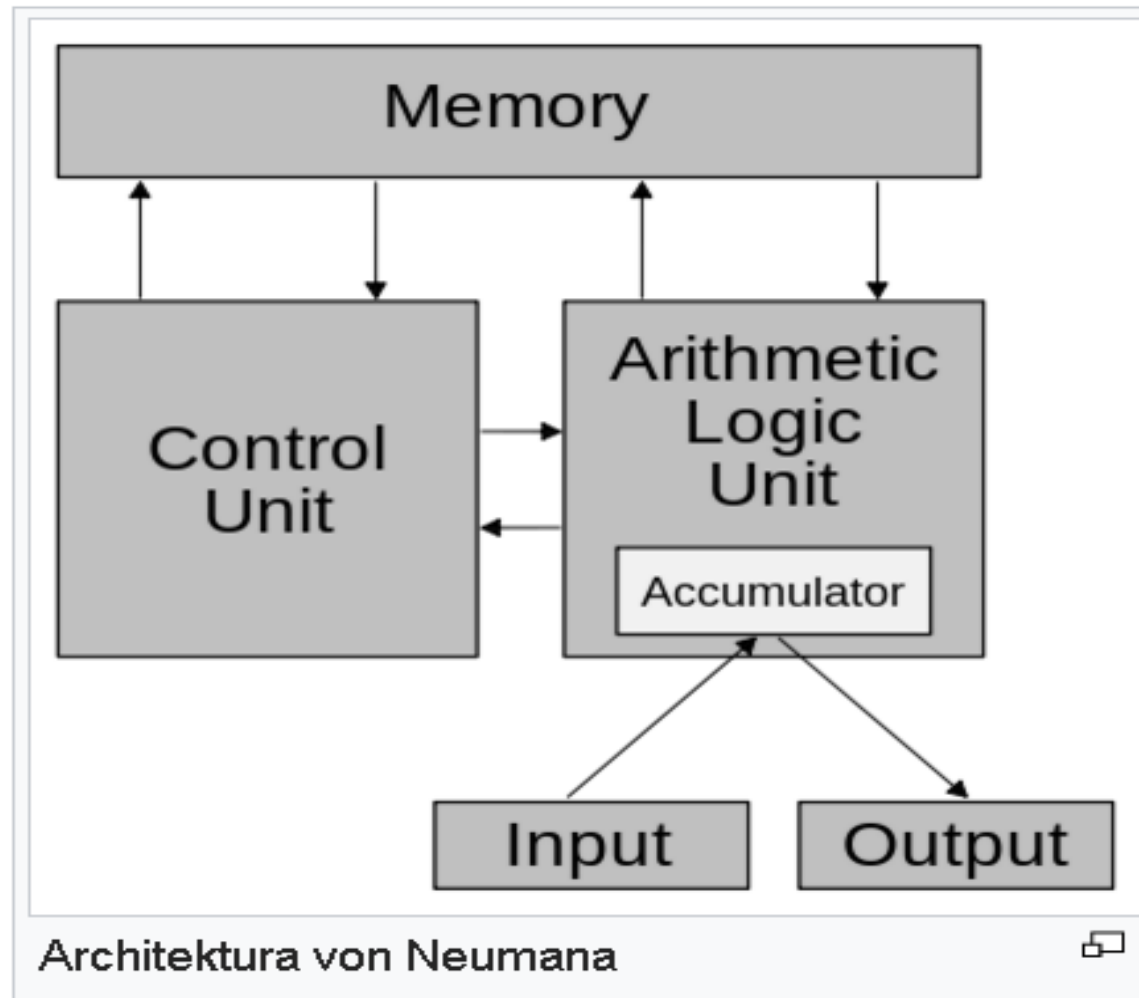


PODSTAWY PROGRAMOWANIA PLC

Program jest sekwencyjnym ciągiem instrukcji
(sekwencja poleceń krok po kroku zmienia stan maszyny)

rzeczywiste komputery działają zgodnie z architekturą von Neumanna



Zmienne jako reprezentacja pamięci

zmienne stanowią abstrakcję komórek pamięci

Charakterystyka zmiennych:

1.nazwa

2.adres

3.wartość

4.typ

5.okres życia

6.zakres widoczności

Adresy zmiennych

program może zawierać różne zmienne o tych samych nazwach, jeśli są umieszczone w różnych podprogramach ta sama zmienna może mieć różne adresy w trakcie różnych wywołań podprogramu

Typ zmiennej

typ to zbiór dopuszczalnych wartości, jakie zmienna może przyjmować w dowolnym momencie wykonania programu typ może definiować precyzję reprezentacji zmiennych (np. dla liczb zmiennoprzecinkowych)

typ definiuje semantykę operatorów, jakie można stosować do zmiennych

Maszyny przechowują dane w postaci liczb binarnych bit (b): pojedyncza cyfra reprezentuje dwa stany: 0 i 1, jest to najmniejsza niepodzielna jednostka informacji cyfrowej bity grupuje się w bajty (B), najczęściej $1B=8b$

nazwa	wartość	użycie	nazwa
kilobajt (kB)	10^3	2^{10}	kibibajt (KiB)
megabajt (MB)	10^6	2^{20}	mebibajt (MiB)
gigabajt (GB)	10^9	2^{30}	gibibajt (GiB)
terabajt (TB)	10^{12}	2^{40}	tebibajt (TiB)
petabajt (PB)	10^{15}	2^{50}	pebibajt (PiB)
eksabajt (EB)	10^{18}	2^{60}	eksbibajt (EiB)
zettabajt (ZB)	10^{21}	2^{70}	zebibajt (ZiB)
jottabajt (YB)	10^{24}	2^{80}	jobibajt (YiB)

Konwersja liczb dziesiętnych do systemu binarnego

Algorytm

dziel liczbę dziesiętną przez 2 zapamiętując reszty z dzielenia
zapisz reszty w odwrotnej kolejności

liczba dziesiętna	wynik dzielenia	reszta
2436	1218	0
1218	609	0
609	304	1
304	152	0
152	76	0
76	38	0
38	19	0
19	8	1
8	4	0
4	2	0
2	1	0
1	0	1



$$2436_{(10)} = 100010000100_{(2)}$$

System szesnastkowy (heksadecymalny)

Podstawą systemu jest liczba 16

liczby zapisywane są za pomocą cyfr 0, 1, ..., 9 i liter A, B, C, D, E, F literom odpowiadają liczby A(10), B(11), C(12), ..., F(15)

kod heksadecymalny bardzo łatwo konwertuje się do kodu binarnego

$$\boxed{1100} \boxed{0010} \boxed{1010} \boxed{0111}_{(2)} = C2A7_{(16)}$$

Conversion steps shown:

- $1100_{(2)} = 12_{(10)} = C_{(16)}$
- $0010_{(2)} = 2_{(10)} = 2_{(16)}$
- $1010_{(2)} = 10_{(10)} = A_{(16)}$
- $0111_{(2)} = 7_{(10)} = 7_{(16)}$

Reprezentacja liczb całkowitych

Wcześniejsze kody umożliwiały przechowywanie liczb naturalnych, w celu zapisania liczb całkowitych należy posłużyć się dodatkowym mechanizmem reprezentacja znak-moduł: (ZM) najbardziej znaczący bit informuje o znaku liczby (0 - liczba dodatnia, 1- liczba ujemna).

Reprezentacja uzupełnienia do dwóch (U2): waga najbardziej znaczącego bitu przyjmuje wartość dodatnią (0) lub ujemną (1), pozostałe wagi zawsze są dodatnie.

Reprezentacja znak-moduł

przykład

znak	64	32	16	8	4	2	1	
0	1	0	1	1	0	0	1	$+89_{(10)}$
znak	64	32	16	8	4	2	1	
1	0	0	0	1	1	0	1	$-13_{(10)}$

Wady:

algorytmy dodawania i odejmowania liczb binarnych muszą uwzględniać zarówno znak, jak i moduł
w reprezentacji występują dwie różne liczby 0

Reprezentacja uzupełnienia do dwóch

przykład

128	64	32	16	8	4	2	1	
0	1	0	1	1	0	0	1	$+89_{(10)}$
-128	64	32	16	8	4	2	1	
1	0	0	0	1	1	0	1	$-115_{(10)}$

Zakres:

na n bitach można zapisać liczby $\langle -2^{n-1}, 2^{n-1}-1 \rangle$

dla 8 bitów: $\langle -128, 127 \rangle$

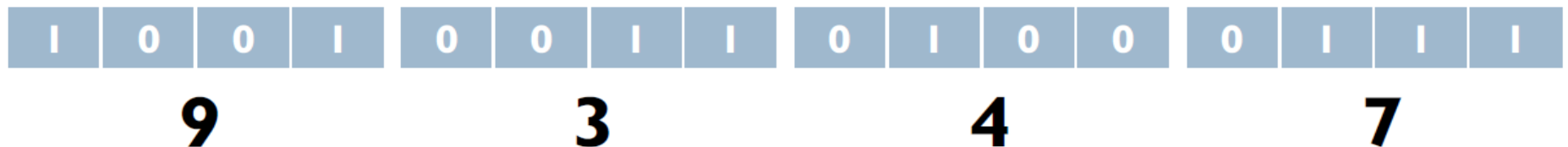
dla 16 bitów: $\langle -32768, 32767 \rangle$

Kod BCD

W wielu przypadkach stosowanie "czystego" kodu binarnego jest nieefektywne ze względu na dużą liczbę konwersji między kodem binarnym i dziesiętnym

BCD (ang. binary coded decimal) to system dziesiętny kodowany dwójkowo

każda cyfra dziesiętna kodowana binarnie na 4 bitach
liczba w kodzie BCD dzielona na 4-bitowe bloki
dekodowane niezależnie



Co to jest algorytm?

... przepis postępowania który prowadzi do uzyskania rozwiązania zadania w skończonej liczbie kroków



przepis kucharski



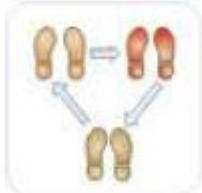
zapis nutowy muzyki



instrukcja montażu mebla



opis dojazdu



układ choreograficzny

Cechy algorytmu

Sposób zapisu:

- język naturalny
- schemat blokowy
- instrukcje programu
- pseudokod
- ...

Sposób zapisu musi pozwalać na wyrażenie

- sekwencyjności
- wyboru
- iteracji
- rekursji

Sekwencyjność

Sekwencyjność odnosi się do konieczności wykonania kroków algorytmu zgodnie z kolejnością ich zapisu

Wybór

opis algorytmu może zawierać wiele alternatywnych ścieżek wiodących do celu

Iteracja

iteracja pozwala powtórzyć fragment algorytmu określoną liczbę razy lub powtarzać fragment algorytmu aż do momentu zajścia pewnego warunku

Rekurencja

rekurencja oznacza podział problemu na mniejsze zagnieżdżone podproblemy o identycznej strukturze i rozwiązanie podproblemów przez algorytm nadrzędny

Rekurencja, zwana także rekursją (ang. recursion, z łac. recurrere, przybiec z powrotem) – odwoływanie się np. funkcji lub definicji do samej siebie.

Pisanie algorytmów

Czy rozumiemy, co potrafi zrobić PLC?

Podjęcie redukcjonistyczne

Podziel problem na mniejsze i prostsze podproblemy.

Zaprojektuj rozwiązania dla poszczególnych modułów, w razie konieczności podziel podproblem na jeszcze mniejsze elementy i zapisz algorytm w wybranym języku programowania.

Top-down design, od ogółu do szczegółu.

Programowanie jest jedno, jest tylko wiele języków

Cechy algorytmu

Skończoność

algorytm zapewnia osiągnięcie rozwiązania w skończonej liczbie kroków (czyli w skończonym czasie),

dokładna liczba kroków nie jest znana a priori,

algorytm posiada warunek zakończenia.

Jednoznaczność

dla tych samych danych wejściowych algorytm musi dawać zawsze te same wyniki,

algorytm jest niezależny od momentu jego uruchomienia, wpływu innych programów, sprzętu, kodowania znaków, itp.

Z czego składa się algorytm?

Dane

obiekty podlegające przekształceniu podczas działania

Wynik

ostateczny rezultat działania algorytmu

Instrukcje

opis sekwencji czynności które muszą zostać wykonane w ściśle określonej kolejności

Sposoby zapisu algorytmu

Lista kroków (opis słowny)

najbardziej naturalny sposób zapisu algorytmu

Zapis graficzny (schemat blokowy)

zestaw symbolicznych elementów
odpowiadających czynnościom w pseudojęzyku
programowania

W konkretnym języku programowania

Przykład listy kroków

1. Czy słuchawka jest odłożona? Jeśli tak, to przejdź do (2), jeśli nie, to odłóż słuchawkę.
2. Podnieś słuchawkę.
3. Wybierz cyfrę 6.
4. Wybierz cyfrę 1.
5. Wybierz cyfrę 6.
6. Wykonaj czynność cztery razy, wybierz cyfrę 2.
7. Czy połączyłeś się z fryzjerem? Jeśli tak, to przejdź do (8), w przeciwnym wypadku przejdź do (9).
8. Umów się na strzyżenie włosów.
9. Odłóż słuchawkę.







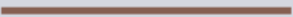
Schematy blokowe

Dla skomplikowanych algorytmów zapis w postaci języka naturalnego szybko staje się nieczytelny i zagmatwany schemat blokowy to graficzny zapis algorytmu operacje przedstawione są za pomocą odpowiednio połączonych bloków.

Połączenia określają kolejność i sposób wykonywania operacji.

Istnieje standardowy, ogólnie przyjęty system oznaczeń graficznych dla schematów blokowych.

Symbole używane w schematach blokowych

symbol	nazwa	opis
	początek, koniec	oznacza miejsce rozpoczęcia i zakończenia algorytmu
	operator	działanie do wykonania
	wejście/wyjście	wprowadzanie danych lub wyświetlanie danych
	element decyzyjny	wybór jednej z alternatywnych dróg działania
	łącznik	połączenie dwóch fragmentów sieci działania
	komentarz	miejsce na komentarz
	linia	połączenie kolejnych symboli działania

Operacja wyboru

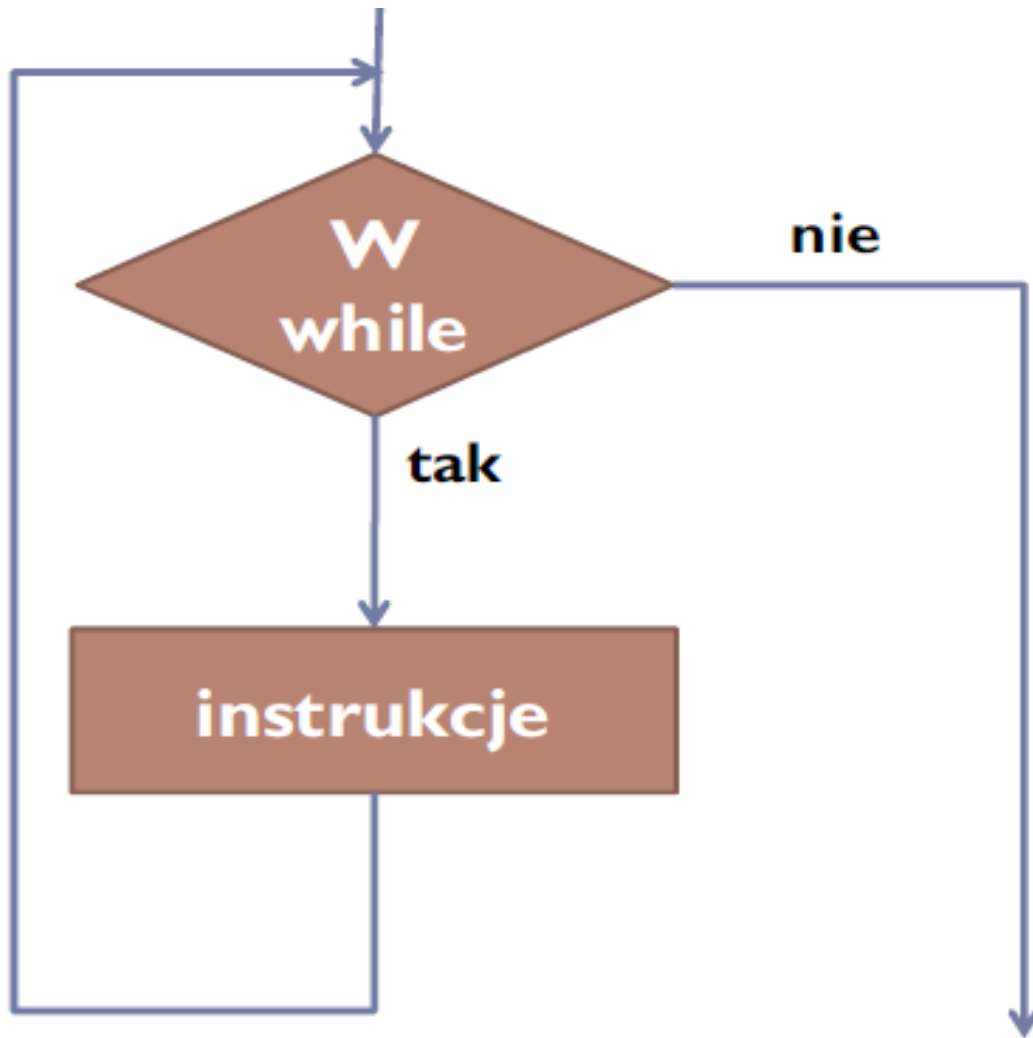
na schemacie blokowym operacje wyboru jednej z alternatywnych dróg działania realizujemy za pomocą skrzynki warunkowej.

wewnątrz skrzynki warunkowej umieszczamy warunek Logiczny

Operatory: =; <; >; < >; <=; >= ; OR; AND

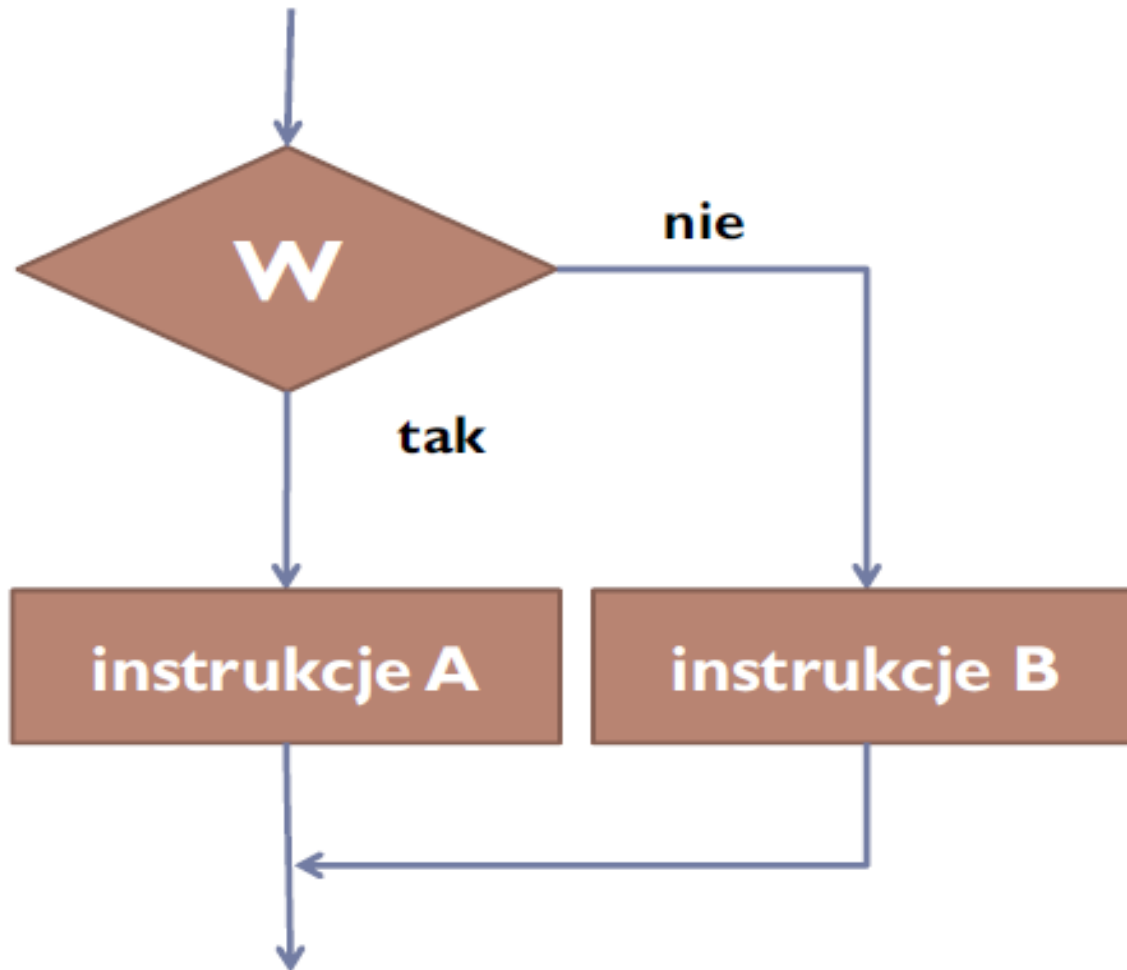


Instrukcja pętli "dopóki"



dopóki spełniony
jest warunek W
to
instrukcje są
powtarzane

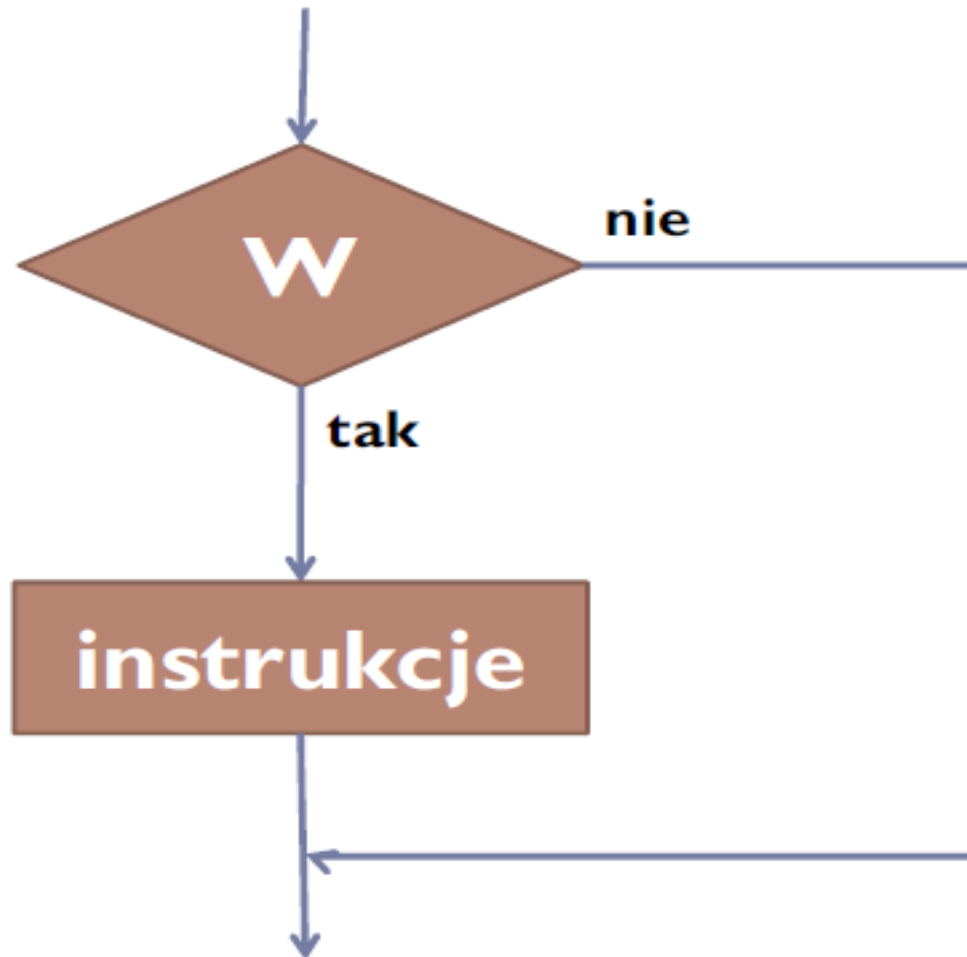
Instrukcja warunkowa



Jeśli spełniony jest warunek W to realizowane są instrukcje A,

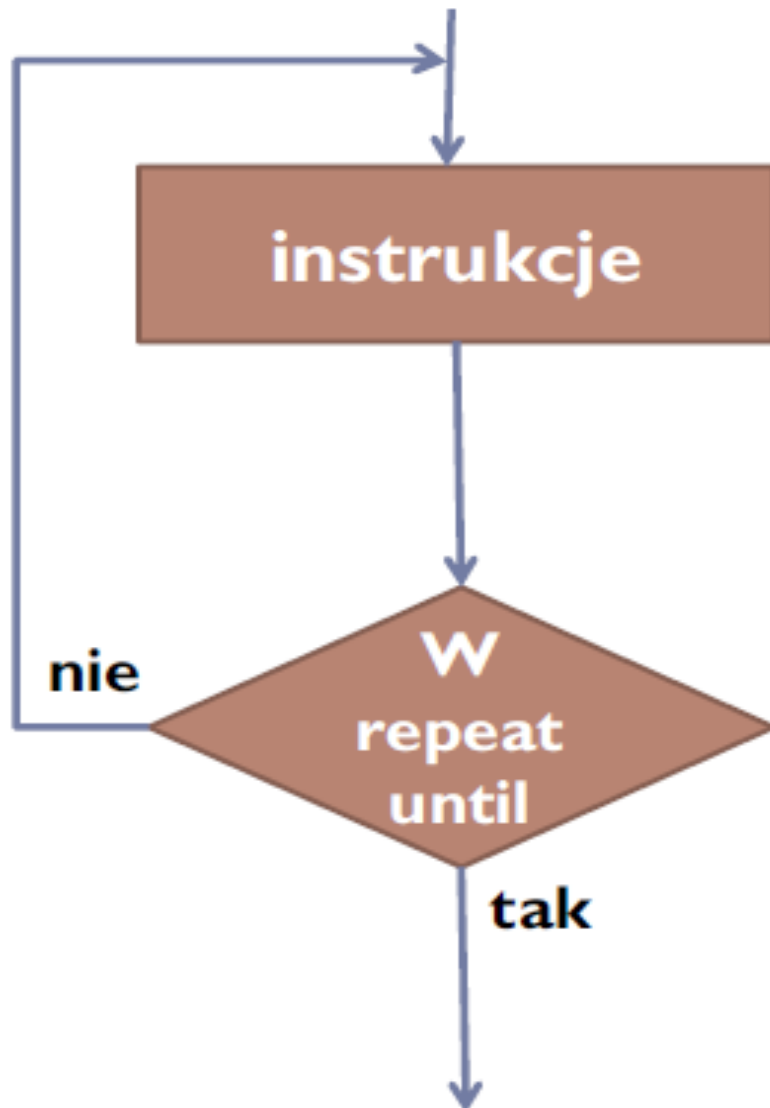
W przeciwnym Wypadku realizowane są instrukcje B

Instrukcja warunkowa



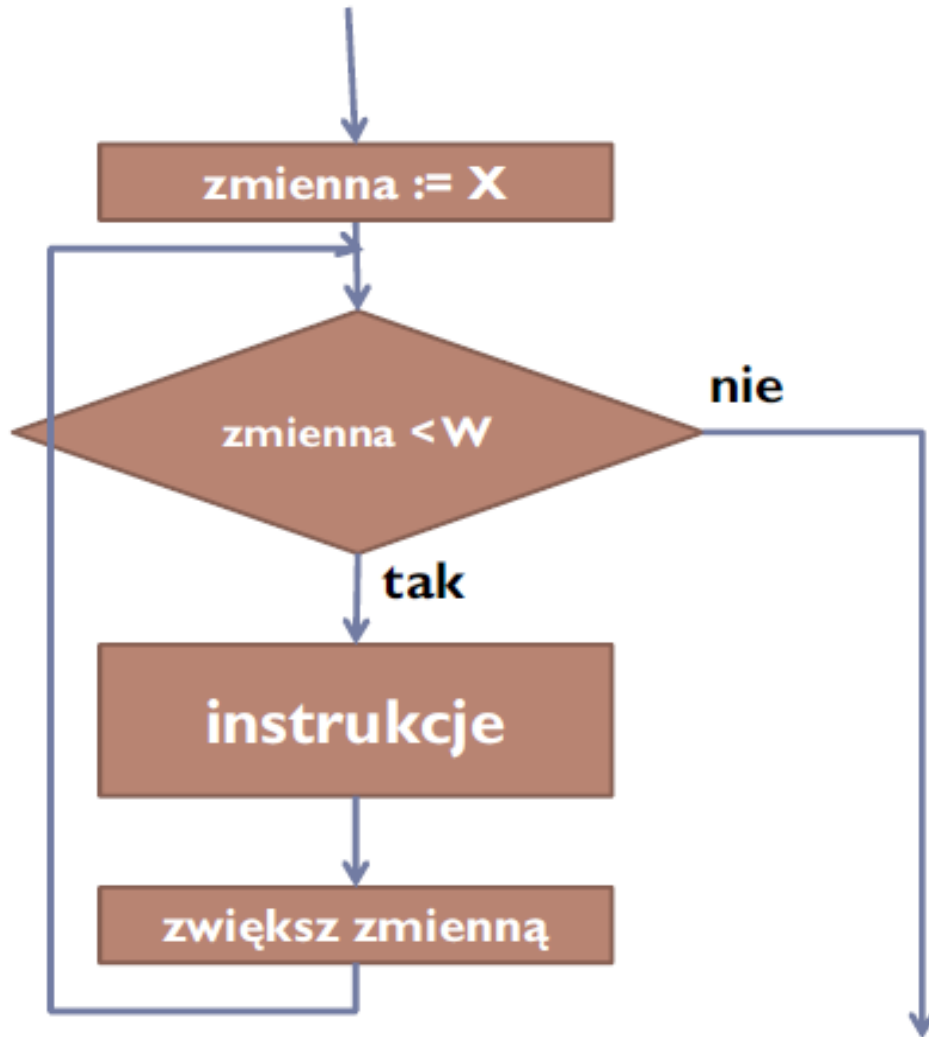
instrukcje są
realizowane jeśli
spełniony jest
warunek W

Instrukcja pętli "powtarzaj aż"



instrukcje są
powtarzane aż do
spełnienia czy jest
warunek W

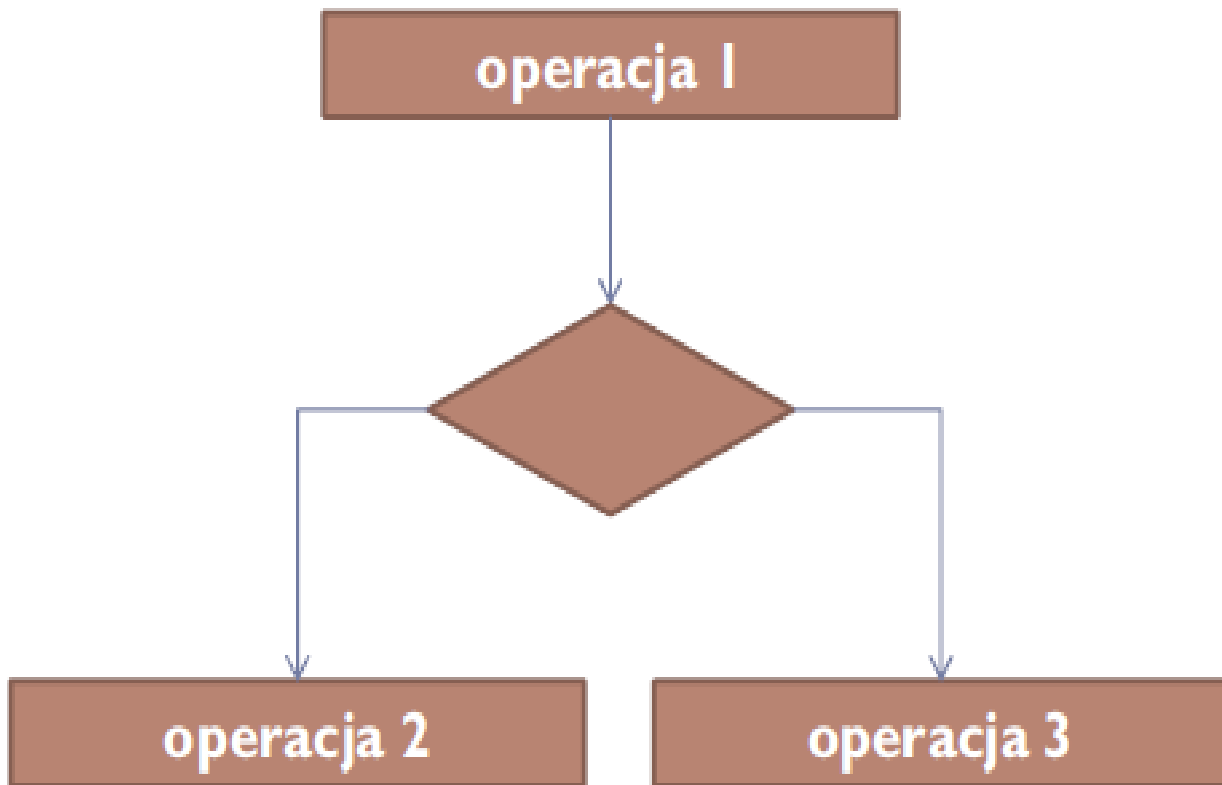
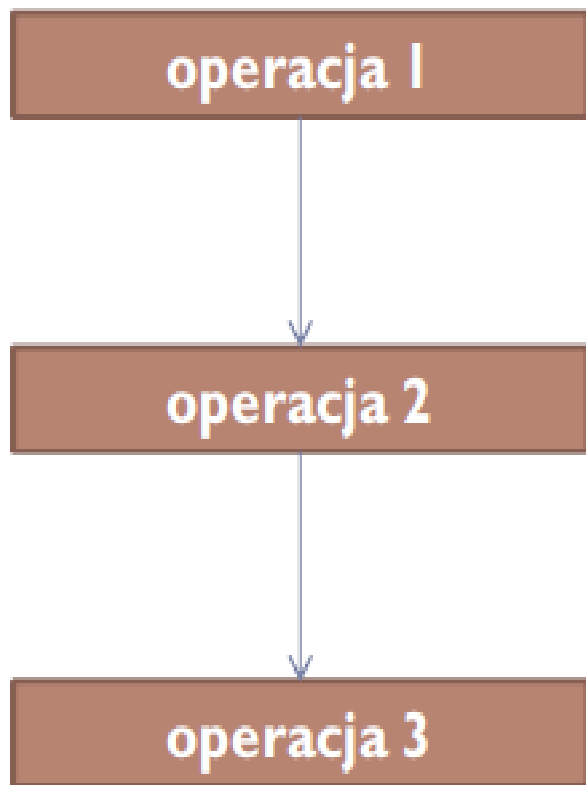
Instrukcja pętli "powtarzaj n razy"



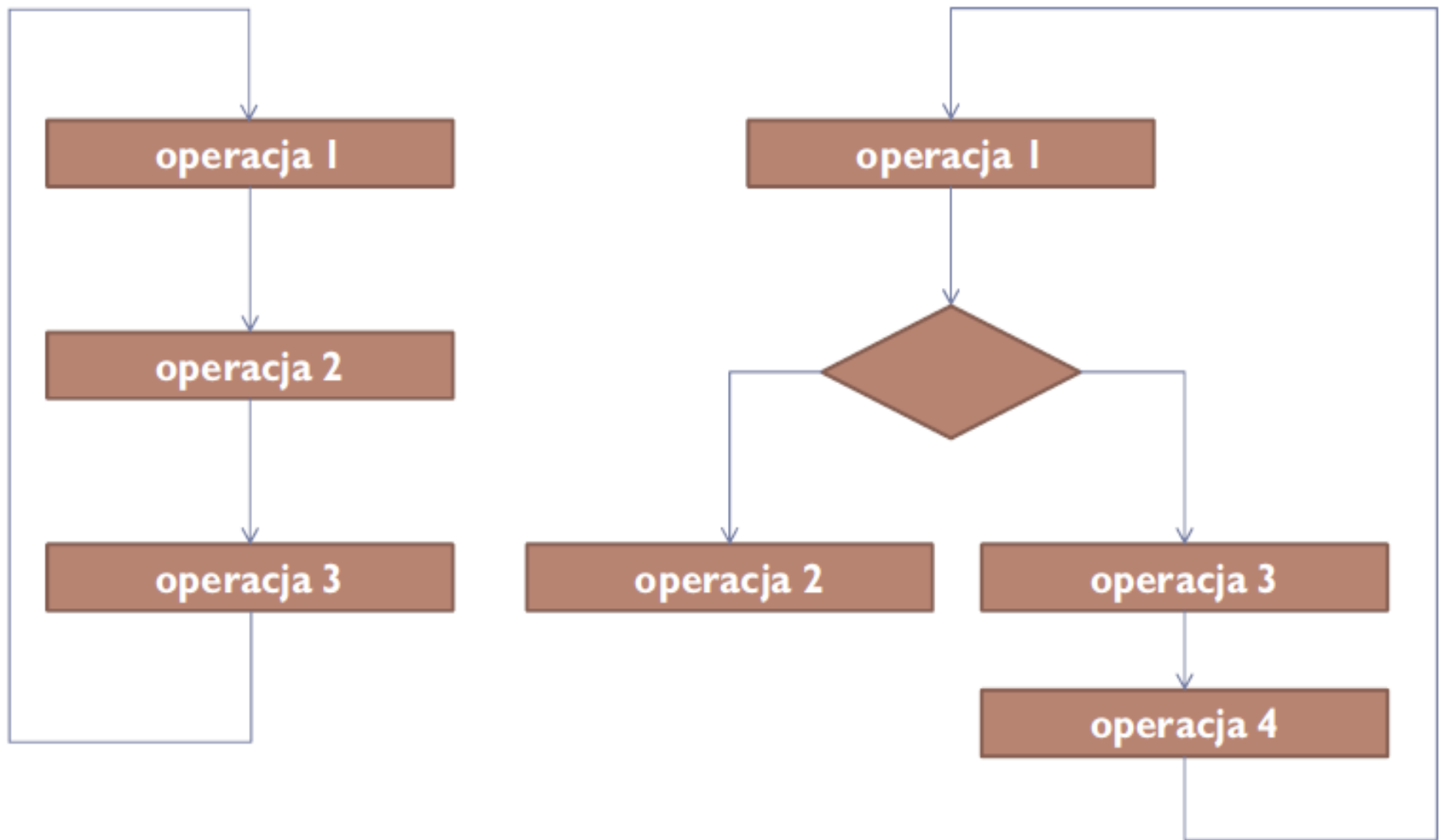
zainicjalizuj zmienną za pomocą wartości
powtarzaj określoną liczbę razy
 $zmienna < W$

$zmienna := X$
jeśli zmienna jest poniżej warunku zmiennej kontrolnej, to wejdź do pętli
jeśli warunek jest fałszywy, to następuje koniec przetwarzania

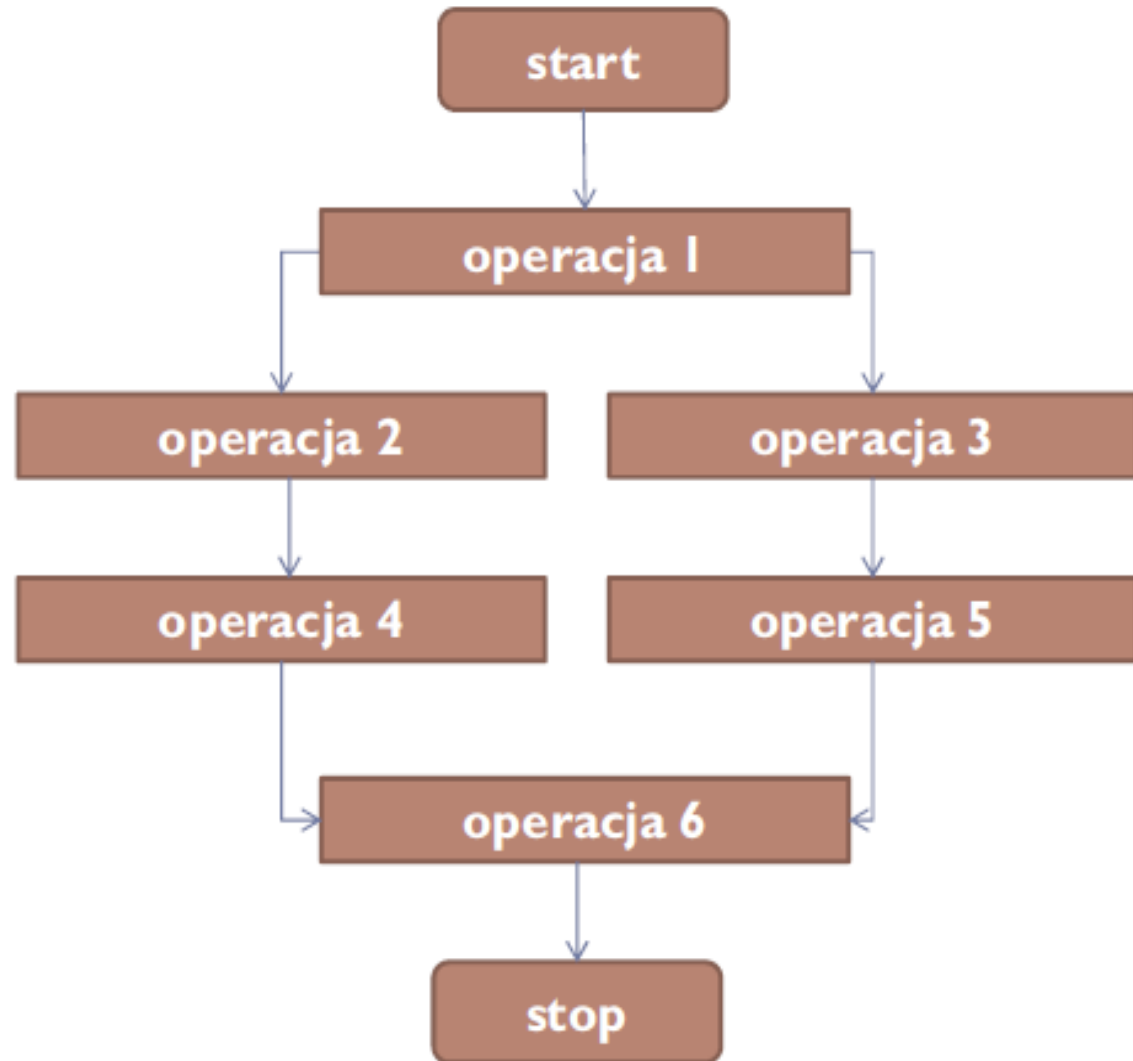
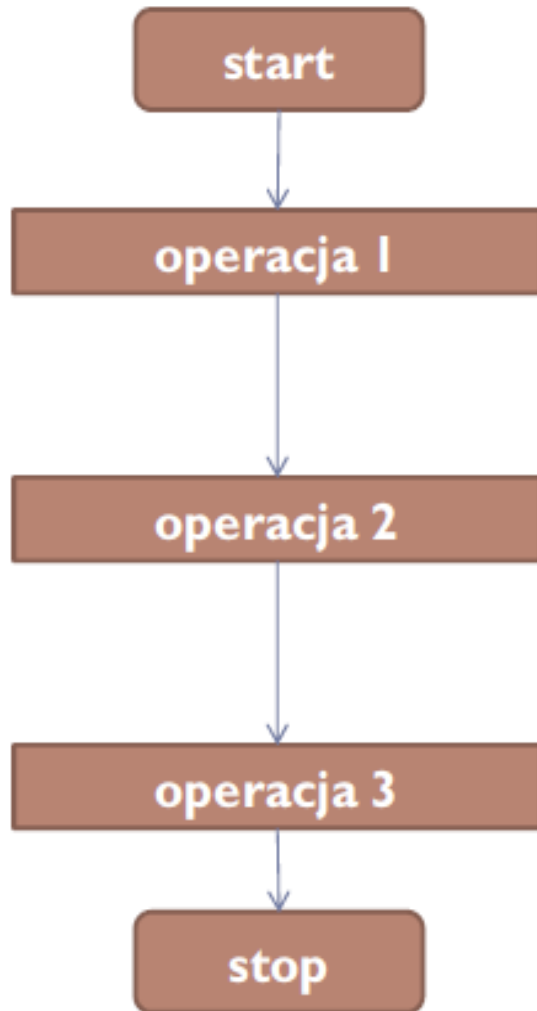
Algorytm prosty i rozgałęziony



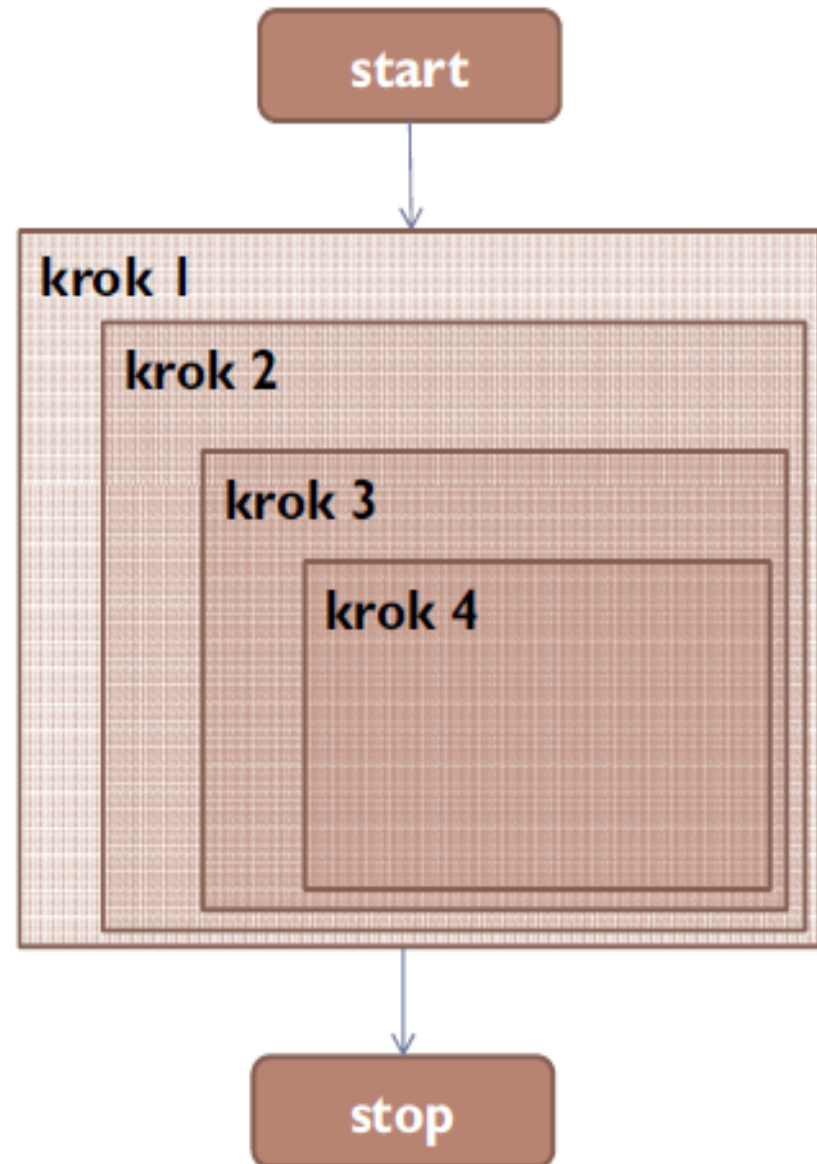
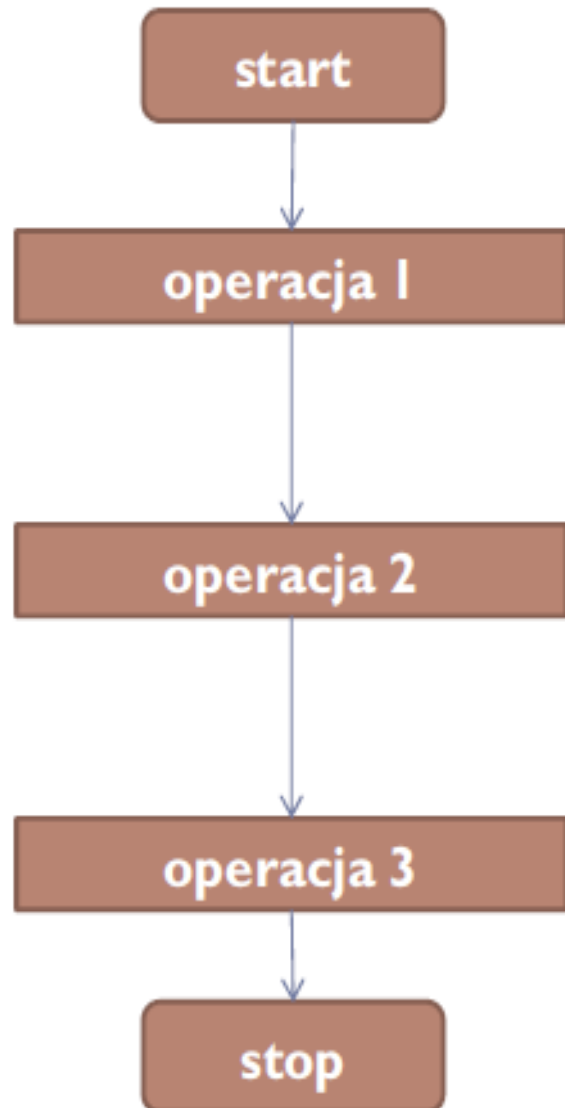
Algorytm cykliczny i mieszany



Algorytm sekwencyjny i równoległy



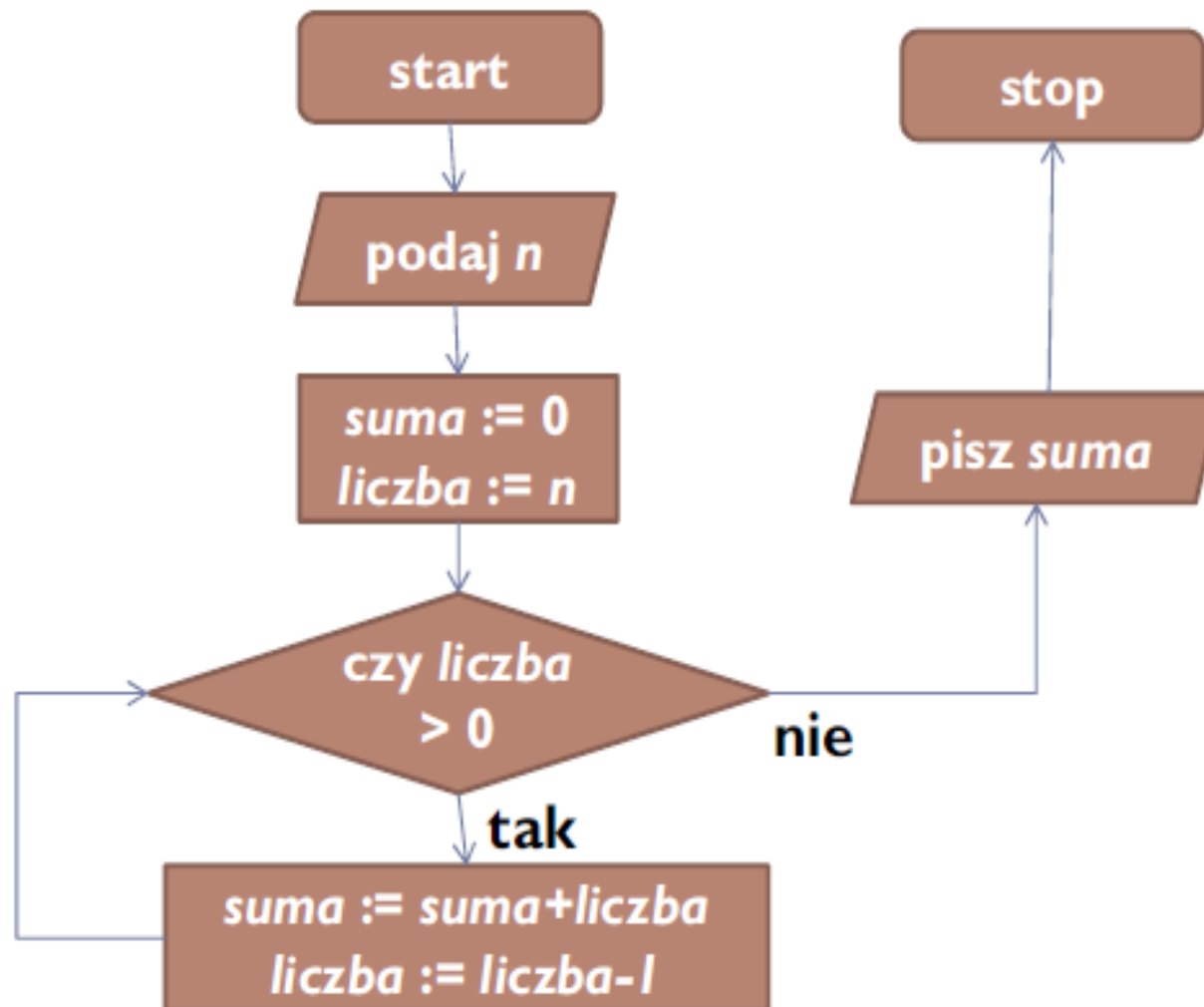
Algorytm iteracyjny i rekurencyjny



Przegląd podstawowych algorytmów

Zadanie:

zsumuj liczby naturalne z zakresu 1.. n

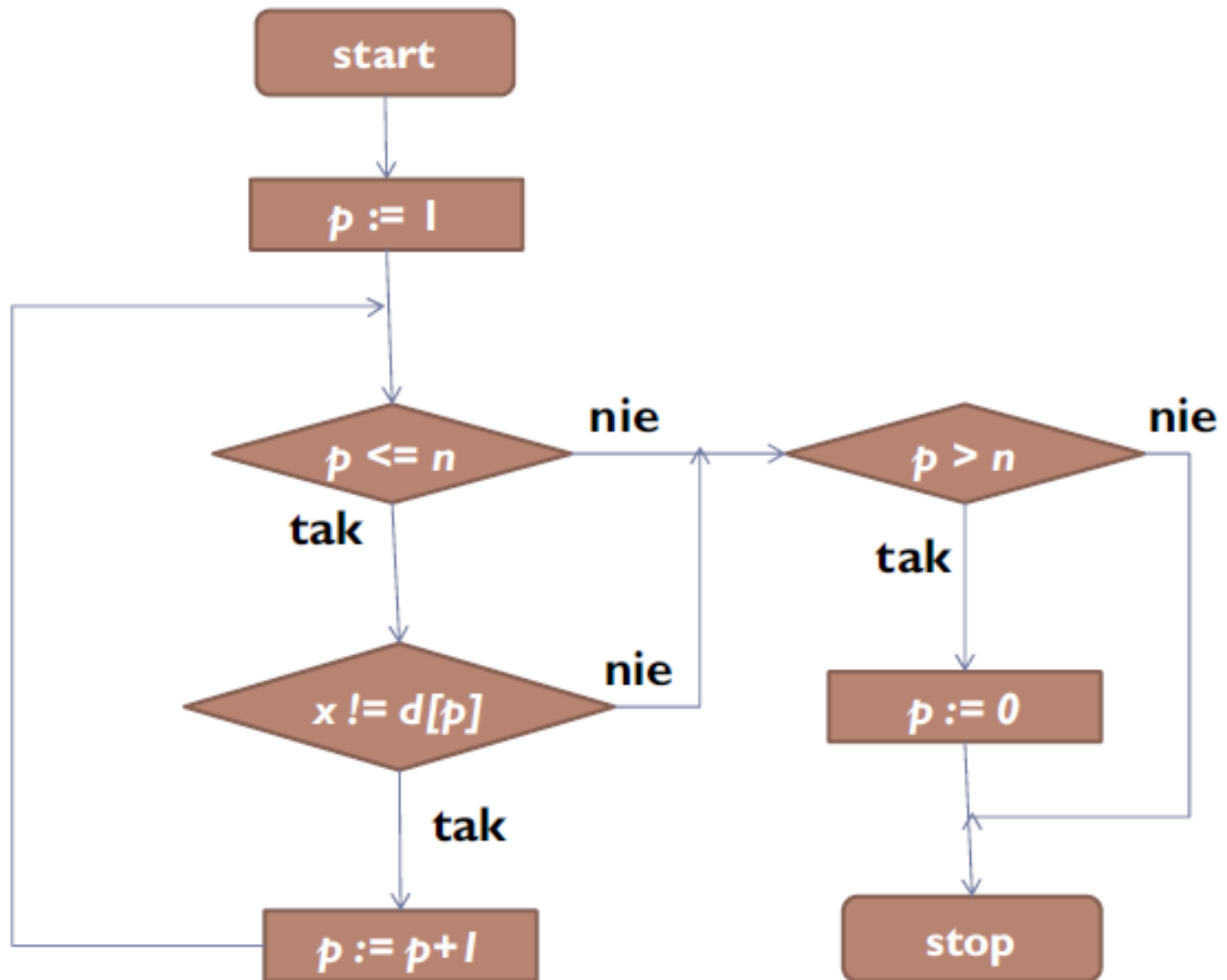


Przeszukiwanie z wartownikiem

generalna zasada wyszukiwania

warunek sprawdzania zakresu ($p \leq n$) jest potrzebny jedynie w przypadku, gdy zbior $d[.]$ nie zawiera poszukiwanego elementu (gwarancja zakończenia pętli) jeśli można zagwarantować, że element zawsze zostanie znaleziony, to warunek staje się zbędny na końcu zbioru $d[.]$ dodajemy poszukiwany element x . Jeśli algorytm zwróci w wyniku pozycję $n+1$, to wiemy, że znaleziony został wartownik (przeszukiwanie jest nieudane).

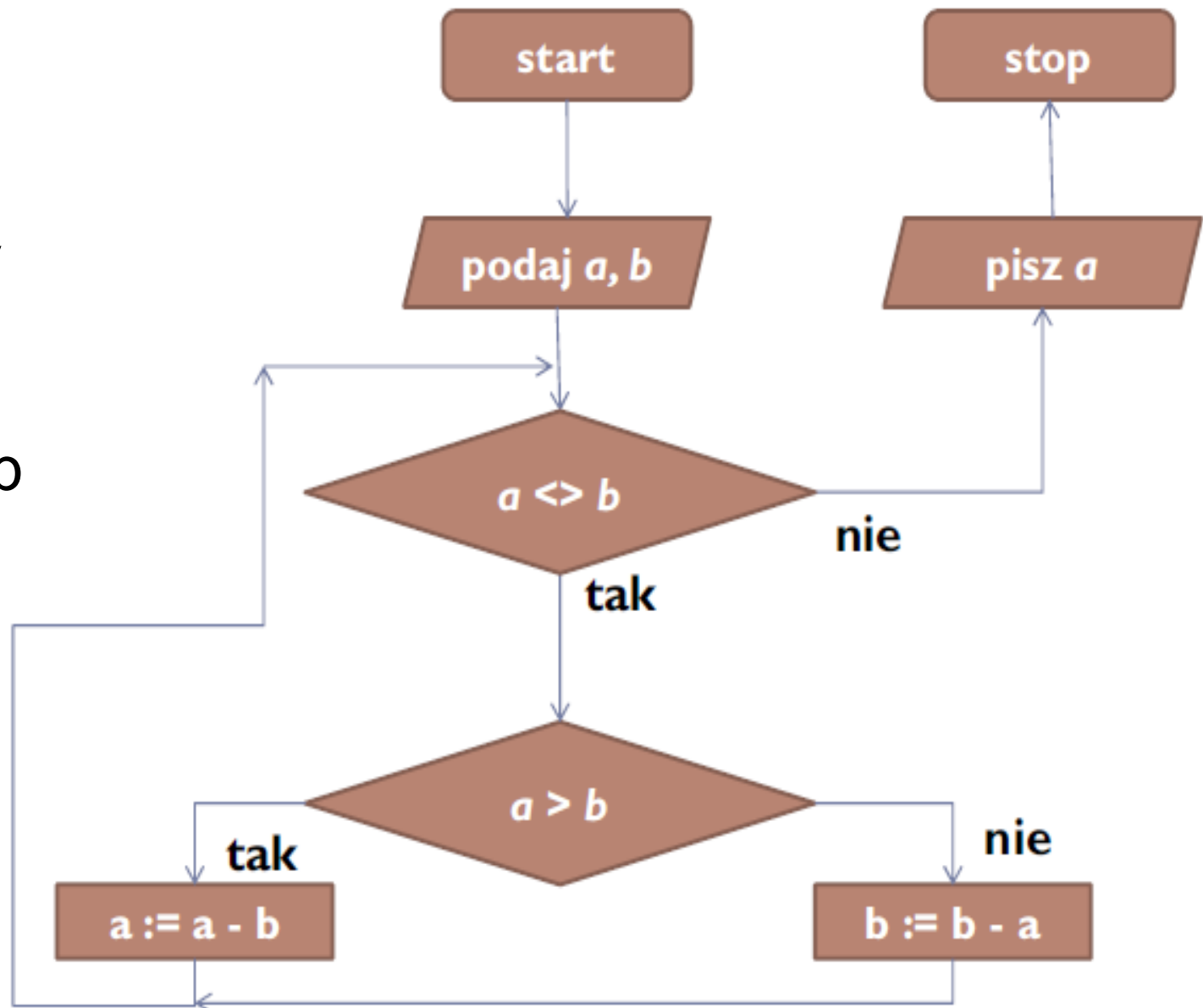
Przeszukiwanie z wartownikiem: schemat blokowy



Algorytm Euklidesa

Zadanie

znajdź
największy
wspólny
dzielnik
dwóch liczb



KONIEC