

Schematy blokowe – wprowadzenie do algorytmów

Krzysztof Gębicz

Cel lekcji

- zrozumie, czym są schematy blokowe,
- pozna ich znaczenie w programowaniu,
- nauczy się rozpoznawać podstawowe symbole,
- pozna zasady budowy schematu krok po kroku,
- będzie potrafił czytać proste schematy i analizować ich logikę.

Co to jest schemat blokowy?

- Schemat blokowy to graficzne przedstawienie algorytmu lub procesu.
- Algorytm = zbiór kroków prowadzących do rozwiązania problemu.
- Zamiast długiego opisu słownego używamy symboli graficznych i strzałek.
- Dzięki temu przedstawiamy algorytm w sposób jasny, uporządkowany i łatwy do odczytania.

Dlaczego schematy blokowe są ważne?

- Ułatwiają zrozumienie problemu – widzimy cały proces w jednym miejscu.
- Porządkują myślenie – zmuszają do uporządkowania kroków w logicznej kolejności.
- Pomagają w komunikacji – różne osoby (uczniowie, programiści, menedżerowie) szybciej zrozumieją schemat niż opis słowny.
- Redukują ryzyko błędów – łatwiej dostrzec brakujący krok lub błędny warunek.

Dlaczego schematy blokowe są ważne?

- Przygotowują do pisania kodu – schemat blokowy to pierwszy krok przed zapisaniem algorytmu w języku programowania.
- Uniwersalność – stosowane w informatyce, biznesie, inżynierii, edukacji.
- Przystępność – zrozumiały nawet dla osób, które nie znają programowania.

Zastosowania w praktyce

- Informatyka – planowanie programów, wizualizacja algorytmów.
- Inżynieria – opisywanie procesów produkcyjnych i technologicznych.
- Biznes – przedstawianie procedur, np. proces obsługi klienta.
- Administracja – wizualizacja dokumentów i obiegu informacji.
- Edukacja – nauka logicznego myślenia i analizowania problemów.

Symbol 1 – Oval (Start/Stop)

- Wygląd: elipsa lub owal.
- Znaczenie: początek lub koniec algorytmu.
- Ważne zasady:
 - Każdy schemat musi mieć dokładnie jeden punkt Start i przynajmniej jeden punkt Stop.
 - Oval nie może pojawiać się w środku schematu jako „czynność” – ma wyłącznie sygnalizować początek lub zakończenie.

Symbol 1 – Oval (Start/Stop)



Symbol 2 – Prostokąt (Czynność)

- Wygląd: prostokąt.
- Znaczenie: krok, w którym wykonywana jest czynność, obliczenie lub operacja.
- Wskazówki:
 - Opisy powinny być krótkie i jednoznaczne.
 - Prostokąty są najczęściej używanymi symbolami w schematach blokowych.

Symbol 2 – Prostokąt (Czynność)

Czynność
"a+b"

Symbol 3 – Równoległobok (Wejście/Wyjście)

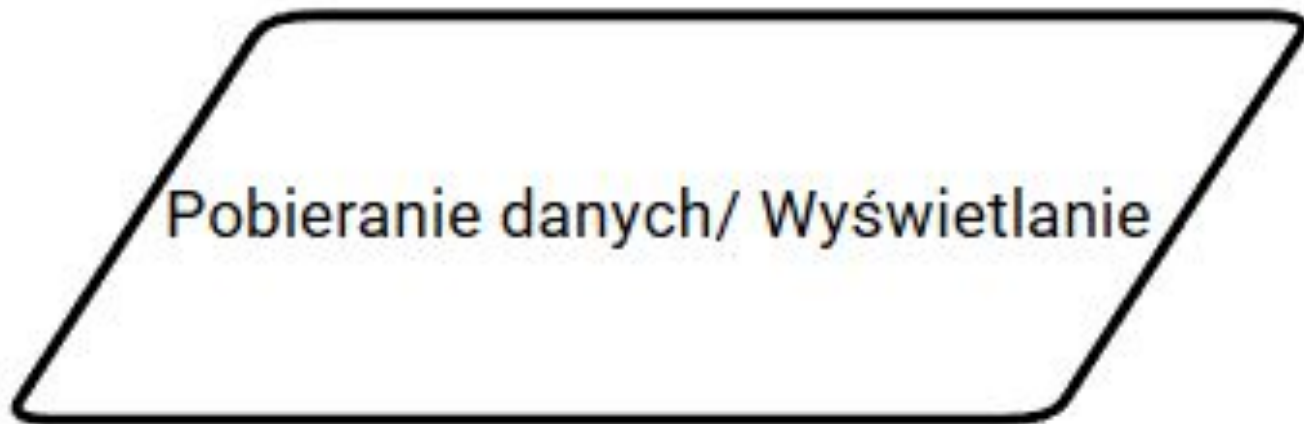
Wygląd: równoległobok, często lekko pochylony.

Znaczenie: interakcja programu z użytkownikiem – pobieranie danych lub wyświetlanie wyników.

Przykłady użycia:

- „Podaj liczbę n” (wejście)
- „Wyświetl wynik” (wyjście)

Symbol 3 – Równoległobok (Wejście/Wyjście)



Symbol 4 – Romb (Warunek/Decyzja)

- Wygląd: romb, z którego wychodzą dwie lub więcej strzałki.
- Znaczenie: miejsce podjęcia decyzji – pytanie, które daje odpowiedź TAK/NIE lub podobną.
- Przykłady:
 - „Czy liczba > 0 ?”
 - „Czy masz wodę?”
- Uwagi: każda odpowiedź prowadzi do innej ścieżki algorytmu.

Symbol 4 – Romb (Warunek/Decyzja)



Symbol 5 – Strzałki (Kolejność działań)

- Wygląd: linie z grotem.
- Znaczenie: określają kierunek przepływu algorytmu.
- Zasady:
 - Schemat czytamy od góry do dołu, od lewej do prawej.
 - Strzałki nie powinny się krzyżować – w dużych schematach używa się łączników.
 - Strzałka zawsze musi prowadzić do kolejnego symbolu.

Symbol 5 – Strzałki (Kolejność działań)

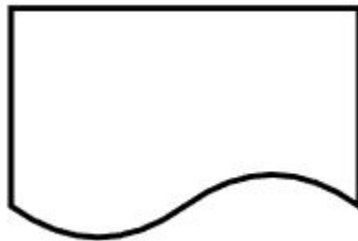


Dodatkowe symbole

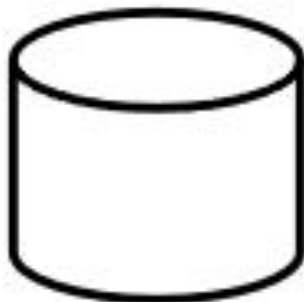
1. Dokument – prostokąt z falowaną dolną linią, symbolizuje wytworzenie dokumentu lub zapis informacji.
2. Baza danych – cylinder, oznacza zapis/odczyt danych w pamięci lub w bazie.

Dodatkowe symbole

1. Dokument



2. Baza danych



Zasady budowania schematów

- Zawsze rozpoczynamy od Start i kończymy na Stop.
- Każdy symbol ma jedno wejście i jedno wyjście (z wyjątkiem rombu).
- Warunki muszą prowadzić do dwóch dróg: TAK i NIE.
- Algorytm powinien być kompletny – każda możliwa ścieżka prowadzi do końca.

Zasady budowania schematów

- Opisy w symbolach mają być krótkie, precyzyjne i jednoznaczne.
- Strzałki nie powinny się płatać ani krzyżować.
- Schemat ma być czytelny – układ od góry do dołu i od lewej do prawej.
- Nie mieszać różnych poziomów szczegółowości – jeśli proces jest zbyt duży, dzielimy go na mniejsze schematy.

Jak powstaje schemat?

1. Zdefiniuj problem i wypisz algorytm słownie.
2. Podziel czynności na: operacje, wejścia/wyjścia, warunki.
3. Zapisz je w odpowiednich symbolach.
4. Połącz symbole strzałkami w logicznej kolejności.
5. Sprawdź, czy każda ścieżka ma początek i koniec.

Przykład teoretyczny 1 – Dodawanie dwóch liczb

Algorytm krok po kroku:

1. START
2. Wprowadź dwie liczby
3. Oblicz sumę
4. Wyświetl wynik
5. STOP

Schemat: prosty układ 5 symboli – owal, równoległobok, prostokąt, równoległobok, owal.

Przykład teoretyczny 2 – Sprawdzenie parzystości

Algorytm krok po kroku:

1. START
2. Wprowadź liczbę
3. Sprawdź: czy liczba mod 2 = 0?
4. Jeśli TAK → „Liczba parzysta”
5. Jeśli NIE → „Liczba nieparzysta”
6. STOP

Schemat: romb jako warunek, dwie ścieżki TAK/NIE prowadzące do różnych wyników

Przykład teoretyczny 3 – Proces codzienny

Algorytm „Robienie herbaty”:

1. START
2. Czy mam wodę? Jeśli NIE → STOP.
3. Zagotuj wodę.
4. Włóż herbatę do kubka.
5. Zalej wrzątkiem.
6. Czy chcę cukier? Jeśli TAK → dodaj cukier.
7. STOP

Schemat: przykład nienumeryczny, łatwy do wyobrażenia i zrozumienia.

Podsumowanie

1. Schemat blokowy = graficzne przedstawienie algorytmu.
2. Najważniejsze symbole: owal (Start/Stop), prostokąt (Czynność), równoległobok (Wejście/Wyjście), romb (Warunek), strzałki (Przepływ).
3. Dodatkowe symbole: łącznik, dokument, baza danych, komentarz.
4. Zasady: prostota, czytelność, poprawna logika, jednoznaczne opisy.