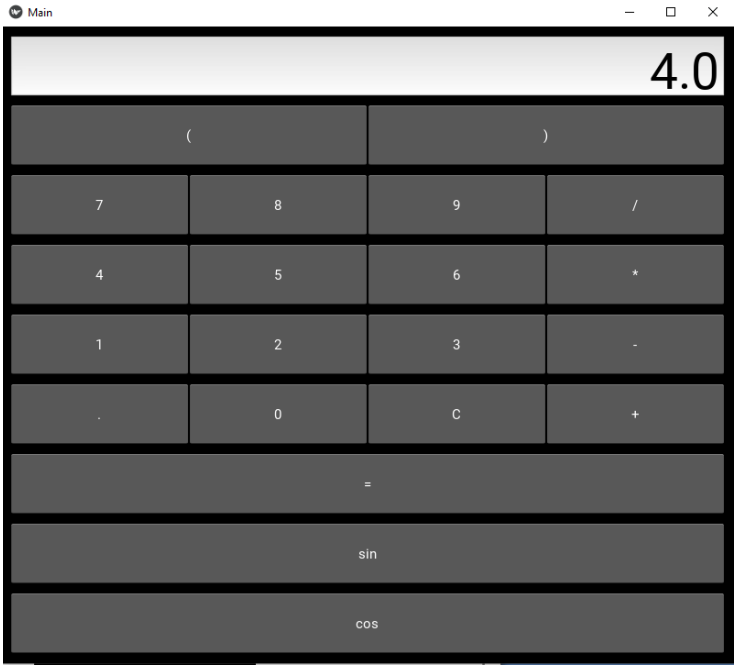

Рисунок 10 – Результат деления

Нахождение значения sin от числа

Для того, чтобы воспользоваться функцией для нахождение sin от числа. Необходимо сначала написать число в полее ввода текста, а потом нажать на клавишу калькудятора “sin”

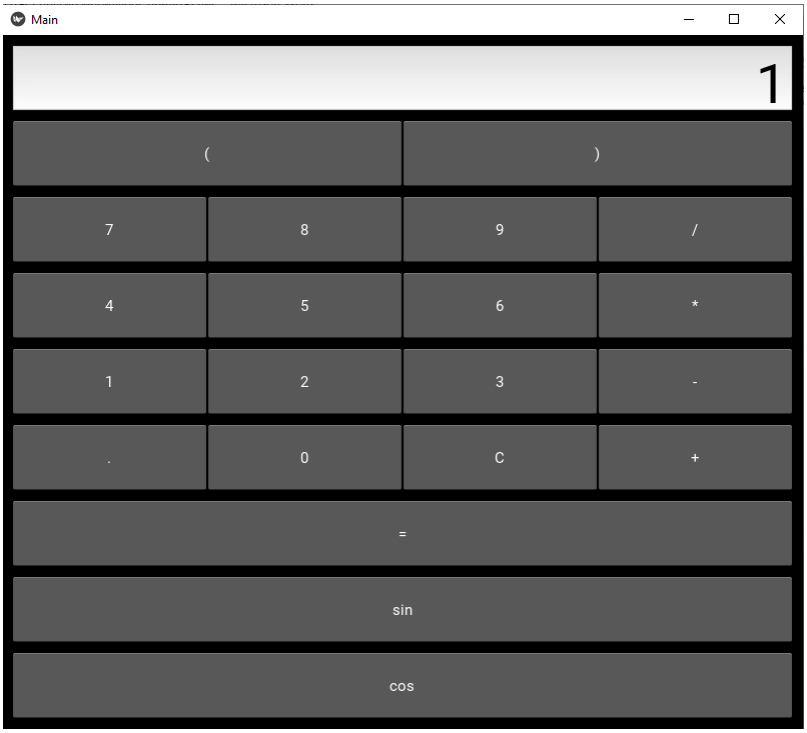

Рисунок 11 – Вписание числа для нахождения sin от числа

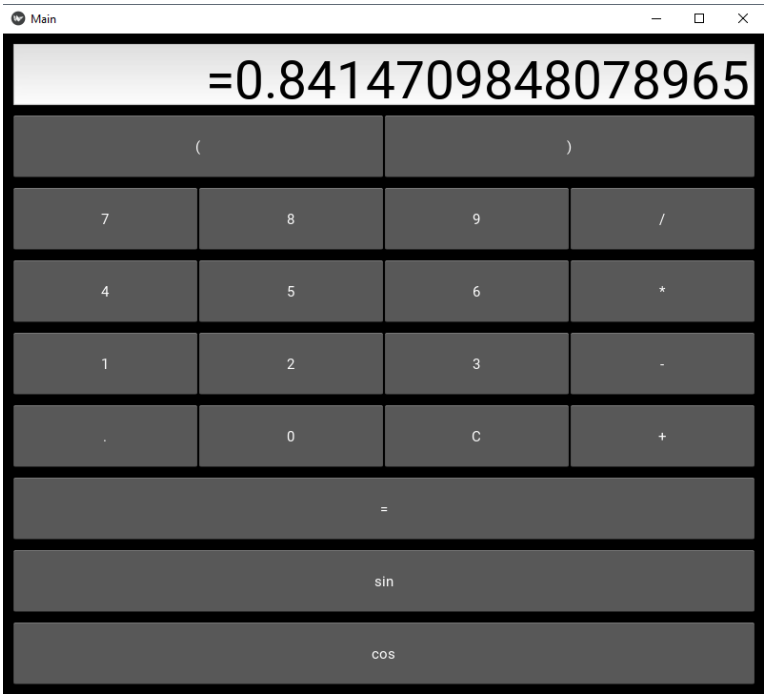

Рисунок 12 – Результат нахолждения sin от числа

Нахождение значения cos от числа

Для того, чтобы воспользоваться этой функцией моего кальлятора, необходило повторить ту же последовательность дейсвий, что и при нахожения значение sin от числа, только вместо клавиши “sin”, необходимо воспользоваться клавишей “cos”

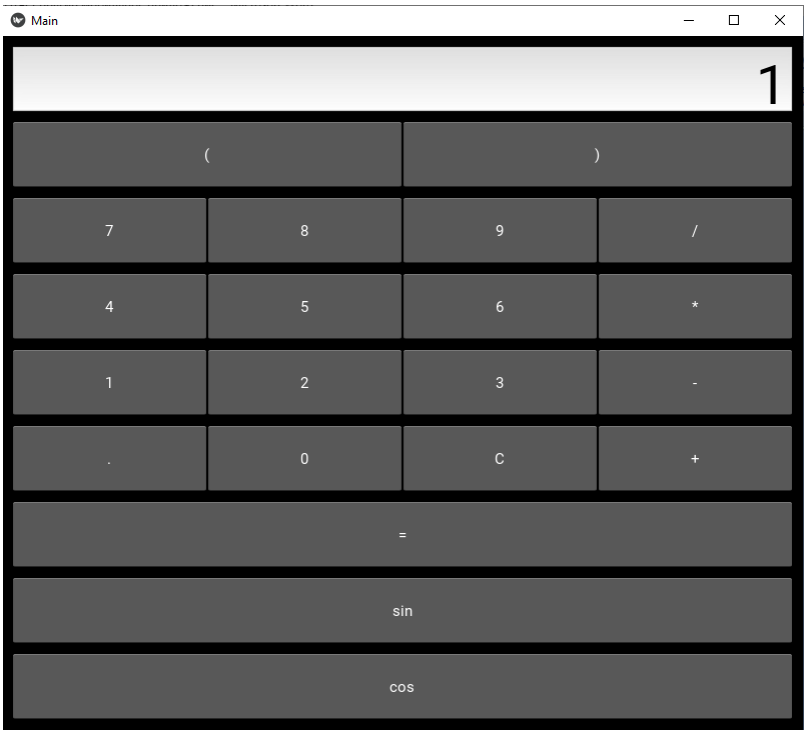

Рисунок 13 –Вписание числа для нахождения cos от числа

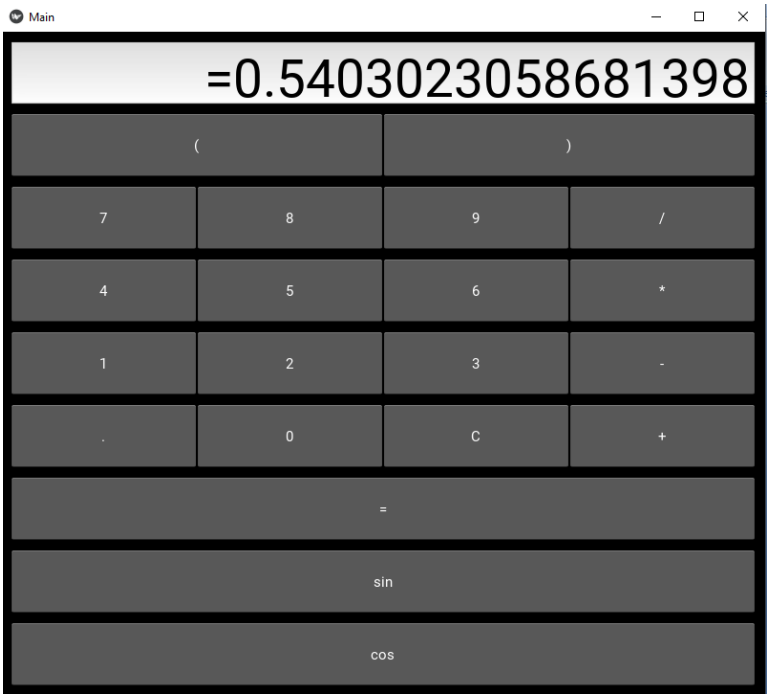

Рисунок 14 – Результат нахождения cos от числа

Вот таким образом работают все функции моего калькулятора

\

Заключение

В результате выполнения технологической практики была разработана мобильное приложение для мобильных телефонов

Приложение написано на языке программирования Python. Python — статически типизированный язык программирования.

PyCharm – интегрированная среда разработки.

Данная среда разработки универсальна, так как позволяет оптимизировать работу будущих приложения для работы не только на смартфонах, но и для планшета, портативных ПК, которые работают на основе рассматриваемой операционной системы.

Список использованных источников

1) Документация по Python. URL: https://www.python.org/doc/ (Дата обращения: 18.06.2021)
2) Как создать мобильное приложение в Python URL: https://proglib.io/p/mobile-python (Дата обращения: 19.06.2021)
3) Документация по установки Kivy. URL: https://kivy.org/doc/stable/gettingstarted/installation.html#install-pip (Дата обращения: 15.06.2021)
4) Галерея примеров Kivy. URL: https://kivy.org/doc/stable/examples/gallery.html#. (Дата обращения: 17.06.2021)
5) Pyton User Manual. URL: http://pylint.pycqa.org/en/latest/ (Дата обращения: 20.06.2021)

Приложение А
(обязательное)

Листинг программы

```
import math

import kivy
kivy.require('1.0.7')

from kivy.app import App
from kivy.uix.button import Button
from kivy.uix.boxlayout import
BoxLayout
from kivy.uix.textinput import
TextInput


class MainApp(App):
    def build(self):
        main_layout =
BoxLayout(orientation="vertical",
padding=10, spacing=10)
        self.solution =
TextInput(multiline=False,
readonly=False, halign="right",
font_size=55,)

main_layout.add_widget(self.solution)
        buttons = [
            ["(", ")"],
            ["7", "8", "9", "/"],
            ["4", "5", "6", "*"],
            ["1", "2", "3", "-"],
            [".", "0", "C", "+"],
        ]
        for row in buttons:
            h_layout = BoxLayout()
            for label in row:
                button =
Button(text=label,
pos_hint={"center_x": 0.5,
"center_y": 0.5})

button.bind(on_press=self.on_button_p
ress)

h_layout.add_widget(button)

main_layout.add_widget(h_layout)



        equals_button =
Button(text="=",
pos_hint={"center_x": 0.5,
"center_y": 0.5})

equals_button.bind(on_press=self.on_s
olution)

main_layout.add_widget(equals_button)

        equals_button =
Button(text="sin",
pos_hint={"center_x": 0.5,
"center_y": 0.5})

equals_button.bind(on_press=self.on_s
in)

main_layout.add_widget(equals_button)

        equals_button =
Button(text="cos",
pos_hint={"center_x": 0.5,
"center_y": 0.5})

equals_button.bind(on_press=self.on_c
os)

main_layout.add_widget(equals_button)


        # return a Button() as a root
widget
        return main_layout


    def on_button_press(self,
instance):
        if instance.text == "C":
            self.solution.text = ""
        else:
            self.solution.text +=
instance.text


    def on_solution(self, instance):
        if self.solution.text:
            try:
                self.solution.text =
str(eval(self.solution.text))
            except:
                self.solution.text =
"error"

    def on_sin(self, instance):
        if instance.text == "sin":
            self.solution.text = ("="
```

```
+
str(math.sin(int(self.solution.text))
))

    def on_cos(self, instance):
        if instance.text == "cos":
            self.solution.text = ("="
+
str(math.cos(int(self.solution.text))
))

if __name__ == '__main__':
    MainApp().run()
```