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

*Хачатрян Артем Анушаванович*

*Иваницкая Алиса Игоревна*

*Манцова Вероника Владимировна*

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

Ростов-на-Дону

2022 г.**Tkinter**

**№3**. Добавить операцию «корень квадратный из x» и «квадрат y» в Radiobutton и в обработчик.

**Решение**. Создаем 3 фрейма: для размещения компоненты для ввода чисел x, y, фрейм для выбора операции и фрейм для вычисления значения и вывода результата. Операцию будем выбирать с помощью виджета Radiobutton, в цикле создаем 6 кнопок Radiobutton, обрабатываем кнопку, создаем ее и метку, к кнопке присоединяем обработчик, затем запускаем цикл обработки сообщений.

**№4**. Вычислите значение функции

Найдите область определения функции, в случае если начальные данные не входят в ООП, выдайте сообщение об этом.

**Решение**. Создаём 2 фрейма: для размещения компонент, задающих x, y и для задания функции и вычисления. Далее создаём обработчик кнопки, который считывает текст из редактора, выполняет его, и результат записывает на метку. Создаём кнопку и метку. Затем запускаем цикл обработки сообщений.

**№18**. Создайте программу с кнопкой. При нажатии на кнопку запускается дополнительное окно Toplevel.

**Решение**. Создаем окно, задаем размеры, создаем новое окно с помощью Toplevel, также задаем размеры окна и запускаем цикл обработки сообщений.

**№24**. С помощью метода размещения grid создайте таблицу редакторов Entry. Заполните их при создании случайными числами. Создайте кнопку, подсчитывающую сумму чисел в редакторах.

**Решение**. Создаём окно с заданными размерами, с помощью цикла и метода grid создаём таблицу и заполняем ее случайными числами, добавляем кнопку, вычисляющую сумму.

**№30**. На основе примера 10 создайте следующую программу. Есть 2 редактора для формул, и в них записано, например, “x+ 1” и “x\* 2”. В программе есть список K размерности 11 (n=0..10), состоящий из нулей, только K[1] =1. В цикле по x от 1 до 10 запустите последовательное вычисление K[x+ 1] += K[x] и K[2\*x] += K[x].

**Решение**. Создаём фрейм для формул, метку с надписью формулы и сами формулы. Далее создаём фрейм для кнопки и саму кнопку. Создаём фреймы для вывода исходного списка и измененного. Также создаем список и функцию для изменения этого списка. Затем запускаем цикл обработки сообщений.

**Файлы**

**№3.** Создайте файл, содержащий случайное (не менее n1 и не более n2) количество случайных целых чисел из промежутка [-m,m]−. Числа в файле разделены пробелами.

**Решение**. Создаем файл, пользователь вводит n1, n2, m, затем randint заполняет данный промежуток целыми числами и в цикле мы числа в файле разделяем пробелами. Закрываем файл.

**№1**7. Переименуйте существующий файл. Проверьте существование файла.

**Решение**. Создаем файл и закрываем, делаем проверку на его существование и переименовываем, затем закрываем.

**№23.** Из двух файлов целых чисел, упорядоченных по возрастанию, создать новый файл чисел, упорядоченных по возрастанию.

**Решение**. Создаем 2 функции: для сортировки списка в порядке возрастания и для создания двух файлов с упорядоченными числами(в порядке возрастания). Создаем список , в который запишем все числа из файлов и создаем третий файл. Циклом в список записываем числа из файлов и сортируем их, затем записываем упорядоченные числа в новый файл и закрываем файлы.

**№26**. Удалить из файла наименьшее нечетное число.

**Решение**. В компоненту, содержащую наименьшее нечетное число, копируем последнюю компоненту файла, а затем удаляем последнюю компоненту с помощью truncate.

**№34**. Сохранение произвольное множество(set) в файл с помощью модуля Pickle. Прочитайте обратно его из этого файла и выведите на экран.

**Решение**. Создаем произвольное множество, создаем файл, сохраняем множество в файл, затем считываем множество из этого файла с помощью load.

**Пакет numpy**

**№2**. Создайте 2 вектора с помощью функции arrange.

**Решение**. Создаем горизонтальный и вертикальный векторы с помощью цикла for.

**№6**. Выдайте на экран информацию о созданном выше массиве: тип (dtype), размерность (ndim, shape), количество элементов (size), itemsize, nbytes.

**Решение**. Задаем несколько массивов разного типа, размера. Создаем еще один массив со всеми массивами. В цикле трансформируем вложенные последовательности в многомерный массив. Затем выводим информацию о созданном массиве.

**№14**. Вычислите норму вектора.

**Решение**. Вводим размер вектора и вводим числа в данный вектор, используем функцию array и с помощью операции norm вычисляем норму вектора.

**№17**. Найдите определитель квадратной матрицы.

**Решение**. Создаем случайную квадратную матрицу, используем функцию array, вычисляем определитель с помощью linalg.det.

**№24**. Определите произвольный массив. Найдите для него среднее значение, дисперсию и медиану.

**Решение**. Создаем произвольный массив, вычисляем среднее значение с помощью nanmean, дисперсию с помощью nanvar и медиану с помощью nanmedian. Выводим полученный результат.

**№26**. Определите, есть ли среди элементов произвольного массива числа большие 100.

**Решение**. Создаем произвольный массив, с помощью any проверяем есть ли в массиве числа больше 100.

**Графика Matplotlib**

Прочитать текст из User’s Guide “Pyplot tutorial” и выполните команды из него.

**Пакет sympy**

**№1(2).** Вычислить

**Решение**. С помощью функции Limit вычисляем предел и затем правосторонний и левосторонний пределы.

**№2(2).** Найти

**Решение**. С помощью Sum вычисляем заданную бесконечную сумму.

**№3(4).** Найти производную функции

**Решение**. С помощью diff находим производную.

**№4(4).** Вычислить интегралы

**Решение**. С помощью integrate вычисляем интеграл.

**№5(3).** Найти все вещественные и комплексные корни полинома

**Решение**. Используем Eq для задания полинома и solve для нахождения корней.

**№6(1).** Нарисовать график функции левой части уравнения:

**Решение**. С помощью plot рисуем график заданного уравнения.

**Git и Github**

Выполнили все задания и отчет, и загрузили git-hab.