

Dokumentacja projektu

Liczby zaprzyjaźnione

Programowanie 3

Mikołaj Pazera, grupa 1B

27 stycznia 2020

Część I

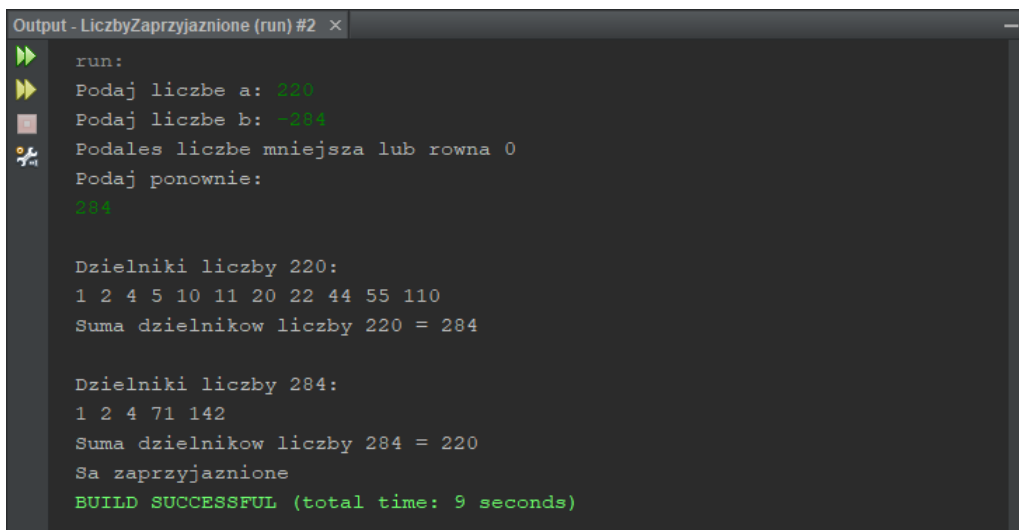
Opis programu

Program służy do sprawdzania czy dana para tworzy tzw. liczby zaprzyjaźnione. Dane pobierane są od użytkownika w programie. Polecenie:

Za Wikipedią: *Liczby zaprzyjaźnione to para różnych liczb naturalnych, takich, że suma dzielników każdej z tych liczb równa się drugiej (nie uwzględniając tych dwóch liczb jako dzielników).*" Np. liczba 284 ma dzielniki: 1, 2, 4, 71, 142, których suma daje 220, a liczba 220 ma dzielniki: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, których suma daje 284. Zatem liczby 220 i 284 tworzą parę liczb zaprzyjaźnionych. Należy napisać program, który dla dowolnej pary różnych liczb naturalnych będzie rozstrzygał, czy para ta tworzy liczby zaprzyjaźnione.

Instrukcja obsługi

Program pobiera dane od użytkownika z klawiatury. Pary liczb są sprawdzane, następnie wszystkie informacje wypisywane są na konsolę. Program jest zabezpieczony przed wpisywaniem nieporpawnych danych. Przykładowe uruchomienie programu wraz z wynikiem jego działania:



```
Output - LiczbyZaprzyjaznione (run) #2 x
run:
Podaj liczbe a: 220
Podaj liczbe b: 284
Podales liczbe mniejsza lub rowna 0
Podaj ponownie:
284

Dzielniki liczby 220:
1 2 4 5 10 11 20 22 44 55 110
Suma dzielnikow liczby 220 = 284

Dzielniki liczby 284:
1 2 4 71 142
Suma dzielnikow liczby 284 = 220
Sa zaprzyjaznione
BUILD SUCCESSFUL (total time: 9 seconds)
```

Część II

Część techniczna

Program napisany jest w języku Java w wersji 1.8.0_221. Dane liczbowe są typu `int`. Zabezpieczenie przed wprowadzeniem niepoprawnych danych jest zrealizowane za pomocą struktury `try {...} catch {...}`. Szukanie dzielników wykonywane jest za pomocą pętli `for`.

Opis działania

Sprawdzanie czy para liczb tworzy liczby zaprzyjzaznione odbywa się za pomocą szukania dzielników obu liczb. Nie jest używany w tym celu konkretny wzór lecz sprawdzanie wszystkich liczb i od 0 do n . Jeśli i jest dzielnikiem liczby n to dodawana jest ona do sumy dzielników

$$i \bmod (n) \equiv 0 \quad (1)$$

Jeśli suma dzielników liczby (z wyłączeniem samej liczby) a jest równa liczbie b (i odwrotnie) to liczby tworzą parę liczb zaprzyjzaznionych. Przykładowo sprawdzenie liczb 220 oraz 286:

Dzielniki liczby 220	Dzielniki liczby 284
1	1
2	2
4	4
71	5
142	10
	11
	20
	22
	44
	55
	110
Suma: 284	Suma: 220

Pseudokod programu

Result: Program Liczby Zaprzyjaźnione

start;

Wczytaj liczbę a;

while *Liczba a < 0 // nie jest liczbą całkowitą* **do**
| Pobierz liczbę a ponownie
end

Wczytaj liczbę b;

while *Liczba b < 0 // nie jest liczbą całkowitą* **do**
| Pobierz liczbę b ponownie
end

Licz dzielniki a;

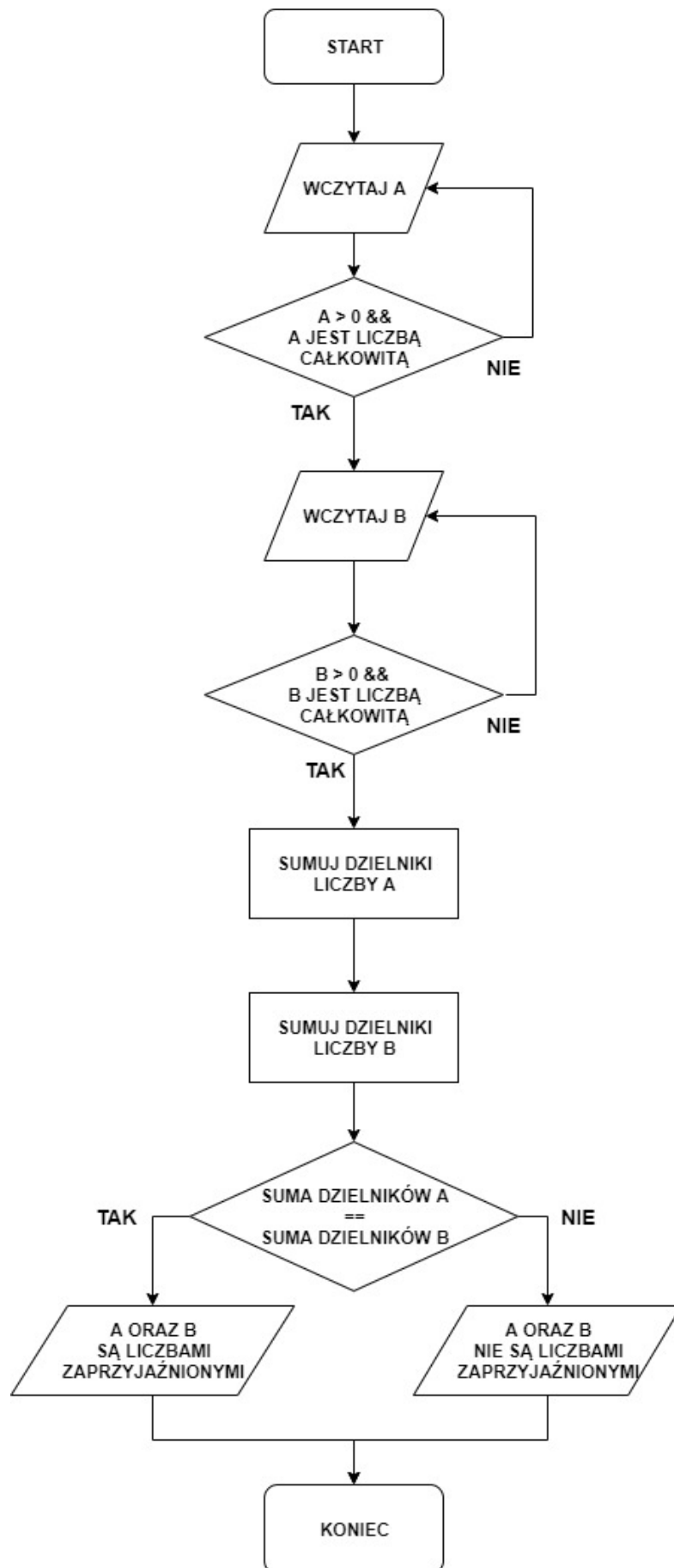
Licz dzielniki b;

if *suma dzielników a = suma dzielników b* **then**
| a oraz b są zaprzyjaźnione
end

else
| a oraz b nie są zaprzyjaźnione
end

Algorithm 1: Pseudokod programu

Schemat blokowy



Pełny kod programu

Listing 1: Plik python_raport.py

```
package liczbyzaprzyjzaznione;
import java.util.*;

public class LiczbyZaprzyjzaznione {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        // Wczytywanie liczb
        System.out.print("Podaj liczbę a: ");
        int a = 0;

        // Sprawdzenie poprawności typu zmiennej
        try{
            // Dopuszczalne tylko liczby dodatnie
            do{
                a = in.nextInt();
                // Wczytaj ponownie jeśli mniejsza od 0
                if(a <= 0){
                    System.out.println("Podajesz liczbę mniejszą lub równą 0\nPodaj ponownie: ");
                }
            }while (a <= 0);
        }
        catch(Exception e){
            System.out.println(e);
            return;
        }
        System.out.print("Podaj liczbę b: ");
        int b = 0;

        // Sprawdzenie poprawności typu zmiennej
        try{
            // Dopuszczalne tylko liczby dodatnie
            do{
                b = in.nextInt();
```

```

        if(b <= 0){
            // Wczytaj ponownie jesli mniejsza od 0
            System.out.println("Podales liczbe mniejsza lub rowna 0\n
            Podaj ponownie: ");
        }
    }while (b <= 0);
}
catch(Exception e){
    System.out.println(e);
    return;
}

// Tworzenie zmiennych
ArrayList<Integer> dzielnikiA = new ArrayList<Integer>();
ArrayList<Integer> dzielnikiB = new ArrayList<Integer>();
int sumaDzA = 0;
int sumaDzB = 0;

// Szukanie dzielnikow, sumowanie oraz zapis do tablicy
System.out.println("\nDzielniki liczby " + a + ": ");
for (int i = 1; i < a; i++){
    if(a % i == 0){
        dzielnikiA.add(i);
        System.out.print(i + " ");
        sumaDzA += i;
    }
}
System.out.println("\nSuma dzielnikow liczby " + a + " = " + sumaDzA);
// Szukanie dzielnikow, sumowanie oraz zapis do tablicy
System.out.println("\nDzielniki liczby " + b + ": ");
for (int i = 1; i < b; i++){
    if(b % i == 0){
        dzielnikiB.add(i);
        System.out.print(i + " ");
        sumaDzB += i;
    }
}
System.out.println("\nSuma dzielnikow liczby " + b + " = " + sumaDzB);
// Wyświetlenie informacji koncowej

```

```
    if(sumaDzA == b && sumaDzB == a){  
        System.out.println("Sa zaprzyjaznione");  
    }  
    else{  
        System.out.println("Nie sa zaprzyjaznione");  
    }  
}  
  
}
```