Студент группы ИС-23 Скляров В.Д.

Отчёт

Практическое занятие №13.

Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием с матриц в IDE PyCharm Community.

Задача 1

Постановка задачи: В матрице найти минимальный элемент в предпоследней строке.

```
#Импортирование библиотеки рандом
import random

#Генерация случайного размера матрицы
number = random.randint(3, 5)

#Генерация матрицы
def matrix(number):
    matrix = [[1*random.randint(10, 40) for x in range(number)] for i in range(number)] #Генерация матрицы
    yield matrix #Возвращение значения

min_number_func = 0 #Флаг

def min_number(st): #Функция, которая выводит матрицу
    global min_number_func #Глобальная переменная
    print("Вывод матрицы:")
    answ = iter(st[0]) #Итерабельный объект
    while True: #Бесконечный цикл
        try: #Обработчик ошибок
            b = next(answ) #Проходит по всем элементам в итерабельном объекте
            if b == st[0][-2]: #Провека на предпоследнюю строку
                min_number_func = min(b) #Минимальное значение
            print(*b) #вывод строк матрицы
        except StopIteration: #Обработчик ошибки - остановка
            print("Минимальное значение в предпоследней строке:", min_number_func) #Вывод итого мин. элемента
            break
```

```
#Вызов 2-ух функций
answer = list(matrix(number))
min_number(answer)
```

протокол работы программы:

выводится матрица рандомный размер матрицы: [[83, 14, 83], [7, 46, 13], [89, 54, 35]]

Вывод:

Минимальное значение в предпоследней строке: 7

Задача 2

Постановка задачи: квадратной матрице элементы на главной диагонали увеличить в 2 раза.

```
#Импортирование библиотек
import random

#Генерация рандомного четного числа.
def random_number():
    number = random.randint(2, 6)
    return number

#Создание матрицы.
def matrix(n):
    matrix_sq = [[random.randint(1, 30) for i in range(number)] for k in range(number)] #Генератор матрицы
    yield matrix_sq #Возвращает значение

#Вывод матрицы.
def print_matrix(matrix): # Функция которая работает с итер. объектом. Выводя результат
    iter_object = iter(matrix[0]) #Итерируемый объект с индексом 0
    print("Вывод матрицы")
    while True:
        try:
            el_matrix = next(iter_object) # Проходит по каждому элементу
```

```
            print(*el_matrix) # Вывод каждый элемент
        except StopIteration: # обработчик ошибок
            print("StopIteration. Операция закончилась.\n")
            break # Остановка цикла


    print("Вывод измененной матрицы\n")


    iter_object_two = iter(matrix[0])
    chet = 0 # Счетчик
    while True:
        try:
            el_matrix_two = next(iter_object_two)
            el_matrix_two[chet] *= 2 # Умножение каждого элемента на главной диагонали
            print(*el_matrix_two)
            chet += 1 # Прибавление к счетчику значений.
        except StopIteration:
            print("StopIteration. Операция закончилась.")
            break # Остановка цикла


number = random_number() # Вызов 1-й функции, генерация ранд. числа
a = list(matrix(number)) # Вызов 2-й функции, генерация матрицы. В list - списке.
print_matrix(a) # Вызов 3-й функции. Вывод матрицы. Вывод измененной матрицы.
```

протокол работы программы:

выводится матрица рандомный размер матрицы: [[83, 14, 83], [7, 46, 13], [89, 54, 35]]

выводится отредактированный список: [[166, 14, 83], [7, 92, 13], [89, 54, 70]]

Вывод: В процессе выполнения практического занятия выработал навыки составления программ с использованием функции def, lambda, генераторов, итераторов и входящих в него функций, циклов, условий в IDE Pycharm Community. Были использованы языковые конструкции: if, import, for, range, list. Выполнены разработка кода, откладка, тестирование, оптимизация программного кода. Готовые

программные коды выложены на GitHub.