

Metodyki i techniki programowania

dr inż. Maciej Kusy

Katedra Podstaw Elektroniki
Wydział Elektrotechniki i Informatyki
Politechnika Rzeszowska

Elektronika i Telekomunikacja, sem. 2

Plan wykładu

- Sprawy organizacyjne
- Dane osobowe, materiały pomocnicze
- Wykaz literatury
- Zakres tematyczny wykładu, ćwiczeń i laboratorium
- Podstawowe pojęcia: algorytmika, algorytm, dane wejściowe, wynik, specyfikacja problemu, program komputerowy
- Sposoby prezentowania algorytmu
- Schemat blokowy i jego elementy
- Zasady przedstawiania algorytmu w postaci schematu blokowego

Dane osobowe, materiały pomocnicze

dr inż. Maciej Kusy

Katedra Podstaw Elektroniki

Wydział Elektrotechniki i Informatyki

Politechnika Rzeszowska

Pokój **303** budynek **A**, ul. W. Pola 2, tel: (17) 865 1113

Adres email: **mkusy@prz.edu.pl**

Strona internetowa:

<https://mkusy.v.prz.edu.pl/>

Materiały pomocnicze oraz karta przedmiotu dostępne na stronie internetowej.

Wybrane pozycje literaturowe

- Cormen T. H., Leiserson Ch. E., Rivest R. L., Stein C.: „*Wprowadzenie do algorytmów*”, WNT, Warszawa, 2004.
- Grębosz J.: „*Symfonia C++ standard. Programowanie w języku C++ orientowane obiektowo*”, EDITION 2000, Kraków, 2005.
- Grębosz J.: „*Pasja C++*”, tom1, tom2, Oficyna Kalimach, Kraków, 1997.
- Gurbiel E., Hardt-Olejniczak G., Kołczyk E., Krupicka H., Sysło M. M.: „*Informatyka. Podręcznik dla liceum ogólnokształcącego, cz. 1, cz.2*”, WSIP, Warszawa, 2002.
- Kernighan Brian W., Ritchie Dennis M.: „*Język ANSI C*”, WNT, Warszawa, 1994.
- **Kusy M.: „*Metodyki i techniki programowania. Laboratorium*”, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2009.**
- Liberty J.: „*C#. Programowanie*”, O'REILLY®, Helion, 2006.
- Lis M.: „*C#. Praktyczny kurs*”, Helion 2007.
- Matulewski J.: „*C# 3.0 i .NET 3.5. Technologia LINQ*”, Helion, 2008.
- Microsoft Visual C# 2005 Express Edition: „*Projektuj sam*”, Edycja polska Microsoft Press, Warszawa 2006.
- Perry S.C.: „*C# i .NET*”, Prentice Hall, Helion, 2006.

Zakres tematyczny przedmiotu

- Program wykładu:
 - teoria algorytmu, schemat blokowy,
 - język **C**, **C#** i środowisko **.NET**.
- Program ćwiczeń:
 - tworzenie i reprezentowanie algorytmów, język C
 - metody reprezentowania informacji,
 - implementacja dynamicznych struktur danych (tablica, stos, lista, kolejka).
- Program laboratorium:
 - zapoznanie się ze środowiskiem Visual Studio 2015,
 - implementacja algorytmów przedstawionych na ćwiczeniach w języku C, operacje bitowe, struktury,
 - definiowanie klas, pól składowych, metod właściwości w języku C#,
 - wprowadzenie interfejsu, dziedziczenia i polimorfizmu,
 - stworzenie aplikacji okienkowej.

Podstawowe pojęcia

- **Algorytmika** – dział informatyki, zajmujący się różnymi aspektami tworzenia i analizowania algorytmów.
- **Dane wejściowe** – zadana wstępnie informacja wykorzystywana przez algorytm podczas rozwiązywania problemu do osiągnięcia celu.
- **Wynik** – efekt działania algorytmu (dane wyjściowe).
- **Specyfikacja problemu** – układ dane wejściowe-wynik. Określenie warunków, jakie powinny spełniać dane wejściowe oraz wynik.
- **Program komputerowy** – implementacja algorytmu za pomocą ciągu instrukcji (sekwencji słów, symboli) w danym języku programowania zrozumiałym dla maszyny/interpretera.

Algorytm

- Synonim przepisu lub instrukcji postępowania.
- Skończony i uporządkowany zbiór jasno zdefiniowanych czynności, który prowadzi do rozwiązania jakiegoś problemu w ograniczonej liczbie kroków.
- Kryteria oceny:
 - poprawność (wykonalność),
 - efektywność (najszybsze rozwiązanie),
 - uniwersalność (rozwiązanie klasy problemów),
 - czytelność (zrozumiały i elegancki zapis),
 - prostota (preferowanie łatwiejszego rozwiązania),
 - skończoność (skończona liczba operacji),
 - porządek (wykonanie operacji w ustalonej kolejności).

Specyfikacja algorytmu

- Sprecyzowanie problemu
- Wyróżnienie danych wejściowych (wykorzystywanych przez algorytm) oraz wyniku (efekt działania algorytmu)
- Określenie, jakie działania mogą być stosowane w poszukiwanym algorytmie

Specyfikacja algorytmu obliczania pierwiastków równania kwadratowego: $a \cdot x^2 + b \cdot x + c = 0$	
Dane wejściowe:	Dowolne liczby rzeczywiste: a , b , c .
Dane wyjściowe:	<ul style="list-style-type: none"> • dwie liczby rzeczywiste x_1, x_2 dla $\Delta > 0$ • jedna liczba rzeczywista x_0 dla $\Delta = 0$ • dwie liczby zespolone x_1, x_2 dla $\Delta < 0$
Działania:	$+$, $-$, \cdot , $/$, $\sqrt{(\text{sqrt})}$

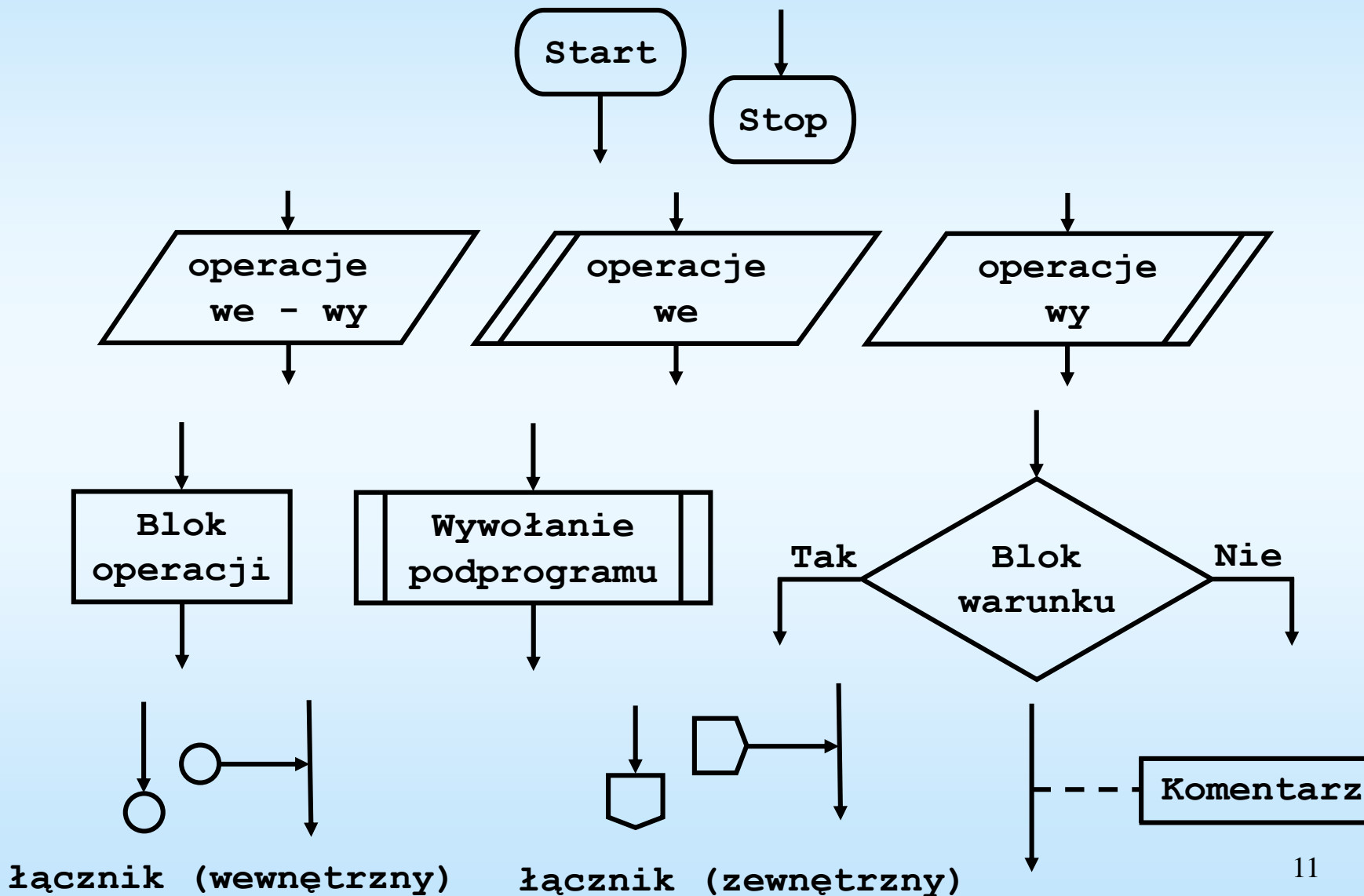
Sposoby prezentowania algorytmów

- Opis słowny – operacje przedstawione za pomocą zwykłego tekstu – ogólne przedstawienie problemu.
- Lista kroków algorytmu – przedstawienie algorytmu w skończonej liczbie punktów (kroków).
- Pseudokod – nazywany często *niby-kod*: może zawierać zarówno instrukcje zaczerpnięte z języka programowania jak i opis słowny.
- Schemat blokowy algorytmu.
- Program komputerowy (zapis algorytmu w danym języku programowania, np. C, C++, C#, Java...).

Schemat blokowy

- Graficzna reprezentacja algorytmu (ang. *flowchart*).
- W schemacie blokowym operacje (lista kroków) jakie wykonuje algorytm są reprezentowane w postaci odpowiednio połączonych figur geometrycznych (bloków).
- Kształt i opis bloku jednoznacznie precyzuje i definiuje wykonywany rodzaj operacji.
- Kolejność wykonywania operacji wyznaczają połączenia między blokami (linie zakończone strzałkami).
- Połączenia pomiędzy blokami nie mogą się krzyżować.

Elementy schematu blokowego



Zasady przedstawiania algorytmu w postaci schematu blokowego

- Operacje algorytmu należy umieszczać w odpowiednich blokach.
- Każdy schemat blokowy ma jeden blok startowy, natomiast bloków zakończenia algorytmu może być kilka.
- Wszystkie bloki muszą być ze sobą połączone.
- Kolejność wykonywania operacji wyznaczają połączenia między blokami.
- Każde połączenie jest zaczepione od danego bloku i dochodzi do następnego bloku lub innego połączenia.
- Do każdego bloku wchodzi i wychodzi jedno (lub dwa w przypadku bloku warunku) połączenie (nie dotyczy to bloku początku i końca).