Języki Skryptowe

dokumentacja projektu
ZADANIE 1 – LICZBY ZESPOLONE
konkurs Algorytmion 2017

Patryk Sroczyński, grupa 4H

1 grudnia 2021

Część I

Opis programu

Liczbą zespoloną nazywamy wyrażenie postaci $a + bi$ gdzie a i b są dowolnymi liczbami rzeczywistymi, a $i$jest jednostką urojoną, spełniającą warunek $i^2 = -1$ (jak widać, i nie może być liczbą rzeczywistą).
Napisz program, który po podaniu dwóch liczb zespolonych x i y oraz liczby naturalnej n, zwracał będzie sumę $x + y$, różnicę $x - y$, iloczyn $x * y$, iloraz x/y liczb zespolonych x i y oraz n-tą potęgę liczby zespolonej x Przykładowo, dla danych: $x = 3 + 2i, y = 2 - 3i, n = 3$, program zwróci kolejno: $5 - i$, $1 + 5i$, $12 - 5i$, i oraz $9 + 46i$
Wskazówka (do dzielenia liczb zespolonych): sprawdź, jaką ciekawą cechę ma iloczyn dwóch liczb zespolonych $a + bi$ oraz $a - bi$.
Sposób wprowadzania liczb zespolonych pozostawiamy w gestii rozwiązującego

Instrukcja obsługi

Aby wykonać program, należy uruchomić skrypt app.bat. Ponadto w folderze powinny znajdować się dwa skrypty pythonowe (app.py oraz web.py) oraz plik z danymi wejściowymi (in.txt) zawierający 5 liczb wpisanych jedna pod drugą.

Część II

Opis działania

Liczby zespolone są rozszerzeniem liczb rzeczywistych R. Zbiór liczb zespolonych oznaczamy symbolem C.
Liczbę urojoną zapisujemy za pomocą jednostki urojonej i. Liczbę i definiujemy następująco:

$$i^2 = -1$$

Liczbę zespoloną ogólnie możemy zapisać:

$$a + bi$$

Gdzie: a to część rzeczywista, b to część urojona, a i to jednostka urojona
Dodawanie, odejmowanie, mnożenie liczb zespolonych wykonujemy tak jak wyrażenia algebraiczne w dziedzinie R, jednocześnie pamiętając, że $i^2 = -1$. Natomiast przy dzieleniu skorzystamy z następującego wzoru:

$$\frac{z1}{z2} = \frac{(ac + bd)}{|z2|^2} + \frac{(bc + ad)}{|z2|^2} i$$

Podczas potęgowania skorzystamy ze wzoru de Moivre'a: $z^n = |z|^n(cosn\phi + isinn\phi)$, gdzie ϕ - argument liczby zespolonej, $|z|$ - moduł liczby zespolonej

Implementacja

Liczby zespolone oparte są na klasie Complex. Klasa ta przyjmuje w swoim konstruktorze dwa parametry. Jest to kolejno część rzeczywista liczby i część urojona. Do wykonywania działań arytmetycznych wykorzystane zostały tzw. magic methods. Są to metody które w zapisie zaczynają się oraz kończą podwójnym podkreśleniem (np. __pow__()). Metody te nie są wywoływane bezpośrednio przez użytkownika, natomiast zostają wykonane automatycznie w zależności od wykonywanych działań. Np. metoda __add__() zostanie wywołana, gdy dodamy dwie liczby zespolone.

```
1  class Complex(object):
2
3      def __init__(self, real, imag = 0.0):
4          self.real = real
5          self.imag = imag
6
7      def __add__(self, other):
8          return Complex(self.real + other.real, self.imag + other.imag)
9
10     def __sub__(self, other):
11         return Complex(self.real - other.real, self.imag - other.imag)
12
13     def __mul__(self, other):
```

```
14          return Complex((self.real * other.real) - (self.imag * other.
                imag),
15                  (self.imag * other.real) + (self.real * other.imag))
16
17      def __truediv__(self, other):
18          m = other.real ** 2 + other.imag ** 2
19          return Complex((self.real * other.real + self.imag * other.imag)
                    / m,
20                  (self.imag * other.real - self.real * other.imag) / m)
21
22      def __pow__(self, n):
23          r = pow(math.hypot(self.real, self.imag), n)
24          phi = n * ((math.pi / 2 - math.atan2(self.real, self.imag)) % (
                math.pi * 2))
25          return Complex(math.cos(phi) * r, math.sin(phi) * r)
```

Pełen kod aplikacji

```
@echo off
setlocal enabledelayedexpansion

:menu
echo =============
echo 1. Start
echo 2. Informacje
echo 3. Backup
echo 4. Zakoncz
echo =============

set /p input=Wybierz opcje:
if %input% EQU 1 goto start
if %input% EQU 2 goto info
if %input% EQU 3 goto backup
if %input% EQU 4 (
    exit
) else (
    cls
    goto menu
)

:start
call app.py
call web.py
cls
goto menu

:info
echo Program wykonuje podstawowe dzialania arytmetyczne w dziedzinie
   liczb zespolonych
echo Program pobiera dane wejsciowe z pliku
echo a nastepnie zwraca wynik w postaci strony
echo Program jest zadaniem konkursowym z konkursu Algorytmion 2017
echo Autor: Patryk Sroczynski
set /p c=Nacisnij enter
cls
goto menu

:backup
if not exist backup%date% (
mkdir backup%date%
copy "*.*" backup%date% > nul
echo Backup zapisany w folderze backup%date%
) else (
    copy "*.*" backup%date% > nul
    echo Backup juz istnieje, pliki zostaly nadpisane
)
set /p c=Nacisnij enter
cls
goto menu
pause
```

```
import math

class Complex(object):

    def __init__(self, real, imag = 0.0):
        self.real = real
        self.imag = imag

    def __add__(self, other):
        return Complex(self.real + other.real, self.imag + other.imag)

    def __sub__(self, other):
        return Complex(self.real - other.real, self.imag - other.imag)

    def __mul__(self, other):
        return Complex((self.real * other.real) - (self.imag * other.
            imag),
             (self.imag * other.real) + (self.real * other.imag))

    def __truediv__(self, other):
        m = other.real ** 2 + other.imag ** 2
        return Complex((self.real * other.real + self.imag * other.imag)
             / m,
             (self.imag * other.real - self.real * other.imag) / m)

    def __pow__(self, n):
        r = pow(math.hypot(self.real, self.imag), n)
        phi = n * ((math.pi / 2 - math.atan2(self.real, self.imag)) % (
            math.pi * 2))
        return Complex(math.cos(phi) * r, math.sin(phi) * r)


numbers = []
try:
    file = open("in.txt", "r")
    for line in file.readlines():
        try:
            numbers.append(float(line))
        except ValueError:
            numbers.append(1)
    file.close()
    try:
        x1 = numbers[0]
    except IndexError:
        x1 = 1
    try:
        i1 = numbers[1]
    except IndexError:
        i1 = 1
    try:
        x2 = numbers[2]
    except IndexError:
        x2 = 1
    try:
        i2 = numbers[3]
```

```
53      except IndexError:
54          i2 = 1
55      try:
56          n = numbers[4]
57      except IndexError:
58          n = 1
59
60      a = Complex(x1, i1)
61      b = Complex(x2, i2)
62      sum = a + b
63      dif = a - b
64      prod = a * b
65      quo = a / b
66      pow = Complex(3, 2)**3
67
68  except IOError:
69      print("Blad otwierania pliku")
70
71  file = open("out.txt", "w")
72  data = str(sum.real) + "\n" + str(sum.imag) + "\n" + str(dif.real) + "\n
        " + str(dif.imag) + "\n" + str(prod.real) + "\n" + str(prod.imag) + "
        \n" + str(quo.real) + "\n" + str(quo.imag) + "\n" + str(pow.real) + "
        \n" + str(pow.imag)
73  file.write(data)
74  file.close()
```

```
import webbrowser

# in.txt
numbers = []
try:
    file = open("in.txt", "r")
    for line in file.readlines():
        try:
            numbers.append(float(line))
        except ValueError:
            numbers.append(1)
    file.close()
    try:
        x1 = numbers[0]
    except IndexError:
        x1 = 1
    try:
        i1 = numbers[1]
    except IndexError:
        i1 = 1
    try:
        x2 = numbers[2]
    except IndexError:
        x2 = 1
    try:
        i2 = numbers[3]
    except IndexError:
        i2 = 1
    try:
        n = numbers[4]
    except IndexError:
        n = 1
except IOError:
    print("Blad otwierania pliku")

#out.txt
results = []
try:
    file = open("out.txt", "r")
    for line in file.readlines():
        try:
            results.append(float(line))
        except ValueError:
            results.append(0)
    file.close()
except IOError:
    print("Blad otwierania pliku")

if len(results) < 9:
    results = [0, 0, 0, 0, 0, 0, 0, 0, 0]

def plusorminus(number):
    if number >= 0:
        return " + " + str(number)
    else:
```

```
56          return " " + str(number)
57
58  try:
59      file = open("index.html", "w", encoding="utf-8")
60      content = """
61  <!DOCTYPE html>
62  <html lang="pl">
63  <head>
64      <meta charset="UTF-8">
65      <meta http-equiv="X-UA-Compatible" content="IE=edge">
66      <meta name="viewport" content="width=device-width, initial-scale=1.0
            ">
67      <title>Liczby zespolone</title>
68      <style>
69          *, body {
70              margin: 0;
71              padding: 0;
72              color: rgb(255, 255, 255);
73              font-size: 24px;
74          }
75          table {
76              width: 50vw;
77              height: 50vh;
78              margin-top: 10vh;
79          }
80          table,
81          td {
82              border: 3px solid #333;
83              text-align: center;
84          }
85          table, th {
86              background: rgb(114, 114, 245);
87          }
88
89          thead,
90          tfoot {
91              background-color: #333;
92              color: #fff;
93          }
94          .center {
95              margin-left: auto;
96              margin-right: auto;
97          }
98      </style>
99  </head>
100 <body>
101     <table class="center">
102         <thead>
103             <tr>
104                 <th colspan="4">Dzia[U+FFFD]ania na liczbach zespolonych</th>
105             </tr>
106         </thead>
107         <tbody>
108             <tr>
109                 <td>Dodawanie</td>
```

```
110                 <td>""" + str(x1) + plusorminus(i1) + """i</td>
111                 <td>""" + str(x2) + plusorminus(i2) + """i</td>
112                 <td>""" + str(round(results[0])) + " " + plusorminus(
                        round(results[1])) + """i</td>
113             </tr>
114             <tr>
115                 <td>Odejmowanie</td>
116                 <td>""" + str(x1) + plusorminus(i1) + """i</td>
117                 <td>""" + str(x2) + plusorminus(i2) + """i</td>
118                 <td>""" + str(round(results[2])) + " " + plusorminus(
                        round(results[3])) + """i</td>
119             </tr>
120             <tr>
121                 <td>Mnozenie</td>
122                 <td>""" + str(x1) + plusorminus(i1) + """i</td>
123                 <td>""" + str(x2) + plusorminus(i2) + """i</td>
124                 <td>""" + str(round(results[4])) + " " + plusorminus(
                        round(results[5])) + """i</td>
125             </tr>
126             <tr>
127                 <td>Dzielenie</td>
128                 <td>""" + str(x1) + plusorminus(i1) + """i</td>
129                 <td>""" + str(x2) + plusorminus(i2) + """i</td>
130                 <td>""" + str(round(results[6])) + " " + plusorminus(
                        round(results[7])) + """i</td>
131             </tr>
132             <tr>
133                 <td>Potegowanie</td>
134                 <td>""" + str(x1) + plusorminus(i1) + """i</td>
135                 <td> </td>
136                 <td>""" + str(round(results[8])) + " " + plusorminus(
                        round(results[9])) + """i</td>
137             </tr>
138         </tbody>
139     </table>
140 </body>
141 </html>
142     """
143     file.write(content)
144     file.close()
145     webbrowser.open_new_tab('index.html')
146 
147 except IOError:
148     print("Blad otwierania pliku")
```