

Języki programowania i GUI

Lista 5 - 2021

1. Dany jest fragment programu napisanego w języku Julia:

```
struct Osoba
    imie::String
    nazwisko::String
end

osoby=[
    Osoba("Jan","Kowalski"),
    Osoba("Ewa","Nowak"),
    Osoba("Artur","Kowalski"),
    Osoba("Adam","Nowak")
]
```

Napisz linię kodu sortującą i wypisującą na ekranie listę osób w kolejności alfabetycznej/antyalfabetycznej: (a) wg imienia używając `by=`, (b) wg imienia używając `lt=`. (c) wg nazwiska i imienia używając `lt=`, (d) wg nazwiska i imienia używając `by=`.

2. Używając operatora `\` rozwiąż w jednej linii układ równań:

$$\begin{cases} 3x + 5y + z = 1 \\ 4x + 2y - 7z = 2 \\ 2x - 3y + 3z = 3 \end{cases}$$

3. Stwórz tablicę o wymiarach 20x40 zawierającą liczby ze zbioru 1:15. Policz ilość wystąpień każdej liczby w tablicy za pomocą: (a) słownika (b) tablicy liczników
4. W podobny sposób zlicz ilość wystąpień każdej litery/cyfry/znaku występującej w pliku zawierającym Twoje rozwiązanie niniejszego zadania.
5. (3pkt) Zaimplementuj struktury Punkt, Koło, Prostokąt opisujące leżące na płaszczyźnie figury o różnych kształtach rozmiarach i położeniach.

Zalecam reprezentowanie punktów na płaszczyźnie za pomocą liczb zespolonych. Wówczas odległość między punktami `a` i `b` wynosi `abs(a-b)` a pole trójkąta `a b c` wynosi `imag(conj(c-a)*(b-a))/2`

Dla każdej z figur zdefiniuj funkcje obliczające pola i obwody.

Dla każdej pary klas zdefiniuj funkcję odległość obliczającą odległość między nimi. Odległością figur `A`, `B` nazywamy długość najkrótszego odcinka którego jeden koniec należy do figury `A` a drugi koniec do figury `B`.

Następnie stwórz tablicę dwunastu figur położonych w pewnych odległościach od siebie i w jednej linii skonstruuj macierz odległości między tymi figurami.

6. Dodatkowo dla każdej figury zdefiniuj funkcję `rysuj(fig)`, która nanosi kontur figury na istniejący wykres. Użyj biblioteki `'Plots'`. Sprawdź na kilku przykładach czy Twoje funkcje działają poprawnie.
7. Napisz klasę `PrimesUpTo(n)` oraz funkcję `iterate(P::PrimesUpTo,state)` tak , aby liczby pierwsze nie większe niż `n` można było wypisać poleceniem

```
for i in PrimesUpTo(n)
    print(i, ' ')
end
```