МИНОБРНАУКИ РОССИИ

## Федеральное государственное автономное образовательное

## учреждение высшего образования

## «ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики, механики и компьютерных наук им. И.И. Воровича

01.03.02 — Прикладная математика и информатика

**Проект 2-го курса**

**ОТЧЕТ**

студентов 2 курса группы №6

*Токарь Анна Александровна ­*

*Беккер Дмитрий Игоревич­ ­*

*Харченко Вячеслав Сергеевич­ ­*

**Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***/к.ф.-м.н.,**доцент А.П. Мелехов* /

Ростов-на-Дону

2022 г.

Отчёт по заданию 1. Создание переносимых графических интерфейсов пользователя (Graphical User Interface, GUI). Tkinter. Отчёт на примере нескольких заданий.

В ходе выполнения заданий Tkinter данные брались из редакторов Entry. Результаты работы выводились на метки Label. Для каждого задания создавалось новое приложение.

Задание 2.

Задача: добавить операции «%» и «//» в Radiobutton и в обработчик.

Ход работы: Подключаем tkinter – from tkinter import \*, создаём главное окно - root = Tk( ), создаем список с именами будущих кнопок калькулятора и сами кнопки, затем проходим этот список (для создания кнопки используется конструктор Button(), а так же Grid – это менеджер геометрии, который размещает виджеты в двухмерной сетке – столбцы и строки). После моделируем логику калькулятора – исключение написания слов, вывод ошибки, например, при делении на 0). В конце очищаем поля вывода для кнопок “C”, “Exit”, “(“, ”)”.

Изображение выглядит как текст, белый, снимок экрана

Автоматически созданное описание Изображение выглядит как стол

Автоматически созданное описание

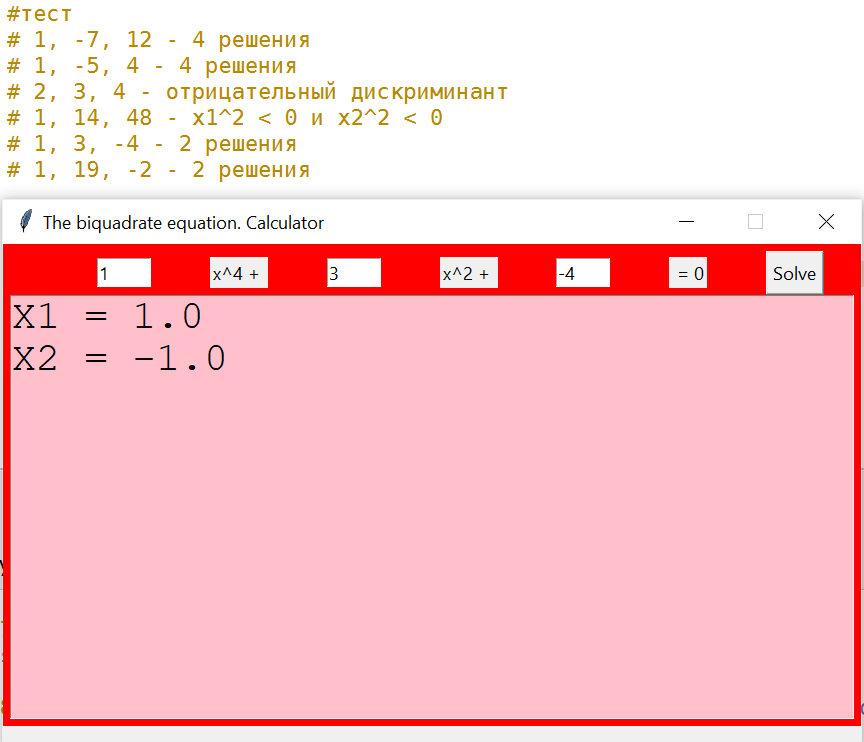
Изображение выглядит как стол

Автоматически созданное описание

Задание 11.

Задача: решить биквадратное уравнение. Пользователь в редакторах задает коэффициенты биквадратного уравнения a, b, c. По щелчку на кнопке 66 считываются эти значения (в защищенных блоках), и программа выдает количество вещественных корней и их значения.

Ход работы: кроме tkinter подключаем from math import sqrt, далее создаём функцию-решение – def decisive\_device(a,b,c). В этой функции рассматриваем все случаи дискриминанта и расписываем вычисления, затем добавляем в программу функцию def handler(), в которой описываем параметры a\_val = float(a.get()), b\_val = float(b.get()), c\_val = float(c.get()). Создаём в ней вставку через функцию insertion решения через функцию decisive\_device. В основной програмее – root=Tk() устанавливаем окно и его размеры, а так же цвет при помощи title, minsize и Frame. Создаём поля для аргументов с помощью Label. В конце добавляем кнопку для ршения – Buttom, место для вывода решения – Text, grid и запускаем главное окно – root.mainloop(). Ниже приведён пример работы программы и тесты для работы с ней.



Отчёт по заданию 2. Файлы. Работа с файловой системой. Отчёт на примере работы заданий.

Задание 18.

Задача: напишите программы, которые выполняют следующие действия: удалите существующий файл. Проверьте существование файла.

Реализация:

import os # подключить модуль os

os.remove('18 задание\_1.txt') ; # удаляем заготовленный файл

# проверяем наличие этого же файла

if os.path.exists('18 задание\_1.txt'):

print("Файл существует")

else:

print("Файл не существует")

# проверяем наличие другого заготовленного файла

if os.path.exists('18 задание\_2.txt'):

print("Файл существует")

else:

print("Файл не существует")

Результат работы:

Файл не существует

Файл существует

Задание 32.

Задача: сохранение произвольный список(list) в файл с помощью модуля Pickle. Прочитайте обратно его из этого файла и выведите на экран.

Реализация:

import pickle

n = int(input("Введите длину будущего массива: "))

arr = [int(input(f"Введите {i} элемент массива: ")) for i in range(n)]

with open("./new\_file.txt", "wb") as f:

for num in arr:

pickle.dump(num, f)

new\_arr = []

with open("./new\_file.txt", "rb") as f:

for i in range(n):

new\_arr.append(pickle.load(f))

print(new\_arr)

Результат работы:

Введите длину будущего массива: 4

Введите 0 элемент массива: 1

Введите 1 элемент массива: 2

Введите 2 элемент массива: 3

Введите 3 элемент массива: 4

[1, 2, 3, 4]

Отчёт по заданию 3. Пакет numpy. Отчёт на примере работы нескольких заданий.

Задание 19.

Задача: найдите собственные значения и собственные вектора квадратной матрицы.

Реализация:

import numpy as np

from numpy import linalg as LA

#импортируем всё, что необходимо из библиотек

# создаем квадратную матрицу размером matrsize

# с числами [-10, 10]

matrsize = int(input('Введите размерность матрицы: '))

print('\nМатрица (%dx%d):'%(matrsize, matrsize))

matrix = np.random.randint(-10, 10, size = (matrsize, matrsize))

print(matrix)

# находим собственные значения и вектора матрицы

vals, vecs = LA.eig(matrix)

print('Собственные значения матрицы:\n', vals, end='\n')

print('Собственные вектора матрицы:\n', vecs)

Задание 23.

Задание 22.

Задача: Агрегирование. Определите произвольный массив. Найдите для него следующие величины (можно по одной из осей или для всего массива):

Сумму, среднее и минимальное значения.

Реализация:

import numpy as np

#импортируем всё, что необходимо из библиотеки

#формируем массив

n = int(input('Введите количество строк: '))

m = int(input('Введите количество столбцов: '))

a = np.random.randint(10, size = (n, m))

print(a)

s = np.sum(a) #находим сумму всех элементов

print("Сумма всех элементов матрицы = ", s)

sr = np.mean(a) #находим среднее значение всех элементов

print("Среднее значение всех элементов = ", sr)

m = np.min(a) #находим минимальное значение в матрице

print("Минимальное значение в матрице = ", m)

Отчёт по заданию 4. Графика Matplotlib.

Задача: Прочитайте текст из User’s Guide “Pyplot tutorial” <https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py> и выполните команды из него в Jupiter. Выберите пункт из User’s Guide <https://matplotlib.org/stable/users/index.html>. Изучите его и выполните команды из него.

Ход работы:

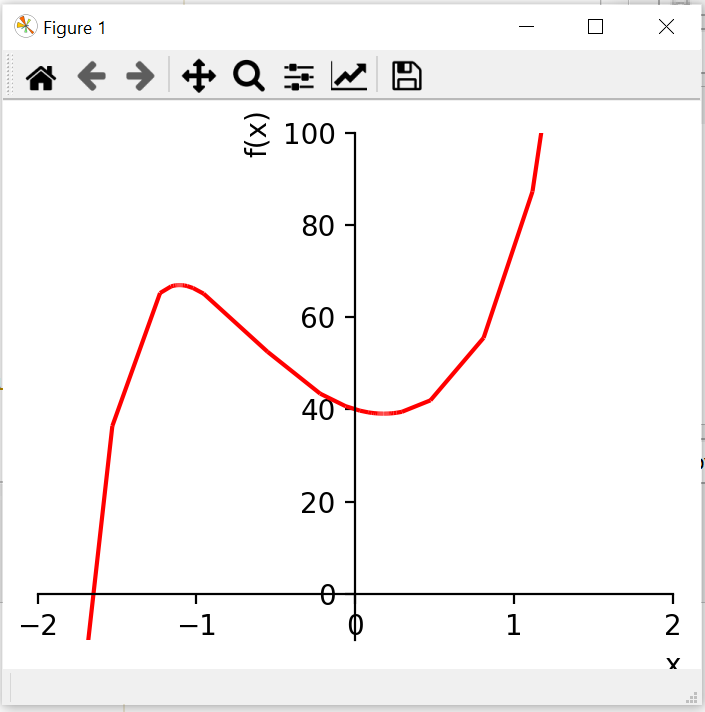
Были просмотрены и опробованы примеры команд представленные в “Pyplot tutorial”.

Были рассмотрены некоторые команды модуля “Pyplot” необходимые для работы с интерактивными рисунками (модуль “Pyplot” предоставляет функции для явного создания рисунков, которые включают интерактивные инструменты, панель инструментов, подсказку и привязки клавиш).

Отчёт по заданию 5. Пакет Sympy: символьные вычисления в Python.

Задача: выполнить 6 заданий с помощью пакета Sympy.

Ход работы: подключаем import sympy и from sympy.plotting import plot. Для вычисления предела используем метод limit, для нахождения бесконечной суммы берём метод summation. Далее для того, чтобы найти производную функции запишем x = Symbol('x'), а затем инициализируем функцию f = x\*\*x. Саму производную найдём через метод diff. Для вычисления интеграла используем integrate. Для того, чтобы найти все вещественные и комплексные корни полиномов и вывести их подключаем rom sympy.simplify.simplify import nthroot и используем метод nroots. Ниже представлена реализация.

Изображение выглядит как текст

Автоматически созданное описание

Отчёт по заданию 6. Контроль версий Git и GitHub.

Задача: Зарегистрироваться на сайте <https://github.com> создать свой репозиторий и выложить в нем сделанные программы и отчет по курсу. Прислать ссылку на репозиторий.

Ход работы: задание было выполнено успешно с помощью вспомогательных сайтов

<https://habr.com/ru/post/541258/https://www.youtube.com/watch?v=Rke_Z1-> и <nvUM&list=PL3LQJkGQtzc5rDeb7FjACNb6sOW300yA0&index=1>, а так же самого сайта GitHub и его документации.

Ссылка на личный GitHub - <https://github.com/Anna-Tokar/AnnTokar>

Список используемой литературы.

(дата обращения к каждой ссылке: 29.05.2022).

1. <https://www.sympy.org/ru/index.html>
2. Sympy tutorialspoint. Simply easy learning
3. <https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py>
4. <https://matplotlib.org/stable/users/index.html>
5. [https://habr.com/ru/post/541258/https://www.youtube.com/watch?v=Rke\_Z1-](https://habr.com/ru/post/541258/https:/www.youtube.com/watch?v=Rke_Z1-)
6. [nvUM&list=PL3LQJkGQtzc5rDeb7FjACNb6sOW300yA0&index=1](file:///C:\Users\ACER\Desktop\nvUM&list=PL3LQJkGQtzc5rDeb7FjACNb6sOW300yA0&index=1)
7. <https://github.com>