

Практическое задание, модуль 2

по курсу "Язык программирования Python"

Обязательные требования:

- Решение должно размещаться сразу после ячейки с заданием в одной или нескольких ячейках;
- Каждое решение должно быть независимо от решений других заданий, т.е. все необходимые импорты должны присутствовать в решении каждого задания;
- В каждом решении результат должен быть выведен на экран;
- Чем более общее решение, тем выше оценка.

Сделано:

Задача 1 +
Задача 2 +
Задача 3 +
Задача 4 +
Задача 5 +-
Задача 6 -

Задание 1 (10 баллов)

Необходимо реализовать односторонний **бинарный клеточный автомат**. Продемонстрировать работу программы на правилах 158 и 118. Сделать не менее 100 итераций. За использование каждого цикла `for` более 1 раза - **штраф** 5 баллов.

```
In [49]: import cellpylib as cpl
```

Для правила 158

```
In [50]: cellular_automaton = cpl.init_simple(100)

cellular_automaton = cpl.evolve(cellular_automaton, timesteps=100, apply_rule=lambda n, c, t: cpl.nks_rule(n))
cpl.plot(cellular_automaton)
```

Для правила 18

```
In [51]: cellular_automaton = cpl.init_simple(100)

cellular_automaton = cpl.evolve(cellular_automaton, timesteps=100, apply_rule=lambda n, c, t: cpl.nks_rule(n))
cpl.plot(cellular_automaton)
```

Задание 2 (20 баллов)

В списке `vecs` - координаты `x`, `y` векторов в двумерном пространстве. Рассчитать:

- Длину ломаной линии, соединяющей последовательно концы векторов;
- Углы между предыдущим и следующим векторами;
- Площади треугольников, вершины которых заданы концами `i`, `i+1`, `i+2` векторов.

Использование любых циклов, кроме встроенных в `Numpy` **запрещено**, за каждый такой цикл - **штраф** 5 баллов.

```
vecs = [[6.44, 5.05],
        [3.59, 7.71],
        [8.81, 2.02],
        [8.71, 7.8 ],
        [5.51, 2.53],
        [6.35, 4.89],
        [7.88, 3.52],
        [4.89, 5.83],
        [1.66, 5.24],
        [3.1 , 8.91],
        [5.11, 8.22],
        [7.06, 7.15],
        [2.15, 2.78],
        [7.25, 8.94],
        [1.62, 4.97],
        [8.86, 4.11],
        [8.59, 8.38],
        [2.18, 7.01],
        [8.24, 8.57],
        [9.97, 1.82]]

In [14]: import numpy as np

In [15]: vecs = [[6.44, 5.05],
                [3.59, 7.71],
                [8.81, 2.02],
                [8.71, 7.8 ],
                [5.51, 2.53],
                [6.35, 4.89],
                [7.88, 3.52],
                [4.89, 5.83],
                [1.66, 5.24],
                [3.1 , 8.91],
                [5.11, 8.22],
                [7.06, 7.15],
                [2.15, 2.78],
                [7.25, 8.94],
                [1.62, 4.97],
                [8.86, 4.11],
                [8.59, 8.38],
                [2.18, 7.01],
                [8.24, 8.57],
                [9.97, 1.82]]

len(vecs)

massiv = np.array(vecs)
```

Пункт 1

```
In [31]: dlina_lom_lin = np.linalg.norm(massiv[1:]-massiv[:-1], axis = 1).sum()
print(dlina_lom_lin)

96.28806530780746
```

Пункт 2

```
In [32]: f2 = massiv[1:2]
f1 = massiv[:-2]

chisl = f1[:,0]*f2[:,0]+f1[:,1]*f2[:,1]
znam = np.linalg.norm(l1,axis=1)*np.linalg.norm(l2,axis=1)

angle_between_vectors = a/b
itog_angle_vectors = np.arccos(angle_between_vectors)*(180/np.pi)
print(itog_angle_vectors)

[25.18833115 23.18678794 11.74909345 4.24605385 5.59259866 12.41214944
 48.35177656 20.80475116 14.28945276 25.45312202 5.85035943 5.5963802
 19.66402009 26.07349619 27.6553106 47.83936013 1.83362526 62.37982712]
```

Пункт 3

```
In [34]: dlina_otrez = massiv[1:]-massiv[:-1]
t1 = dlina_otrez[:-1]
t2 = dlina_otrez[1:]
ploshad = np.abs(np.cross(t1,t2)/2)
print(ploshad)

[ 1.16565 14.8013  9.5115  1.5626  2.3808  0.281  4.6127  5.50225
 4.18915  0.4026  6.8876  3.9793  7.2169 16.7923 15.3413 13.8703
 0.8487 21.8016 ]
```

Задание 3 (10 баллов)

В архиве `evolution.zip` находятся 360 файлов с именами `evolution_NNN.txt`, где `NNN = 000...359` - номер орбиты. В первом столбце каждого файла записаны расстояния (в км) периферии орбиты (ближайших к планете точек) на каждый из 225 дней.

Необходимо:

- При помощи `Python` распаковать `zip` файл.
- Сформировать `DataFrame`, в котором 225 столбцов и 360 строк. Каждая строка соответствует орбите, а столбец - дню. В ячейках записаны расстояния периферии.
- Найти номера орбит, расстояние периферии которых всегда больше 6350 км.
- Изобразить цветовую карту (`contourf`) по оси абсцисс - номер орбиты, по оси ординат - номер дня, цветом отображены расстояния периферии. Рядом с цветовой картой изобразить `colorbar`.
- Изобразить поверх цветовой карты изолинии (`contour`), соответствующие уровням 6100, 6250, 6350 км.

```
In [83]: from zipfile import ZipFile
import pandas as pd
import matplotlib.pyplot as plt

In [84]: df1 = []

with ZipFile("evolution.zip", 'r') as myzip:
    for i in range(360):
        with myzip.open("evolution_{:03d}.txt".format(i)) as text:
            df = pd.read_table(text, sep = "\s+", usecols = ['Sc,Venus,RMAG'])
            df.columns = [i]
            df = df.transpose()
            df1.append(df)

df = pd.concat(df1)
y = df.columns
x = df.index
z = df.values.T

points = plt.contourf(x,y,z)

a = plt.contour(points, levels = [6100,6250,6350])

plt.colorbar(points)

plt.show()
```

Задание 4 (15 баллов)

Рассчитать и изобразить:

- Вариант 1: **множество Мандельброта**
- Вариант 2: **фрактал Ляпунова**
- Вариант 3: **бассейн Ньютона**

Примерный план решения задачи:

- написать функцию для расчета в одной точке плоскости
- написать функцию для расчета на прямоугольной равномерной решетке
- выполнить расчет и получить матрицу значений
- при помощи `imshow` изобразить полученную матрицу
- использовать декоратор `@njit` модуля `numba` для ускорения вычислений

Вариант:

```
In [65]: 20 % 3 + 1

Out[65]: 3

In [90]: import numpy as np
import math
import matplotlib.pyplot as plt
from math import sqrt
from numba import njit

@njit
def raschet_1():
    W = 854
    H = 480
    X0 = W // 2
    Y0 = H // 2
    PXL = 0.1
    EPS = 0.01
    PI5 = 0.628318530717959

    img = np.zeros((H, W))
    for i in range(H):
        for j in range(W):
            x = X0 + i * PXL
            y = (j - X0) * PXL
            z = complex(x, y)
            if x > 0 or y:
                t = complex(0, 0)
                while sqrt((z.real - t.real) ** 2 + (z.imag - t.imag) ** 2) >= EPS:
                    t = z
                    z = 0.8 * z + 0.2 * pow(z, -4)
                    a = int(math.phase(z) / PI5)
                    col = 0

                if a == 0:
                    col = 5
                elif a == 2 or a == 1:
                    col = 4
                elif a == 4 or a == 3:
                    col = 3
                elif a == -4 or a == -3:
                    col = 2
                elif a == -2 or a == -1:
                    col = 1
                img[i][j] = col
            return img

@njit
def raschet_2():
    iter = 50
    ma = 1000000
    mi = 1 / 1000000
    img = np.zeros((800, 800))

    for y in range(-400, 400):
        for x in range(-400, 400):
            n = 0
            z = complex(x * 0.005, y * 0.005)
            d = z
            while t = z
                p = (((t.real) ** 2 + (t.imag) ** 2) ** 2
                    x = z / 3 * t.real + ((t.real) ** 2 - (t.imag) ** 2) / (3 * p)
                    z = complex(x, z.imag)
                    y = z / 3 * t.imag * (1 - t.real / p)
                    x_d = abs(t.real - z.real)
                    y_d = abs(t.imag - z.imag)
                    d = complex(x_d, y_d)
                    n += 1
                    if n >= 9 % 255:
                        img[400 + x][400 + y] = col
            return img

def first():
    img_1 = raschet_1()
    img_2 = raschet_2()

    ax = plt.subplot(1, 2, 1)
    ax.imshow(img_1)
    ax = plt.subplot(1, 2, 2)
    ax.imshow(img_2)

    plt.show()

if __name__ == '__main__':
    first()
```

```
0
0 200 400 600 800
200 400 600 800
0 200 400 600 800
```

Задание 5 (15 баллов)

В файле `fin.csv` находится информация о доходах и расходах физического лица.

Требуется используя модуль `Pandas` рассчитать:

- среднемесячный доход для каждого года
- среднемесячный расход по каждой категории для каждого года
- сводную таблицу суммарных расходов: строки - Категория/Год, столбцы - Месяц.
- для каждой пары (Категория, Год) изобразить столбчатую диаграмму: суммарные расходы в каждом месяце (использовать таблицу из п.3).
- ежедневный баланс (т.е. сколько денег есть у физ. лица каждый день от начала и до конца периода, представленного в файле) и изобразить график.

```
In [42]: import pandas as pd
import numpy as np

In [114]: df = pd.read_csv("fin.csv")
df[["Год"]]=df[["BPEMЯ"]].astype(str).str[:4]
df[["Месяц"]]=df[["BPEMЯ"]].astype(str).str[5:7]
df
```

```
Out[114]:
```

	ВРЕМЯ	СУММА	КАТЕГОРИЯ	Год	Месяц
0	2017-04-28 01:58:00	-205.0	Еда	2017	04
1	2017-04-28 22:43:00	-153.0	Интернет и телефон	2017	04
2	2017-04-28 22:43:00	-600.0	Интернет и телефон	2017	04
3	2017-04-29 13:59:00	-265.0	Еда	2017	04
4	2017-04-29 13:59:00	-5000.0	Путешествия	2017	04
...
2071	2019-09-14 17:56:00	-10996.0	Одежда и обувь	2019	09
2072	2019-09-14 19:11:00	-120.0	Еда	2019	09
2073	2019-09-15 12:16:00	-1300.0	Еда	2019	09
2074	2019-09-15 20:03:00	-1481.0	Еда	2019	09
2075	2019-09-16 19:28:00	-1568.0	Еда	2019	09

2076 rows x 5 columns

```
In [117]: df[["КАТЕГОРИЯ"]].unique()

Out[117]: array(['Еда', 'Интернет и телефон', 'Путешествия', 'Доходы',
        'Спорты', 'Программы', 'Транспорт', 'Подарки', 'Развлечения',
        'Комплектующие', 'Дом', 'Стройматериалы', 'Автомобиль',
        'Красота и здоровье', 'Одежда и обувь', 'Спорт', 'Хобби',
        'Здоровье', 'Обучение', 'Украшения'], dtype=object)
```

Пункт 1

```
In [110]: for item in ["2017", "2018", "2019"]:
    god = df[df[["КАТЕГОРИЯ"]=="Доходы"]&(df.Год == item)]
    per = god["СУММА"].sum()/len(set(god.Месяц))
    print("Среднемесячный доход за", item, "год - ", per, "\n")
```

Среднемесячный доход за 2017 год - 151685.66666666666

Среднемесячный доход за 2018 год - 186374.91666666666

Среднемесячный доход за 2019 год - 200122.88888888888

Пункт 2

```
In [113]: list_kat = list(set(df.КАТЕГОРИЯ))
for item in ["2017", "2018", "2019"]:
    for elements in list_kat:
        god = df[(df[["КАТЕГОРИЯ"]== elements)&(df.Год == item)]
        if elements != "Доходы":
            if god["СУММА"].sum() != 0:
                per = god["СУММА"].sum()/len(set(god.Месяц))
                print("Среднемесячный расход за ", item, ", в категории:", elements.upper(), "составляет = ", per, "\n")
```

Среднемесячный расход за 2017, в категории: ТРАНСПОРТ составляет = -2506.0
Среднемесячный расход за 2017, в категории: ДОМ составляет = -44376.142857142855
Среднемесячный расход за 2017, в категории: СПОРТ составляет = -13496.666666666666
Среднемесячный расход за 2017, в категории: КОМПЬЮТЕР И ПРОГРАММЫ составляет = -13725.4
Среднемесячный расход за 2017, в категории: ПУТЕШЕСТВИЯ составляет = -52241.4
Среднемесячный расход за 2017, в категории: ДОХОДЫ составляет = -1200.0
Среднемесячный расход за 2017, в категории: ЕДА составляет = -19732.333333333332
Среднемесячный расход за 2017, в категории: УКРАШЕНИЯ составляет = -3466.6666666666665
Среднемесячный расход за 2017, в категории: ОБУЧЕНИЕ составляет = -3267.0
Среднемесячный расход за 2017, в категории: ПОДАРОКИ составляет = -35919.142857142855
Среднемесячный расход за 2017, в категории: СТРОЙМАТЕРИАЛЫ составляет = -1100.0
Среднемесячный расход за 2017, в категории: ДОМ составляет = -4793.3636363636364
Среднемесячный расход за 2017, в категории: ОДЕЖДА И ОБУВЬ составляет = -9066.4
Среднемесячный расход за 2017, в категории: АВТОМОБИЛЬ составляет = -1800.0
Среднемесячный расход за 2017, в категории: ЗДОРОВЬЕ составляет = -928.25
Среднемесячный расход за 2017, в категории: ИНТЕРНЕТ И ТЕЛЕФОН составляет = -1936.2
Среднемесячный расход за 2017, в категории: ТРАНСПОРТ составляет = -5188.222222222223
Среднемесячный расход за 2017, в категории: ОБУЧЕНИЕ составляет = -3027.5
Среднемесячный расход за 2017, в категории: ПУТЕШЕСТВИЯ составляет = -3469.1666666666665
Среднемесячный расход за 2018, в категории: ДОХОДЫ составляет = -5225.2
Среднемесячный расход за 2018, в категории: ЕДА составляет = -27533.083333333332
Среднемесячный расход за 2018, в категории: УКРАШЕНИЯ составляет = -4800.0
Среднемесячный расход за 2018, в категории: ОБУЧЕНИЕ составляет = -2524.777777777778
Среднемесячный расход за 2018, в категории: КОМПЛЕКТУЮЩИЕ составляет = -12178.285714285714
Среднемесячный расход за 2018, в категории: КРАСОТА И ЗДОРОВЬЕ составляет = -4000.0
Среднемесячный расход за 2018, в категории: ОДЕЖДА И ОБУВЬ составляет = -8145.777777777778
Среднемесячный расход за 2018, в категории: ПОДАРОКИ составляет = -19432.363636363636
Среднемесячный расход за 2018, в категории: ИНТЕРНЕТ И ТЕЛЕФОН составляет = -1173.4444444444443
Среднемесячный расход за 2018, в категории: РАЗВЛЕЧЕНИЯ составляет = -2350.0
Среднемесячный расход за 2018, в категории: ТРАНСПОРТ составляет = -5188.222222222223
Среднемесячный расход за 2018, в категории: ДОМ составляет = -44966.555555555555
Среднемесячный расход за 2018, в категории: УКРАШЕНИЯ составляет = -4000.0
Среднемесячный расход за 2018, в категории: КОМПЬЮТЕР И ПРОГРАММЫ составляет = -512.33333333333334
Среднемесячный расход за 2018, в категории: ПУТЕШЕСТВИЯ составляет = -36687.5
Среднемесячный расход за 2018, в категории: ДОХОДЫ составляет = -3451.375
Среднемесячный расход за 2018, в категории: ЕДА составляет = -29928.88888888889
Среднемесячный расход за 2018, в категории: ОДЕЖДА И ОБУВЬ составляет = -7502.0
Среднемесячный расход за 2018, в категории: ПОДАРОКИ составляет = -10450.222222222223
Среднемесячный расход за 2018, в категории: ЗДОРОВЬЕ составляет = -2144.0
Среднемесячный расход за 2018, в категории: ИНТЕРНЕТ И ТЕЛЕФОН составляет = -1383.75
Среднемесячный расход за 2018, в категории: РАЗВЛЕЧЕНИЯ составляет = -6733.333333333333

Задание 6 (30 баллов)

Ивану Петровичу по наследству досталась квартира. Пол в одной из комнат размером 4 на 5 метров требует серьезного ремонта. Как выяснилось, неровности пола варьируются от 3 мм до 5 см. Иван Петрович решил для выравнивания применить наливной пол, но не знает как посчитать количество мешков, чтобы оформить заказ. Для начала ему необходимо посчитать объем неровностей. В youtube Иван Петрович увидел рекомендацию, что нужно сделать несколько замеров глубины неровностей относительно нулевой отметки, усреднить эти числа и умножить на площадь (метод среднего слоя), а затем учесть расход сухой смеси. Насколько Иван Петрович ошибется, если воспользуется этим советом?

Для решения задания этого вопроса предлагается рассчитать величину погрешности:

- В случаях, когда замеры проводятся на равномерной сетке
 - 3x3;
 - 5x5;
 - 9x9;
- Вычисление объема производится:
 - метод среднего слоя;
 - интерполяция сплайновой поверхностью 2 порядка и вычисление интеграла (объема под поверхностью);

Так как задача актуальна для любого Ивана Петровича, то необходимо рассмотреть случайное равномерное распределение неровностей пола и для каждого варианта замера и расчета провести по 1000 экспериментов. Полученные значения погрешности необходимо усреднить и посчитать дисперсию. Построить сравнительную столбчатую диаграмму для средних погрешностей и их дисперсий.

Допущения:

- Считать, что неровности пола случайны и распределены равномерно.
- Для точной величины взять замеры, выполненные на равномерной сетке 17x17, интерполяция поверхностью 3 порядка.
- Расход сухой смеси считать за 10 кг на кв. м. при толщине слоя наливного пола 10 мм.
- В одном мешке - 25 кг сухой смеси.
- Интерполяцию можно выполнить при помощи `scipy.interpolate.RectBivariateSpline`. Этот класс позволяет построить поверхность заданного порядка по набору измерений, а также вычислить объем под поверхностью (т.е. взять

