

**II Liceum Ogólnokształcące im. Mikołaja Kopernika z Oddziałami
Dwujęzycznymi i Międzynarodowymi
Bolesława Prusa 33, 64-100 Leszno**

Silnia - rekurencja

Praca przygotowana przez

Patrycję Klein, Zofię Kubicę, Katarzynę Turbańską

Klasa 2e

Pod kierunkiem

Dominika Sיעińskiego

Leszno, 9 maja 2018

Silnia - rekurencja

TEZA: Jest możliwe wytłumaczenie i zrozumienie tematu projektu.

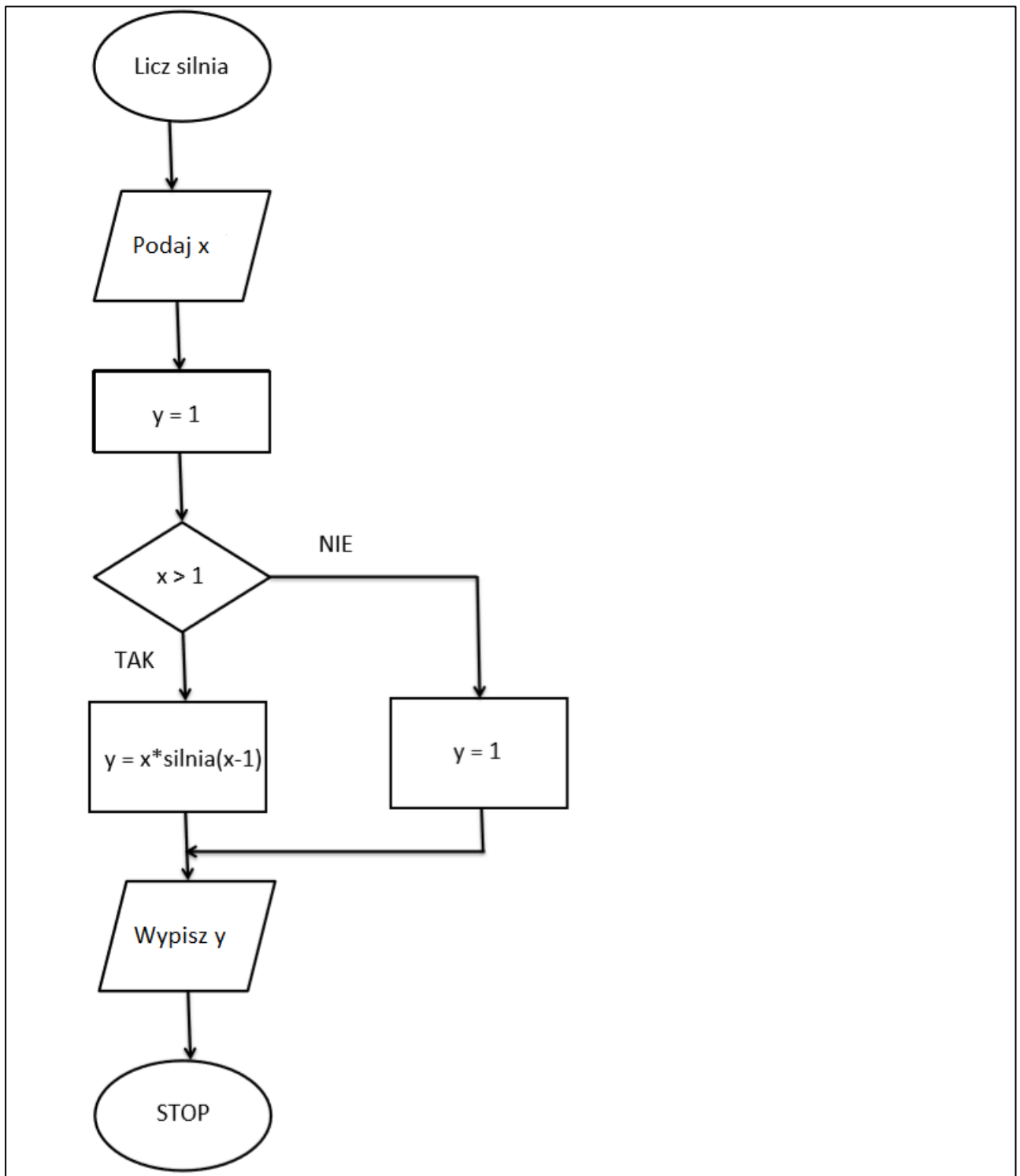
CZĘŚĆ TEORETYCZNA

1. Zasada działania algorytmu.

Podajemy liczbę, z której chcemy obliczyć silnię. Liczba ta przekazywana jest do funkcji silnia. Jeśli podana przez nas liczba wynosi 1 lub 0, zostaje zwrócona liczba 1, ponieważ $\text{silnia}(1) = 1$ oraz $\text{silnia}(0) = 1$. Jeśli liczba jest większa od 1, to komputer mnoży podaną przez nas liczbę przez silnię z liczby o jeden mniejszej. Robi to do momentu, w którym do funkcji silnia zostanie przekazana liczba 1, ponieważ $\text{silnia}(1) = 1$. W tym momencie zostaje zwracany iloczyn wszystkich wcześniejszych silni. Następnie zostaje wyświetlony otrzymany wynik.

Silnia - rekurencja

2. Rysunek ze schematem blokowym.



Rysunek nr 1

Przedstawia schemat blokowy silni obliczanej rekurencyjnie

Silnia - rekurencja

3. Rysunek z pseudokodem.

Start

silnia(x)

y ← 1

func silnia(x)

 if x>1

 then y ← x*silnia(x-1)

 else y ← 1

 endif

return y

endfunc

wypisz y

Koniec

Rysunek nr 2

Przedstawia pseudokod silni obliczanej rekurencyjnie

Silnia - rekurencja

4. Kod algorytmu w języku Swift.

```
func silnia(_ liczba: Int)->Int
{
  if (liczba > 1) {return liczba*silnia(liczba -1)}
  else {return 1}
}

print("Podaj liczbe")
var ile=Int(readLine()!)!
print(silnia(ile))
```

Rysunek nr 3

Przedstawia kod w języku Swift silni obliczanej rekurencyjnie

Silnia - rekurencja

5. Kod algorytmu w języku Python.

```
def silnia(x: int) -> int:
    if x > 1:
        return x*(silnia(x-1))
    else:
        return 1

print("Podaj liczbe")
x = int(input())
print(silnia(x))
```

Rysunek nr 4

Przedstawia kod w języku Python silni obliczanej rekurencyjnie

Silnia - rekurencja

6. Kod algorytmu w języku Java.

```
import java.util.Scanner;

public class silna3 {

    static int silnia(int n) {

        if (n > 1)

            return (n*silnia(n-1));

        else

            return 1;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Podaj liczbe:");

        int liczba = sc.nextInt();

        System.out.println(liczba + " ! = " + silnia(liczba));

    }

}
```

Rysunek nr 5

Przedstawia kod w języku Java silni obliczanej rekurencyjnie

Silnia - rekurencja

7. Kod algorytmu w języku C++.

```
#include <iostream>

#include <cstdlib>

using namespace std;

int silnia (int n)

{

    if (n<2) return n;

    return n*silnia(n-1);

}

int main()

{

    int n;

    cout << "Podaj liczbe" << endl;

    cin >> n;

    cout << silnia(n) << endl;

    return(0);

}
```

Rysunek nr 6

Przedstawia kod w języku C++ silni obliczanej rekurencyjnie

Silnia - rekurencja

WNIOSKI

Silnia liczby naturalnej n jest to iloczyn wszystkich liczb naturalnych dodatnich nie większych od n .

Funkcja zapisana za pomocą rekurencji jest to funkcja odwołująca się do samej siebie.

Ukazana powyżej w różnych językach programowania silnia zapisana za pomocą rekurencji jest algorytmem zarówno łatwym do wytłumaczenia jak i do zrozumienia. Schemat tej funkcji jest krótki, przejrzysty, prosty i nieskomplikowany. Do funkcji silni wprowadzamy naturalną liczbę dodatnią, jak wynika to z definicji silni. Jeśli liczba ta jest mniejsza od 2, silnia zwraca wartość 1. W przeciwnym wypadku odwołuje się do samej siebie i zwraca iloczyn podanej liczby oraz funkcji silni o jeden mniejszej od podanej liczby. Wykonuje to do momentu, w którym do funkcji silnia zostanie przekazana liczba 1, z której zostaje zwrócona wartość 1. Wtedy zostaje zwracany iloczyn wszystkich wcześniejszych silni.

Sam proces zapisu algorytmu w różnych językach programowania również nie stanowi większego problemu, jeśli zna się podstawy danego języka. Wystarczy jedynie zrozumieć zasadę działania algorytmu.

Dlatego też z pozycji nauczyciela, którą przyjęliśmy w trakcie części praktycznej naszego zadania, możemy stwierdzić, iż jest to temat prosty i przyjemny do wytłumaczenia. Sprzyja temu zarówno długość algorytmu, jego przejrzystość oraz prostota. Jedynym kłopotem okazały niekiedy zaskakujące pytania stawiane przez uczniów, na które nie zawsze byliśmy w stanie odpowiedzieć.

Z naszych obserwacji wynika jednak, że uczniowie pragnący zrozumieć dane zagadnienie nie mają z tym większych trudności, a i pozostali są w stanie w stopniu zadawalającym przekazać dalej otrzymane od nas informacje. Na wszystkie pytania zadane po zakończeniu prowadzonej przez nas lekcji otrzymaliśmy satysfakcjonujące nas odpowiedzi.

Silnia - rekurencja

Spis zdjęć, rysunków, tabel i wykresów:

1. Rysunek przedstawiający schemat blokowy silni obliczanej rekurencyjnie
rysunek własny
2. Rysunek przedstawiający pseudokod silni obliczanej rekurencyjnie
rysunek własny
3. Rysunek przedstawiający kod w języku Swift silni obliczanej rekurencyjnie
rysunek własny
4. Rysunek przedstawiający kod w języku Python silni obliczanej rekurencyjnie
rysunek własny
5. Rysunek przedstawiający kod w języku Java silni obliczanej rekurencyjnie
rysunek własny
6. Rysunek przedstawiający kod w języku C++ silni obliczanej rekurencyjnie
rysunek własny

Zgadzam się na udostępnienie mojego projektu na stronie www.code.kopernik-leszno.pl, aby kolejni uczniowie mogli skorzystać z tego materiału przy nauce programowania.