

Material do lekcji 2 (PR)

Temat: Rekurencja – silnia

Cele lekcji:

1. Uczeń zna pojęcie rekurencji
2. Uczeń wie kiedy ciąg jest zdefiniowany rekurencyjnie
3. Uczeń wymienia klasyczne przykłady zastosowania rekurencji
4. Uczeń stosuje rekurencję do obliczania silni liczby naturalnej

Przebieg lekcji:

1. Zapoznanie z celami lekcji
2. Ciąg liczb zdefiniowany rekurencyjnie

Definicja

Ciąg jest zdefiniowany rekurencyjnie, jeśli spełnione są następujące warunki:

- a) określony jest pewien skończony zbiór początkowych wyrazów ciągu (na przykład kilka pierwszych lub pierwszy wyraz);
- b) kolejne wyrazy ciągu definiowane są za pomocą poprzednich elementów ciągu.

Warunek (a) definiuje wyrazy początkowe ciągu. Pozostałe wyrazy ciągu są określane za pomocą wzoru (b), czyli rekurencyjnie. Wzór definiujący taki ciąg nazywamy wzorem, równaniem lub zależnością rekurencyjną.

W algorytmice i programowaniu rekurencja występuje wówczas, gdy **procedura lub funkcja wywołuje samą siebie**, czyli uruchomienie procedury lub funkcji następuje wewnątrz niej samej.

Algorytmów rekurencyjnych raczej nie przedstawia się za pomocą schematów blokowych. Wymagałoby to oddzielnego konstruowania procedur czy funkcji rekurencyjnych w postaci schematów. Można jednak zastosować listę kroków, definiując funkcję i wskazując jej parametry.

3. Silnia liczby naturalnej – podstawy matematyczne

Definicja

Silnia liczby naturalnej n to iloczyn kolejnych liczb naturalnych $1, 2, 3, \dots, n$, co oznaczamy $n!$. Na przykład $4! = 1 \cdot 2 \cdot 3 \cdot 4$. Należy jednak zwrócić uwagę na to, że najmniejszą liczbą, dla której definiuje się silnię, jest 0 . $0!$ równe jest 1 .

Podstawą wyznaczenia $n!$ jest formalna definicja silni, która ma następującą postać:

$$n! = \begin{cases} 1 & \text{dla } n = 0 \\ (n-1)! \cdot n & \text{dla } n > 0 \end{cases} \quad (1.12)$$

Powyższą definicję ilustruje przykład:

$$4! = 3! \cdot 4 = 2! \cdot 3 \cdot 4 = 1! \cdot 2 \cdot 3 \cdot 4 = 0! \cdot 1 \cdot 2 \cdot 3 \cdot 4 = 1 \cdot 1 \cdot 2 \cdot 3 \cdot 4$$

Skonstruujmy rekurencyjny **algorytm wyznaczający silnię liczby naturalnej**.

Specyfikacja:

Dane: Liczba naturalna: $n \geq 0$.

Wynik: Silnia liczby naturalnej n : $n!$.

Definicję rekurencyjną, na podstawie której tworzymy funkcję rekurencyjną wyznaczającą $n!$, można przedstawić następująco:

$$\begin{cases} a_0 = 1 \\ a_n = a_{n-1} \cdot n & \text{dla } n > 0 \end{cases} \quad (1.13)$$

4. Programowanie rekurencyjne – obliczanie silni.

Zadanie do wykonania:

- wyznaczanie wyrazów ciągu Fibonacciego – zapoznaj się z definicją (str. 39)

```

#include <cstdlib>
#include <iostream>
using namespace std;

long long silnia(int n)
{
    if(n<2)
        return 1; //silnia z 0 i 1 wynosi 1

    return n*silnia(n-1); //wywołanie funkcji przez samą siebie ze zmniejszonym argumentem
}

int main()
{
    int n;

    cout<<"Podaj liczbę: ";
    cin>>n;
    cout<<"n! = "<<silnia(n)<<endl;

    system("pause");
    return 0;
}

```

Przeanalizujmy działanie programu dla $n = 4$;

$$\begin{aligned}
 \text{wynik} &= \text{silnia}(4) = 4 \cdot \text{silnia}(3) = 4 \cdot 3 \cdot \text{silnia}(2) = \\
 &4 \cdot 3 \cdot 2 \cdot \text{silnia}(1) = 4 \cdot 3 \cdot 2 \cdot 1 = 24
 \end{aligned}$$

<https://forum.pasja-informatyki.pl/310927/zadanie-rekurencja>

<http://www.algorytm.edu.pl/rekurencja-zadania/75-n-ty-wyraz-ciagu.html>