

Практикум 4. Комплексные числа (2).

Цель работы – изучение логических переменных, операций отношения, логических операций, логического индексирования.

Продолжительность работы - 2 часа.

Оборудование, приборы, инструментарий – работа выполняется в компьютерном классе с использованием пакета Anaconda.

Порядок выполнения

1. Упражнения выполняются параллельно с изучением теоретического материала.
2. После выполнения каждого упражнения результаты заносятся в отчёт.
3. При выполнении упражнений в случае появления сообщения об ошибке рекомендуется сначала самостоятельно выяснить, чем оно вызвано, и исправить команду; если многократные попытки устранить ошибку не привели к успеху, то проконсультироваться с преподавателем.
4. Дома доделать упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые Вы не успели выполнить во время аудиторного занятия.
5. После выполнения упражнений выполнить дополнительные упражнения для самостоятельной работы и ответить на контрольные вопросы и (см. ниже).
6. Подготовить **отчёт**, в который включить упражнения из раздела «Краткие теоретические сведения и практические упражнения» и упражнения для самостоятельной работы. Отчёт представить либо в формате интерактивного питон-скрипта (ipynb), либо в виде документа Microsoft Word. Файл следует назвать по следующей схеме **pin_10_Ivanov_P_01_s1** (группа, фамилия, инициалы, номер лабораторной, семестр). Отчет должен содержать по каждому выполненному упражнению: № упражнения, текст упражнения; команды, скопированные из командного окна, с комментариями к ним и результаты их выполнения, включая построенные графики; тексты python функций; выводы.

Краткие теоретические сведения и практические упражнения

1. Логические переменные, операции отношения.

Логические переменные могут принимать одно из двух значений: True - истина и False - ложь.

Логический массив можно создать несколькими способами:

- 1) можно воспользоваться логическими константами True и False;
- 2) можно объявить 0 или 1 логическими константами задав их тип как bool.

Пример 1.

```
import numpy as np
x = np.array([True, True, False])
print(x)
y = np.array([1,1,0], dtype=bool)
print(y)
```

3) Значения логического класса также можно получить как результат операций отношения (сравнения), список которых приведен в таблице 1.

Операции реализуются с помощью встроенных функций (столбец 2 табл. 1), но записывают их обычно для вызова с помощью символов (столбец 1 табл. 1). Операндами x,y операций отношения могут быть только массивы одинаковой размерности; операция производится над элементами с одинаковыми индексами.

Таблица 1. Операции отношения		
Символ операции	Реализующая функция	Описание
==	np.equal(x,y)	Проверка скаляров на равенство. Поэлементная проверка массивов одинаковой размерности на равенство
!=	np.not_equal(x,y)	Поэлементная проверка массивов одинаковой размерности на неравенство
<	np.less(x,y)	Поэлементная проверка массивов одинаковой размерности на «меньше»
>	np.greater(x,y)	Поэлементная проверка массивов одинаковой размерности на «больше»
<=	np.less_equal(x,y)	Поэлементная проверка массивов одинаковой размерности на «меньше или равно»
>=	np.greater_equal(x,y)	Поэлементная проверка массивов одинаковой размерности на «больше или равно»

Функция np.array_equal(x,y) возвращает 1, если массивы x и y равны, и 0 в противном случае.

Пример 2.

```
import numpy as np
A = np.array([1, 2, 3])
B = np.array([-1, 2, 4])
print(A!=B)
print(A<B)
x = np.array([[2,3,4],[-1,-1,-1]])
y = np.array([[-2,3,5],[2,3,-7]])
print(x)
print(y)
print(x>=y) # поэлементное сравнение массивов
```

```
h = np.array([[2, 3, -4],[0, -2, 3]])
print(h)
print(h<0) # проверим какие из элементов отрицательные
```

```
x=np.array([1,2])
y=np.array([2,4])
z=np.array([2,4,3])
print(np.array_equal(x,y))
print(np.array_equal(y,z))
print(np.array_equal(y,z[:2]))
```

Упражнение 1. Найти все значения корней $\sqrt[8]{256}$. Используя операции отношения, выяснить:

- 1) какие из найденных корней изображаются на комплексной плоскости точками, лежащими в левой полуплоскости;
- 2) какие из найденных корней изображаются на комплексной плоскости точками, лежащими на действительной оси;
- 3) какие из найденных корней имеют главный аргумент, больший $\frac{\pi}{3}$.

Замечание: отношение равенства не всегда дает удовлетворительный результат из-за возможных ошибок округления. На замену этому отношению целесообразнее применять отношение **np.isclose(x,y)**, которое проверит что расстояние между x и y не превосходит малой величины, сравнимой с вычислительной погрешностью. Попробуйте использовать **np.isclose(x,y)** в заданиях упражнения 1.

2. Логические операции

Над переменными логического класса можно совершать логические операции, приведенные в таблице 2.

Таблица 2. Логические операции.		
Символ операции	Логическая операция	Описание
&	and	Логическое умножение скаляров (результат истина, только если x и y истинны).

		<p>Логическое покомпонентное умножение массивов одинаковой размерности.</p> <p>Логическое умножение массива на скаляр.</p>
	or	<p>Логическое сложение скаляров (результат ложь, только если x и y ложны).</p> <p>Логическое покомпонентное сложение массивов одинаковой размерности.</p> <p>Логическое сложение массива со скаляром.</p>
~	not	<p>Логическое отрицание скаляра или всех элементов массива</p>
^	xor	<p>Сложение по модулю 2 скаляров (результат ложен в двух случаях: если x и y оба ложны или x и y оба истинны).</p> <p>Покомпонентное сложение по модулю 2 массивов одинаковой размерности.</p> <p>Покомпонентное сложение по модулю 2 массива и скаляра.</p>

Пример 3.

```
import numpy as np
x = np.array([0,0,1,1], dtype=bool)
y = np.array([0,1,0,1], dtype=bool)
t = np.array([1<2, 5>3])
print(' x = '+str(x))
print(' y = '+str(y))
print('x and y = '+str(x&y))
print('x or y = '+str(x|y))
print('x xor y = '+str(x^y))
print(' not x = '+str(~x))
```

Пример 4.

```
import numpy as np
z1 = 28-4j
z2 = 20+10j
print('z1='+str(z1))
```

```
print('z2='+str(z2))
print('Both of |z_{1}|<29 and |z_{2}|<29 --> '+
str((np.abs(z1)<29) & (np.abs(z2)<29)) )
# проверим, верно ли, что модули обоих чисел меньше 29.
```

```
print('Any of |z_{1}|>32 or |z_{2}|>32 --> '+
str((np.abs(z1)>32) | (np.abs(z2)>32)) )
# верно ли, что модуль хотя бы одного из них больше 32.
```

Упражнение 2. Пусть $z_1 = 2 - 3i$, $z_2 = 5 + i$. Используя операции отношения и логические операции (использовать `np.isclose(x,y)` вместо равенства), выяснить, какие из приведенных ниже утверждений верные, а какие нет:

$$1) \begin{cases} \operatorname{Arg} z_1 > \pi / 3, \\ \operatorname{Arg} z_2 \leq 5\pi / 6; \end{cases} \quad 2) \begin{cases} \operatorname{Re}(3z_1 + 2z_2) < 8, \\ \operatorname{Im}(z_1 - z_2) \neq 2. \end{cases}$$

3. Логическое индексирование

Для того чтобы выделить из исходного массива данные нового массива или изменить группу элементов в исходном массиве, удовлетворяющих заданному условию, целесообразно использовать логическое индексирование. Для выполнения логического индексирования необходимо с помощью логических операций или операций отношения сформировать логический массив, содержащий одинаковое число элементов с исходным массивом. Сформированный логический массив состоит из нулей и единиц, где единичные элементы соответствуют выполнению условия, а нулевые значения – невыполнению этого условия. Если логическое индексирование используется справа от знака присваивания или является одиночной конструкцией, то создается новый массив из тех элементов исходного массива, индексы которых совпадают с индексами истинных элементов логического массива. В том случае если логическое индексирование используется слева от знака присваивания, то элементам исходного массива, индексы которых совпадают с индексами истинных элементов логического массива, присваивается значение, находящееся справа от знака присваивания.

Пример 5.

```
import numpy as np
A = np.array([1,2,3,4])
print('A '+str(A))
```

```

print('A>=2 '+str(A[A>=2]))
# выбираем элементы из массива A, которые >= 2
L = np.array([0,1,1,0], dtype=bool)
B = A[L]
# прочитаем элементы с помощью логических индексов L
A[L] = -5
# перезапишем на эти позиции в массиве A новые значения
print('B=A(L) '+str(B))
print('A(L)=-5 '+str(A))

```

Упражнение 3. Сформировать массив с элементами $(1+3i)^k$, где $k=1,2,\dots,8$.
Используя операцию логического индексирования, выбрать из массива числа, удовлетворяющие условиям:

а) $\operatorname{Re}(z) > 2$; б) $\operatorname{Im}(z) \leq \operatorname{Re}(z)$.

Упражнение 4. Сформировать массив с элементами $3 - ki$, где $k=1,2,\dots,9$.
Используя операцию логического индексирования, заменить в этом массиве числа, удовлетворяющие условию $4 < |z| \leq 7$, на 0.

Пример 6. На комплексной плоскости построить множество точек, координаты которых удовлетворяют условию $|z+i| \geq 1,5$.

В программе используем функцию **rand**, которая позволяет генерировать массив случайных чисел, равномерно распределенных на отрезке $[0;1]$. Функция `rand (n,m)` возвращает прямоугольную матрицу размерности $n \times m$ со случайными числами.

Построение множества точек, комплексной плоскости, удовлетворяющих неравенству :

```

import numpy as np
import numpy.random as rng
import matplotlib.pyplot as plt

x = 3 - 6*rng.rand(10000) # 10_000 точек x на отрезке [-3;3]
y = 3 - 6*rng.rand(10000) # 10_000 точек y на отрезке [-3;3]
z = x+ 1j*y
L = (np.abs(z+1j)>=1.5) # логический индекс для отбора значений
plt.plot(x[L],y[L],'.')
plt.axis('equal')
plt.grid(True)
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
plt.title('$| z+i | \geq 1.5 $')
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')

```

Упражнение 5. На комплексной плоскости построить множество точек, координаты которых удовлетворяют условиям : $1 \leq |z - 1 - i| \leq 2$.

Задания для самостоятельной работы

1. Выполнить упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые не успели сделать в аудитории.

1. Самостоятельно выполнить упражнения:

Упражнение С1. Найти все значения корней $\sqrt[8]{-256}$. Используя операции отношения, выяснить:

1) какие из найденных корней изображаются на комплексной плоскости точками, лежащими в левой полуплоскости;

2) какие из найденных корней изображаются на комплексной плоскости точками, лежащими на действительной оси;

3) какие из найденных корней имеют главный аргумент, больший $\frac{2\pi}{3}$.

Упражнение С2. Пусть $z_1 = 2 - 3i$, $z_2 = 5 + i$. Используя операции отношения и логические операции, выяснить, какие из приведенных ниже утверждений верные, а какие нет:

1) $5 \leq |z_1| < 6$;

4) только одно из чисел z_1 , z_2 по модулю меньше 4.

Упражнение С3. Сформировать массив с элементами $(1 - 3i)^k$, где $k = 1, 2, \dots, 8$. Используя операцию логического индексирования, выбрать из массива числа, удовлетворяющие условиям:

а) $\operatorname{Re}(z) > 2$; б) $\operatorname{Im}(z) \leq \operatorname{Re}(z)$.

Упражнение С4. Сформировать массив с элементами $2 - ki$, где $k = 1, 2, \dots, 9$. Используя операцию логического индексирования, заменить в этом массиве числа, удовлетворяющие условию $2 < |z| \leq 5$, на 0.

Упражнение С5. На комплексной плоскости построить множество точек, координаты которых удовлетворяют условиям (программы оформить, используя скрипты:

$$\begin{cases} |\operatorname{Arg} z| \geq \frac{\pi}{4}, \\ |z - 1| \leq 2. \end{cases}$$

2. Ответить на контрольные вопросы:

- 1) Каким образом можно задать логический массив?
- 2) Какие логические операции можно совершать над переменными логического класса?
- 3) Можно ли к числовым переменным применить операции логического класса?
- 4) Для чего применяется логическое индексирование?