```python
import sympy
import math
import mpmath
import cmath
```

Действия над комплексными числами.

Пример №1

```python
x = complex(1, 3)
y = complex(2, -1)
z = x * y
print(z)
g = complex(1, -2)
print(g)
t = complex(10, 0)
print(t)
h = t / g
print(h)
p = complex(-1, -1)
n = p * p
print(n)
C = z + h + n
print(C)
```

```
(5+5j)
(1-2j)
(10+0j)
(2+4j)
2j
(7+11j)
```

Пример №2

```python
x = complex(0, 1)
y = pow(x, 2)
print(y)
```

```
(-1+0j)
```

Пример №3

```python
x = complex(1, 3)
y = complex(2, -1)
z = x * y
print(z)
g = complex(1, -2)
print(g)
t = complex(10, 0)
print(t)
h = t / g
print(h)
```

```
p = complex(-1, -1)
n = p * p
print(n)
C = z + h + n
print(C)
```

```
(5+5j)
(1-2j)
(10+0j)
(2+4j)
2j
(7+11j)
```

Пример №4

```
x = sympy.Symbol("x")
print(sympy.solve(x ** 2 - 2 * x + 5))
```

```
[1 - 2*I, 1 + 2*I]
```

Пример №5

```
x = complex(1, -2)
i = complex(0, 1)
f = x ** 4 + (2 + i) / x - (-3 + 2 * i)
print(f)
```

```
(-4+23j)
```

Пример №6

```
print((1 + i)**8/(1+i)**6)
```

```
(-0+2j)
```

Пример №7

```
mpmath.mp.dps = 3
x = sympy.Symbol('x')
y = sympy.Symbol('y')
i = complex(0, 1)
fl = (2 + i) * x + y * (2 - i) - 6
f2 = (2 - i) * x + (3 - 2 * i) * y - 8
print(sympy.nsolve((fl, f2), (x, y), (-1, 1)))
```

```
Matrix([[-0.0588 - 0.765*I], [1.82 + 1.71*I]])
```

Пример №8

```
print(sympy.solve(x**2 - 3 + 4 * i))
```

```
[-2.0 + 1.0*I, 2.0 - 1.0*I]
```

Пример №9

```
x = sympy.Symbol("x")
i = complex(0, 1)
print(sympy.solve((2 + i) * x ** 2 - (5 - i) * x + 2 - 2 * i))
```

```
[0.8 - 0.4*I, 1.0 - 1.0*I]
```

Пример №10

```
x = sympy.Symbol("x")
i = complex(0, 1)
print(sympy.solve(x ** 2 - 3 + 4 * i))
```

```
[-2.0 + 1.0*I, 2.0 - 1.0*I]
```

Пример №11

```
i = complex(0, 1)
print(-(3 + 5 * i) ** 10 - 25 * (3 * i - 9) / 2 + 8 * i)
```

```
(28984688.5+34989570.5j)
```

Пример №12

```
z = complex(2, 2 * math.sqrt(3))
print(abs(z))
print(cmath.phase(z))
print(round(math.degrees(cmath.phase(z))))
```

```
3.9999999999999996
1.0471975511965976
60
```

Пример №13

```
zl = complex(-4, -9)
z2 = complex(1, -8)
print(complex(zl - sympy.conjugate(z2)) / complex(z2 +
sympy.conjugate(zl)))
i = complex(0, 1)
print((1 + 2 * i) * (-1 + 5 * i) / (6 - i))
z = complex(1, 2)
p = (1 + 3 * i) * z**2 + (-5 + 6 * i) * z + (2 - i)
print(p)
```

```
(-0.19999999999999982+5.6000000000000005j)
(-1.8648648648648647+0.1891891891891892j)
(-30-10j)
```

Примеры решения задач

Приведите число $z = -3 + 3\sqrt{3}i$ к тригонометрическому виду.

```
z = -3 + 3 * math.sqrt(3) * 1j
fi = round(math.degrees(cmath.phase(z)))
print(fi)
r = abs(z)
print(r)
```

```
120
6.0
```

Пусть $z_1 = -1 + 4i, z_2 = 1 + i$. Вычислите $ \frac{z_{1}}{z_{2}} + \frac{z_{2}}{z_{1}} $

```
z1 = -1 + 4j
z2 = 1 + 1j
print((z1 / z2.conjugate()) + z2 / z1)
```

```
(-2.323529411764706+1.2058823529411764j)
```

Вычислите значение многочлена $ P(z) = (-4+4i)z^2 + (-1 +3i)z + (-2-3i)$ в точке $ z = 1+3i $

```
z = 1 + 3j
p = (-4 + 4j) * (z * z) + (-1 + 3j) * z + (-2 - 3j)
print(p)
```

```
(-4-59j)
```

Найдите комплексные корни уравнения $ x^2 +8x + 20 = 0 $

```
x = sympy.Symbol("x")
print(sympy.solve(x ** 2 + 8 * x + 20))
```

```
[-4 - 2*I, -4 + 2*I]
```

Вычислите модуль и аргумент числа $ z = -6$

```
z = complex(-6, 0)
print(round(math.degrees(cmath.phase(z))), abs(z))
```

```
180 6.0
```

Приведите число $ z = 6 - 6i $ к тригонометрическому виду.

```
z = complex(6, 6)
print(round(math.degrees(cmath.phase(z))))
r = abs(z)
print(r)
c = r * (math.cos(-45) + 1j * math.sin(-45))
print(c)
```

```
45
8.48528137423857
(4.4575048871930445-7.220155828003307j)
```

Вычислите значение выражения $ \frac{(5+6i)(-1+6i)}{5-i} $ и представьте реузльтат в виде $ a + bi $

```
print(((5+6j)*(-1+6j))/(5-1j))
```

```
(-8.807692307692307+3.0384615384615383j)
```

Пусть $ z_{1} = -1 + 2i, z_{2} = -1 + 5i $. Вычислите $ \frac{\overline{z_{1}} - z_{2}}{z_{1}+\overline{z_{2}}} $

```
z1 = -1 + 2j
z2 = -1 + 5j
print((z1.conjugate() - z2) / (z1 + z2.conjugate()))
```

```
(1.6153846153846154+1.0769230769230769j)
```

Вычислите значение выражения $ \frac{(2-4i)(3-4i)}{2 + 5i} $ и представьте результат в виде $ a + bi $

```
print(((2-4j)*(3-4j))/(2+5j))
```

```
(-4.137931034482759+0.3448275862068966j)
```

Решение собственной задачи с использованием комплексных чисел

Необходимо найти корни кубического уравнения: x^3+3x^2-3x+1.

```
x = sympy.Symbol('x')
a = 1
b = 3
c = 3

eq = [x**3 + 3*x**2 - 3*x + 1]

roots = sympy.solve(eq, set=True)

print(f"The roots are: {roots}")
```

```
The roots are: ([x], {(-1 - 2**(1/3)/(-1/2 - sqrt(3)*I/2) -
2**(2/3)*(-1/2 - sqrt(3)*I/2),), (-2**(2/3) - 2**(1/3) - 1,), (-1 -
2**(2/3)*(-1/2 + sqrt(3)*I/2) - 2**(1/3)/(-1/2 + sqrt(3)*I/2),)})
```