

Schematy blokowe

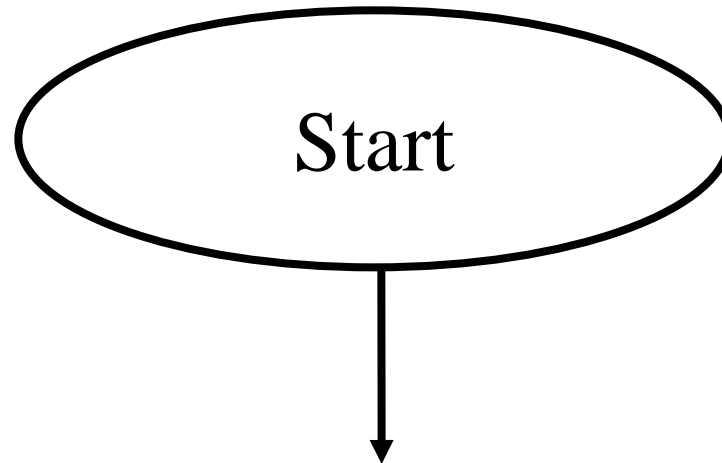
Algorytmy

Marek Pudełko

Metody zapisu algorytmów

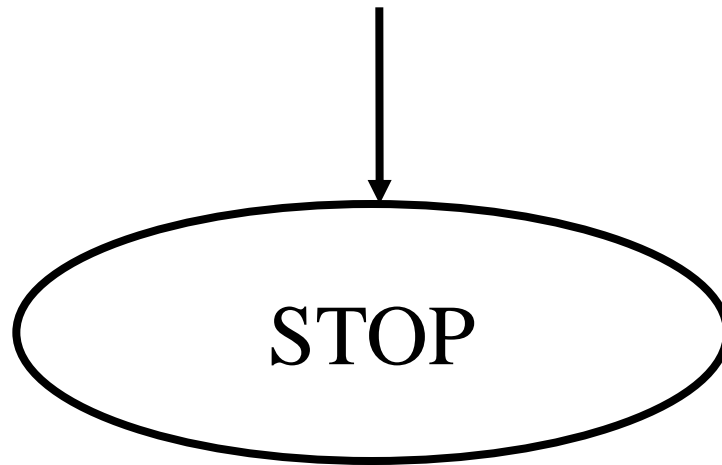
- Algorytmy można zapisywać w postaci słownej, listy kroków lub symbolicznej - używając metajęzyków.
- Metajęzyk to język bardzo ogólny - opisujący algorytmy, by na jego podstawie można było je zaimplementować w każdym języku.
- Wykorzystuje się w tym przypadku schematy blokowe.
 - Stosowane są proste figury geometryczne, np. prostokąt, romb, koło, równoległobok itd...
 - Łączą je strzałki symbolizujące kolejność operacji.
 - W tych figurach umieszczamy warunki oraz proste instrukcje.

Blok startu



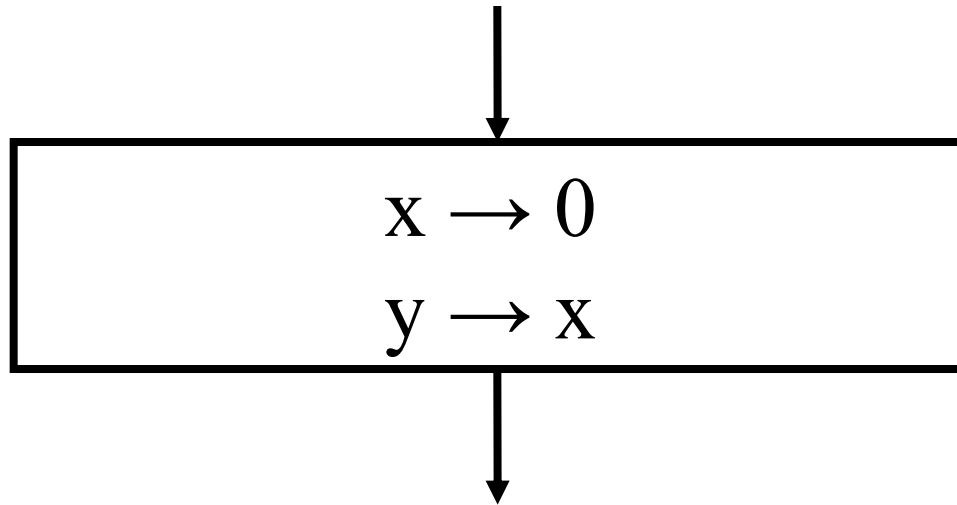
- Blok start wskazuje początek wykonywania algorytmu.
- Każdy algorytm ma dokładnie jeden start.
- Do startu nie wchodzi żadna strzałka, a wychodzi tylko jedna.

Blok zakończenia



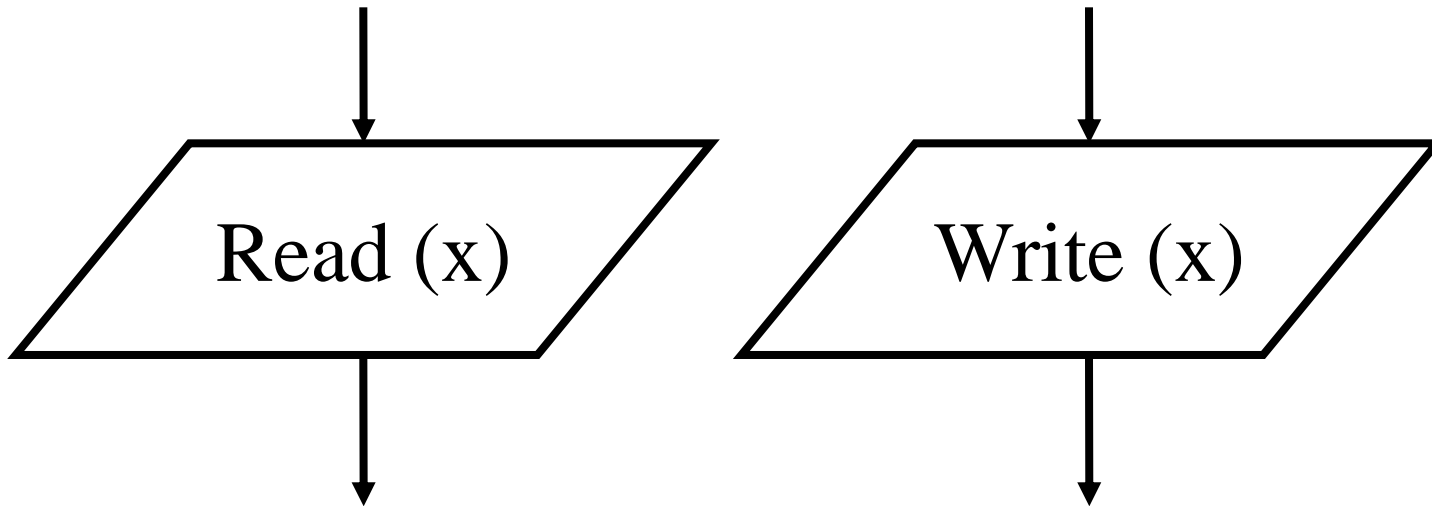
- Blok stop wskazuje koniec wykonywania algorytmu.
- Każdy algorytm ma dokładnie jeden stop (choć dla czytelności można narysować ich kilka).
- Do stopu wchodzi tylko jedna strzałka, a nie wychodzi żadna.

Blok operacji



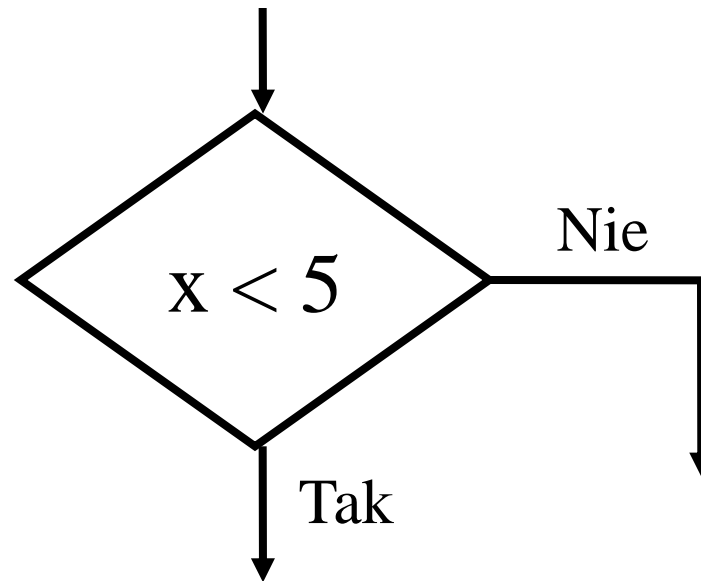
- Blok operacji informuje o wykonywanym poleceniu.
- Wewnątrz niego wpisuje się komentarz określający daną operację.
 - Jeśli kilka operacji tworzy logiczną całość, to wszystkie one mogą być umieszczone w jednym bloku.
- Każdy blok ma jedno wejście i jedno wyjście.

Blok wymiany danych



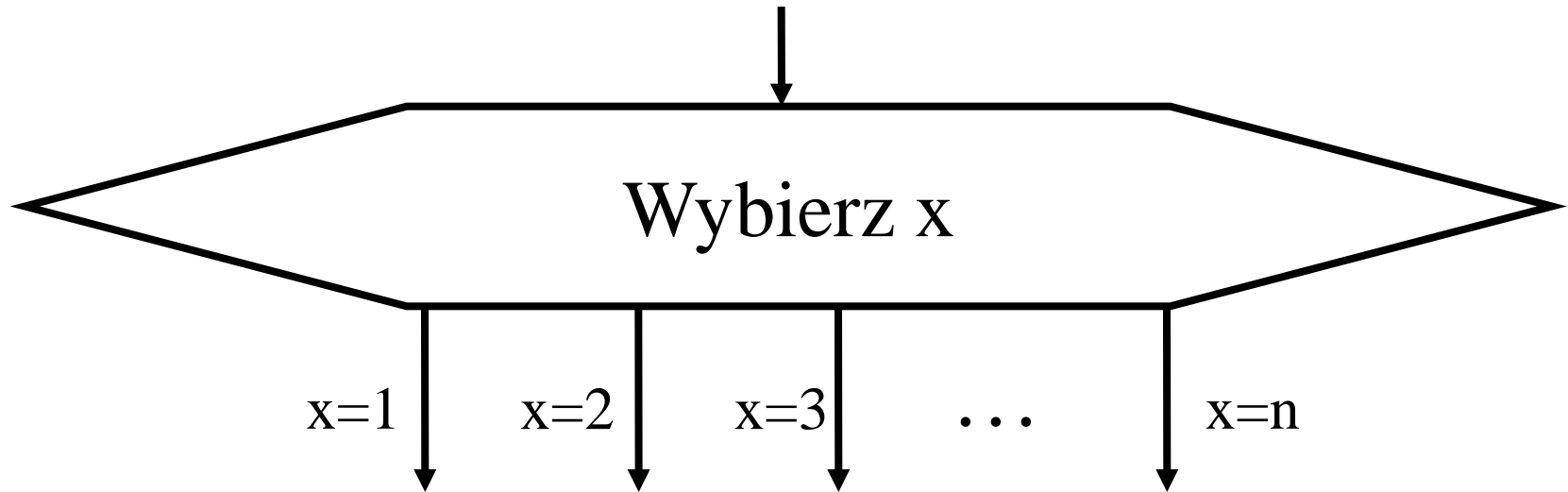
- Blok wymiany danych jest stosowany do odczytu lub zapisu danych.
- W jego obrębie należy umieścić stosowną instrukcję np. `Read (x)` lub `Write (x)` (można też stosować opis słowny np. "*Drukuj x na ekran*").
- Każdy blok ma jedno wejście i jedno wyjście.

Blok decyzyjny



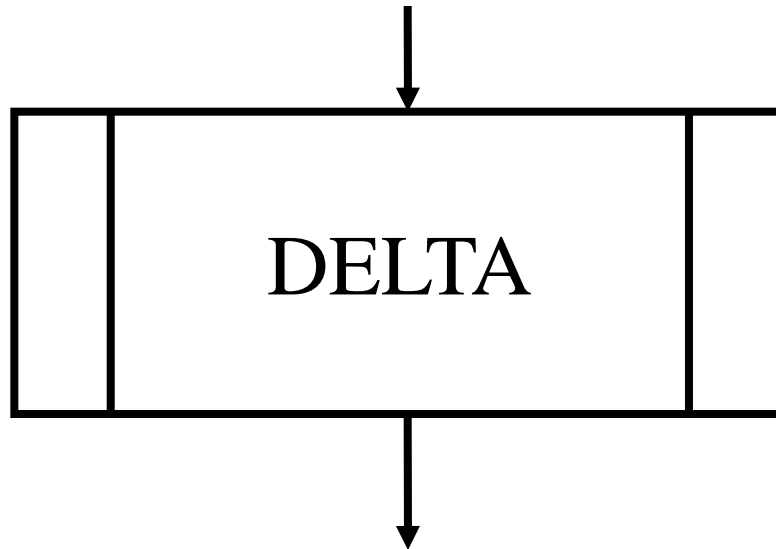
- Blok decyzyjny to instrukcja alternatywy. Wewnątrz niej znajduje się odpowiedni warunek.
- Każdy romb ma dokładnie jedną strzałkę wchodzącą oraz dokładnie dwie strzałki wychodzące.
 - Gdy warunek jest spełniony (strzałkę wychodzącą z tego wierzchołka należy opatrzyć etykietą „**TAK**”)
 - gdy warunek nie jest spełniony (Strzałka **NIE**).

Blok wielowyboru



- Blok wielowyboru umożliwia wybór, gdy mamy co najmniej trzy możliwości.
- Każdy heksagon ma dokładnie jedną strzałkę wchodzącą oraz wiele strzałek wychodzących.

Blok procedury



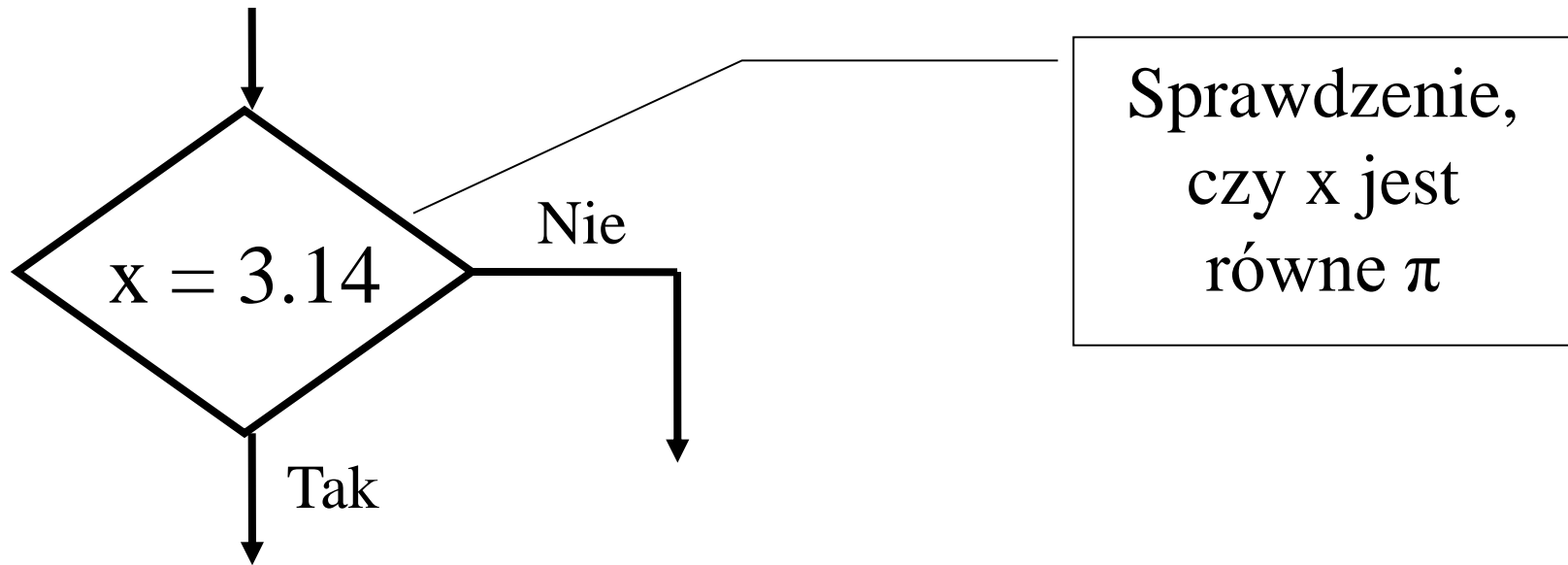
- Blok procedury symbolizuje proces, który został już kiedyś zdefiniowany i który można użyć wielokrotnie (podprogram).
- Warunkiem użycia jest wcześniejsze zdefiniowanie procesu.
- Każdy blok procedury ma jedno wejście i jedno wyjście.

Strzałka



- Strzałka wskazuje kierunek wykonywania poleceń i przesyłu danych.

Komentarz



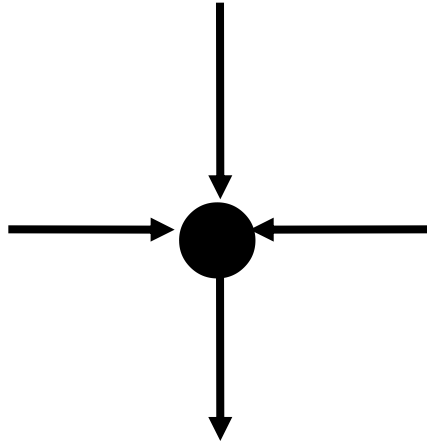
- Komentarz umożliwia umieszczenie opisu wykonywanej operacji.

Łącznik stronicowy



- Łącznik ten pozwala na łączenie fragmentów algorytmu, które nie są blisko siebie, a znajdują się na jednej kartce.
- Łączniki zwiększają czytelność algorytmu.
- Oba łączniki mają ten sam numer.

Punkt koncentracji



- Punkt koncentracji oznacza miejsce, do którego wpływa kilka strzałek.
- Rysuje się go w postaci punktu.

Reguły rysowania schematów blokowych

1. Schemat powinien być prosty i czytelny. W razie złożonego schematu, należy go podzielić na mniejsze części i zamieścić je na osobnych arkuszach.
2. W schematach składni dobrze jest używać powszechnie znanych operatorów matematycznych.
 - Schemat blokowy powinien być jednakowo dobrze rozumiany przez programistów znających różne języki programowania.
3. Do rysowania schematów dobrze jest używać szablonów.
 - Polepsza to ich czytelność.
4. Niezbędne jest komentowanie zarówno operacji, jak i kolejności ich wykonywania.
 - Komentarze powinny być krótkie, lecz dostatecznie dokładnie wyjaśniające znaczenie opisywanych elementów.

Reguły rysowania schematów blokowych

5. Należy unikać rysowania przecinających się ścieżek sterowania.
 - W razie konieczności lepiej wprowadzić odpowiednie łączniki, które pozwolą na wyeliminowanie niektórych linii wskazujących kolejność działań w algorytmie.
6. Należy dokładnie numerować arkusze, na których został rozrysowany schemat blokowy.
7. Uwzględnianie możliwości poprawek na arkuszach.
 - Wskazane jest tworzenie arkuszy tak, aby możliwe było naniesienie poprawek bez konieczności przerysowywania całego schematu.

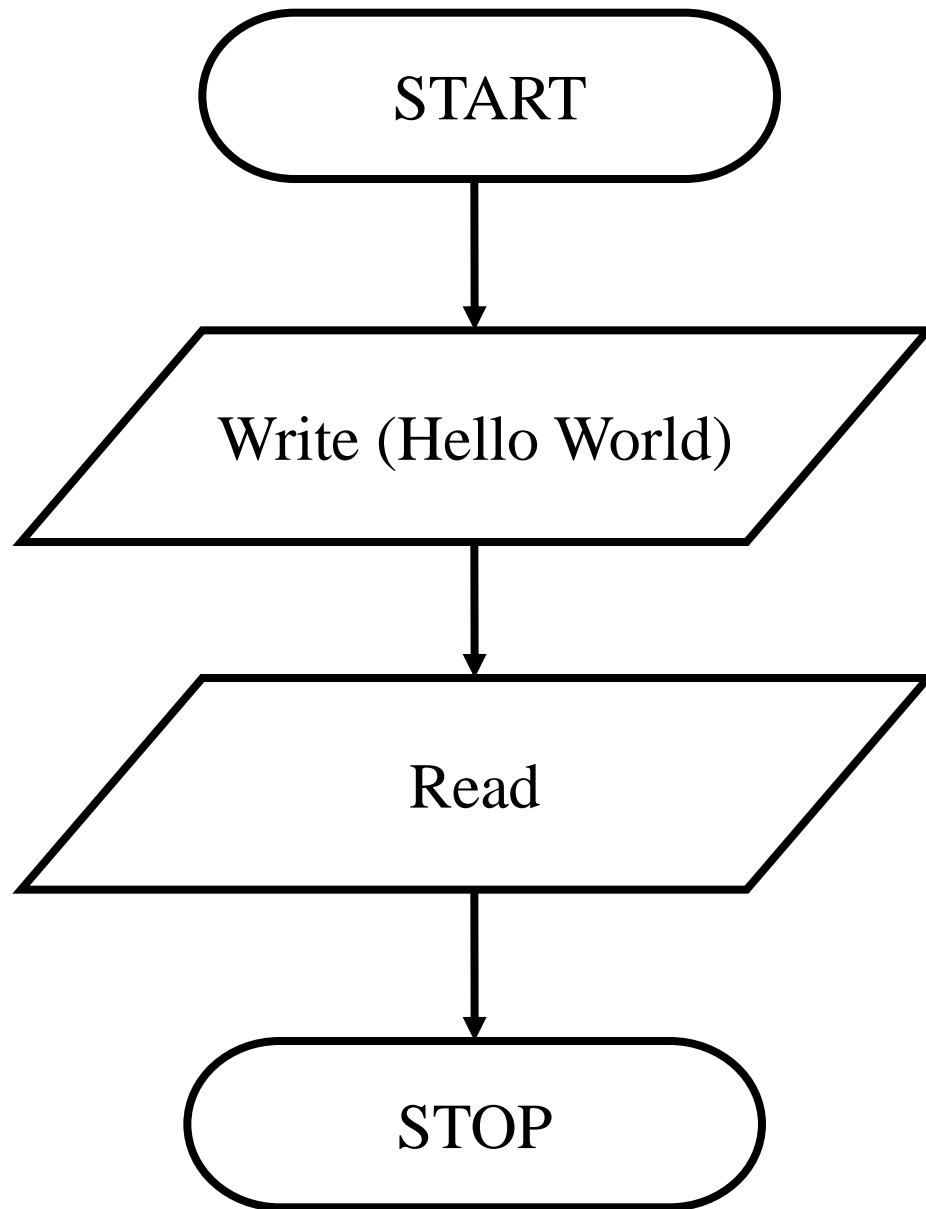
Reguły rysowania schematów blokowych

8. Należy unikać zarówno zbyt dużej szczegółowości jak i zbytniej ogólności schematów. Oczywiście operacje można zapisać w formie pseudokodu. (np. „*Odczytaj z pliku liczbę x*” - „**Fileread(x)**”).
9. Nie należy umieszczać zbyt dużej liczby operacji w jednym bloku.
10. Operacja warunkowa IF zawsze prowadzi do konieczności rozważenia dwóch dróg: gdy warunek jest spełniony i gdy nie jest.
11. Po zbudowaniu schematu blokowego nie może być strzałek, które znikąd nie wychodzą, lub donikąd nie dochodzą.
12. Bloki są ze sobą połączone.
13. Każda operacja jest umieszczona w bloku.

Algorytm Hello World

Zapis słowny

1. START
2. Zainicjuj program
3. Napisz na ekranie tekst „Hello World!”
4. Poczekaj na naciśnięcie klawisza ENTER.
5. STOP



Realizacja w Pascalu

```
program pierwszy;  
begin  
    writeln('Hello World!');  
    readln;  
end.
```

Realizacja w C/C++

```
#include <cstdlib>
#include <stdio.h>
#include <iostream>
    using namespace std;

int main(int argc, char *argv[])
{
    cout<<"Hello World";
    getchar();
    return 0;
}
```

Powtórzenie

1. Co to jest algorytm?
2. Jak możemy zapisywać algorytmy?
3. Do czego służą schematy blokowe?
4. Wymień poszczególne symbole blokowe – narysuj je i omów znaczenie.
5. Jakie znasz reguły rysowania schematów blokowych?