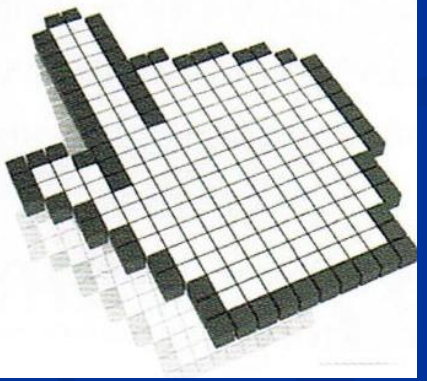
Temat:

Od problemu do wyniku – projektowanie rozwiązania problemu za pomocą umownego strukturalnego języka programowania

1

Od problemu do wyniku



Algorytm - przepis opisujący krok po kroku rozwiązanie problemu lub osiągnięcie jakiegoś celu

Podaj przykłady algorytmów?

Algorytm wyrażony w języku określonego typu nazywa się programem.

Przykłady języków programowania:

Basic

Turbo Pascal

C++

i inne

Podaj różnicę między algorytmem a programem?

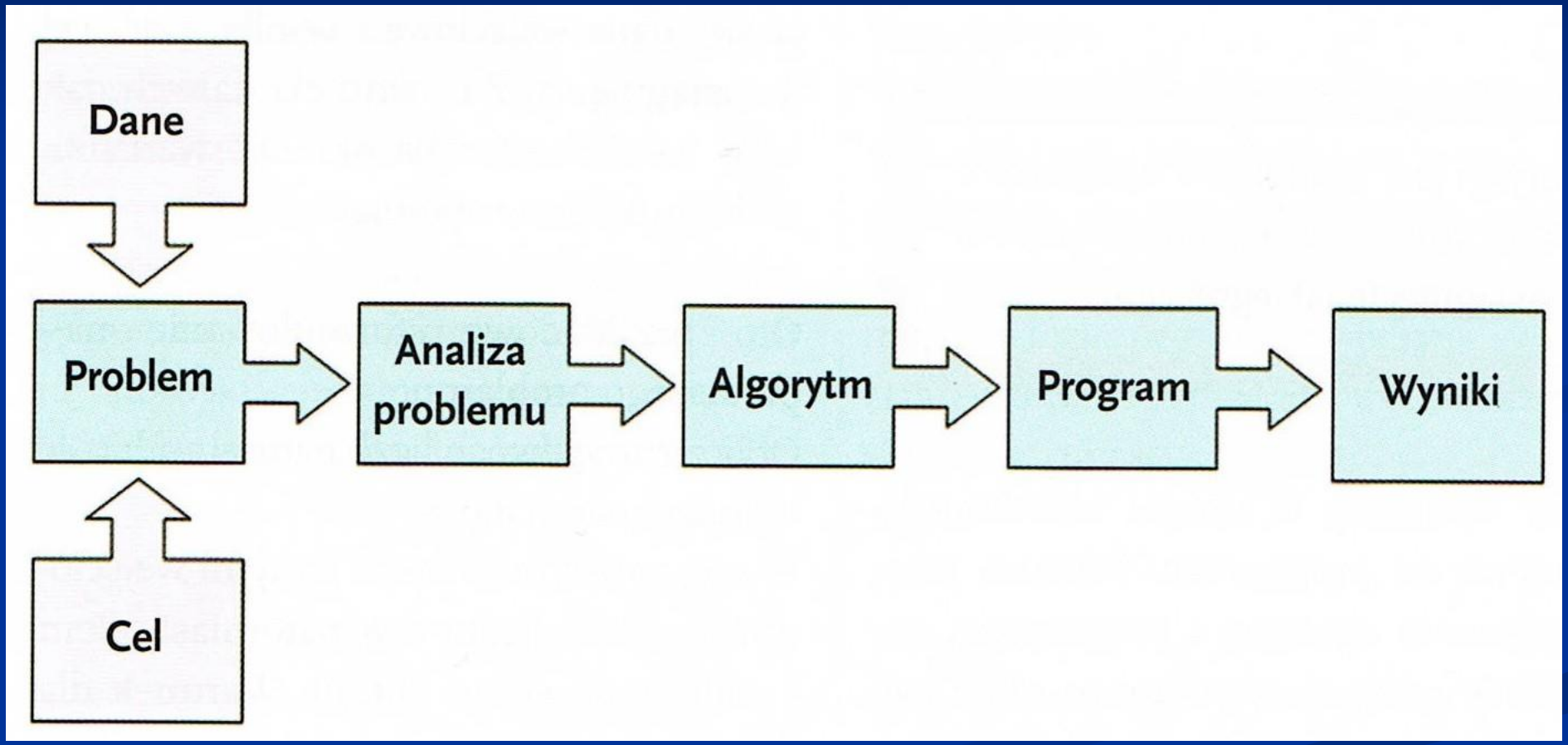
Algorytm i program ma 3 własności:

- Poprawność - poprawne wyniki

- Skończoność - algorytm wykonuje skończoną liczbę kroków

- Sprawność (jest lepszy, liczy szybciej, zajmuje mniej miejsca w pamięci)

1 Od problemu do wyniku



Algorytmy można przedstawiać w różny sposób, np. poprzez:

- schematy blokowe (gimnazjum)

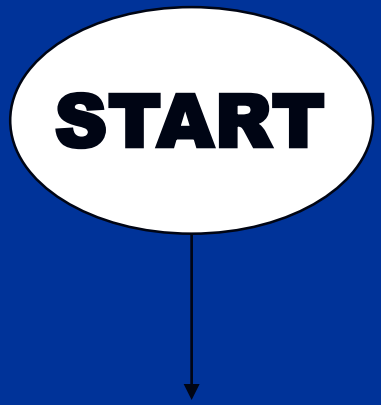
- umowny, strukturalny język programowania,

- itp.

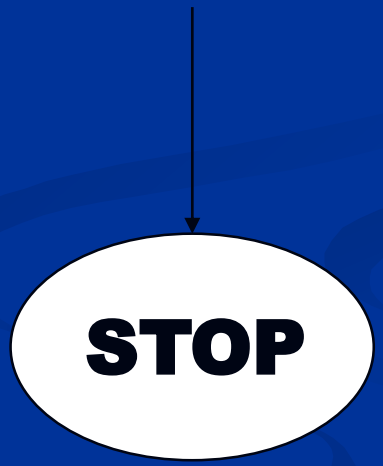
Schematy blokowe (powtórzenie)

Skrzynki graniczne: START/STOP

WARIANT 1



Początek algorytmu

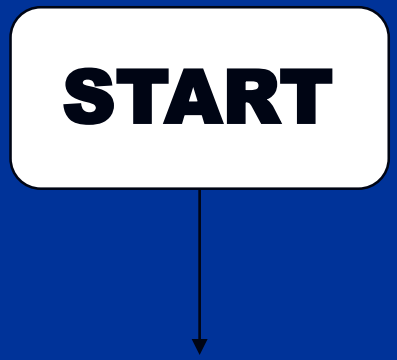


Koniec algorytmu

Schematy blokowe (powtórzenie)

Skrzynki graniczne: START/STOP

WARIANT 2



Początek algorytmu

Koniec algorytmu

Schematy blokowe (powtórzenie)

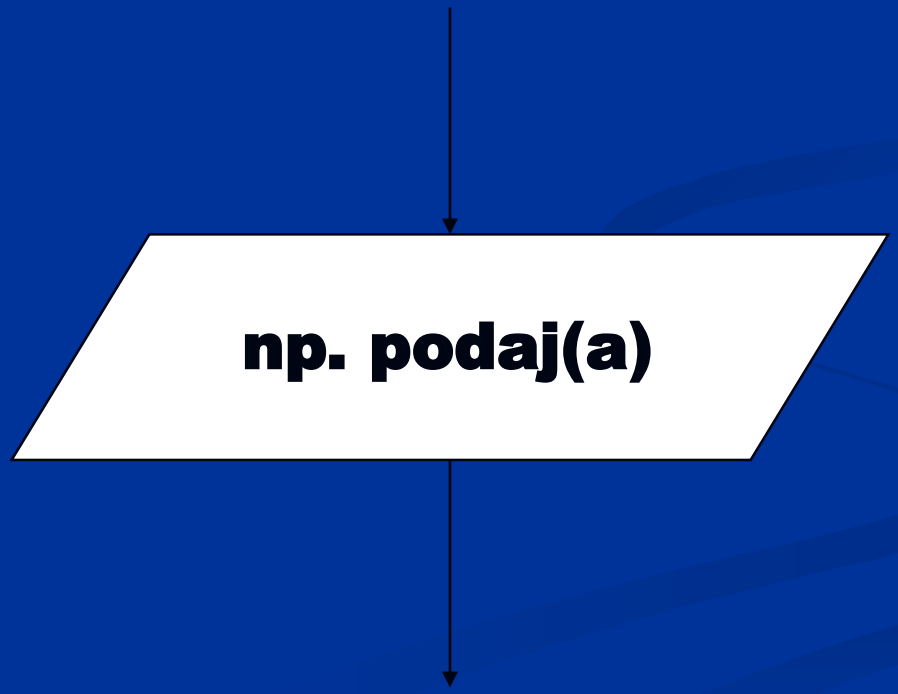
Skrzynka operacyjna (wykonawcza):

- kształt prostokąt
- wykonuje operacje (obliczenia)
- wprowadzenie stałych (instrukcja przypisania)

Schematy blokowe (powtórzenie)

Skrzynka wejścia:

- kształt równoległoboku
- wprowadzenie danych



Schematy blokowe (powtórzenie)

Skrzynka wyjścia:

- kształt równoległoboku
- wyprowadzenie danych (pisanie wyników, drukowanie)

Schematy blokowe (powtórzenie)

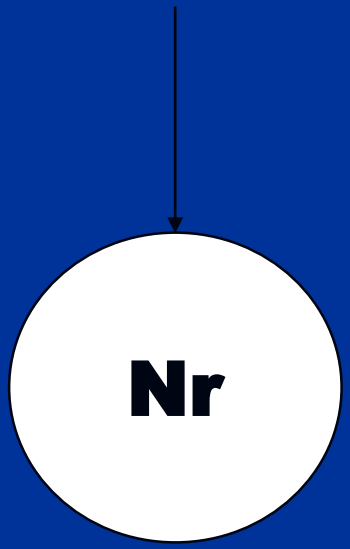
Skrzynka warunkowa (decyzyjna):

- kształt rombu
- sprawdzenie warunku (wykonywane jest skrzydło albo TAK albo NIE)

Schematy blokowe (powtórzenie)

Skrzynki łącznikowa:

- kształt okrąg
- łączy (kontynuuje) algorytm napisany na kilku stronach



Koniec algorytmu na danej stronie

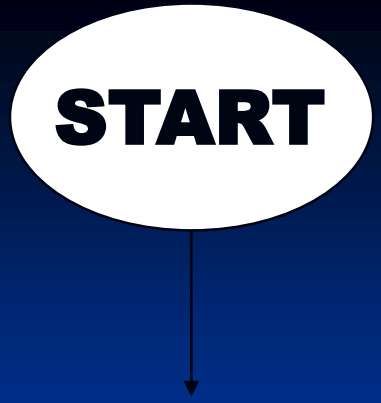
Kontynuowanie algorytmu z danej stronie

Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

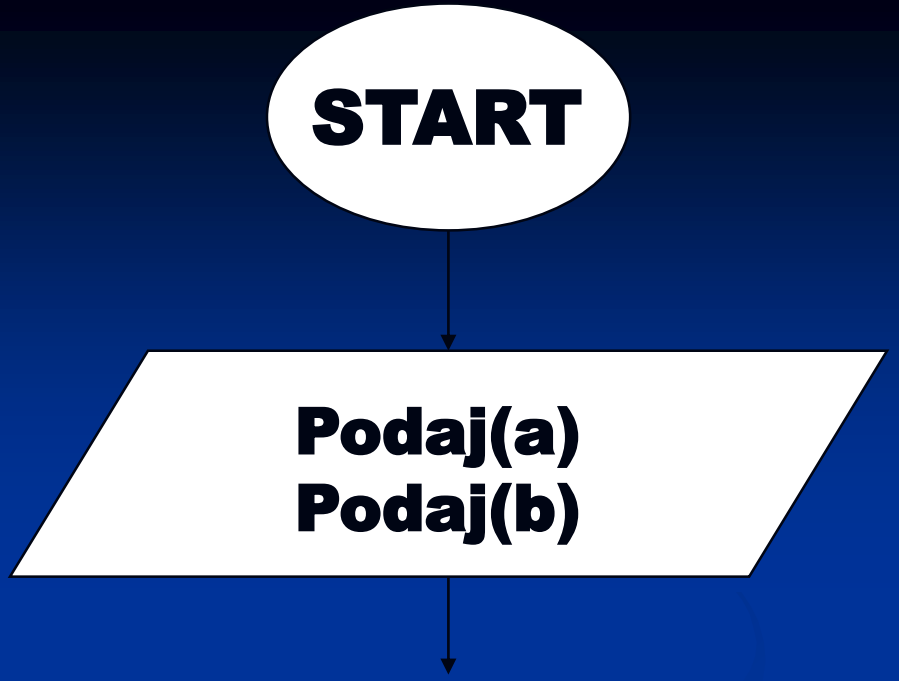
Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b



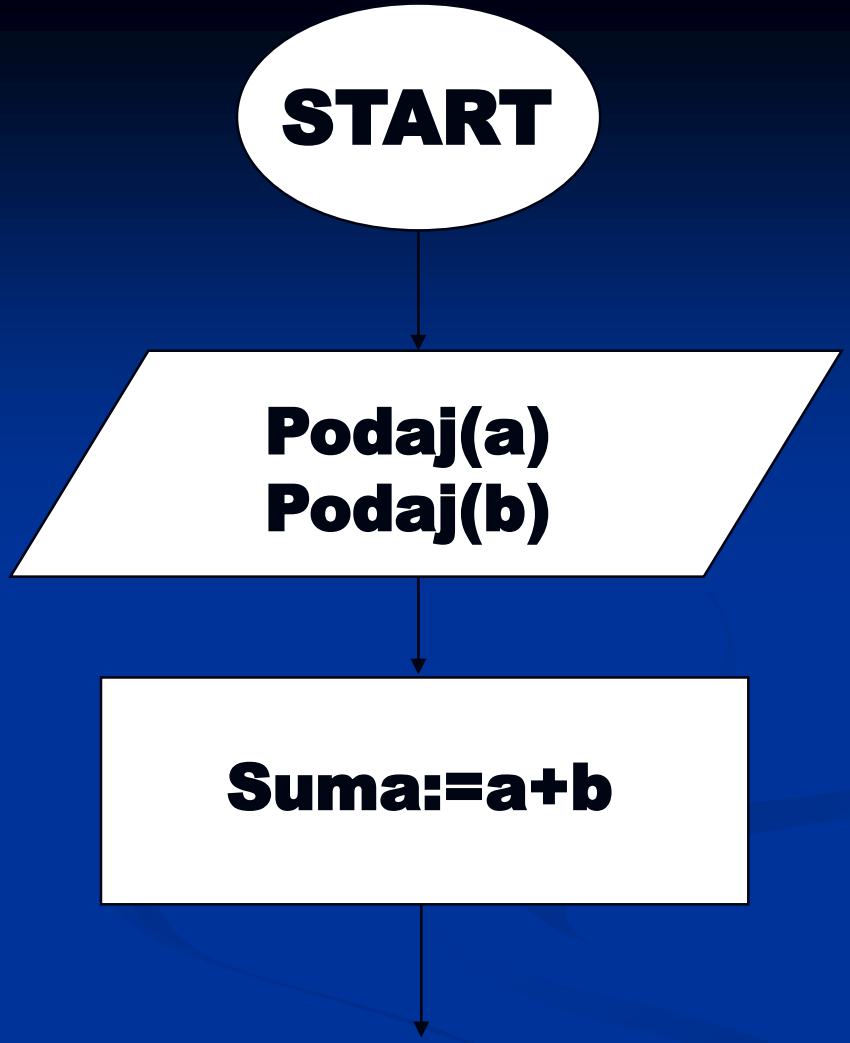
Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b



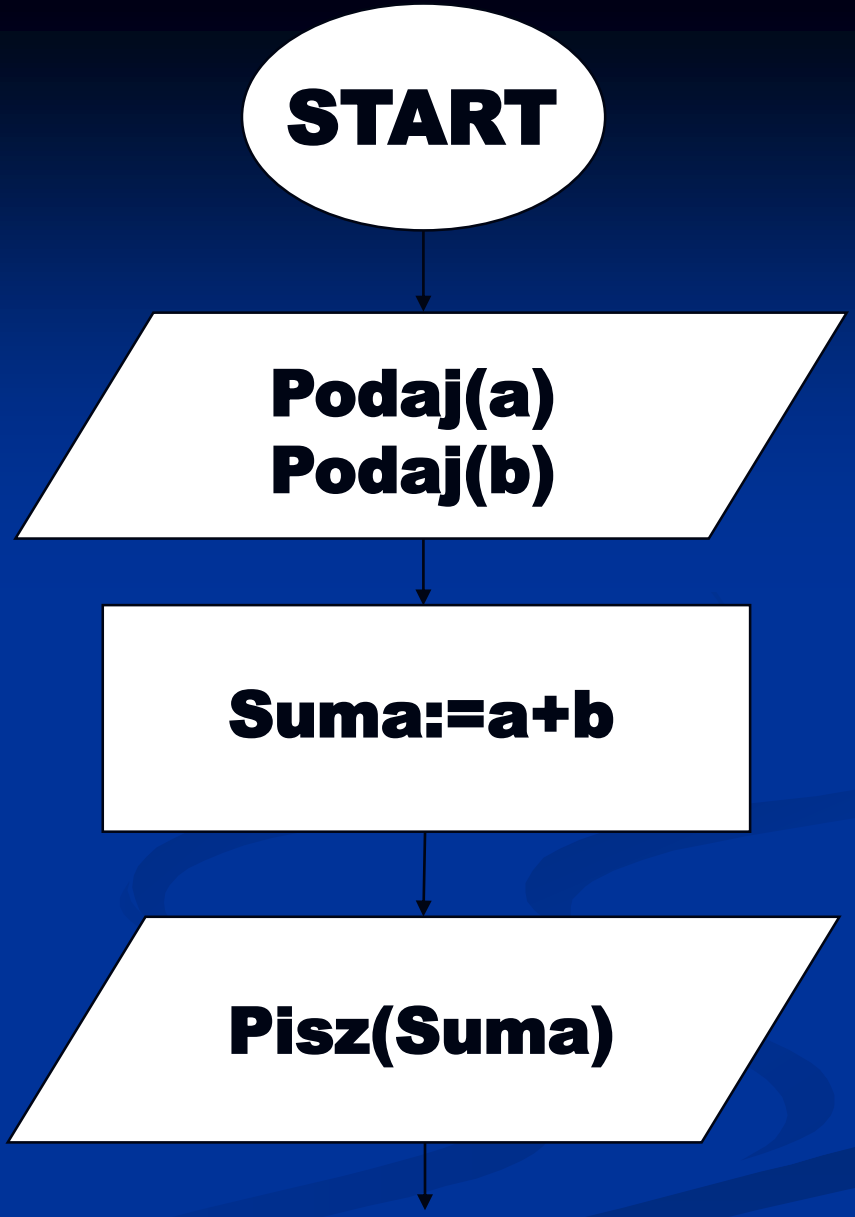
Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b



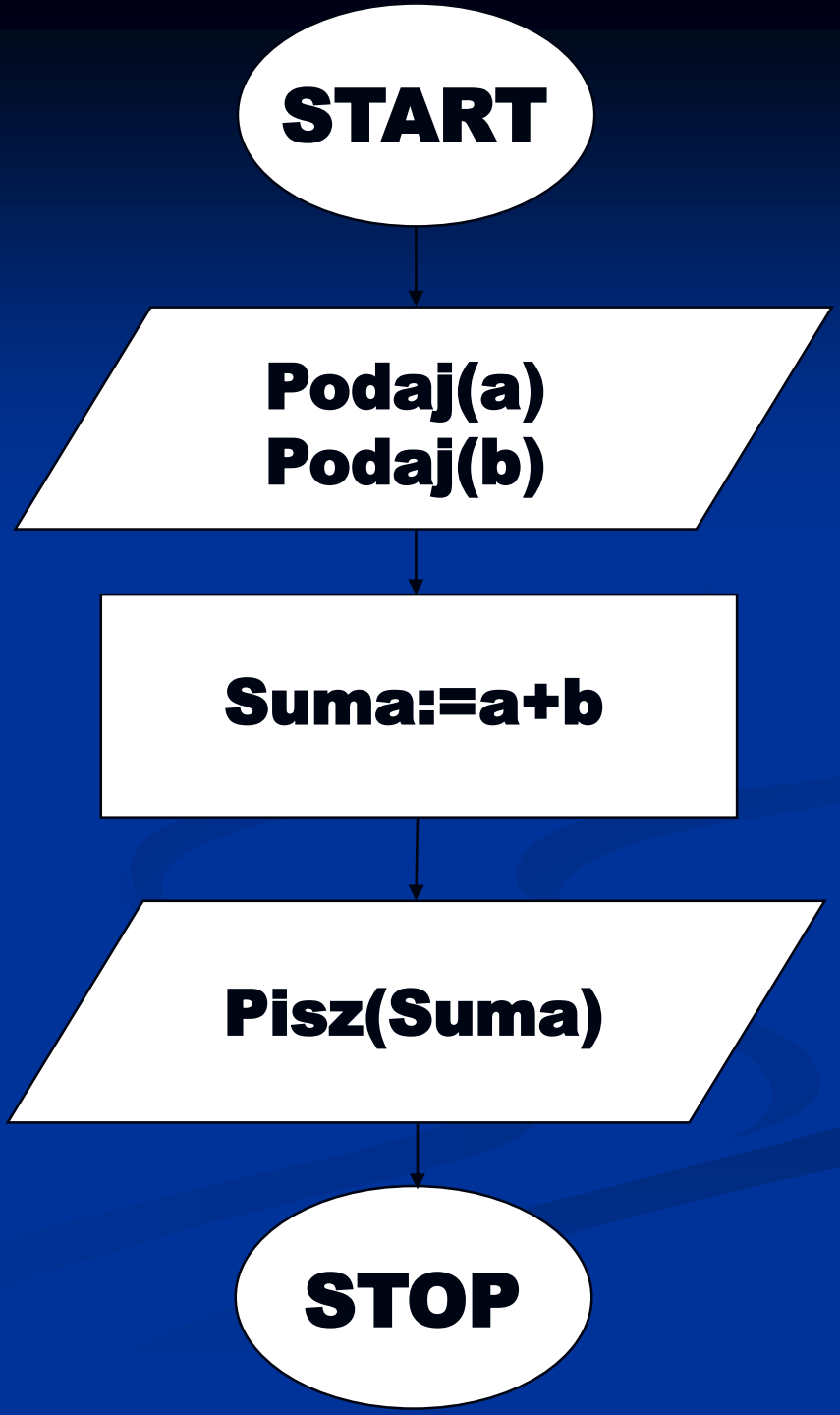
Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b



Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b



Ćwiczenie:

Narysuj algorytm, który obliczy pole powierzchni koła o zmiennym promieniu r

2

Projektowanie rozwiązania problemu za pomocą umownego strukturalnego języka programowania

Instrukcja przypisania

z := w

z – zmienna w - wyrażenie

Instrukcja przypisania

z := w z – zmienna w - wyrażenie

Przykłady

c:=5 nadanie zmiennej *c* wartości 5

Instrukcja przypisania

z := w z – zmienna w - wyrażenie

Przykłady

x : = 5 nadanie zmiennej *x* wartości 5

I := I + 1 *zmiennej I przypisz nową wartość równą poprzedniej wartości I powiększonej o 1*

Przykładowo I=4 to nową wartością I=5

Instrukcja przypisania

z := w z – zmienna w - wyrażenie

Przykłady

x : = 5 nadanie zmiennej *x* wartości 5

I := I + 1 *zmiennej I przypisz nową wartość równą poprzedniej wartości I powiększonej o 1*

Przykładowo I=4 to nową wartością *I=5*

***Zwróć* uwagę, że znakiem przypisania jest**

:=

~~=~~

Instrukcje wejścia / wyjścia

wprowadzanie danych i wyprowadzanie wyników

Podaj (x) wprowadzenie do programu wartości, która zostanie przypisana zmiennej *x*

Pisz (x) wyprowadzenie z programu wyniku, który został przypisany zmiennej *x*

Pisz ('Wynik') - wyprowadzenie napisu Wynik

Pisz ('x') wyprowadzenie napisu x, a nie wartości zmiennej *x*

Instrukcja złożona

POCZĄTEK

Instrukcja_1
Instrukcja_2 **=> pojedyncza instrukcja**
Instrukcja_3 itd.

KONIEC

Instrukcja warunkowa (decyzyjna)

Wyróżnia się 2 przypadki:

- instrukcję warunkową prostą:

  JEŚLI warunek TO akcja_1

- instrukcję warunkową z alternatywą:

  JEŚLI warunek TO akcja_1

  W PRZECIWNYM RAZIE akcja_2

Warunek jest wyrażeniem logicznym, któremu przypisuje się wartość logiczną prawda (Tak) lub fałsz (Nie)

Przykład 1

JEŚLI liczba > 0 **TO**
  Pisz ('liczba dodatnia')
W PRZECIWNYM RAZIE
Pisz ('liczba ujemna lub równa zeru')

Przykład 2

JEŚLI a<>0 **TO** x := -b/a

JEŚLI a=b **TO Pisz** ('obie liczby są równe')

Operatory matematyczne:

=	równe	nie mylić :=
<>	różne	
<	mniejsze	
>	większe	
<=	mniejsze lub równe	
>=	większe lub równe	

Inne operatory logiczne:

NIE (ang. NOT) **negacja**

I (ang. AND) **iloczyn logiczny**

LUB (ang. OR) **suma logiczna**

Instrukcje iteracyjne

służą do deklarowania wielokrotnego wykonywania w programie tych samych operacji tzw. (pętli)

Metoda 1

Instrukcja powtarzaj

POWTARZAJ akcję **AŻ** warunek

Powtarzaj czynności opisane przez akcję, aż stwierdzisz, że warunek logiczny jest prawdziwy.

Powtarzanie akcji trwa dopóty, dopóki warunek logiczny jest fałszywy.

Przykład

```
ile := 1
POWTARZAJ
   Pisz ('Będę się uczył informatyki')
      ile := ile + 1
AŻ ile >100
```

Przykład

```
ile := 1
POWTARZAJ
   Pisz ('Będę się uczył informatyki')
      ile := ile + 1
AŻ ile >100
```

OPIS:

Pisz sto razy tekst:

Będę się* uczył *informatyki

Gdy zmienna ile przekroczy wartość 100

zakończ

Metoda 2

Instrukcja dopóki

DOPÓKI warunek **WYKONUJ** akcję

Opis:

Dopóki warunek logiczny jest prawdziwy, wykonuj czynności opisane przez akcję

Jaka jest różnica między instrukcjami POWTARZAJ i DOPÓKI

W pierwszym przypadku jest wykonywana akcja, a dopiero potem sprawdzany warunek. Jeśli jest fałszywy, następuje powrót.

W drugim przypadku jest sprawdzany warunek i jeśli jest prawdziwy, dopiero wtedy akcja jest wykonywana.

Jeśli w instrukcji DOPÓKI trzeba wykonać nie jedną, lecz kilka akcji, to należy je ująć klamrą POCZĄTEK ... KONIEC

Przykład

```
ile := 1
DOPÓKI  ile  <=100  WYKONUJ
  POCZĄTEK
     Pisz   ('Będę się uczył informatyki')
     ile := ile + 1
  KONIEC
```

Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

Jak będzie wyglądać schemat blokowy i zapisy w umownym strukturalnym języku programowania?

Umowny strukturalny język programowania

Schemat blokowy

Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

START

Umowny strukturalny język programowania

Schemat blokowy

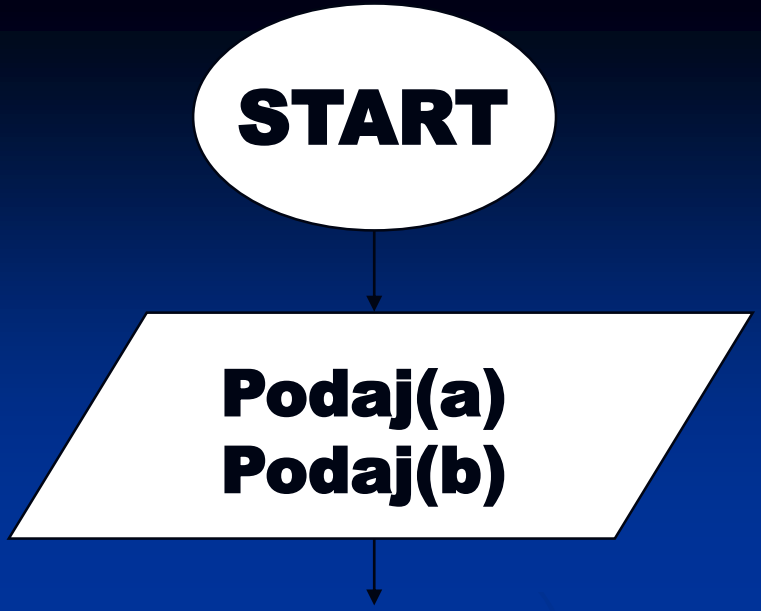
Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

START

Podaj(a)
Podaj(b)

Umowny strukturalny język programowania



Schemat blokowy

Ćwiczenie:

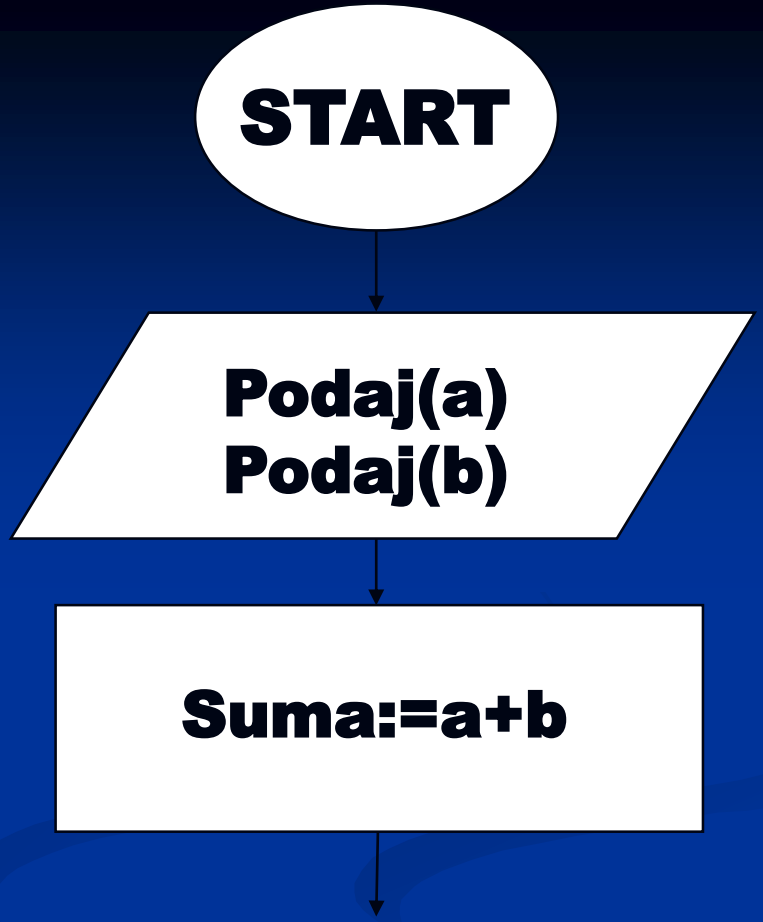
Narysuj algorytm, który obliczy sumę dwóch liczb a i b

START

Podaj(a)
Podaj(b)

Suma:=a+b

Umowny strukturalny język programowania



Schemat blokowy

Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

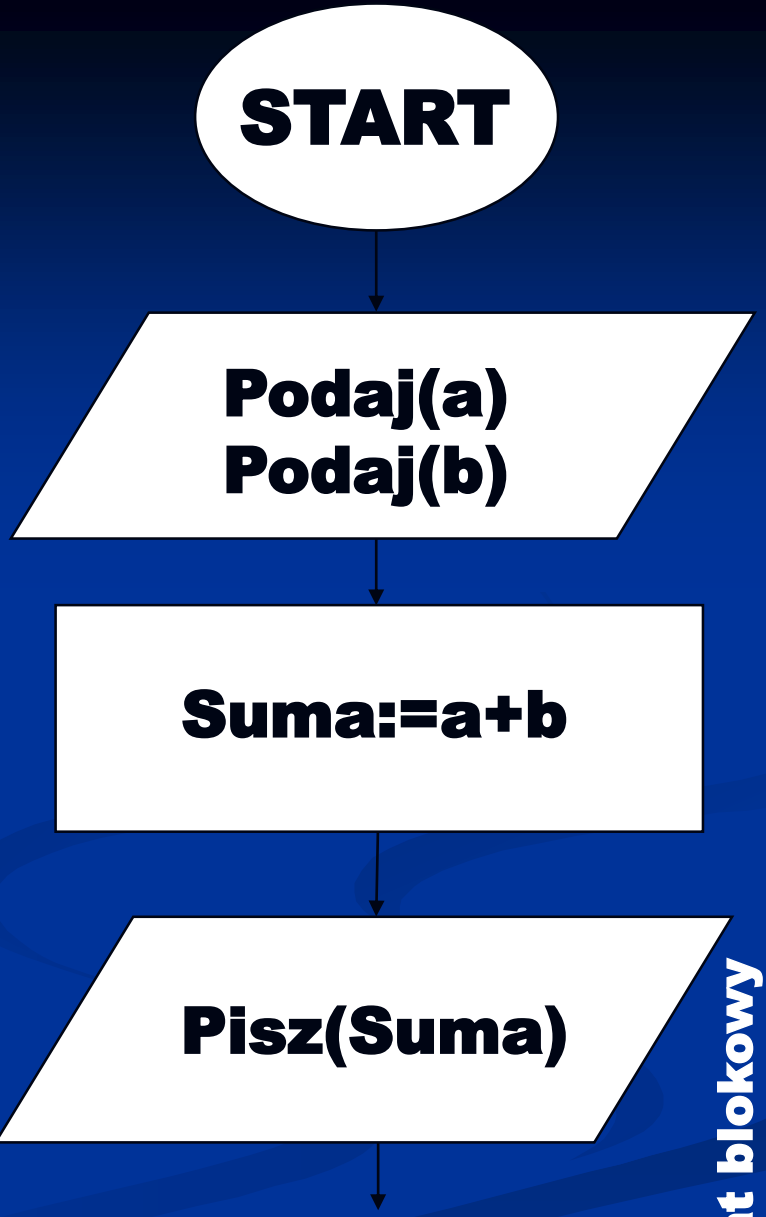
Umowny strukturalny język programowania

START

Podaj(a)
Podaj(b)

Suma:=a+b

Pisz(Suma)



Schemat blokowy

Ćwiczenie:

Narysuj algorytm, który obliczy sumę dwóch liczb a i b

START

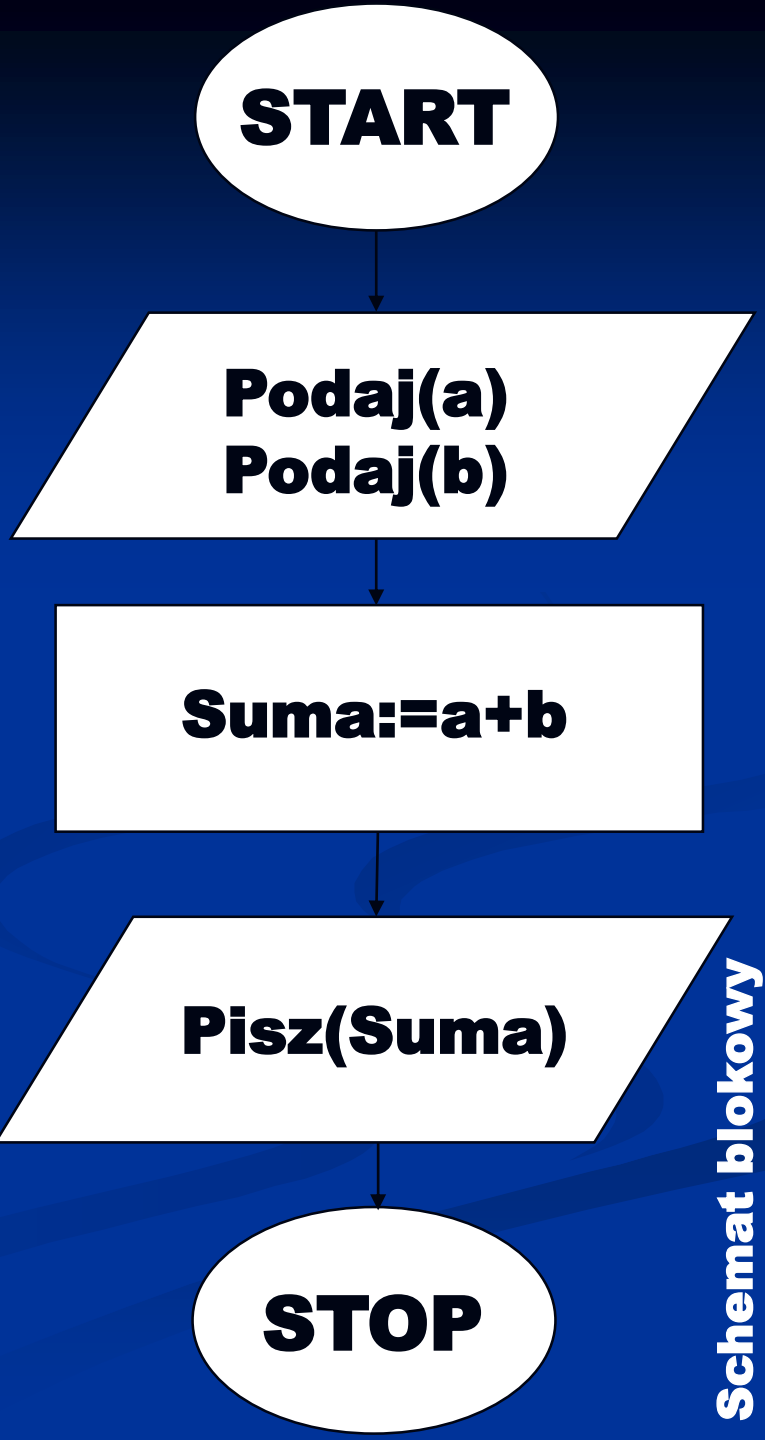
Podaj(a)
Podaj(b)

Suma:=a+b

Pisz(Suma)

STOP

Umowny strukturalny język programowania



Schemat blokowy