

Работа № 1 Основы Jupyter Notebook

1.1. Введение

Jupyter-ноутбуки применяют для визуализации данных в *big data* и *data science*.

Jupyter-ноутбук — это среда разработки, где сразу можно видеть результат выполнения кода и его отдельных фрагментов.

Отличие от традиционной среды разработки в том, что код можно разбить на куски и выполнять их в произвольном порядке. Представьте, что вы можете написать кусочек кода на салфетке и сказать салфетке: «Выполнись».

В такой среде разработки можно, например, написать функцию и сразу проверить её работу, без запуска программы целиком. А ещё можно поменять порядок выполнения кода. Можно отдельно загрузить файл в память, отдельно проверить его содержимое, отдельно обработать содержимое.

А ещё в *jupyter-ноутбуках* есть вывод результата сразу после фрагмента кода. Например, можно прямо в середине кода увидеть построенный график, получить предварительные цифры или любую другую визуализацию.

Чаще всего, когда говорят о *jupyter-ноутбуках*, имеют в виду работу с *Python*. Но на самом деле можно работать и с другими:

- Ruby
- Perl
- R
- bash-скрипты

Для этого используют специальные магические команды, которые так и называются — *magic-команды*. Они позволяют запускать код на других языках, а ещё существенно расширяют возможности обычного Python.

1.2. Форматирование текста

При наборе текста в ячейке типа Markdown для его стилизации можно использовать следующие приемы:

1. заголовки создаются с помощью символа «#» и следующего за ним пробела; один такой символ сделает крупный заголовок верхнего уровня, два — заголовок поменьше и т.д. до шестого уровня.

Пример:

Это третий уровень

2. символ звездочки «*» с двух сторон текста нужен для оформления этого текста курсивом, а с помощью \ перед каждой * можно предотвратить выделение курсивом: *Jupiter-Notebook* или _Не курсив_

3. два символа звездочки «**» с двух сторон текста позволяют придать тексту полужирное начертание, а с помощью \ перед каждой * можно предотвратить выделение курсивом: **Жирный** или **Не жирный**

4. три символа звездочки «***» с двух сторон текста позволяют оформить текст жирным курсивом, а с помощью \ перед каждой * можно предотвратить выделение курсивом: **Жирный курсив** или ***Не жирный курсив***

5. для создания маркированного списка, каждый его элемент начинают с символов «*», «–» или «+» и пробела:

- 1
- 2
- 3

6. нумерованные списки начинают с цифры и точки, причём цифры могут быть любыми и идти не по порядку(примером нумерованного списка является это перечисление).

7. символ «<u>» с левой стороны и «</u>» с другой стороны текста можно оформить его подчеркиванием: подчеркнуто

8. символ ` с двух сторон текста нужен для оформления этого текста как оператор языка: for a in A

9. с помощью функционала HTML можно окрашивать текст в различные цвета; делается это добавлением (без скобок) <(font color="red")> в начале и <(/font)> в конце: красный!

10. с помощью трех ` и названию языка после них перед частью текста и трех ` в конце текста можно оформить его как фрагмент программного кода:

```
def count_by(arr, fn=lambda x: x):
    key = {}
    for el in map(fn, arr):
        key[el] = 0 if el not in key else key[el]
        key[el] += 1
    return key
```

11. с помощью «<» можно оформить текст как цитату:

Jupyter-ноутбук — это среда разработки, где сразу можно видеть результат выполнения кода и его отдельных фрагментов.

1.3. Создание списков

Это было разобрано в разделе с форматированием текста, но следует расширить эту информацию:

Для создания маркированного списка, каждый его элемент начинают с символов «*», «–» или «+» и пробела.

- 1

- 2
- 3

нумерованные списки начинают с цифры и точки, причём цифры могут быть любыми и идти не по порядку(примером нумерованного списка является это перечисление).

1. 1
2. 2
3. 3

многоуровневые списки разделяются на уровни; чтобы сделать многоуровневый список, нужно будет на следующей строке сделать отступ с помощью TAB:

1. 1
 - A. 1
 - B. 2
 - C. 3
2. 2
 - A. 1
 - B. 2
 - C. 3

1.4. Добавление таблиц

Разметка Markdown позволяет вставлять таблицы. Для разметки таблицы используются всего два символа: вертикальная черта «|» и дефис «-». Дефисы работают примерно так же, как в случае с горизонтальной линией: отделяют заголовки от других строк, при этом количество символов значения не имеет. Вертикальная черта служит границей между столбцами.

Пример:

views	tag	time
122400	cat	33 day
51400	fox	30 day
99000	dog	37 day

1.5. Вставка изображений и видео

Синтаксис Markdown для вставки картинок очень похож на разметку, используемую для вставки ссылок. Разница в восклицательном знаке перед первыми квадратными скобками:

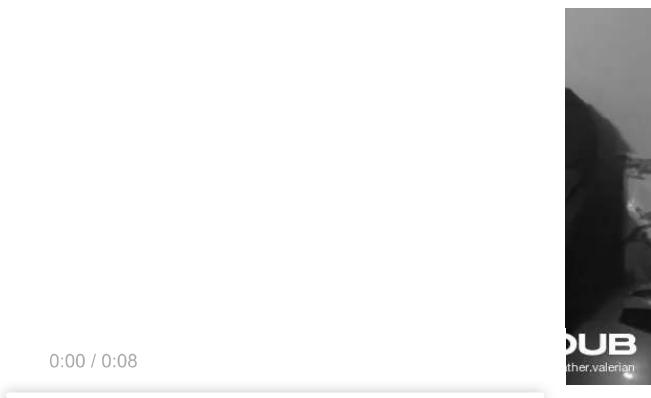
![Alt-текст изображения](Источник изображения«Необязательный заголовок изображения»)

Пример картинки:



Html-теги можно использовать и для вставки в ноутбук видеофайла, например можно использовать тег <video controls src="...">

Пример видео:



1.6. Вставка ссылок на интернет ресурсы

Чтобы добавить гиперссылку, нужно добавить текст ссылки, заключив его в квадратные скобки и рядом в круглых скобках поместить URL ресурса и при необходимости текст заголовка (между квадратными и круглыми скобками не должно быть пробела): [Текст ссылки](URL «Необязательный заголовок»)

Пример:

Посмотрите сайт с забавными и познавательными видео [сайт](https://coub.com/) (<https://coub.com/>)

Посмотрите сайт с забавными и познавательными видео [сайт](https://coub.com/) (<https://coub.com/>)

Сайт с забавными и познавательными видео <https://coub.com/> (<https://coub.com/>)

1.7. Вставка фрагментов программного кода

Это было разобрано в разделе с форматированием текста, но продублирую эту информацию:

С помощью трех «`» и названию языка после них перед частью текста и трех «`» в конце текста можно оформить его как фрагмент программного кода:

```
def count_by(arr, fn=lambda x: x):
    key = {}
    for el in map(fn, arr):
        key[el] = 0 if el not in key else key[el]
        key[el] += 1
    return key
```

1.8. Запись формул в формате LaTex

Формулы, вставляемые в ячейки типа *Markdown*, должны быть подготовлены в соответствии с синтаксисом *LaTeX*.

LaTeX – система набора и верстки текстов с формулами, стандарт для обмена и публикации научных документов.

При подготовке формул используются следующие специальные символы, которые не выводятся на экран:

- «{» – начало группы;
- «}» – конец группы;
- «_» – нижние индексы;
- «^» – верхние индексы;
- «\» – сигнальный символ команд.

Пример стандартной формулы:

$$x + \left(y^2 + \frac{x^2 + z}{1 + z^2} \right)$$

Пример сложной формулы:

$$\frac{(x_1 x_2) \\ \times (x'_1 x'_2)}{(y_1 y_2 y_3 y_4)}$$

Работа № 2 Программирование линейных алгоритмов

2.1. Задание

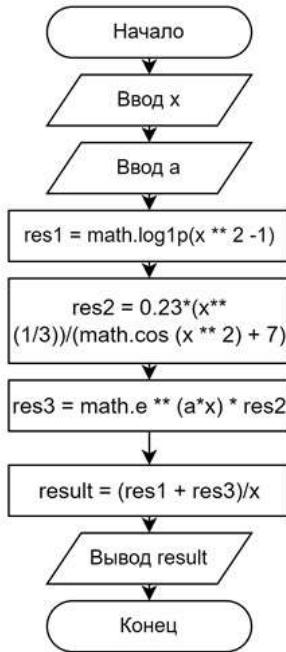
Получение базовых знаний об объектах программы на языке *Python*, а также получение практических навыков написания программ с линейной структурой.

Для задания в соответствии со своим вариантом составить алгоритм и написать программу, реализующую ввод исходных данных, вычисление значения и вывод результата. Оформить отчет, содержащий текст варианта задания, схему алгоритма, код программы и результаты выполнения.

Во всех вариантах задания входных параметров (x , a) задаются произвольно при вводе исходных данных.

$$y = \frac{\ln(x^2) + e^{ax} * \frac{0.23 * \sqrt[3]{x}}{\cos(x^2) + 7}}{x}$$

2.2. Блок схема



2.3. Листинг программы

```

import math

x = float(input("Введите x: "))
a = float(input("Введите a: "))

res1 = math.log1p(x ** 2 - 1)
res2 = 0.23*(x**(1/3))/(math.cos (x ** 2) + 7)
res3 = math.e ** (a*x) * res2

result = (res1 + res3)/x

print('Y =', result)
  
```

2.4. Результат работы программы в таблице

x	a	Y
3	1	1.097156
6	2	1650.311330
0.5	12	15.710831

2.5. Контрольный расчет (MathCAD)

$x := 3 \quad a := 1$ $y := \frac{\ln(x^2) + e^{a \cdot x} \cdot \frac{0.23 \cdot \sqrt[3]{x}}{\cos(x^2) + 7}}{x} = 15.711$	$x := 0.5 \quad a := 12$ $y := \frac{\ln(x^2) + e^{a \cdot x} \cdot \frac{0.23 \cdot \sqrt[3]{x}}{\cos(x^2) + 7}}{x} = 15.711$
$x := 6 \quad a := 2$ $y := \frac{\ln(x^2) + e^{a \cdot x} \cdot \frac{0.23 \cdot \sqrt[3]{x}}{\cos(x^2) + 7}}{x} = 1650.311$	

3 Работа № 3 Условный оператор if

3.1. Задание

Освоение технологии проектирования алгоритма и программы разветвляющейся структуры с использованием оператора условного перехода на языке Python.

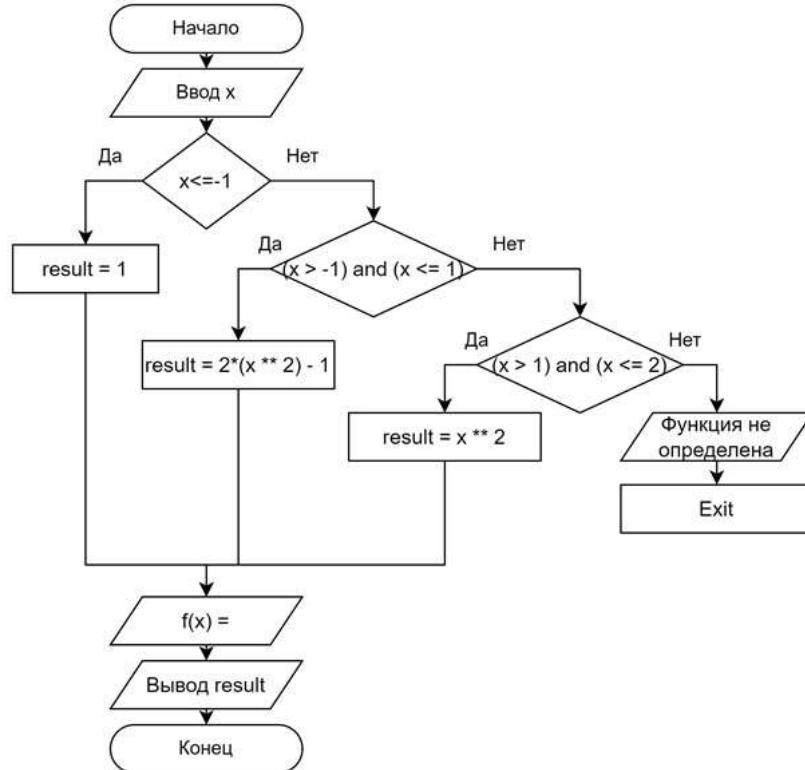
Для задания в соответствии со своим вариантом составить алгоритм и написать программу, реализующую выполнение определенных действий в зависимости от выполнения заданного условия.

Вычислить значение функции $f(x)$ для вводимого значения x :

$$f(x) = \begin{cases} 1, & x \leq -1; \\ 2x^2 - 1, & -1 < x \leq 1; \\ x^2, & 1 < x \leq 2. \end{cases}$$

При $x > 2$ функция $f(x)$ не определена.

3.2. Блок схема



3.3. Листинг программы

```
import math

x = float(input("Введите x: "))

if (x<=-1):
    result = 1
elif (x > -1) and (x <= 1):
    result = 2*(x ** 2) - 1
elif (x > 1) and (x <= 2):
    result = x ** 2
else:
    print('Функция не определена')
    exit()

print('f(x) =', result)
```

3.4. Результат работы программы в таблице

x	$f(x)$
1.5	2.25
-1.5	1

3 Функция неопределена

4 Работа № 4 Множественное ветвление

4.1. Задание

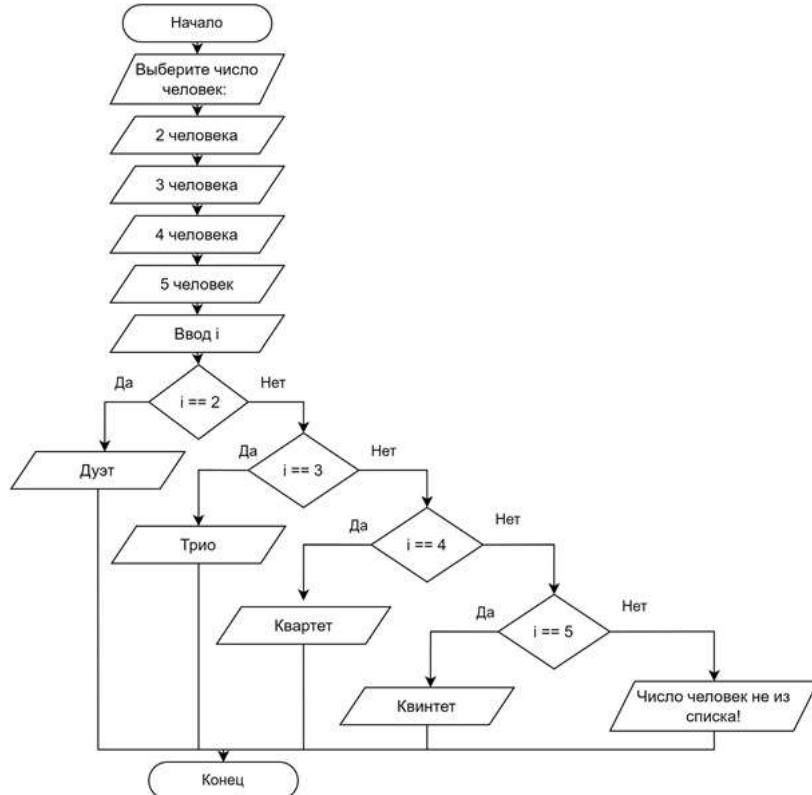
Изучение оператора if-elif-else для организации многоальтернативного ветвления.

Для задания в соответствии в соответствии со своим вариантом составить алгоритм и написать программу, реализующую выполнение определенных действий в зависимости от результатов проверки заданного набора условий.

Для введенного числа исполнителей вывести название ансамбля из данного числа людей:

1. 2 человека - дуэт;
2. 3 человека - трио;
3. 4 человека - квартет;
4. 5 человек - квинтет;

4.2. Блок-схема



4.3. Листинг программы

```
print('Выберите число человек: ')
print('2 человека')
print('3 человека')
print('4 человека')
print('5 человек')

i = int(input('Число человек: '))
if i == 2:
    print('дуэт')
elif i == 3:
    print('трио')
elif i == 4:
    print('квартет')
elif i == 5:
    print('квинтет')
else:
    print('Число человек не из списка!')
```

4.4. Результат работы программы в таблице

x	result
5	квинтет
1	Число человек не из списка!
3	Трио

5 Работа № 5 Цикл while

5.1. Задание

Знакомство с простейшими приемами программирования циклических программ.

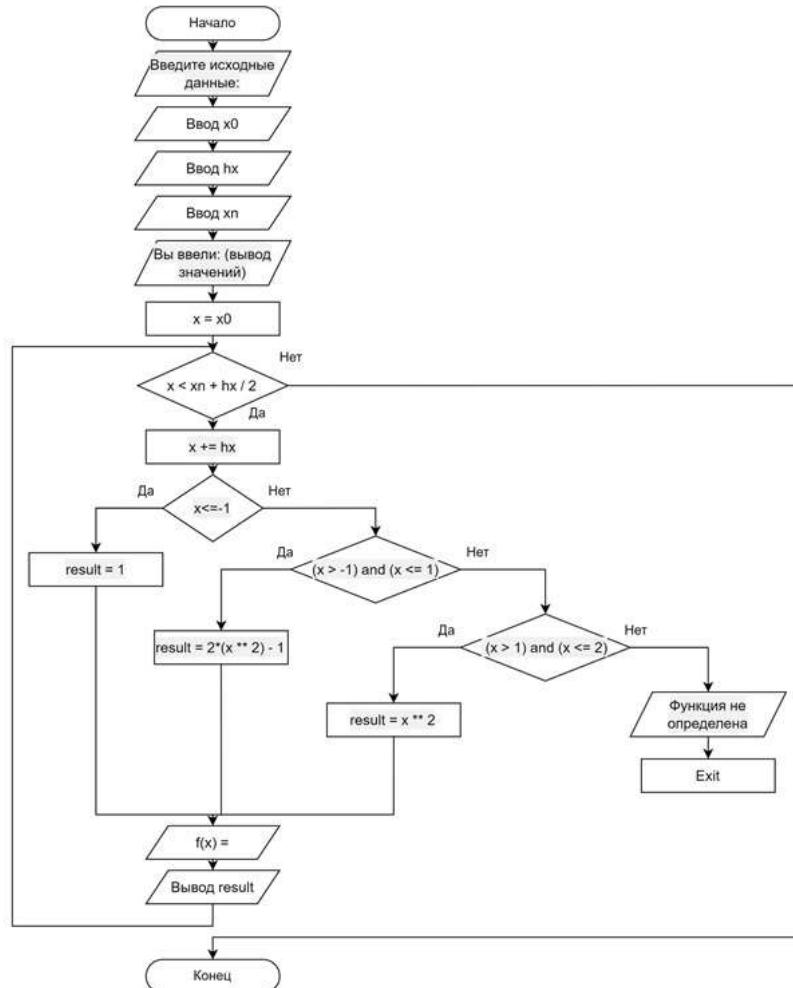
Для задания в соответствии со своим вариантом составить алгоритм и написать программу, имеющую структуру цикла с предусловием и осуществляющие обработку тела цикла при изменении аргумента от начального значения до конечного значения с постоянным шагом.

Вычислить значение функции $f(x)$ для $x = x_0 + h \cdot x_n$:

$$f(x) = \begin{cases} 1, & x \leq -1; \\ 2x^2 - 1, & -1 < x \leq 1; \\ x^2, & 1 < x \leq 2. \end{cases}$$

При $x > 2$ функция $f(x)$ не определена.

5.2. Блок-схема



5.3. Листинг программы

```
import math

print("Введите исходные данные:")
print("x0 = ", end='')
x0 = float(input())
print("hx = ", end='')
hx = float(input())
print("xn = ", end='')
xn = float(input())
print("Вы ввели:")
print("x0 = %.2f hx = %.2f xn = %.2f" % (x0, hx, xn))
x = x0

result = 0

while x < xn + hx / 2:
    x += hx
    if (x<=-1):
        result = 1
    elif (x > -1) and (x <= 1):
        result = 2*(x ** 2) - 1
    elif (x > 1) and (x <= 2):
        result = x ** 2
    else:
        print('Функция не определена')
        exit()
print('f(x) =', result)
```

5.4. Результат работы программы в таблице

Nº	x0	hx	xn	f(x)
1	-1	0.5	3	-0.5
2	-1	0.5	3	-1.0
3	-1	0.5	3	-0.5
4	-1	0.5	3	1.0
5	-1	0.5	3	2.25
6	-1	0.5	3	4.0

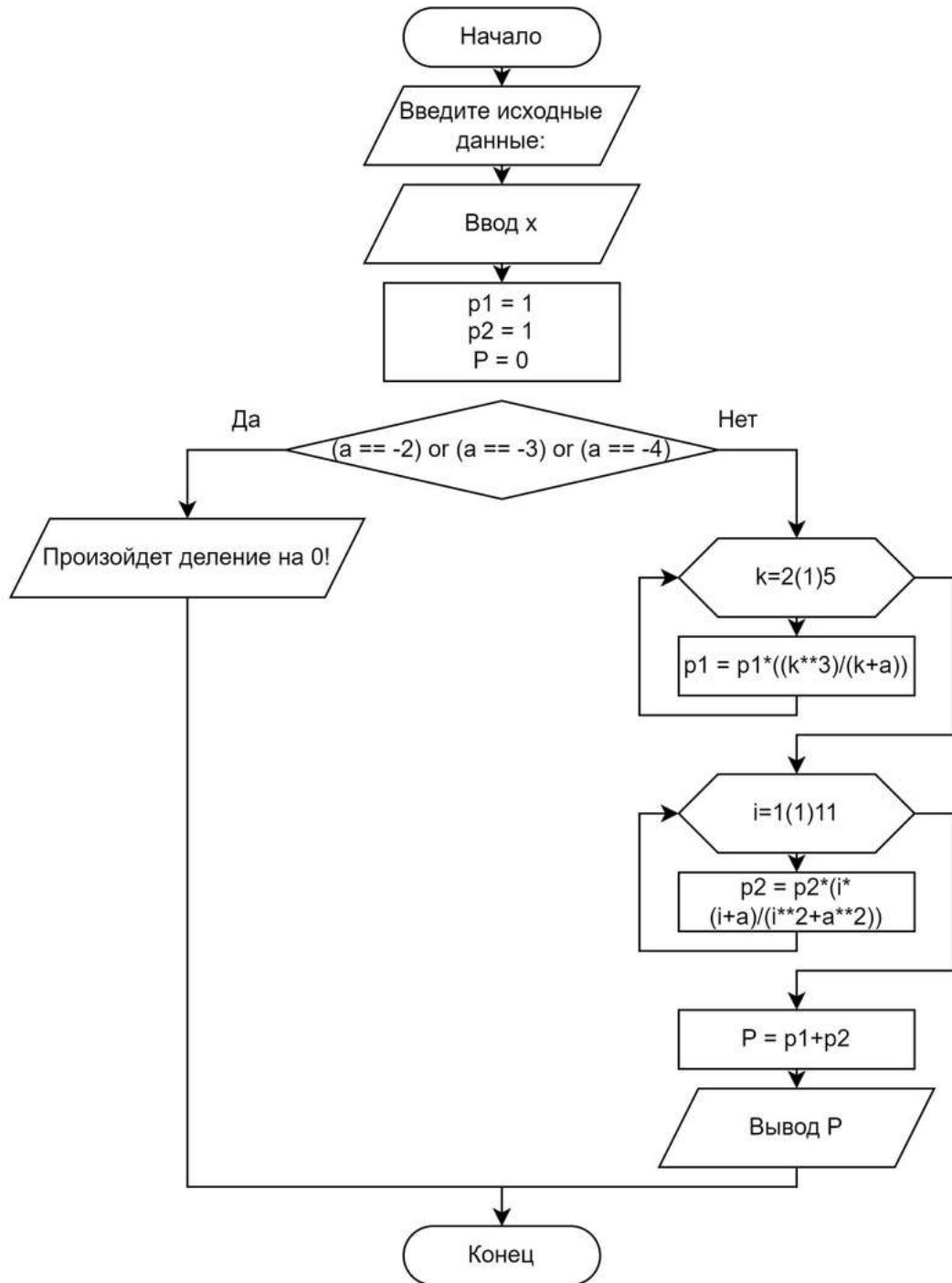
6 Работа № 6 Цикл for

6.1. Задание

Во всех вариантах заданий подготовьте ноутбук с отчетом к лабораторной работе. Вычислить произведение:

$$P = \prod_{i=1}^{10} \frac{i(i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a}$$

6.2. Блок-схема



6.3. Листинг программы

Ввод [14]:

```
import math

print("Введите исходные данные:")
print("a = ", end='')
a = float(input())

p1 = 1
p2 = 1
P = 0

if (a == -2) or (a == -3) or (a == -4):
    print("Произойдет деление на 0!")
else:
    for k in range(2,5):
        p1 = p1*((k**3)/(k+a))
    for i in range(1,11):
        p2 = p2*(i*(i+a)/(i**2+a**2))
    P = p1+p2
    print('P = ', P)
```

Введите исходные данные:

```
a = 3
P = 66.84160757289789
```

6.4. Результат работы программы в таблице

№	a	P
1	3	66.84160
2	5	27.56992
3	-5	-2304.0
4	-3	Произойдет деление на 0!

6.5. Контрольный расчет в MathCAD

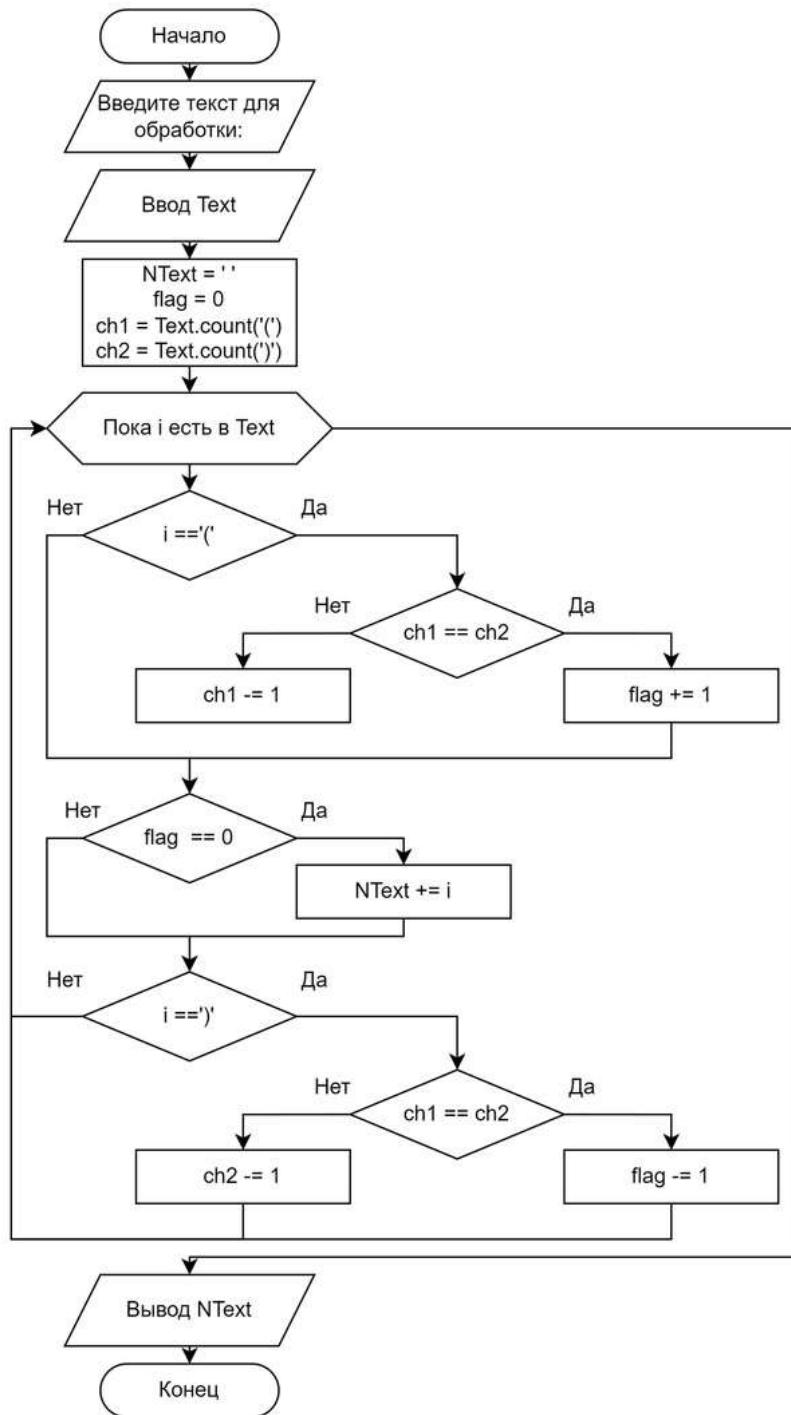
$$a:=3$$
$$P:=\prod_{i=1}^{10} \frac{i \cdot (i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} = 66.842$$
$$a:=5$$
$$P:=\prod_{i=1}^{10} \frac{i \cdot (i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} = 27.57$$
$$a:=-5$$
$$P:=\prod_{i=1}^{10} \frac{i \cdot (i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} = -2.304 \cdot 10^3$$
$$a:=-3$$
$$P:=\prod_{i=1}^{10} \frac{i \cdot (i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} = ?$$

7 Работа № 7 Строки

7.1. Задание

задача 2. Питончик Пончик, он же Питончик, считает, что текст, записанный в скобках, можно не читать (как вот тут, например). Помогите ленивому Питончику укоротить время чтения: напишите программу, которая в тексте будет удалять все, что находится внутри скобок и сами эти скобки, возвращая очищенный текст (скобки могут быть вложенными). При разработке алгоритма решения нужно учесть, что самая последняя открывающая скобка обязательно иметь после себя закрывающую.

7.2. Блок-схема



7.3. Листинг программы

Ввод [2]:

```
print("Введите текст для обработки:")
Text = input()
NText = ''
flag = 0
ch1 = Text.count('(')
ch2 = Text.count(')')

for i in Text:

    if i == '(':
        if ch1 == ch2:
            flag += 1
        else:
            ch1 -= 1

    if flag == 0:
        NText += i

    if i == ')':
        if ch1 == ch2:
            flag -= 1
        else:
            ch2 -= 1

print ( NText )
```

Введите текст для обработки:

Падал (Лишнее) прошлогодний (Лишнее)снег (Лишнее)
Падал прошлогодний снег

7.4. Результат работы программы в таблице

№	Вход	Результат
1	Падал (Лишнее)Лишнее) прошлогодний снег (Лишнее)	Падал прошлогодний снег
2	(Лишнее(Лишнее)Лишнее) Падал прошлогодний (Лишнее(Лишнее(Лишнее)))снег	Падал прошлогодний снег
3	Падал (Лишнее) прошлогодний (Лишнее)снег (Лишнее)	Падал прошлогодний снег

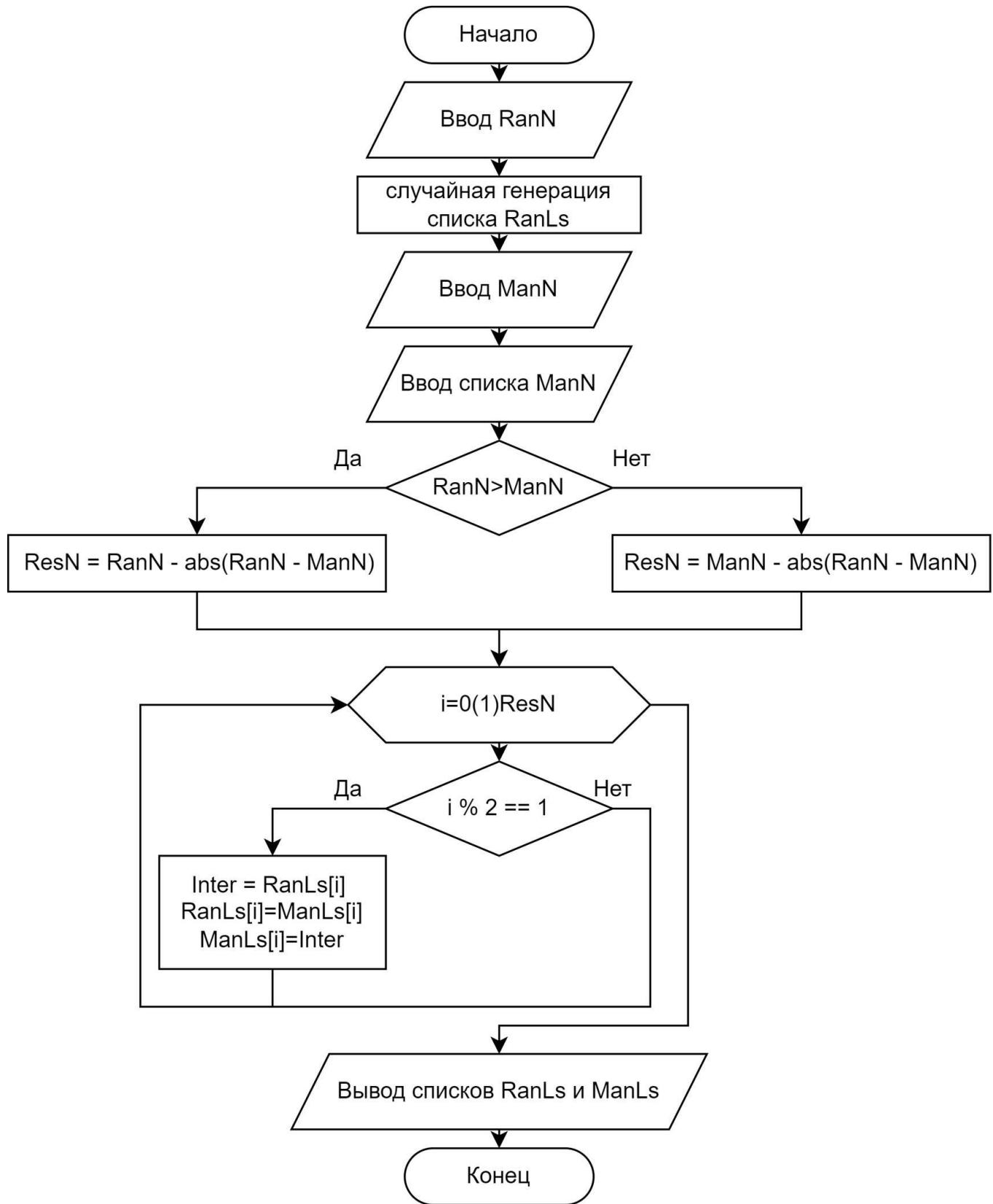
8 Работа № 8 Списки

8.1. Задание

Во всех задачах необходимо обработать два списка: А - сгенерированный случайным образом список целых чисел, В - введенный с клавиатуры список действительных чисел.

Поменять местами элементы списков, стоящие на четных позициях(т.е. должны поменяться 2, 4, 6 и т.д элементы местами). Списки могут быть разного размера. Если какой-то список длиннее, то его "хвост" останется неизменным.

8.2. Блок-схема



8.3. Листинг программы

Ввод [20]:

```
from random import randint
RanN = int(input('Число элементов генерируемого списка = '))
RanLs = [randint(-100, 100) for _ in range(RanN)]
print('Сгенерированный список: \n', RanLs)

ManN = int(input('Число элементов списка с ручным вводом = '))
ManLs = []
for i in range(ManN):
    ManLs.append(int(input()))
print('Список с ручным вводом: \n', ManLs)

if RanN>ManN:
    ResN = RanN - abs(RanN - ManN)
else:
    ResN = ManN - abs(RanN - ManN)

for i in range(0, ResN):
    if i % 2 == 1:
        Inter = RanLs[i]
        RanLs[i]=ManLs[i]
        ManLs[i]=Inter

print('\n')
print('Сгенерированный список: \n', RanLs)
print('\n')
print('Список с ручным вводом: \n', ManLs)
```

Число элементов генерируемого списка = 3

Сгенерированный список:

[-4, 64, -71]

Число элементов списка с ручным вводом = 5

12

42

531

5436

24

Список с ручным вводом:

[12, 42, 531, 5436, 24]

Сгенерированный список:

[-4, 42, -71]

Список с ручным вводом:

[12, 64, 531, 5436, 24]

8.4. Результат работы программы в таблице

№	RanList	ManList	Result - RanList	Result - ManList
1	[-38, 40, -46, 6, 55]	[123, 432, 231]	[-38, 432, -46, 6, 55]	[123, 40, 231]
2	[32, -14, -52, -29, 55]	[15, 78, 42, 179, 321]	[32, 78, -52, 179, 55]	[15, -14, 42, -29, 321]
3	[-4, 64, -71]	[12, 42, 531, 5436, 24]	[-4, 42, -71]	[12, 64, 531, 5436, 24]

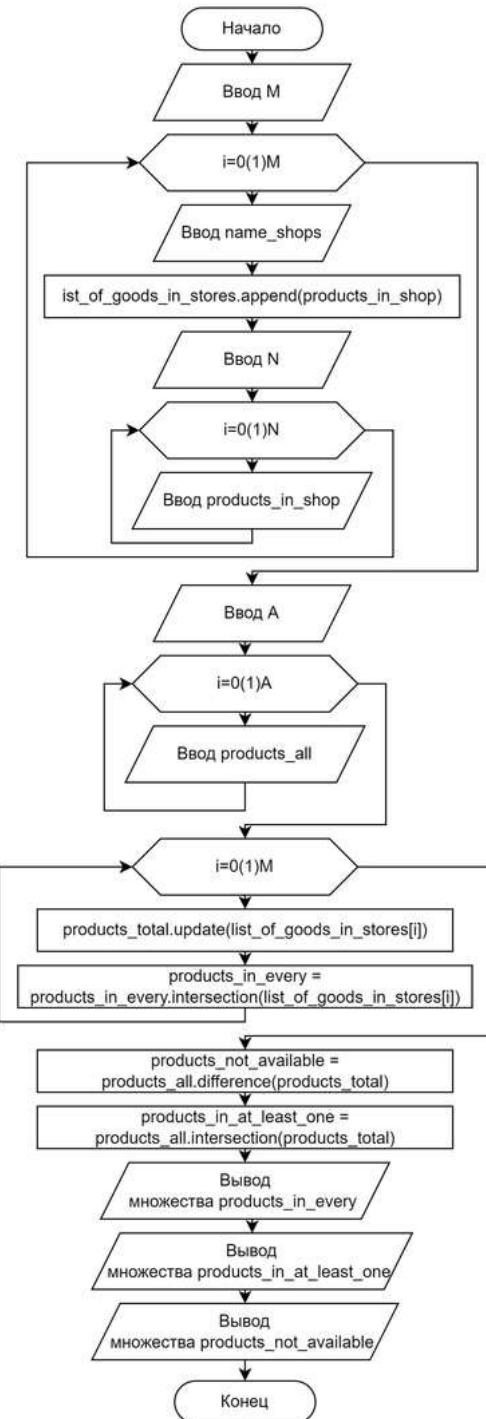
9 Работа № 9 Множества

9.1. Задание

В программном решении вашей задачи требуется использовать конструкцию множества.

Задан некоторорый набор товаров. Определить для каждого из товаров, какие из них имеются в каждом из магазинов, какие товары есть хотя бы в одном магазине, и каких товаров нет ни в одном магазине.

9.2. Блок-схема



9.3. Листинг программы

Ввод [48]:

```
M = int(input('Сколько у нас магазинов?: '))
shops = set()

products_in_every = set()
products_in_at_least_one = set()
products_not_available = set()
products_total = set()
list_of_goods_in_stores = []

for i in range(M):
    name_shops = input(f'\n Название {i+1} магазина: ')
    products_in_shop = set()
    list_of_goods_in_stores.append(products_in_shop)
    N = int(input(f'Сколько у нас товаров в магазине {i+1}) - {name_shops}?: '))
    for j in range(N):
        products_in_shop.add(input(f' Название {j+1} товара: '))

A = int(input('\n Сколько у нас всего товаров?: '))
products_all = set()
for i in range(A):
    products_all.add(input(f' Название {i+1} товара: '))
    products_in_every = products_all

for i in range(M):
    products_total.update(list_of_goods_in_stores[i])
    products_in_every = products_in_every.intersection(list_of_goods_in_stores[i])

products_not_available = products_all.difference(products_total)
products_in_at_least_one = products_all.intersection(products_total)

print ('\n Имеются в каждом из', M , 'магазинов: ')
for product in products_in_every:
    print (' ', product)

print ('\n Есть хотя бы в одном из', M , 'магазинов: ')
for product in products_in_at_least_one:
    print (' ', product)

print ('\n Товаров нет ни в одном из', M , 'магазинов: ')
for product in products_not_available:
    print (' ', product)
```

Сколько у нас магазинов?: 2

Название 1 магазина: А

Сколько у нас товаров в магазине 1) - А?: 2

Название 1 товара: А

Название 2 товара: В

Название 2 магазина: Н

Сколько у нас товаров в магазине 2) - Н?: 2

Название 1 товара: А

Название 2 товара: С

Сколько у нас всего товаров?: 4

Название 1 товара: А

Название 2 товара: В

Название 3 товара: С

Название 4 товара: Д

Имеются в каждом из 2 магазинов:

А

Есть хотя бы в одном из 2 магазинов:

С

В

А

Товаров нет ни в одном из 2 магазинов:

Д

9.4. Результат работы программы

```
Сколько у нас магазинов?: 2
Название 1 магазина: Блинофф
Сколько у нас товаров в магазине 1) - Блинофф?: 2
Название 1 товара: Блинчики с творогом
Название 2 товара: Блинчики с мясом

Название 2 магазина: Эмиль Бонте
Сколько у нас товаров в магазине 2) - Эмиль Бонте?: 2
Название 1 товара: Блинчики с творогом
Название 2 товара: Блинчики с курицей и грибами

Сколько у нас всего товаров?: 4
Название 1 товара: Блинчики с мясом
Название 2 товара: Блинчики с творогом
Название 3 товара: Блинчики с курицей и грибами
Название 4 товара: Блинчики с брусничным джемом

Имеются в каждом из 2 магазинов:
Блинчики с творогом

Есть хотя бы в одном из 2 магазинов:
Блинчики с курицей и грибами
Блинчики с мясом
Блинчики с творогом

Товаров нет ни в одном из 2 магазинов:
Блинчики с брусничным джемом
```

10 Работа № 10 Словари

10.1. Задание

В тех вариантах, где моделируется работа с базой данных, необходимо программно заполнить словарь информацией (чтобы меньше вводить данных при тестировании). Предусмотрите систему меню с пунктами:

- просмотр всех записей в БД;
- добавление N записей;
- удаление записи по ключу;
- поиск необходимой информации;
- завершение работы с БД;

Отображаемая информация должна быть в таблице.

Задача 2. Дан русский текст. Текст может состоять из любых символов, вам необходимо транслитерировать (то есть заменить все русские буквы на английские) только русские буквы, а остальные оставить на месте. Строчные буквы заменяются на строчные, заглавные заменяются на заглавные. Если заглавная буква превращается при транслитерации в несколько букв, то заглавной должна оставаться только первая из них (например, «Ц» - «Cz»). Правила трансляции для больших букв А-А, Б-В, Г-Г, Д-Д, Е-Е, Ё-Ё, Ж-Ж, З-З, И-И, Й-Й, К-К, Л-Л, М-М, Н-Н, О-О, П-Р, Р-Р, С-С, Т-Т, У-У, Ф-Ф, Х-Х, Ц-Ц, Ч-Ч, Ш-Ш, Щ-Щ, Ъ-\", Ы-\"Y, Ь-\", Э-Е', Ю-Юи, Я-Яа . Например, это Юность моя -> e'то Yunost' myoa . Примечание. Специальные правила замены предусмотренные стандартом: Буква "ц" заменяется на букву "с" если за ней следуют буквы е, и, у, я. Во всех остальных случаях буква "ц" заменяется на последовательность "cz".

10.2. Листинг программы

Ввод [64]:

```
unit_dict = {' ': ' ', '\n': '\n', 'А': 'A', 'Б': 'B', 'В': 'V', 'Г': 'G', 'Д': 'D', 'Е': 'E', 'Ё': 'Yo', 'Ж': 'Zh', 'З': 'Z', 'И': 'I',
flag_l = {'e', 'i', 'y', 'j'}

input_str = "Ци дзень Пинь цй \n Я люблю питон"
res, flag = " ", 0

for char in input_str:
    if char == 'ц':
        res += 'c'
        flag = 1
    if (char in flag_l) and (flag == 1):
        res += 'z'
        flag = 2
    if (char in unit_dict) and (flag != 2) and (flag != 1):
        res += unit_dict[char]
    else:
        flag = 0
print(res)
```

```
Czi Dzen` Pin` cj
Ya lyublyu piton
```

10.3. Результат работы программы

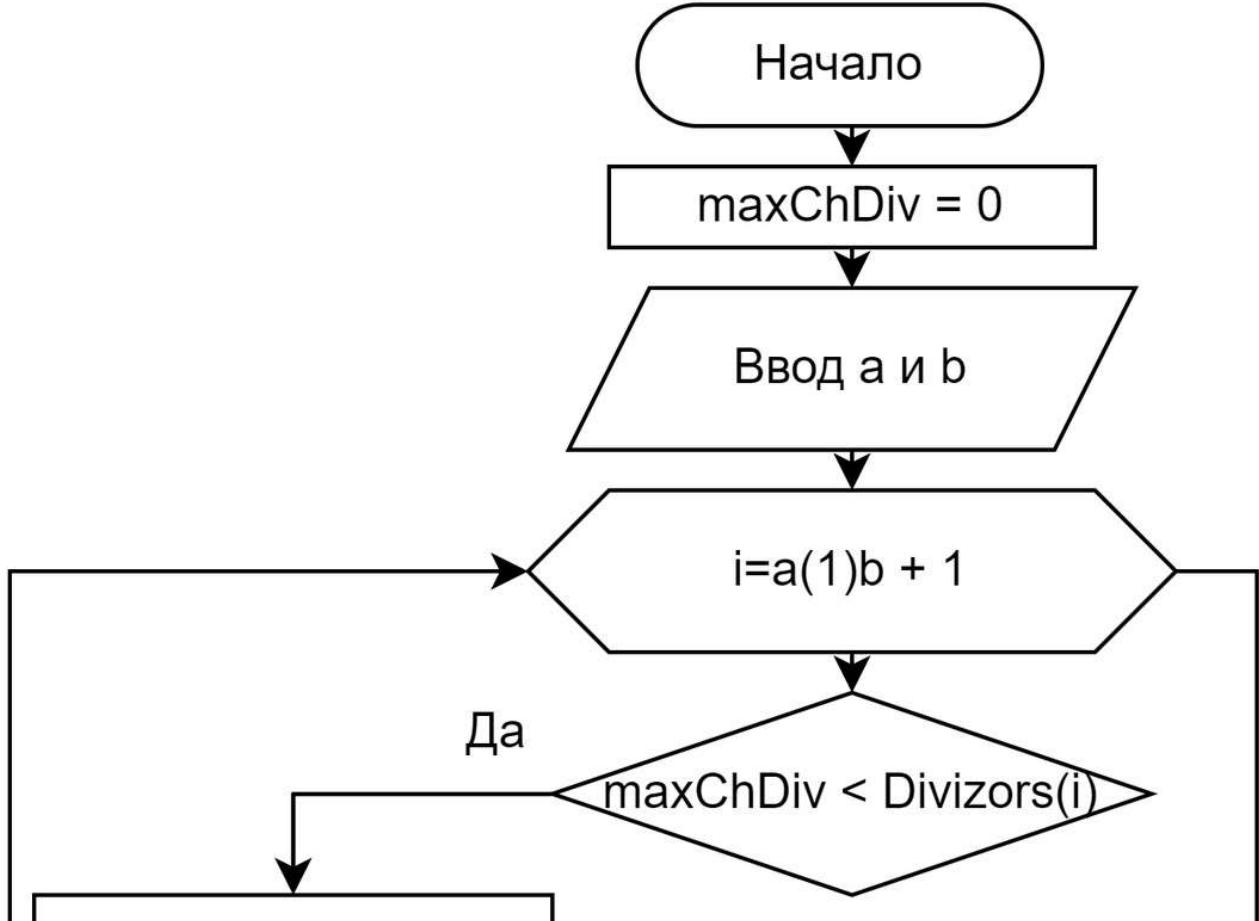
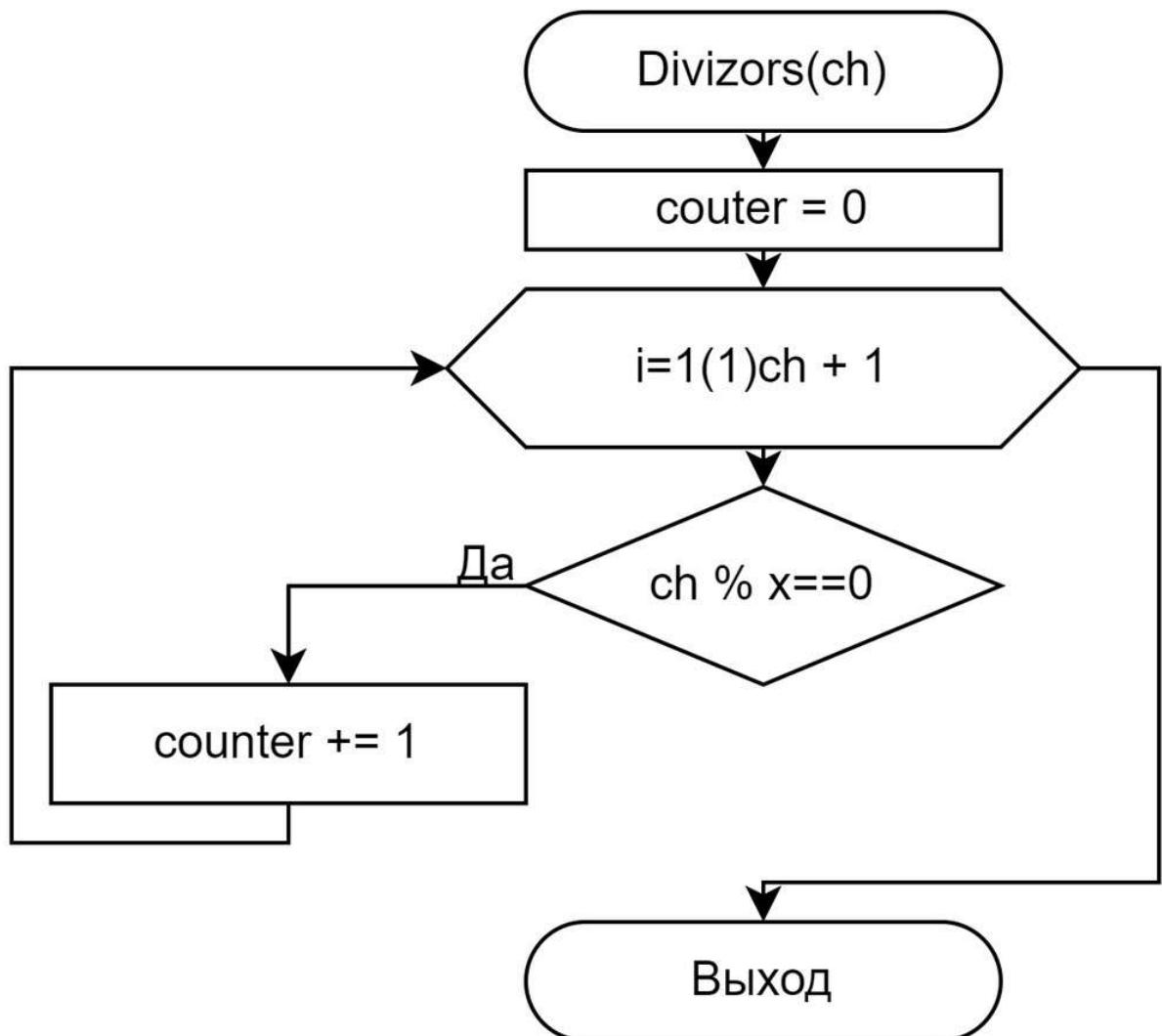
11.1 Функции (часть 1)

11.1. Задание

При выполнении работы необходимо, воспользовавшись принципами исходящего программирования (проектированием приложений "сверху-вниз", выделить необходимые для решения задачи функции.

Составить программу для нахождения чисел их интервала $[a, b]$, имеющих наибольшее количество делителей.

11.2. Блок - схема



11.3. Листинг программы – *Divizors*

Ввод [35]:

```
def Divizors(ch):
    counter=0
    for x in range(1,ch+1):
        if ch % x==0:
            counter += 1
    return counter

maxChDiv = 0
a = int(input("Введите число, соответствующее началу интервала (A)"))
b = int(input("Введите число, соответствующее концу интервала (B)"))
for i in range (a, b+1):
    if (maxChDiv < Divizors(i)):
        maxChDiv = Divizors(i)
        maxCh = i

print(maxChDiv)
print(maxCh)
```

Введите число, соответствующее началу интервала (A)1
Введите число, соответствующее концу интервала (B)100
12
60

Ввод [31]:

```
ch=int(input("please eenter a number "))
a=[]
for x in range(1,ch+1):
    if ch%x==0:
        a.append(x)
print(a)

please eenter a number 60
[1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60]
```

11.4. Результат работы программы в таблице

Nº	A	B	MaxChDiv	MaxCh
1	1	100	12	60
2	100	200	18	180
3	200	300	20	240

12 Текстовые файлы

12.1. Задание

Предусмотреть два режима работы: обработка тестового файла и сгенерированного, причем уже в этой программе используем функции.

Сформировать текстовый файл Data.txt из заглавных букв X, Y, Z латинского алфавита и десятичных цифр. Найти номер строки, в которой находится самое большое число, и определить в этой строке максимальное количество подряд идущих букв Y. Если строк с максимальным числом несколько, то выбрать ту, которая находится в файле раньше.

12.2. Листинг программы

Ввод [109]:

```
from random import randrange

def ranGen():
    ranChorStr = randrange(0, 1+1)
    if ranChorStr == 1:
        return chr(randrange(ord('X'), ord('Z')+1))
    else:
        return str(randrange(0, 9+1))

choice = int(input("Вы хотите брать данные из готового файла(1)"
                   "или из рандомно сгенерированного(2)? Введите номер выбора: "))
if choice == 1:
    n = int(input('Введите количество строк '))
    f = open('DataRandom.txt', 'w')
    for _ in range(n):
        cnt = randrange(10, 100+1)
        for _ in range(cnt):
            f.write(ranGen())
        f.write('\n')
    f.close()
    f = open('DataRandom.txt', 'r')
else:
    f = open('DataReady.txt', 'r')

maxLine, line, Ch, maxCh, counterY, maxY = 0, 0, 0, 0, 0, 0
for s in f:
    line += 1
    for c in s:
        if c.isdigit():
            Ch = Ch*10 + int(c)
            if maxCh < Ch:
                maxCh = Ch
                maxLine = line
                YinMaxL = s
        else:
            Ch = 0

    for c in YinMaxL:
        if c.isalpha() and c == 'Y':
            counterY+=1
            if maxY < counterY:
                maxY = counterY
        else:
            counterY = 0

print("Максимальное число:", maxCh, "Строка:", maxLine, "Максимальное количество Y подряд:", maxY)
f.close()
```

Вы хотите брать данные из готового файла(1)или из рандомно сгенерированного(2)? Введите номер выбора: 2
Максимальное число: 999 Стока: 1 Максимальное количество Y подряд: 5

12.3 Результат работы программы

```
Вы хотите брать данные из готового файла(1)или из рандомно сгенерированного(2)? Введите номер выбор
а: 1
Введите количество строк 5
Максимальное число: 9083537 Стока: 1 Максимальное количество Y подряд: 2
```

X2X8X0Z64Z91Z86XX5XXZ27Y0YYXY9YXZZ988YYZ7XX710X95ZY08ZYZZ7Y01Z79Z7Y7ZY57X9083537ZZ8XX1Y8Z5
 88YX99Y4X41X83XXY64X61X73ZY2XYXXY21X36ZX6
 7X77X1ZYZ6XZXYYY9Z74X9Y1X6594055X27ZZ69906XZXXZ5
 YX2X6XX3XYXX
 Z1YZ0XY5Z98ZZ51ZZYYY91YZZX5XXZY9ZYYY74XY1Y54X40Y9XXX5XXX1Y1X8ZYX2Z199YXZY1YY0X762X4Z09XXZZ28XX

Вы ходите брать данные из готового файла(1)или из рандомно сгенерированного(2)? Введите номер выбора:
 а: 2
 Максимальное число: 999 Стока: 1 Максимальное количество Y подряд: 5

999YYYYYYYXXXXZZZZZZYYYY
 789YYYYYYYXXXXZZZZZZYYYY
 999YYYYYYYXXXXZZZZZZYYYY

13 Библиотека NumPy

13.1. Задание

В программном решении вашей задачи требуется использовать функции библиотеки NumPy для обработки двумерного массива (введенного или тестового, заранее введенного). при выполнении заданий НЕ использовать циклы для перебора элементов.

Задача 2. Выполните над массивом предложенные действия:

- 1). найдите среднее значение из всех элементов массива;
- 2). определить в массиве максимальный элемент каждого столбца;
- 3). отсортировать матрицу k-ому столбцу;
- 4). определить в массиве количество элементов, принадлежащих отрезку [-5;5];
- 5). обнулите крайние столбцы в массиве.

13.2. Листинг программы

```

import numpy as np

def f(i, j):
    return j >= (1 - 2)

l = int(input('Введите длину массива: '))
h = int(input('Введите высоту массива: '))

choise = int(input('Выберете вариант использования ( 1 - введенный, 2 - рандомно сгенерированный): '))
if (choise == 1):
    a = np.array([int(input()) for i in range(h * l)])
    a = a.reshape(h, l)
else:
    a = np.random.randint(-100, 100, (l, h))

print(a)
# 1 задание - найдите среднее значение из всех элементов массива;
print('Среднее значение: ', np.average(a))
# 2 задание - определить в массиве максимальный элемент каждого столбца;
print('Максимальное значение по столбцам: ', a.max(axis=0))
# 3 задание - отсортировать матрицу k-ому столбцу;
k = int(input('Введите столбец сортировки массива: '))
print(a[:, k - 1].argsort())
# 4 задание - определить в массиве количество элементов, принадлежащих отрезку [-5;5];
print('Количество элементов, принадлежащих отрезку: ', len(a[(-5 <= a) & (a <= 5)]))
# 5 задание - обнулите крайние столбцы в массиве.
mask = np.array(np.fromfunction(f, a.shape), dtype=bool)
a[mask] = 0
print('Матрица после обнуления: ')
print(a)

```

13.3 Результат работы программы

```
Введите длину массива: 5
Введите высоту массива: 5
[[ 67  71 -65   1 -31]
 [ 83  99  35  80   4]
 [ 44 -78 -85  72   8]
 [-73 -14 -43 -89 -33]
 [-13  81  52 -59 -25]]
Среднее значение: 3.56
Максимальное значение по столбцам: [83 99 52 80 8]
Введите столбец сортировки массива: 1
[[-73 -14 -43 -89 -33]
 [-13  81  52 -59 -25]
 [ 44 -78 -85  72   8]
 [ 67  71 -65   1 -31]
 [ 83  99  35  80   4]]
Количество элементов, принадлежащих отрезку: 2
Матрица после обнуления:
[[ 67  71 -65   0   0]
 [ 83  99  35   0   0]
 [ 44 -78 -85   0   0]
 [-73 -14 -43   0   0]
 [-13  81  52   0   0]]
```

13.4 Листинг задач с Яндекса

B

```
import math
import sys

for line in sys.stdin:
    list1 = list(map(int, line.split(" ")))
    print(math.gcd(*list1))
```

D

```
import math

list1 = list(map(float, input().split(" ")))

print(math.pow(math.prod(list1), 1 / len(list1)))
```

F

```
import numpy as np

def multiplication_matrix(n):
    npArr1 = np.array([i for i in range(1, n + 1)])
    npArrRes = np.array([i * npArr1 for i in range(1, n + 1)])

    return npArrRes
```

H

```
import numpy as np

def snake(m, n, direction='H'):

    if direction == 'H':
        npArr1 = np.arange(1, m * n + 1).reshape(n, m)
        result = np.zeros((n, m), dtype=np.int16)
        flag = 1
        for i in range(n):
            if flag == 1:
                result[i] = np.sort(npArr1[i])
            else:
                result[i] = np.sort(npArr1[i])[::-1]
            flag *= -1
        return result
    else:
        npArr1 = np.arange(1, m * n + 1).reshape(m, n)
        result = np.zeros((m, n), dtype=np.int16)
        flag = -1
        for i in range(m):
            if flag == 1:
                result[i] = np.sort(npArr1[i])
            else:
                result[i] = np.sort(npArr1[i])[::-1]
            flag *= -1
        return np.rot90(result)
```

J

```
import numpy as np

def stairs(v):
    return np.array([np.roll(v, i) for i in range(len(v))])
```

13.5 Правильность задач с Яндекс

Статистика



5 / 10

Вам даны задачи с разной сложностью

Вы можете отправлять неограниченное количество решений

Будет зачтено ваше лучшее решение задачи

Название	Баллы	Статус
A. Математика — круто, но это не точно	0 / 1	Не решена
B. Потоковый НОД	1 / 1	Полное решение
C. Есть варианты?	0 / 1	Не решена
D. Среднее не арифметическое	1 / 1	Полное решение
E. Шаг навстречу	0 / 1	Не решена
F. Матрица умножения	1 / 1	Полное решение
G. Шахматная подготовка	0 / 1	Не решена
H. Числовая змейка 3.0	1 / 1	Полное решение
I. Вращение	0 / 1	Не решена
J. Лесенка	1 / 1	Полное решение

14 Библиотека Mathplotlib

14.1.1 Задание

Задание 1. На трех различных графиках постройте функции заданные:

- а) в декартовых координатах;
- б) в полярных координатах
- в) параметрическими координатами

В случае построения графика в декартовых координатах постройте на одном чертеже графики сразу нескольких функций (достаточно двух). Для построенных графиков настройте различные форматы линии, маркеров, осей (дополнительных и вспомогательных), отображение легенд, заголовка графика, подписей осей. На одном из графиков отобразите аннотации. Еще на одном выведите текст с аналитическим заданием построенной функции.

14.2.1 Листинг и результат программы

Ввод [24]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

fig = plt.figure()
fig.set(facecolor='white')

ax = fig.add_axes([0, 0, 1, 1])
ax = plt.gca()
x = np.arange(-7.5, 12, 0.25)
y = x**3+(5/2)*x**2-2*x+3/2

ax.spines['right'].set_color('gray')
ax.spines['top'].set_color('gray')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))

ax.plot(x, y,
        color = '#FF7F50',
        linewidth = 3,
        marker = '^',
        markeredgecolor = 'black',
        markeredgewidth = 1,
        markersize = 10,
        markerfacecolor = 'white',
        label = r'$y = x^3 - \frac{5}{2}x^2 - 3x + \frac{3}{2}$')

ax.scatter(x, (x**2)*np.sin(2*x),
           color = 'salmon',
           marker = 'd',
           label = r'$y = (x^2) * \sin(2x)$')

ax.set(facecolor = 'white',
       xlim=[-7.5, 12],
       ylim=[-40, 30],
       title = 'Пример графиков в декартовой системе координат',
       xlabel = 'Ось абсцис (XAxis)',
       ylabel = 'Ось ординат (YAxis)')

ax.annotate(r"$График_1 - y = x^3 - \frac{5}{2}x^2 - 3x + \frac{3}{2}$", xy = (-4.25,-20), xytext=(-3,-27),
           arrowprops=dict(facecolor="black", connectionstyle="angle3",arrowstyle="->"))

ax.annotate(r"$График_2 - y = (x^2) * \sin(2x)$", xy = (4.5,10), xytext=(5,13.5),
           arrowprops=dict(connectionstyle="angle3",arrowstyle="-"))

ax.legend(loc = 'lower left')

ax.minorticks_on()

ax.grid(which='major', color = 'aqua', linewidth = 2)
ax.grid(which='minor', color = 'peachpuff', linewidth = 1)

plt.show()
```



Ввод [23]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

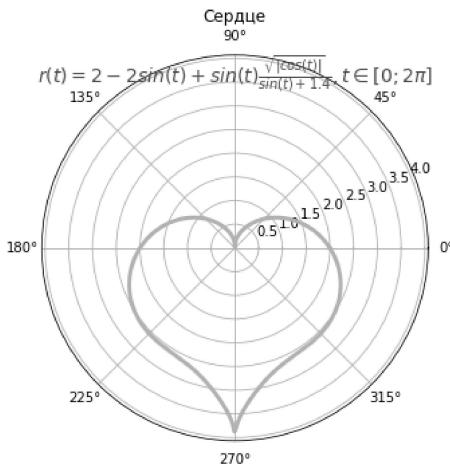
x = np.arange(-10,10.01,0.01)
phi = np.arange(0,2*np.pi,0.01)

fig = plt.figure()

ax = fig.add_axes([0,0,1,1], polar = True)
ax.plot(phi,
         2-2*np.sin(phi) + np.sin(phi) * ((abs(np.cos(phi)))**(0.5))/(np.sin(phi)+1.4),
         color = 'aqua',
         linewidth = 3)

ax.set_title('Сердце')

plt.text(0.5,0.95,"$ r(t) = 2 - 2\sin(t) + \sin(t)\frac{\sqrt{|\cos(t)|}}{\sin(t)+1.4}, t \in [0; 2\pi] $", fontsize = 14, transform = ax.transA
plt.grid(True)
plt.show()
```



Ввод [22]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

phi = np.arange(0,2*np.pi,0.01)

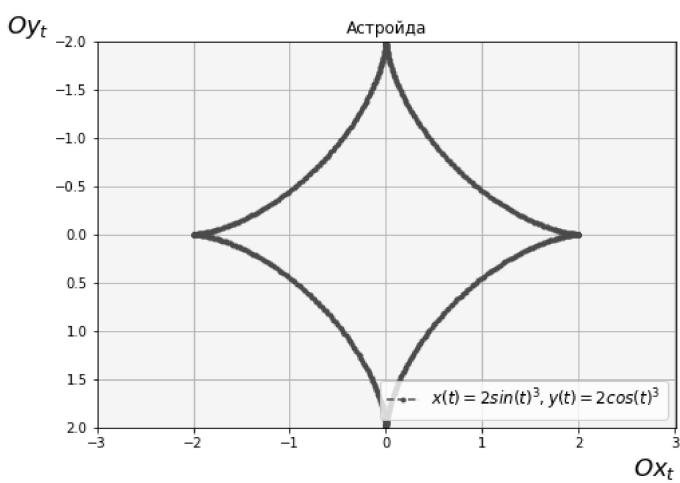
fig = plt.figure()

ax = fig.add_axes([0,0,1,1])
ax.plot(2*np.sin(phi)**3,
        2*np.cos(phi)**3,
        '.-g',
        label = r'$x(t) = 2\sin(t)^3$, $y(t) = 2 \cos(t)^3$')

ax.set(facecolor = 'oldlace',
       xlim=[-3, 3.01],
       ylim=[-2, 2],
       title = 'Астриода',)

ax.set_xlabel('$Ox_t$',loc = 'right', family = 'monospace',size = 18)
ax.set_ylabel('$Oy_t$',loc = 'top', family = 'monospace',size = 18, rotation = 'horizontal')

plt.legend(loc = 'lower right', fontsize = 12)
plt.grid(True)
plt.show()
```



14.1.2 Задание

Задание 2. Используя `subplots` или `subplots` нарисуйте графическое окно, состоящее из трех областей. В одной из них постройте несколько геометрических плоских фигур (например, `Circle`, `Polygon`, `Ellipse`, `Arc`, `Rectangle`), во второй продемонстрируйте эффект рисования графиков от руки, в третьей – отобразите пиксельную картинку с использованием функции `imshow`.

14.2.2 Листинг и результат программы

Ввод [21]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon, Arc
from matplotlib.gridspec import GridSpec

%matplotlib inline

fig = plt.figure(figsize=(14,6))

plt.subplot(1,3,1)

picture = plt.imread('Pictures/Grap.jpg')
plt.axis('off')
n,m = picture.shape[0], picture.shape[1]
plt.imshow(picture)

plt.xkcd(scale = 2, length = 100)
plt.subplot(1,3,2)

x = np.arange(-10,10.01,0.01)
plt.plot(x,(x**2)/5 + 3/2)
plt.title(r'Парабола')
plt.xlabel('$0x$',loc = 'left',size = 12)
plt.ylabel('$0y$',loc = 'bottom', size = 12, rotation = 'horizontal')
plt.rcdefaults()

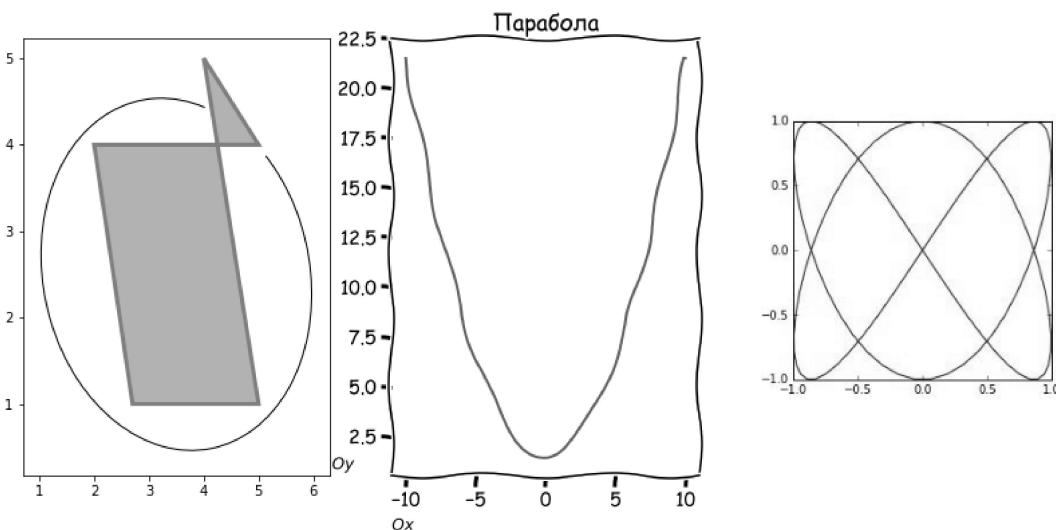
x = np.random.rand(100)
y = np.random.rand(100)

plt.subplot(1,3,1)

ax = plt.gca()
points = [[2.7,1], [2, 4], [5, 4], [4, 5], [5, 1]]
Polygon = Polygon((points), fill=200, facecolor = '#00FFFF', edgecolor='gray', linewidth=3, alpha = 1)
ax.add_patch(Polygon)

Arc = Arc(xy=(3.5, 2.5), width=4, height=5, angle=75, theta1=0, theta2=325)
ax.add_patch(Arc)
ax.autoscale()

plt.show()
```



14.1.3 Задание

Задание 3. Используя класс *GridSpec* продемонстрируйте вариант сложной компоновки графиков на одном чертеже. Выводимая в этом задании информация должна быть статистического характера (доход, рост, вес, оценки и прочее).

14.2.3 Листинг и результат программы

Ввод [20]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

%matplotlib inline

gs = plt.GridSpec(3,3)
fig = plt.figure(figsize=(12,16))

ax = fig.add_subplot(gs[0,0:2])
labels = np.array(['Россия', 'USA', 'Китай', 'Корея', 'Япония', 'Канада'])
values = np.array([6,3,2,4,5,7])
man = np.array([2,1,0,1,2, 3])
woman = values - man

ax.set_title("Забытые на чемпионате шайбы")
ax.bar(labels,
       values,
       width = [0.8,0.6,0.2,0.8,1,0.2],
       bottom = 100,
       color = ['m', '#00FFFF', 'gray','black','peachpuff','aqua'],
       alpha = 0.5,
       edgecolor = 'k',
       linewidth = 3)

ax1 = fig.add_subplot(gs[:,2])
ax1.set_title("Шайб в день")

x = [f'Day{i+1}' for i in range(7)]
y = [11, 9,-2,-3,7,-3,1]
y_error = [4,3,1,1,2,1,1]

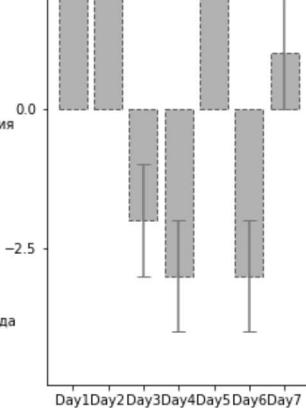
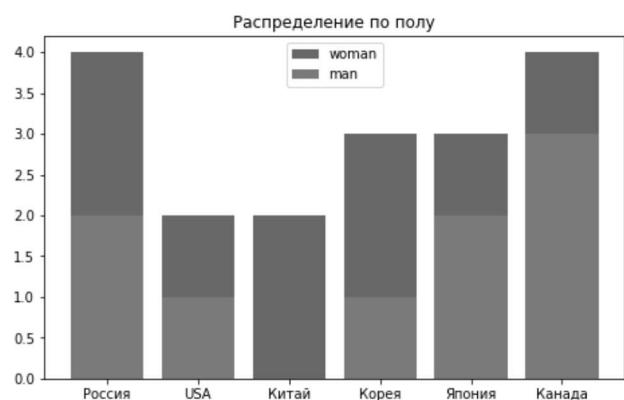
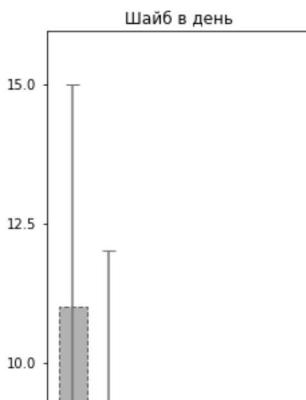
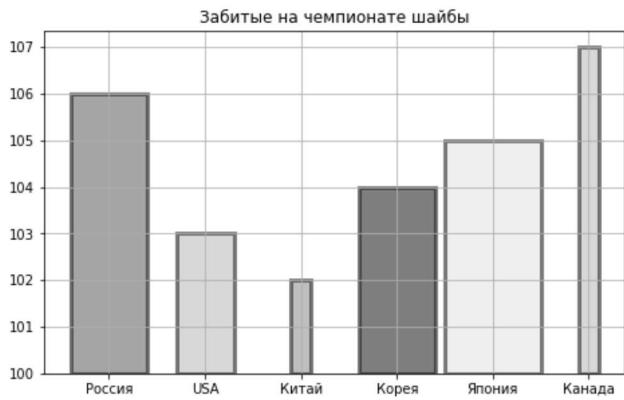
ax1.bar(x, y,
         width = 0.8,
         color = 'cyan',
         linestyle = '--',
         linewidth = 1,
         edgecolor = 'teal',
         yerr = y_error,
         ecolor = 'steelblue',
         capsize = 5)

ax2 = fig.add_subplot(gs[2,1])
ax2.set_title("Забытые на чемпионате шайбы")
exp = (0.2, 0.1, 0, 0, 0, 0.3)
ax2.pie(values, labels = labels, autopct = '%.2f', explode = exp, shadow = True)

ax3 = fig.add_subplot(gs[2,0])
ax3.set_title("Забытые на чемпионате шайбы")
ax3.bart(labels, values)

ax4 = fig.add_subplot(gs[1, :2])
ax4.set_title("Распределение по полу")
x = np.arange(len(labels))
ax4.bar(labels, woman, label = 'woman', color = '#FF1493')
ax4.bar(labels, man, label = 'man', color = '#1E90FF')
plt.xticks(range(len(labels)), labels)
ax4.legend()

ax.grid()
plt.show()
```



14.1.4 Задание

Задание 4. Отобразите трехмерный график, отформатированный в соответствии с выбранной по желанию цветовой картой, и линии уровня для этой поверхности.

14.2.4 Листинг и результат программы

Ввод [19]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

def make_data():
    x = np.linspace (-10, 10, 100)
    y = np.linspace (-10, 10, 100)

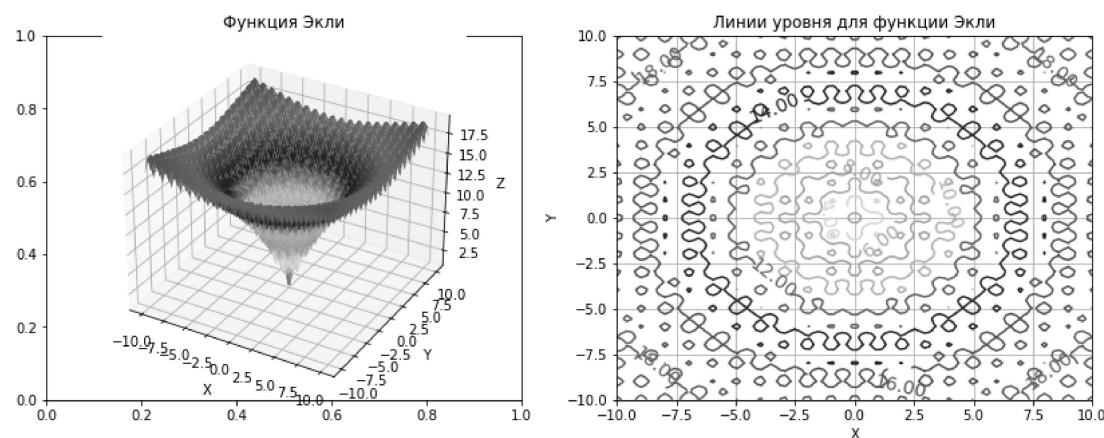
    X, Y = np.meshgrid(x,y)
    Z = -20.0 * np.exp(-0.2 * (0.5 * (X**2 + Y**2))**0.5) - np.exp(0.5 * (np.cos(2 * np.pi * X) + np.cos(2 * np.pi * Y))) + np.e + 20
    return X, Y, Z

if __name__ == '__main__':
    x, y, z = make_data()
    fig, ax = plt.subplots(1, 2, figsize = (14, 5))

    ax[0] = fig.add_subplot(1, 2, 1, projection = '3d')
    ax[0].plot_surface(x, y, z, rstride = 1, cstride = 1, linewidth = 0, cmap = mpl.cm.hsv)
    ax[0].set(xlabel = 'X', ylabel = 'Y', zlabel = 'Z', title = 'Функция Экли')

    contour = ax[1].contour(x, y, z, levels = 10, cmap = mpl.cm.hsv)
    ax[1].clabel(contour, fontsize = 14, fmt = '%.2f')
    ax[1].set(xlabel = 'X', ylabel = 'Y', title = 'Линии уровня для функции Экли')
    ax[1].grid()

plt.show()
```



14.1.5 Задание

Задание 5. С использованием библиотеки *Animation* смоделируйте «случайное блуждание» точки по поверхности (новое положение точки должно не сильно отличаться от предыдущего) на протяжении некоторого числа итераций.

14.2.5 Листинг и результат программы

Ввод [25]:

```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation

%matplotlib notebook

def rotate(v, alph):
    sin = np.sin(alph)
    cos = np.cos(alph)
    R = np.array([[cos, -sin], [sin, cos]])
    return np.dot(R, v)

def F(X, Y):
    return -20.0 * np.exp(-0.2 * (0.5 * (X ** 2 + Y ** 2)) ** 0.5) - np.exp(
        0.5 * (np.cos(2 * np.pi * X) + np.cos(2 * np.pi * Y))) + np.e + 20

def animate(i):
    global dt, angle, drspeed, speed, dx, dy

    angle += np.random.normal(np.pi / 6, np.pi / 6)
    drspeed = rotate(drspeed, angle)
    velocity = drspeed * speed

    dx += velocity[0] * dt
    dy += velocity[1] * dt
    z = F(dx, dy)

    dot.set_offsets([dx, dy])
    dot.set_3d_properties(z, 'z')
    return dot,

x = np.linspace(-10, 10, 100)
y = np.linspace(-10, 10, 100)
X, Y = np.meshgrid(x, y)
Z = F(X, Y)

fig = plt.figure()
ax = plt.axes(xlim=(-10, 10), ylim=(-10, 10), projection='3d')
ax.plot_surface(X, Y, Z, cmap=mpl.cm.hsv, edgecolor='none')

dx = 0
dy = 0
speed = 30
drspeed = np.array([1.0, 0.0])
angle = 0
dot = ax.scatter([dx], [dy], [F(dx, dy)], c='r', s=150)

frames = 300
interval = 1
dt = interval / 300.0

anim = animation.FuncAnimation(fig, animate, frames=frames, interval=interval, blit=True)

plt.show()
```

Figure 1

