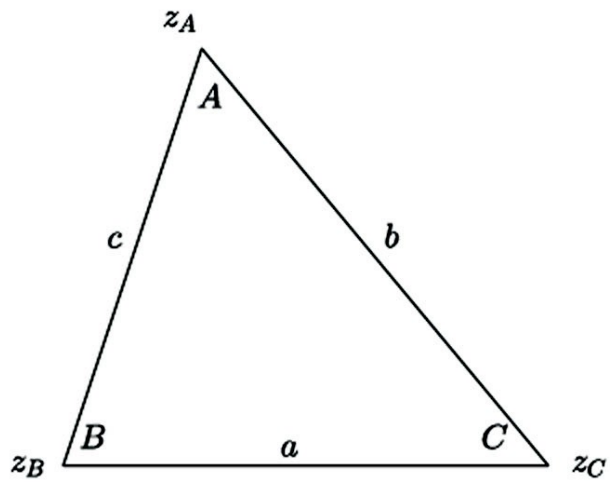
Tema 6

Triunghiuri şi rapoarte complexe.

Ne vom ocupa, în cele ce urmează, de problema triangulaţiei topografice: fiind cunoscute poziţiile a două puncte, *B* şi *C*, punctele care formează *baza* de triangulaţie, se cere să se determine poziţia punctului *A*, punctul nou, măsurând anumite elemente ale triunghiului ΔABC.



Pentru a utiliza notaţiile consacrate din geometria triunghiului, vom nota aici cu z_A, z_B şi z_C afixele vârfurilor triunghiului ΔABC, cu *a*, *b* şi *c* lungimile laturilor *BC*, *CA* şi *AB*, iar cu *A*, *B* şi *C* mărimea unghiurilor triunghiului, măsurată în radiani şi cuprinsă strict în intervalul $(0,\pi)$.

Cu numere complexe avem relaţiile:

$$a=|z_C-z_B|,b=|z_A-z_C|,c=|z_B-z_A|$$

şi

$$A=|arg\ \omega_A|,B=|arg\ \omega_B|,C=|arg\ \omega_C|,$$

unde am notatcu ω_A, ω_B şi ω_C rapoartele

$$\omega_A=\frac{z_C-z_A}{z_B-z_A},\omega_B=\frac{z_A-z_B}{z_C-z_B},\omega_C=\frac{z_B-z_C}{z_A-z_C},$$

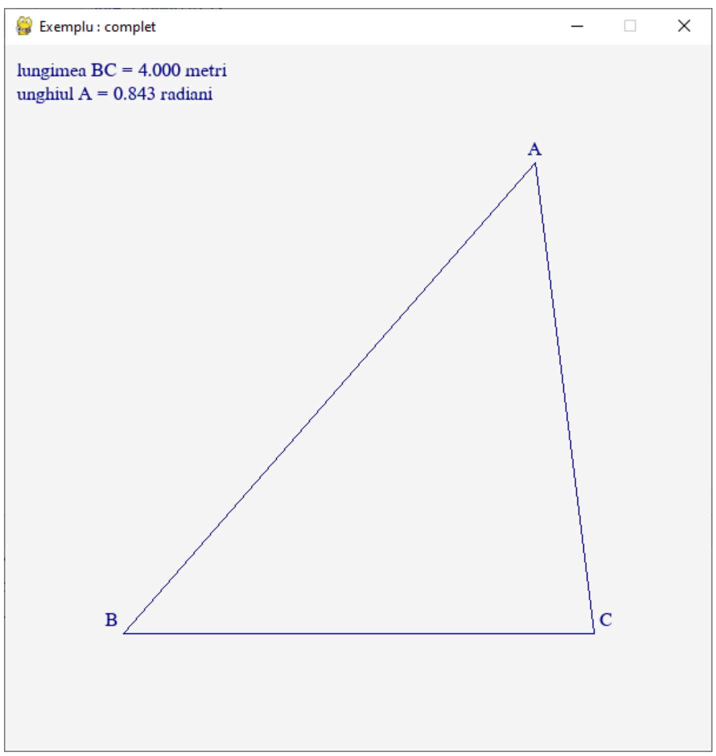
pe care le vom numi în continuare *rotori.* Denumirea este justificată de formula de calcul

$$z_A=z_B+\omega_B(z_C-z_B),$$

de exemplu, care spune că punctul A se obţine din punctul C printr-o rotaţie în jurul lui B de unghi egal cu argumentul lui ω_B şi o scalare de mărime cât modulul lui ω_B.

Programul următor exemplifică implementările mărimilor precizate mai sus:

```
import ComplexPygame as C
import Color
def Exemplu():
    zA, zB, zC = 1.5 + 5j, -2 + 1j, 2 + 1j
    a = C.rho(zC - zB)
    omegaA = (zC - zA) / (zB - zA)
    uA = abs(C.theta(omegaA))
    C.setXminXmaxYminYmax(-3, 3, 0, 6)
    C.drawNgon([zA, zB, zC], Color.Navy)
    C.setText("A", zA)
    C.setText("B", zB - 0.1)
    C.setText("C", zC + 0.1)
    C.setTextIJ(f"lungimea BC = {a:.3f} metri", 10, 30)
    C.setTextIJ(f"unghiul A = {uA:.3f} radiani", 10, 50)

if __name__ == '__main__':
    C.initPygame()
    C.run(Exemplu)
```



Cazul bază-unghi-latură. Începem cu cazul cel mai simplu, cel numit în topografie *radiere*: fiind date punctele de bază z_B şi z_C, se cere să se afle z_A cunoscând unghiul B şi lungimea c. Rezolvarea este imediată, deoarece rotorul vârfului B este gata determinat:

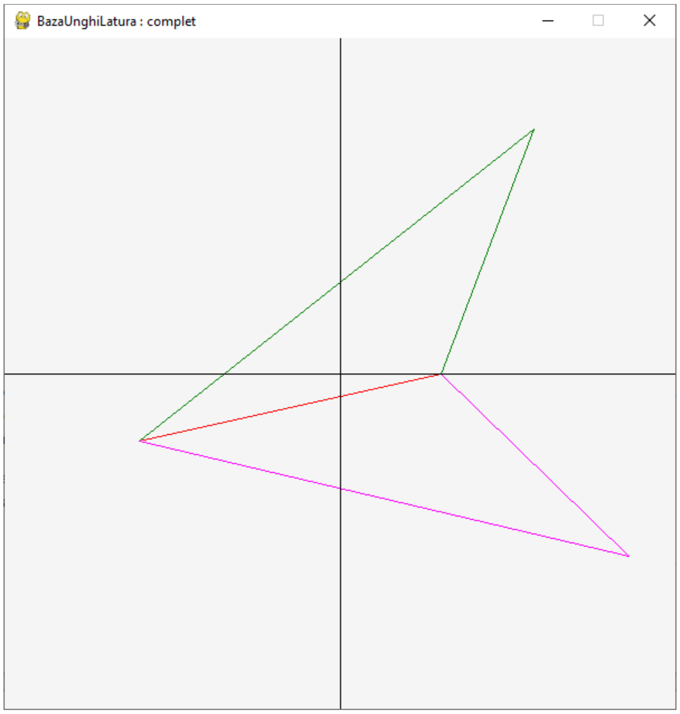
$$\left|\omega_B\right|=\frac{\left|z_A-z_B\right|}{\left|z_C-z_B\right|}=\frac{c}{a}, arg\,\omega_B=B,$$

şi prin urmare z_A se calculează cu relaţia

$$z_A=z_B+\omega_B\left(z_C-z_B\right).$$

Următoarea funcţie construieşte de două ori triunghiul ΔABC, prima dată pe stânga bazei când aceasta este parcursă în sensul de la B la C, şi apoi pe dreapta:

```
def BazaUnghiLatura():
    def bazaUnghiLatura(zB, zC, uB, c, peStg=True):  # returneaza varful zA
        if not peStg:
            uB = -uB
        a = C.rho(zC - zB)
        omegaB = C.fromRhoTheta(c / a, uB)
        return zB + omegaB * (zC - zB)

    C.setXminXmaxYminYmax(-5, 5, -5, 5)
    C.fillScreen()
    zB = -3 - 1j
    zC = 1.5
    uB = math.pi / 7
    c = 7.5
    C.drawLine(zB, zC, Color.Red)
    zA = bazaUnghiLatura(zB, zC, uB, c)
    C.drawLine(zB, zA, Color.Green)
    C.drawLine(zC, zA, Color.Green)
    zA = bazaUnghiLatura(zB, zC, uB, c, False)
    C.drawLine(zB, zA, Color.Magenta)
    C.drawLine(zC, zA, Color.Magenta)
    C.setAxis()
```



Observați că în metoda **bazaUnghiLatura()** parametrul **peStg** are setată valoarea implicită **True,** deci la apelare este necesar să fie precizat numai dacă dorim să aibă valoarea **False.**

Exercițiul 1. Puneți în evidență locul geometric al vârfului A al triunghiului ΔABC, atunci când baza BC este fixă, iar unghiul B și lungimea c variază după legea:

$$\begin{cases} c=t\,, \\ B=\arccos\dfrac{1}{t} \end{cases} \qquad t\in(1,+\infty).$$

Justificați rezultatul observat.

Cazul bază-unghi-unghi. Considerăm cazul când se cunoaște baza și unghiurile B și C de la baza triunghiului. Evident că și unghiul A este cunoscut, $A=\pi-(B+C)$. Pentru determinarea vârfului A se poate folosi rotorul oricărui vârf al triunghiului, noi îl preferăm, pentru uniformitatea prezentării, tot pe $\omega_B=\dfrac{z_A-z_B}{z_C-z_B}$. Acum, pentru a-i calcula modulul, avem nevoie de *Teorema sinusurilor*:

$$\frac{a}{\sin A}=\frac{b}{\sin B}=\frac{c}{\sin C}=2R\,.$$

Prin urmare numărul complex ω_B este determinat din relațiile

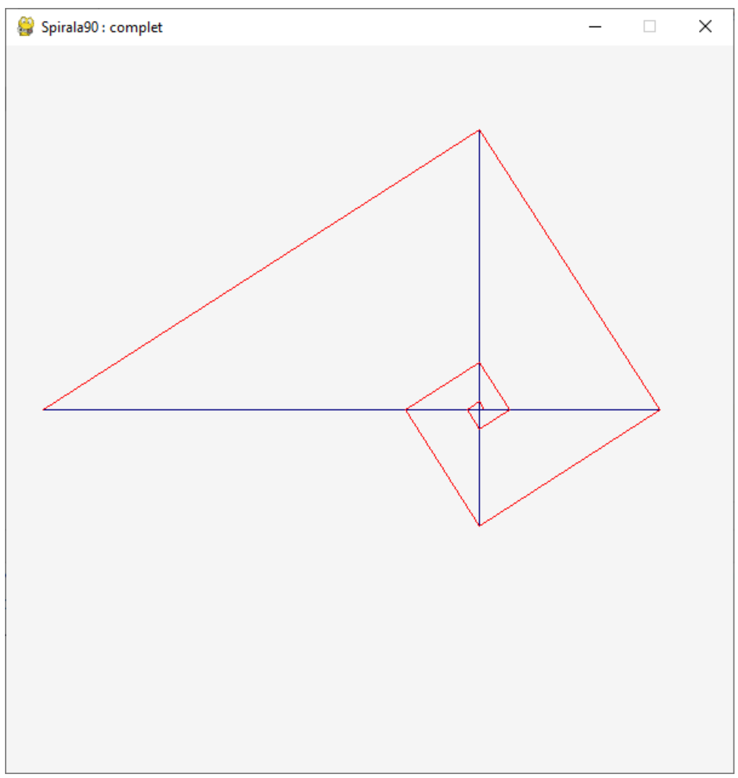
$$|\omega_B| = \frac{|z_A - z_B|}{|z_C - z_B|} = \frac{c}{a} = \frac{\sin C}{\sin A}, \quad arg\,\omega_B = B,$$

iar z_A se calculează cu tot relaţia $z_A = z_B + \omega_B (z_C - z_B)$.

Exerciţiul 2. Implementaţi metoda

```
def bazaUnghiUnghi(zB, zC, uB, uC, peStg=True):
    if not peStg:
        uB, uC = -uB, -uC
    omega = ...
    return zB + omega * (zC - zB)
```

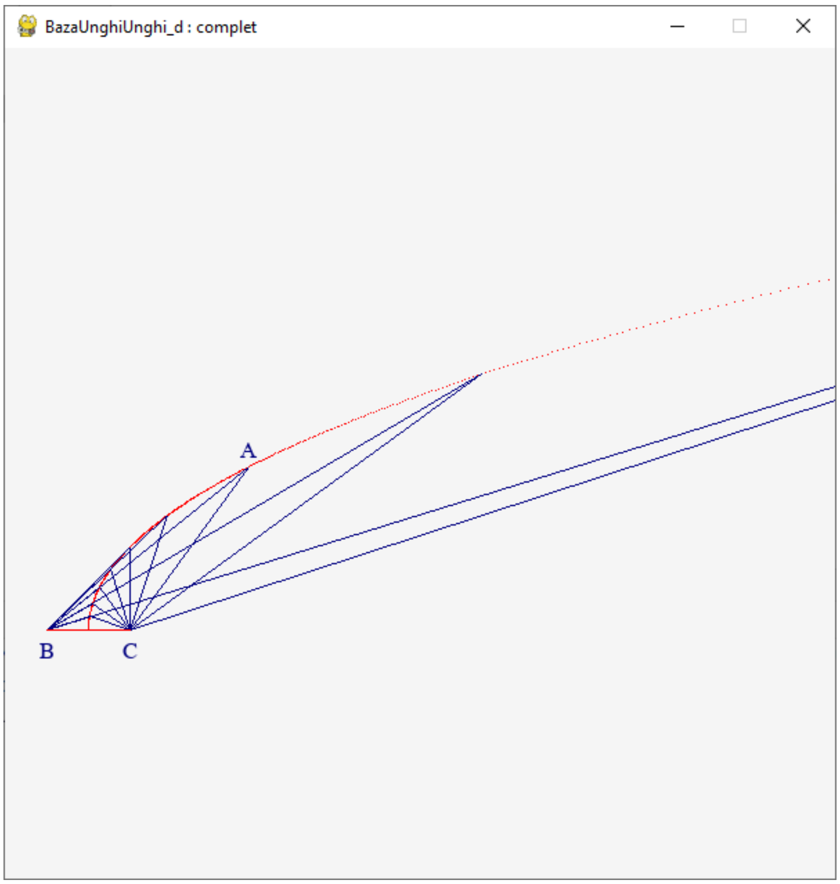
care determină, din datele prezentate, cele două poziţii posibile ale vârfului *A*. Utilizaţi acestă metodă pentru a desena următorul şir de triunghiuri dreptunghice, toate cu aceleaşi unghiuri:



Exerciţiul 3. Punctele *B* şi *C* fiind fixate în mod arbitrar, puneţi în evidenţă, în fiecare dintre cazurile următoare, locul geometric al vârfului *A* al triunghiului $\Delta\, ABC$ când măsura unghiurilor *B* şi *C* variază după legea

$$(a)\ \begin{cases} B = 2t \\ C = \frac{\pi}{2} - t, \end{cases} \quad t \in \left(0, \frac{\pi}{2}\right); \qquad (b)\ \begin{cases} B = \frac{\pi}{3} + t \\ C = \frac{\pi}{3} - t, \end{cases} \quad t \in \left(\frac{-\pi}{3}, \frac{\pi}{3}\right);$$

$$(c)\ \begin{cases} B=\arctan(3t) \\ C=\arctan(2t), \end{cases} \quad t\in(0,+\infty r); \qquad (d)\ \begin{cases} B=\arctan(\sin t) \\ C=t, \end{cases} \quad t\in(0,\pi).$$



Cazul bază-latură-latură. Considerăm, în final, cazul când sunt date vârfurile B şi C şi se cunosc lungimile b şi c. Evident că şi latura a este cunoscută, $a=|z_C-z_B|$. Folosim, de exemplu, iarăşi rotorul $\omega_B=\frac{z_A-z_B}{z_C-z_B}$. Acum modulul este cunoscut:

$$|\omega_B|=\frac{|z_A-z_B|}{|z_C-z_B|}=\frac{c}{a}.$$

iar argumentul se determină din *Teorema cosinusului*:

$$b^2=a^2+c^2-2ac\cos B.$$

Prin urmare

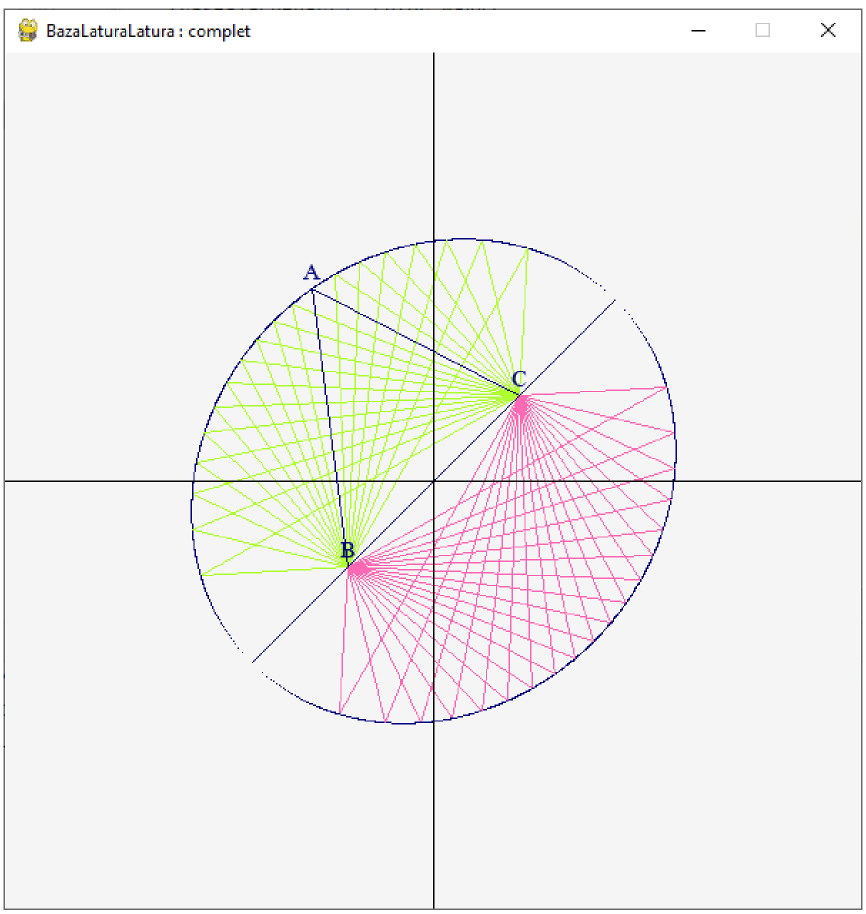
$$arg\,\omega_B = B = \arccos\frac{a^2+c^2-b^2}{2ac},$$

iar $z_A = z_B + \omega_B(z_C - z_B)$.

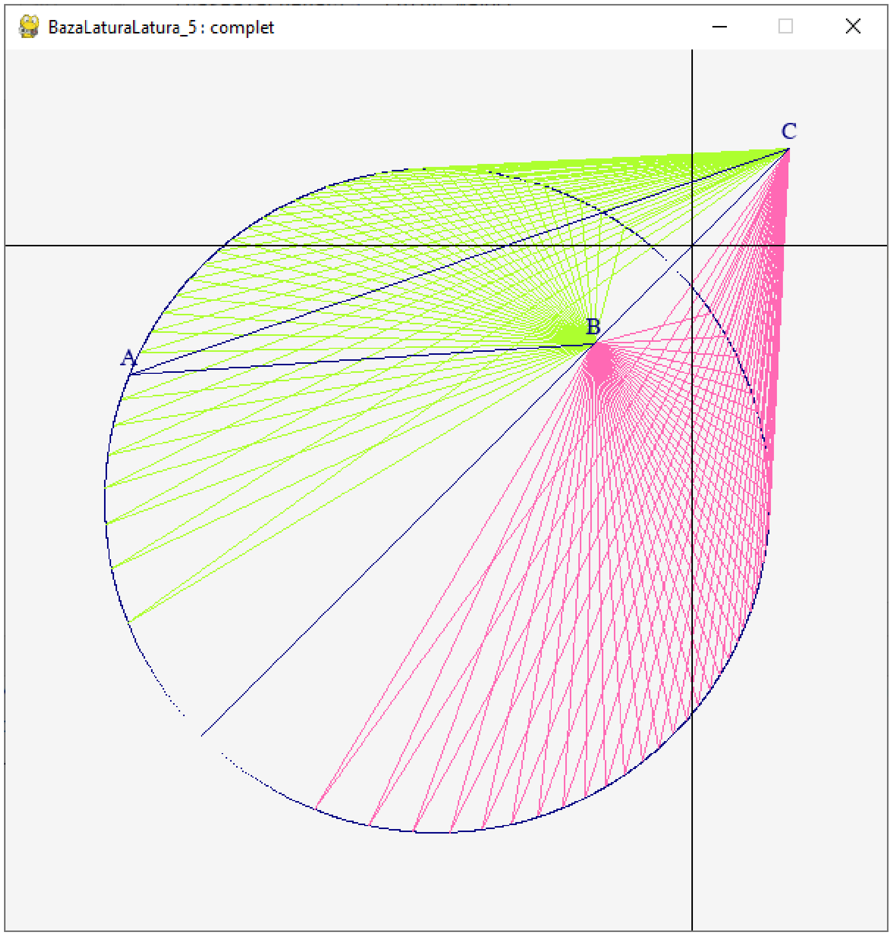
Exercițiul 4. Implementați metoda

```
def bazaLaturaLatura(zB, zC, b, c, peStg=True):
    ...
    return zB + omega * (zC - zB)
```

care determină, din datele prezentate mai sus, cele două poziții posibile ale vârfului A și puneți în evidență locul geometric al punctului z_A în cazul în care $z_B = -1-i$, $z_C = 1+i$, iar lungimile b și c variază după legea: $\begin{cases} b = 3+t \\ c = 3-t, \end{cases} \quad t \in (-\sqrt{2}, \sqrt{2})$.



Exercițiul 5. Același enunț ca mai sus, cu altă lege de variație: $\begin{cases} b=3t \\ c=2t, \end{cases} \quad t\in\left(\frac{2\sqrt{2}}{5}, 2\sqrt{2}\right).$



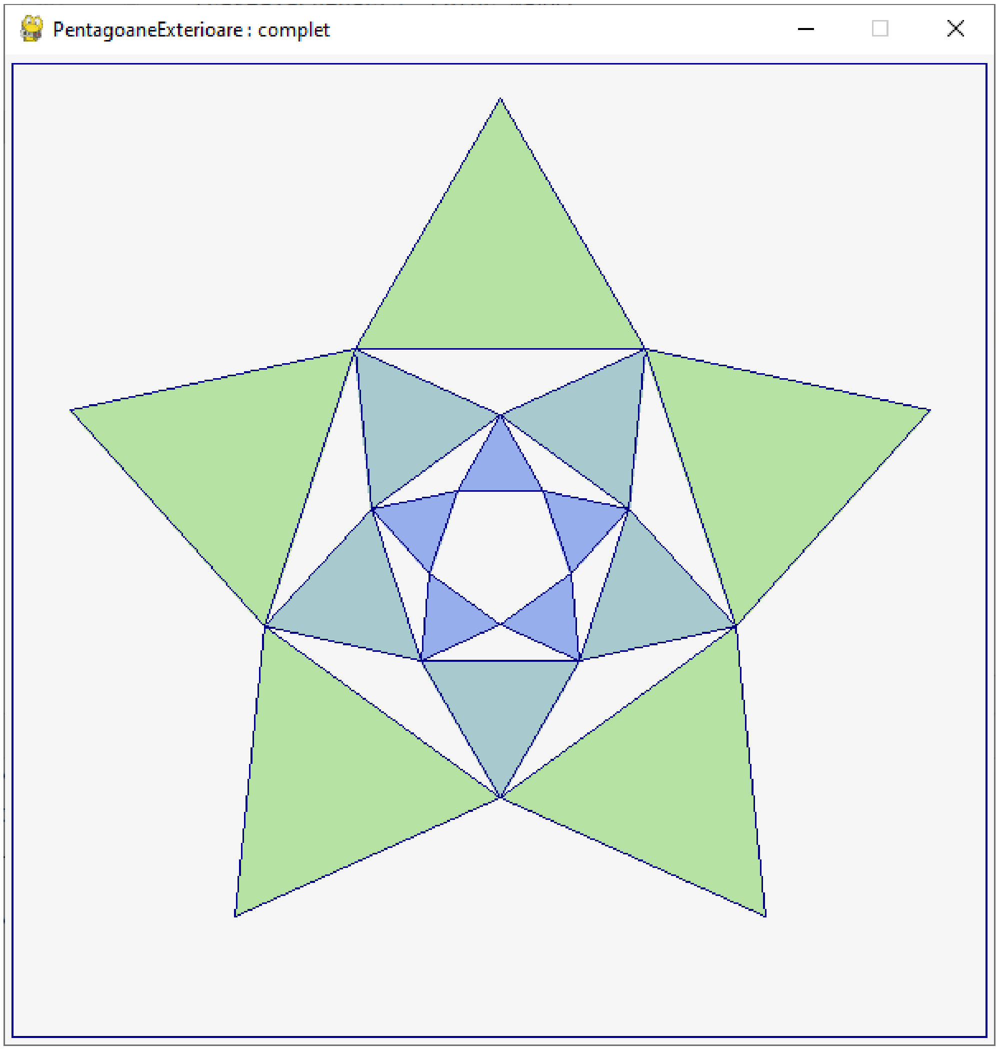
Cazul triunghiului isoscel. Dacă triunghiul ΔABC este isoscel, cu $AB \equiv AC$, atunci este mai elegant să folosim rotorul vârfului A, deoarece acesta este un număr complex unitar,

$$|\omega_A| = \frac{|z_C - z_A|}{|z_B - z_A|} = 1,$$

prin urmare produsul cu ω_A reprezintă o rotație de unghi A, fără scalare. Afixul z_A al apexului A se determină din definiția rotorului $\omega_A = \frac{z_C - z_A}{z_B - z_A}$, și obținem

$$z_A = \frac{z_C - \omega_A z_B}{1 - \omega_A}.$$

Exercițiul 6. Desenul următor a fost obținut construind triunghiuri echilaterale pe exteriorul laturilor pentagonului interior și repetând construcția pentru pentagonul exterior obținut:

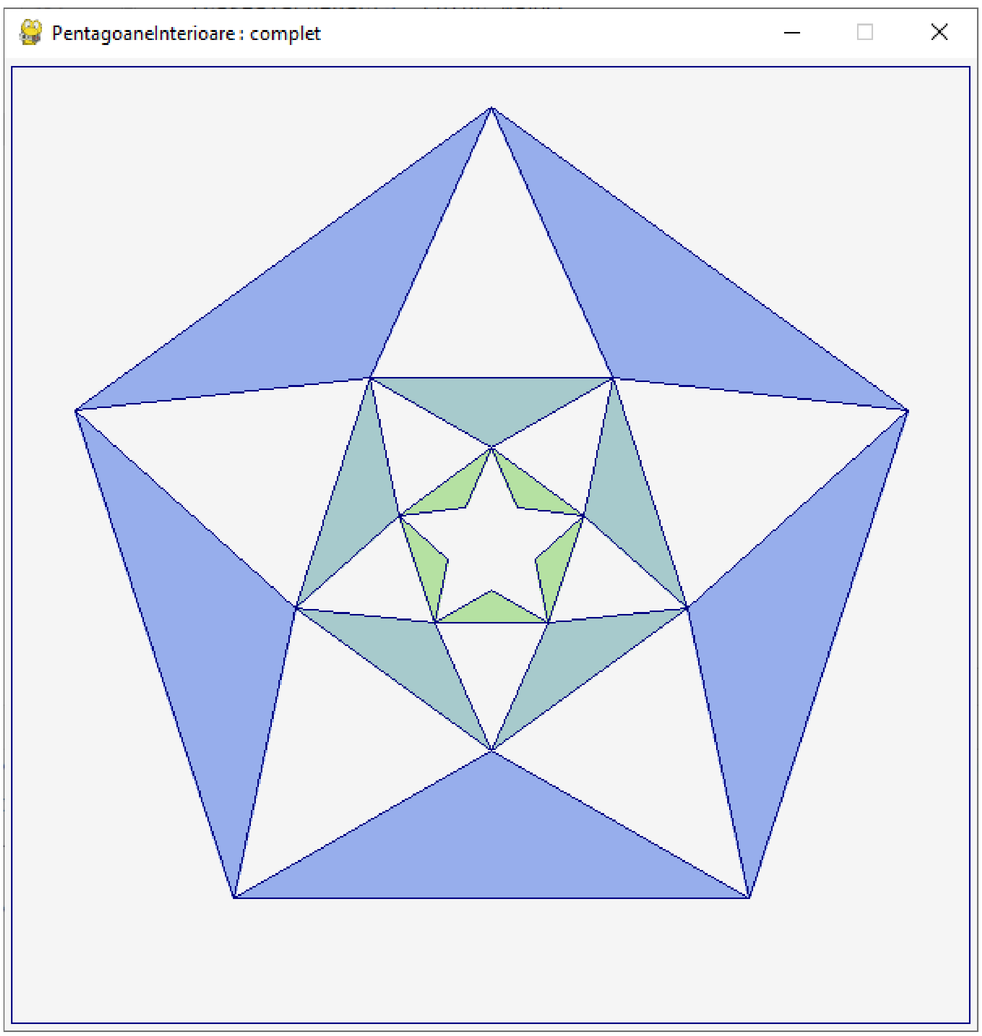


Studiaţi codul funcţiei care a realizat acest desen:

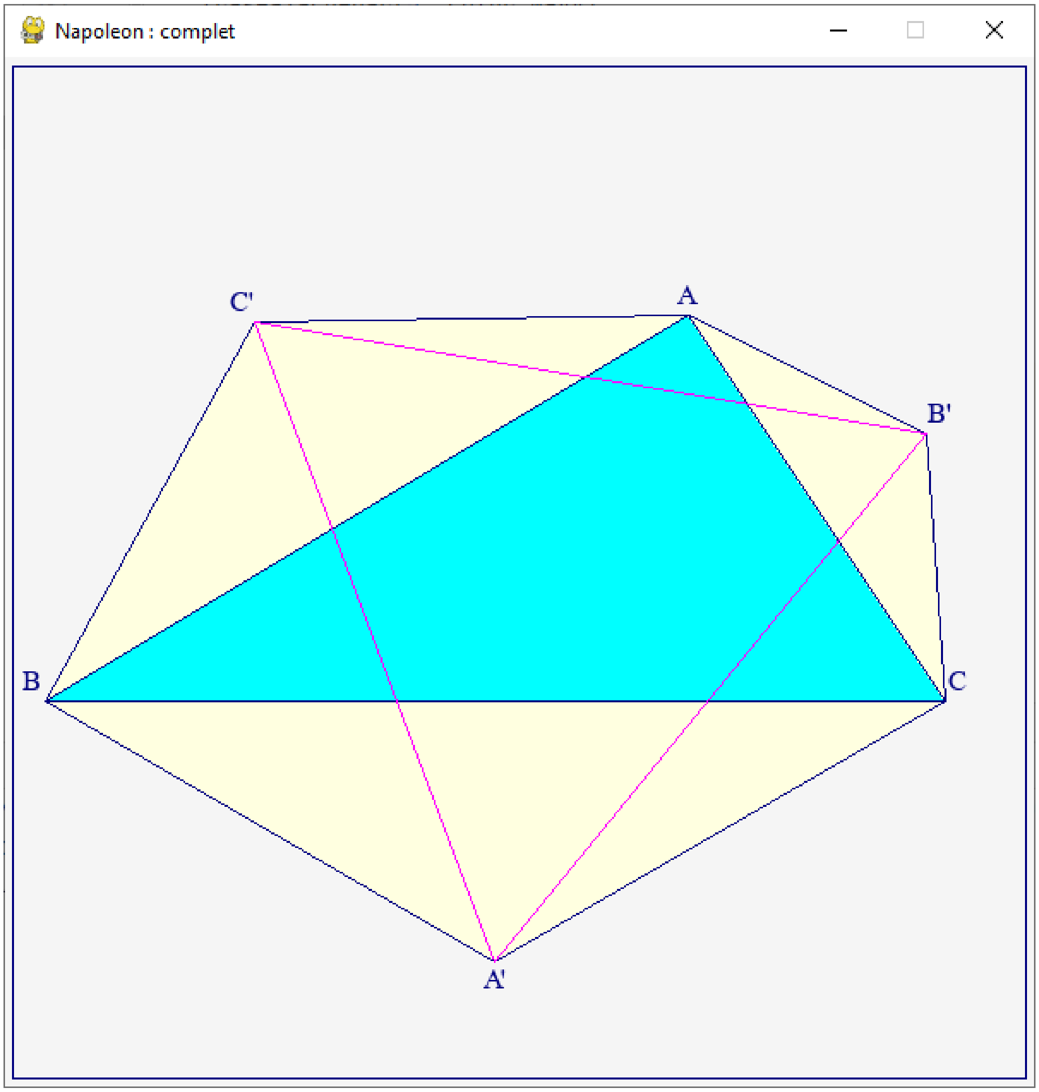
```
def PentagoaneExterioare():
    def bazaApex(zB, zC, uA, peStg=True):
        """
        calculeaza apexul zA al triunghiului isoscel zB zA zC
        """
        omegaA = C.fromRhoTheta(1, uA)
        if not peStg:
            omegaA = omegaA.conjugate()
        return (zC - omegaA * zB) / (1 - omegaA)

    def transformaPoligon(p, col):  # pentagonul p trece in rez
        N = len(p)
        rez = [None] * N
        for k in range(N - 1):
            rez[k] = bazaApex(p[k], p[k + 1], math.pi / 3, False)
            C.fillNgon([p[k], p[k + 1], rez[k]], col)
            C.drawNgon([p[k], p[k + 1], rez[k]], Color.Navy)
        rez[N - 1] = rez[0]
        return rez
```

```
C.setXminXmaxYminYmax(-10, 10, -10, 10)
C.fillScreen()
N = 5
delta = 2 * math.pi / N
q = 0
a = -1.5j
pp = [q + C.fromRhoTheta(1, k * delta) * (a - q) for k in range(N + 1)]
for k in range(3):
    pp = transformaPoligon(pp, Color.Index(100 + 20 * k))
return
```

Încercaţi să modificaţi acest program astfel încât să obţineţi desenul următor, în care s-a pornit de la pentagonul exterior şi s-au construit triunghiuri isoscele pe interiorul laturilor, apoi s-a repetat construcţia pentru pentagonul nou obţinut:



Exercițiul 7. Desenați și rezolvați *problema lui Napoleon* : pe fiecare latură a unui triunghi oarecare ΔABC construiește în exterior câte un triunghi isoscel cu unghiul de la vârf de $120°$ și cu baza pe latura triunghiului dat. Cu notațiile din figura următoare, arătați că triunghiul $\Delta A'B'C'$ este echilateral.



Indicație: Folosim rotorul $\omega=\cos\frac{2\pi}{3}+i\sin\frac{2\pi}{3}$ și avem $z_{B'}=\frac{1}{1-\omega}(z_A-\omega z_C)$ și celelalte. Folosind relațiile $\omega^3=1$ și $\omega^2+\omega+1=0$ obținem $z_{C'}-z_{B'}=\frac{1}{1-\omega}(\omega^2 z_A+\omega z_C+z_B)$, etc.