SCHEMATY BLOKOWE

Do tej pory do opisu algorytmów używaliśmy języka potocznego. Były to jednak algorytmy proste, które ma wykonywać człowiek. W przypadku algorytmów skomplikowanych ten zapis będzie nieczytelny, nie sprawdzi się. Dlatego teraz poznamy nowy, bardziej przejrzysty sposób - schematy blokowe

**Schemat blokowy to graficzny zapis algorytmu rozwiązania zadania, przedstawiający opis i kolejność wykonywania czynności realizujących dany algorytm.**

W schemacie blokowym poszczególne operacje przedstawione są za pomocą odpowiednio połączonych skrzynek (klocków, bloków). Połączenia określają kolejność i sposób wykonywania operacji realizujących dany algorytm. W literaturze informatycznej przyjęto pewne standardowe oznaczenia poszczególnych działań (są to figury geometryczne), ale można również używać innych oznaczeń (muszą one jednak być takie same dla określonego typu operacji).

Przykłady skrzynek (bloków) do prezentacji algorytmu w postaci graficznej:

|  |  |  |  |
| --- | --- | --- | --- |
| Elementy schematu blokowego | | | |
| **Symbol graficzny** | **Nazwa skrzynki (bloku)** | **Funkcja** | **Opis** |
| Skrzynki graniczne | Skrzynka graniczna | Początek algorytmu lub koniec | mają kształt owalu. Ze skrzynki START wychodzi tylko jedno połączenie, skrzynka STOP nie ma połączenia wychodzącego. |
| Skrzynka graniczna |
| Skrzynka operacyjna | Skrzynka operacyjna | Wykonywanie różnych działań, np. sumowania | ma kształt prostokąta. |
| Skrzynka wejścia / wyjścia | Skrzynka wejścia / wyjścia | Wprowadzanie (czytanie) danych lub wyprowadzanie (drukowanie, pisanie) wyników | jest równoległobokiem, wchodzi i wychodzi z niej jedno połączenie. |
| Skrzynka graniczna |
| Skrzynka warunkowa | Skrzynka warunkowa | Sprawdzanie warunku, np. czy N > 0 | mają kształt rombu. Ze skrzynki wychodzą tylko dwa połączenia: jedno oznaczone TAK, a drugie NIE. |
|  |  |  |

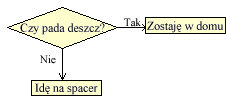
ZASADY BUDOWANIA SCHEMATU BLOKOWEGO

* Każda operacja powinna być umieszczona w skrzynce (bloku)
* Schemat powinien posiadać tylko jedną skrzynkę "Start" i przynajmniej jedną skrzynkę "Stop"
* Skrzynki powinny być ze sobą połączone
* Ze skrzynki powinno wychodzć jedno połączenie; wyjątek stanowią skrzynki "Stop" (z której nie wychodzą już żadne połączenia) oraz "warunkowa" (z której wychodzą dwa połączenia opisane Tak i Nie - w zależności od tego czy warunek jest spełniony czy też nie; można wyjść jedną z dwóch dróg).

### ALGORYTMY WARUNKOWE

Przypomnij: co to jest instrukcja warunkowa ? Z sytuacjami warunkowymi stykamy się w każdej dziedzinie życia codziennego. Na pytanie "Czy pada deszcz ?" odpowiedź może brzmieć "tak" lub "nie". W zależności od tego, czy warunek jest spełniony, czy nie, wybieramy inne rozwiązanie.

Przykład sytuacji warunkowej:



ALGORYTMY ITERACYJNE

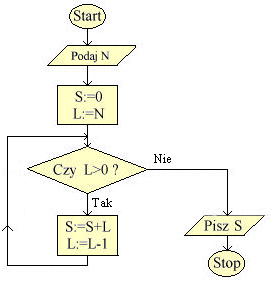
Przypomnij : co to są instrukcje iteracyjne ? Czasami trzeba wykonać te same operacje na wielu liczbach. W takich przypadkach nie jest konieczne wielokrotne opisywanie działań lub rysowanie takich samych skrzynek. Stosujemy wówczas iterację. Mówimy także, że działania te wykonywane są w pętli. Liczba powtórzeń tych działań może być z góry określona lub zależeć od spełnienia warunku. Iteracja to najczęściej spotykana technika algorytmiczna.

**Iteracja to technika algorytmiczna polegająca na wykonaniu tej samej instrukcji dla n zmiennych.**

Iteracja oszczędza czas programisty, który ten musiałby spędzić wpisując instrukcję n razy, zależnie od liczby zmiennych. Liczba powtórzeń w iteracji jest zwykle z góry określona lub zależy od spełnienia określonego warunku.

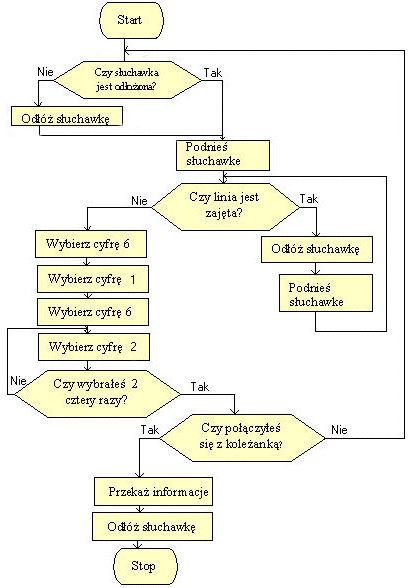
**Przykład**Analiza schematu algorytmu. Przeanalizuj poniższy schemat blokowy i odpowiedz na pytania:

1. Jakie są dane wejściowe do zadania i jakich użyto zmiennych pomocniczych?
2. Jaki cel chcesz osiągnąć (wynik)?
3. Jaki algorytm zastosowano w operacji dodawania liczb ?
4. W którym miejscu wykonywane są dzivałania w pętli ?
5. Które operacje powtarzają się wielokrotnie ?
6. Co określa liczbę powtórzeń ?
7. Kiedy kończy się działanie w pętli ?



W analizowanym algorytmie występuje przypisanie typu S:=S + L. Oznacza ono przypisanie zmiennej "s" (sumie) poprzedniej wartości pamiętanej w zmiennej "s" (poprzedniej sumie) zwiększonej o wartość pamiętaną w zmiennej "l" (kolejną liczbę) - w ten sposób powtarzanie operacji przypisania, realizowane jest dodawanie kolejnych liczb naturalnych. Przypisania l:=l-1 ma tu bardzo istotne znaczenie. Jest to tzw. licznik, w którym następuje obliczanie, ile zostało jeszcze powtórzeń do wykonania. W tym algorytmie liczba powtórzeń została określona na początku instrukcją l:=n.

Gdyby nie umieszczono tego działania nastąpiło by tzw. zapętlenie algorytmu, czyli po sprawdzeniu warunku l>0 działanie w schemacie przebiegałoby zawsze drogą TAK. W realizacji algorytmów iteracyjnych ważne jest prawidłowe określenie sposobu zakończenia działań. Można to zrobić za pomocą licznika, który odlicza kolejne kroki iteracji (liczbę powtórzeń).



Sposób rozwiązania zadania, czyli algorytm, można podać w kilku punktach. Mówimy wówczas, że algorytm jest opisany za pomocą **listy kroków** (ciągu kroków). Zapis taki polega na podaniu kolejnych wykonywanych operacji, składających się na całe rozwiązanie.  
Algorytm można przedstawić również w postaci graficznej, za pomocą schematu blokowego. **Schemat blokowy** to graficzne przedstawienie ciągu kroków algorytmu, często stosowane jako ideowy rysunek poprzedzający tworzenie programu. Sposób i kolejność działań programu określane są przez wzajemny układ i sposób łączenia bloków (skrzynek) ze sobą. Każde działanie (krok) ma w schemacie blokowym swoje standardowe oznaczenie (patrz tabela wyżej).