

## ЗАДАНИЯ

### Задача 1

Дан двумерный массив размером  $m \times n$ . Сформируйте новый массив, заменив положительные элементы единицами, а отрицательные нулями. Выведите оба массива.

### Задача 2

Дана целая квадратная матрица  $n$ -го порядка. Определите, является ли она магическим квадратом, т.е. такой матрицей, в которой суммы элементов во всех строках и столбцах одинаковы.

### Задача 3

Требуется упорядочить по возрастанию элементы каждой строки матрицы размером  $n \times m$ .

## Лабораторная работа 8. Создание классов и объектов в Python. Конструктор класса

Класс содержит имя студента `full_name`, номер группы `group_number` и список полученных оценок `progress`. В программе вводится список студентов. Далее список сортируется по имени, потом выводятся студенты, имеющие неудовлетворительные оценки.

```
class Student:
    def __init__(self, full_name="", group_number=0, progress=[]): # конструктор
        self.full_name = full_name # имя

        self.group_number = group_number # номер группы
        self.progress = progress # оценки
    def __str__(self): # печатаемое представление экземпляра класса
        txt = 'Студент: ' + self.full_name + ' Группа: ' + self.group_number
        txt += ' Оценки:'
        for x in self.progress:
            txt += ' ' + str(x) # добавляем список оценок
        return txt
#-----
def SortParam(st): # функция определяющая атрибут для сортировки
    return st.full_name
#-----
```

```

st_size = 5 # количество студентов

students = [] # создание пустого списка
for i in range(st_size): # цикл для ввода st_size студентов
    print("Введите полное имя студента: ")
    full_name = input() # ввод фамилии
    print("Введите номер группы: ")
    group_number = input() # ввод группы
    n=5
    print('Введите ',n,' оценок в столбик: ') # у каждого студента n оценок
    progress = []
    for i in range(n):
        score = int(input()) # ввод оценок
        progress.append(score) # добавление оценок
    # создание экземпляра класса Student:
    st = Student(full_name, group_number, progress)
    students.append(st) # добавление экземпляра в список

print("Students list:")
for st in students: # вывод полного списка студентов
    print(st)

# сортировка по фамилии, ключ сортировки определяется функцией SortParam:
students = sorted(students, key=SortParam)

print("Sorted students:")
for st in students: # вывод отсортированного списка
    print(st)

print("bad students:")
n=0 # счетчик количества неуспевающих
for st in students: # вывод неуспевающих
    for val in st.progress:
        if val<3 : # есть плохая оценка
            print(st) # выводим студента с плохой оцекой
            n += 1
            break
if n == 0:
    print("no matches were found.")

```

## Лабораторная работа 9. Наследование. Множественное наследование

Класс ForeignPassport является производным от класса Passport. Метод PrintInfo существует в обоих классах. PassportList представляет собой список, содержащий объекты обоих классов.

```
class Passport():
    def __init__(self, first_name, last_name, country, date_of_birth,
numb_of_pasport):
        self.first_name = first_name
        self.last_name = last_name
        self.date_of_birth = date_of_birth
        self.country = country
        self.numb_of_pasport = numb_of_pasport

    def PrintInfo(self):
        print("\nFullname: ",self.first_name, " ",self.last_name)
        print("Date of birth: ",self.date_of_birth)
        print("County: ",self.country)
        print("Passport number: ",self.numb_of_pasport)

class ForeignPassport(Passport):
    def __init__(self, first_name, last_name, country, date_of_birth,
numb_of_pasport,visa):
        super().__init__(first_name, last_name, country, date_of_birth,
numb_of_pasport)
        self.visa = visa

    def PrintInfo(self):
        super().PrintInfo()
        print("Visa: ",self.visa)

PassportList=[]
request = ForeignPassport('Ivan', 'Ivanov', 'Russia', '12.03.1967', '123456789',
'USA')
PassportList.append(request)

request = Passport('Иван', 'Иванов', 'Россия', '12.03.1967', '45001432')
PassportList.append(request)

request = ForeignPassport('Peter', 'Smit', 'USA', '01.03.1990', '21435688',
'Germany')
PassportList.append(request)

for emp in PassportList:
    emp.PrintInfo()
```