

**II Liceum Ogólnokształcące im. Mikołaja Kopernika z Oddziałami
Dwujęzycznymi i Międzynarodowymi; Leszno, ul. Bolesława Brusa 33**

**Jest możliwe wytłumaczenie i zrozumienie
tematu projektu.**

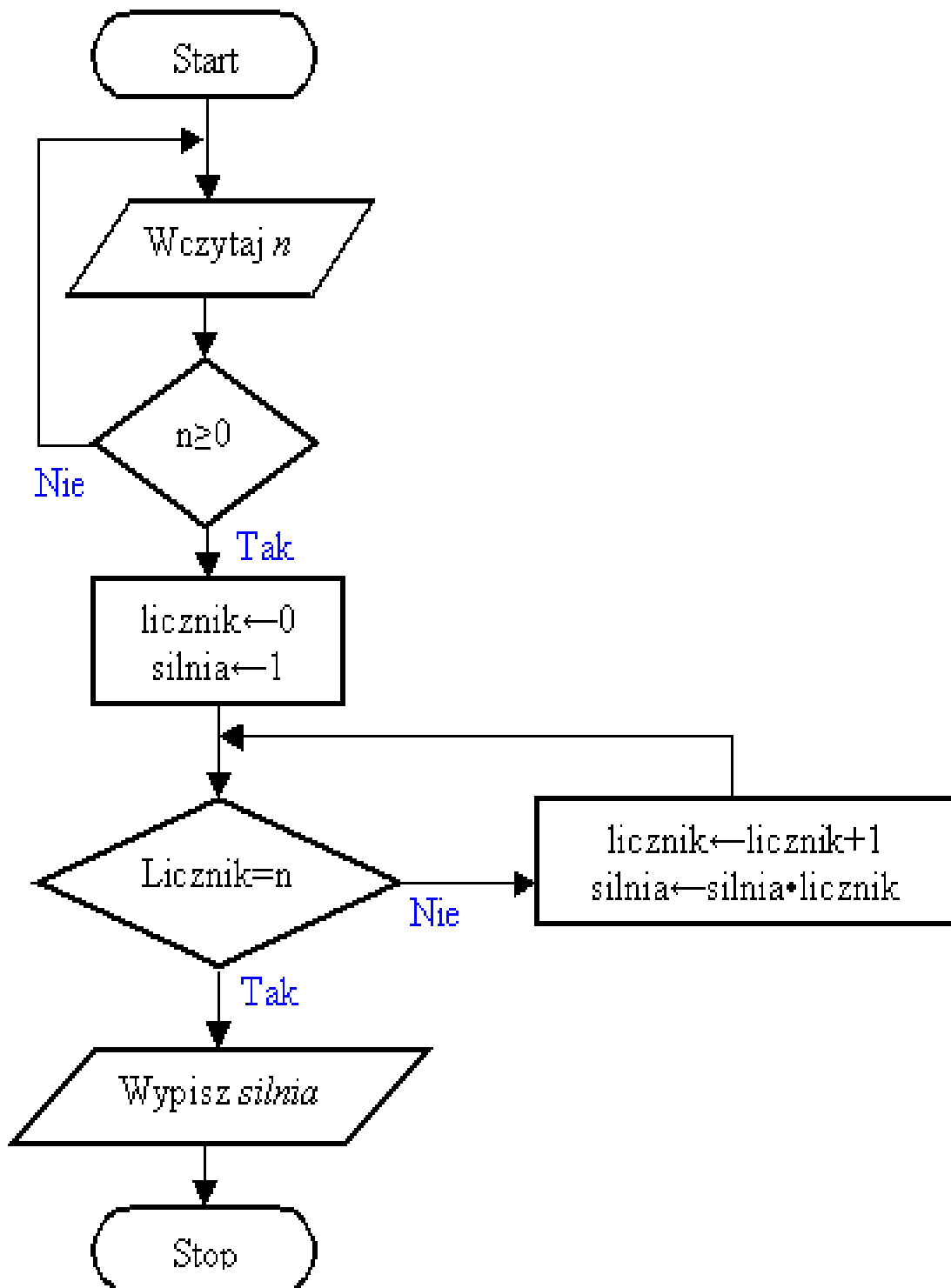
**Praca przygotowana przez
Noemi Kopaczyńską, Julię Klupczyńską i Jerzego Tomaszewskiego
Klasa 1B2 i 1A
pod kierunkiem
pana Dominika Sיעińskiego**

Leszno, 12 kwietnia 2018 roku

Silnia - iteracja

Silnia danej liczby to funkcja, dzięki której można zapisać różne wzory i zależności w skondensowany sposób. Jest ona iloczynem kolejnych liczb naturalnych licząc od liczby jeden. Silnię zapisujemy w sposób „n!”, gdzie „n” jest liczbą naturalną. Na przykład silnię z liczby 3 zapisujemy „3!” i wynosi ona : $1 * 2 * 3 = 6$. Jednak ta liczba szybko rośnie, gdyż silnia z liczby 8 wynosi aż 40 320, a silnia z liczby 100 składa się aż ze 157 cyfr! Silnię można liczyć tylko dla liczb naturalnych, czyli liczb całkowitych większych od zera. Wyjątkiem jest samo zero, ponieważ ma ono swoją silnię. Została ona przyjęta jako 1. Silnia jest funkcją pozwalającą zapisać w skondensowany sposób wzory i zależności pojawiające się w różnych działach matematyki od analizy matematycznej (np. mianownik każdego składnika wzoru Taylora ma postać k!) przez geometrię n-wymiarową (np. stosunek miary n-wymiarowego równoległoscianu do miary sympleksu rozpiętego na wszystkich wierzchołkach równoległoscianu z wyjątkiem jednego jest równy n!), na kombinatoryce skończywszy (np. liczba wszystkich permutacji zbioru n-elementowego jest równa n!).⁽¹⁾ Silnia jest głównie wykorzystywana przy liczeniu rachunków prawdopodobieństwa. Znaleźć ją można w elementach kombinatoryki. Silnia jest używana w permutacji (Ciąg utworzony ze wszystkich n elementów zbioru nazywamy jego permutacją. Liczbę wszystkich permutacji danego n-elementowego zbioru obliczamy wg wzoru $P_n=n!$)⁽²⁾. Oprócz permutacji, silnia ma swoje zastosowanie również w wariacjach bez powtórzeń, które są również elementem kombinatoryki (Wariacją bez powtórzeń nazywamy ciąg k wyrazów, nie powtarzających się, które są elementami danego zbioru o liczności n. Ilość wszystkich wariacji obliczamy ze wzoru $V_n^k = n!/(n-k)!$)⁽³⁾. Kiedy silnia jest rozpisana można zauważyć z jakich czynników składa się liczba z której jest ona liczona, znajomość takiego rozkładu jest przydatna zwłaszcza przy skracaniu ułamków (w sytuacji gdy w liczniku i mianowniku występują silnie). Okazuje się, że silnia ma również inne zastosowanie niż czysto matematyczne. Za jej pomocą można wyliczyć ile jest kombinacji na przykład ustawienia jakiegoś przedmiotu, lub ile jest możliwości wyboru obiektów spośród zbioru obiektów. To ostatnie jest zwłaszcza przydatne gdy na przykład potrzebujesz się spakować i masz „n” par spodni w szafie, a możesz spakować tylko „k”. Aby rozwikłać to jakże trudne zagadnienie wystarczy zastosować wzór $n!/(k!(n-k)!)$. Wygląda na to, że silnia może być również przydatna w życiu codziennym. Ciekawostką na temat silni są liczby, które w języku angielskim nazywają się „factorion”. Ich nietypowość polega na tym, że jeśli dodać silnie wszystkich cyfr zapisu dziesiętnego to otrzyma się początkową liczbę. „Factorionem” są oczywiście cyfry 1 oraz 2. Oprócz nich tylko dwie inne liczby spełniają ten warunek i są nimi 145 oraz 40 585.

Silnia - iteracja



Obraz nr 1

przedstawia algorytm liczenia silni zapisany za pomocą schematu blokowego

Silnia - iteracja

Na powyższym obrazie przedstawiono sposób działania algorytmu liczącego silnię przy pomocy schematu blokowego. Taki zapis pozwala na łatwiejsze ukazanie sposobu jego działania. Jak widać, algorytm na początku pyta o „n”, z którego ma wyliczyć silnię (n!). Jak zostało to wspomniane wcześniej, silnię można liczyć tylko z liczb dodatnich, a dla liczby 0 wynosi 1. Dlatego algorytm sprawdza, czy „n” wynosi więcej niż 0. Jeśli to się zgadza, algorytm tworzy dwie zmienne. Jedna z nich na końcu staje się wynikiem, a druga służy do sprawdzania, czy działanie zostało już zakończone (sprawdza, czy „licznik=n”). Na końcu wyświetla wynik, który jest silnią z wartości „n”.

PSEUDO- KOT

- Wczytaj n
- JEŻELI $n < 0$ IDŹ DO 1
- LICZNIK = 0, SILNIA = 1
- JEŻELI LICZNIK \neq n WTEDY
POWIĘKSZ LICZNIK O 1,
SILNIA = SILNIA * LICZNIK, IDŹ DO 4
- WYPISZ SILNIA

Obraz nr 2

przedstawia algorytm liczenia silni zapisany za pomocą pseudokodu

Silnia - iteracja

Powyższy obraz przedstawia algorytm liczący silnię zapisany w formie pseudokodu. Ukazuje on to samo co schemat blokowy, czyli sposób działania algorytmu. Jednak tutaj jest on przedstawiony bardziej przy pomocy słów, co sprawia, że łatwiej go zrozumieć osobom, które mniej znają się na programowaniu.

```
func silnia(_ liczba: Int)->Int
{
    var wynik = 1
    for x in 2...liczba
    {
        wynik=wynik*x
    }
    return wynik
}
```

```
print("Podaj silnia ile chcesz policzyc")
var ile=Int(readLine()!)
if ile=0 {
    print("1")
} else {
    print(silnia(ile))
}
```

Obraz nr 3

przedstawia algorytm liczenia silni zapisany w języku Swift

Silnia - iteracja

Na powyższym obrazie wdać zapis omawianego algorytmu w języku programowania Swift. Algorytm wykonuje zadania przedstawione w schemacie blokowym i pseudokodzie. W pierwszej kolejności stworzona zostaje funkcja, pod którą następnie podstawia się zmienną „ile”.

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {
    unsigned int n;
    cout<<"Podaj n: ";
    cin>>n;
    unsigned int i=1, wynik=1;
    while(i<=n){
        wynik=wynik*i;
        i++;
    }
    cout<<n<<"! = "<<s<<endl;
}
```

Obraz nr 4

przedstawia algorytm liczenia silni zapisany w języku C++

Silnia - iteracja

Ten algorytm wykonuje to samo, co zapisany w języku Swift, lecz w nieco inny sposób. W tym wypadku nie zostaje stworzona funkcja, tylko program pyta o „n”, z którego następnie wylicza bezpośrednio silnię. Mimo innego zapisu, wynik zawsze będzie taki sam jak w języku Swift.

```
import java.util.Scanner
public class Silnia_Java {
    public static void main (String[] args) {
        int n;
        Scanner odczyt=new Scanner(System.in);
        System.out.print("n = ");
        n=odczyt.nextInt();
        int i=1, s=1;
        while(i<=n){
            s=s*i;
            i++;
        }
        System.out.print(n);
        System.out.print("! = ");
        System.out.println(s);
    }
}
```

Obraz nr 5

przedstawia algorytm liczenia silni zapisany w języku Java

Silnia - iteracja

Mimo, iż ten algorytm wygląda na najbardziej skomplikowany z wszystkich przedstawionych, nie robi on nic więcej niż pozostałe. Jego zadaniem jest wyliczenie silni i robi to pytając o zmienną „n”, a następnie wyliczając z niej silnię i wyświetlając ją w formie : „n! = s”, gdzie „s” jest wynikiem, a „n” – liczbą, z której wylicza silnię.

```
def silnia_iteracja(n):  
  
    wynik=1  
    if n in (0,1):  
        return 1  
    else:  
        for i in range(2,n+1):  
            wynik=wynik*i  
  
    return wynik
```

Obraz nr 6

przedstawia algorytm liczenia silni zapisany w języku Python

Na obrazie nr 6 widać sposób zapisania omawianego algorytmu w języku Python. Zapis ten nie różni się wiele od zapisu w formie pseudokodu czy schematu blokowego poza tym, że użyte zostały specjalne sformułowania używane w tym języku. Jest to najprostszy z przedstawionych algorytmów, gdyż język Python umożliwia zapisanie wielu rzeczy w bardzo

Silnia - iteracja

prosty sposób.

Źródła :

1. Informacja ze strony : <https://pl.wikipedia.org/wiki/Silnia>
2. Informacja ze strony :
https://pl.wikibooks.org/wiki/Matematyka_dla_liceum/Rachunek_prawdopodobie%C5%84stwa/Elementy_kombinatoryki
3. Informacja ze strony :
https://pl.wikibooks.org/wiki/Matematyka_dla_liceum/Rachunek_prawdopodobie%C5%84stwa/Elementy_kombinatoryki

Spis zdjęć, rysunków, tabel i wykresów :

1. Obraz przedstawiający algorytm liczenia silni zapisany za pomocą schematu blokowego
zdjęcie pobrane ze strony <http://www.algorytm.org/kurs-algorytmiki/schematy-blokowe.html>
dnia 29.04.2018 roku
2. Obraz przedstawiający algorytm liczenia silni zapisany za pomocą pseudokodu
rysunek własny
3. Obraz przedstawiający algorytm liczenia silni zapisany w języku Swift
zrzut ekranu - wykonanie własne
4. Obraz przedstawiający algorytm liczenia silni zapisany w języku C++
zrzut ekranu - wykonanie własne
5. Obraz przedstawiający algorytm liczenia silni zapisany w języku Java
zrzut ekranu - wykonanie własne
6. Obraz przedstawiający algorytm liczenia silni zapisany w języku Python
zrzut ekranu - wykonanie własne

Silnia - iteracja

Wnioski :

Przygotowując nasz projekt, osobiście zgłębiliśmy się w temat silni. Na podstawie własnych odczuć jesteśmy w stanie zgodzić się z częścią tezy zakładającą, że zrozumienie tego zagadnienia nie należy do niemożliwych. Jednakże całkowite potwierdzenie owej tezy wymagało od nas dodatkowych działań. Należało bowiem wytłumaczyć, czym jest silnia, komuś innemu. Grupą doświadczalną służącą do tego celu została połowa naszej klasy. Podeszli do tematu z zaangażowaniem i zainteresowaniem, zaciekawieni słuchali tego, co mieliśmy im do przekazania. Przeprowadziliśmy dla nich około dwudziestopięć minutową lekcję na temat silni. Po osiągnięciu sukcesu doszliśmy do wniosku, iż można się zgodzić z całą tezą naszego projektu, czyli, że zagadnienie będące tematem pracy jest możliwe do zrozumienia i wytłumaczenia. Aby łatwiej im było sobie wyobrazić w jaki sposób działa algorytm, przedstawiliśmy go na początku za pomocą schematu blokowego, a następnie słownie został przeczytany pseudokod, był on jednocześnie przyrównywany ze schematem blokowym widniejącym na tablicy, tak aby każdy mógł zauważyć, że obie rzeczy odnoszą się do silni. Chcąc upewnić się, czy nasz mały wykład został na pewno dobrze zrozumiany, pod koniec zadaliśmy klasie kilka pytań. Ku naszemu zadowoleniu, uczniowie odpowiadali chętnie i poprawnie na wszystkie nasze pytania. Postanowiliśmy zatem rzucić im wyzwanie - poprosiliśmy ich o napisanie algorytmu liczącego silnię w języku Scratch. Po kilku minutach zażartej walki z programem jednej z naszych koleżanek udało się odnieść sukces - ukończyła algorytm, a za swoje chęci i wysiłek została hojnie nagrodzona przez naszego nauczyciela informatyki, pana Dominika Siecińskiego, oceną celującą. Zapoznawszy się z wynikami pracy pozostałych uczniów, z radością odkryliśmy, że wielu z nich niemalże udało się ukończyć pracę, co było dla nas bardzo miłym zaskoczeniem. Osiągnięty sukces pozwolił naszym myślą wybiec w przyszłość, zastanawiając się nad wyborem kariery nauczycielskiej, skoro zostaliśmy wysłuchani i rozumiani w przypadku tłumaczenia tak zawilego i skomplikowanego zagadnienia, jakim jest silnia. Byliśmy dumni, że temat został zrozumiany oraz, że nasze starania wzbogaciły naszych kolegów i koleżanki o nową wiedzę. Podsumowując, możemy w pełni zgodzić się z tematem pracy, gdyż udało się nam go zrozumieć, oraz wytłumaczyć i rozumianie przekazać naszym rówieśnikom.

Silnia - iteracja

Dodatek :

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