ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Калининградский государственный технический университет»

(ФГБОУ ВО «КГТУ»)

Факультет автоматизации производства и управления

Кафедра систем управления и вычислительной техники

**Лабораторная работа №7**

**«Обработка матриц»**

**По дисциплине**

**«Высокоуровневые технологии программирования»**

**Вариант: 3**

|  |  |
| --- | --- |
| Работу принял:  Преподаватель  Высоцкий Леонид Григорьевич  (оценка)  Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Работу выполнил:  студент 2-го курса  учебной группы 19-ИЭ-1  Богданов Максим Дмитриевич    Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Калининград  
2021

**Лабораторная работа № 7**

Обработка матриц

Цель работы: изучение стандартных средств Python для представления и обработки матричных данных.

**Задание на выполнение лабораторной работы:**

В соответствии с вариантом создать проект отображения средствами компонента Canvas (допускается использование других виджетов) матрицы заданного размера. Элементы матрицы формируются генератором случайных чисел в диапазоне -100..100. Занесение исходных значений в матрицу производится по двойному щелчку мыши по канве. Если в матрице находится несколько искомых значений, то все они выделяются требуемым образом.

Дана целочисленная матрица {A*ij*} *i*=1..n, *j*=1..m (n,mє(7..11)). Конкретный размер задается опциями контекстного меню. При нажатии клавиши F2 строка и столбец, на пересечении которых находится элемент, наиболее близким к нулю, выделяются разноцветными рамками.

**Структура проекта**

На рисунке 1 представлен внешний вид программы в виде макета в соответствии с вариантом.

Canvas: c

Рисунок 1. Макет интерфейса программы

Button: btnStart

Описание элементов макета:

Canvas: c – Канва на которой будет проходить основная визуальная работа программы.

Функции и события, используемые в программе:

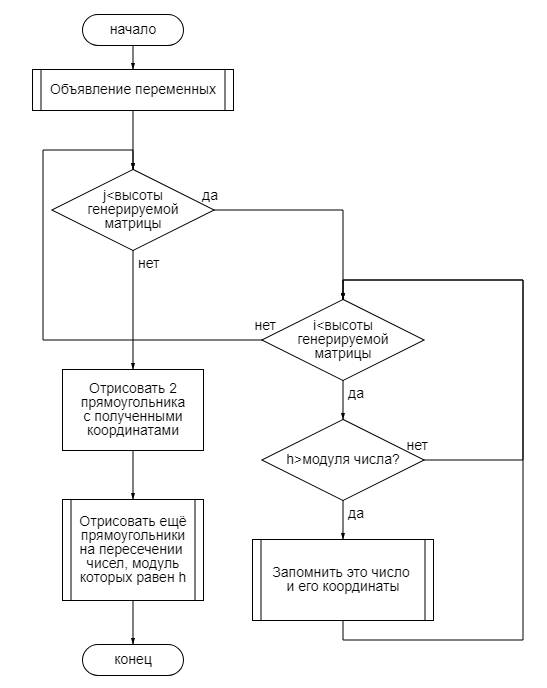
def start: функция, которая генерирует и выводит матрицу на канву.

def ramki: событие, которое рисует рамки, на пересечении которых находятся элементы наиболее близкие к нулю.

**Блок-схема начальной установки, блок-схемы процедур:**

|  |  |
| --- | --- |
| Блок-схема начальной установки: | Блок-схема функции start(): |
|  |  |

Блок-схема функции ramki():

****

**Листинг программы:**

from tkinter import \*

from random import \*

root = Tk()

root.resizable(0, 0)

root.geometry('400x400')

root.title('Лабораторная работа №7')

c = Canvas(width = 396, height = 396, bg = 'lightgrey')

c.place(x = 0, y = 0)

def start(event):

c.delete('all')

X = 30

Y = 30

global arr

arr=[]

for j in range(varJ.get()):

arr.append([])

for i in range(varI.get()):

num = randint(-100,100)

arr[j].append(num)

c.create\_text(X, Y, text = num, font = 'Verdana 10')

X+=30

Y+=30

X = 30

def ramki(event):

h = 101

minI = -1

minJ = -1

for j in range(len(arr)):

for i in range(len(arr[j])):

if (h > abs(arr[j][i])):

h = abs(arr[j][i])

minJ = j

minI = i

if h!=101:

cvet=choice(COLORS)

c.create\_rectangle(minI\*30+15, 15, minI\*30+45, varJ.get()\*30+15, width = 2, outline = cvet)

c.create\_rectangle(15,minJ\*30+15 , varI.get()\*30+15, minJ\*30+45, width = 2, outline = cvet)

for j in range(len(arr)):

for i in range(len(arr[j])):

if (h == abs(arr[j][i])):

cvet=choice(COLORS)

c.create\_rectangle(i\*30+15, 15, i\*30+45, varJ.get()\*30+15, width = 2, outline = cvet)

c.create\_rectangle(15,j\*30+15 , varI.get()\*30+15, j\*30+45, width = 2, outline = cvet)

COLORS = ['azure', 'alice blue', 'lavender','lavender blush', 'misty rose', 'dark slate gray', 'dim gray', 'slate gray', 'light slate gray', 'gray', 'light grey', 'midnight blue', 'navy', 'cornflower blue', 'dark slate blue', 'slate blue', 'medium slate blue', 'light slate blue', 'medium blue', 'royal blue', 'blue', 'dodger blue', 'deep sky blue', 'sky blue', 'light sky blue', 'steel blue', 'light steel blue','light blue', 'powder blue', 'pale turquoise', 'dark turquoise', 'medium turquoise', 'turquoise', 'cyan', 'light cyan', 'cadet blue', 'medium aquamarine', 'aquamarine', 'dark green', 'dark olive green','dark sea green', 'sea green', 'medium sea green', 'light sea green', 'pale green', 'spring green', 'lawn green', 'medium spring green', 'green yellow', 'lime green', 'yellow green', 'forest green', 'olive drab', 'dark khaki', 'khaki', 'pale goldenrod', 'light goldenrod yellow', 'light yellow', 'yellow', 'gold', 'light goldenrod', 'goldenrod', 'dark goldenrod', 'rosy brown', 'indian red', 'saddle brown', 'sandy brown', 'dark salmon', 'salmon',]

varI = IntVar()

varJ = IntVar()

menu = Menu(root, tearoff = 0)

shirina = Menu(root, tearoff = 0)

visota= Menu(root, tearoff = 0)

root.bind("<F2>", ramki)

root.bind("<Button-3>", lambda event: menu.post(event.x\_root, event.y\_root))

c.bind('<Double-Button-1>', start)

menu.add\_cascade(label = "Ширина", menu = shirina)

menu.add\_cascade(label = "Высота", menu = visota)

shirina.add\_checkbutton(label = "7",variable = varI, onvalue = 7, offvalue = 7)

shirina.add\_checkbutton(label = "8",variable = varI, onvalue = 8, offvalue = 8)

shirina.add\_checkbutton(label = "9",variable = varI, onvalue = 9, offvalue = 9)

shirina.add\_checkbutton(label = "10",variable = varI, onvalue = 10,offvalue = 10)

shirina.add\_checkbutton(label = "11",variable = varI, onvalue = 11,offvalue = 11)

visota.add\_checkbutton(label = "7",variable = varJ, onvalue = 7, offvalue = 7)

visota.add\_checkbutton(label = "8",variable = varJ, onvalue = 8, offvalue = 8)

visota.add\_checkbutton(label = "9",variable = varJ, onvalue = 9, offvalue = 9)

visota.add\_checkbutton(label = "10",variable = varJ, onvalue = 10, offvalue = 10)

visota.add\_checkbutton(label = "11",variable = varJ, onvalue = 11, offvalue = 11)

arr=[]

root.mainloop()