

Dr inż. Andrzej Czerepicki

# Języki i paradygmaty programowania

Studia podyplomowe

Wykład 1.

# Wprowadzenie do algorytmów

# Plan wykładu

- Definicja algorytmu
- Przykłady powszechnych algorytmów
- Cechy algorytmów
- Historia algorytmów
- Sposoby zapisu algorytmu
  - zapis słowny
  - zapis w postaci schematów blokowych
  - zapis w postaci kodu źródłowego programu
- Klasyfikacja algorytmów

# 1. Definicja algorytmu

- Algorytm – ciąg instrukcji, koniecznych do wykonania określonego zadania
  - Algorytmy powstały znacznie wcześniej niż pojawiła się możliwość ich implementacji za pomocą maszyn (znane są algorytmy sprzed 3 tys lat)
  - Termin „algorytm” pochodzi od nazwiska perskiego matematyka Al-Chuwarismi (IX wiek)
  - Nie istnieje jednoznacznej ogólnie przyjętej definicji algorytmu, w zależności od źródła definicje algorytmu mogą się różnić

## 2. Przykłady powszechnie znanych algorytmów

- Algorytm niekoniecznie musi kojarzyć się z komputerem:
  - Instrukcja montażu mebli
  - Przepisy kulinarne
  - Pobranie pieniędzy z bankomatu
    - Z punktu widzenia klienta banku a bankomatu są to dwa odmienne algorytmy
  - Wysłanie SMS z komórki
  - Obliczenie NWD dwóch liczb(Algorytm Euklidesa)
  - itp.

# 3. Cechy algorytmu

- Skończoność algorytmu
  - Algorytm powinien się zakończyć po wykonaniu pewnej ilości kroków, niezależnie od tego czy zadanie zostało rozwiązane
- Uniwersalność algorytmu
  - Algorytm powinien działać na dowolnych danych wejściowych (np.  $F(x) = X * X$  ) a nie tylko na wybranych zestawach (  $F(x) = 2 + 2$  )

### 3. Cechy algorytmu (c.d.)

- Jednoznaczność i wykonalność instrukcji
  - Każda instrukcja powinna mieć jednoznacznie określony sposób wykonania oraz być wykonywalna na danym urządzeniu
- Określona kolejność wykonania instrukcji
  - Zmiana kolejności instrukcji może prowadzić do powstania zupełnie innego algorytmu

### 3. Cechy algorytmu (c.d.)

- Przykład ciągu instrukcji nie spełniającej warunku **skończoności** (więc formalnie nie będącej algorytmem)
  - 1) nalać kieliszek wódki
  - 2) wypić
  - 3) przejść do pkt. 1
- Jak należy poprawić ww. ciąg instrukcji, by spełniał warunek skończoności?



### 3. Cechy algorytmu (c.d.)

- Przykład zbioru instrukcji, nie spełniającego warunku **określonej kolejności**:
  - 1) zakochać się
  - 2) wyjść za mąż
  - 3) urodzić dziecko
- (\*) w zależności od sytuacji kolejność zdarzeń może być dowolna
  - Odpowiednio będziemy mieli do czynienia z różnymi wynikami, t.j. różnymi algorytmami

## 4. Historia algorytmów

- Algorytm pojawił się tysiące lat przez stworzeniem maszyny go implementującej
- III wiek p.n.e. - Algorytm Euklidesa
  - Obliczenie NWD (największy wspólny dzielnik dwóch liczb naturalnych)
  - Najstarszy znany matematyczny algorytm
- IX wiek – Algorytmy operacji matematycznych nad liczbami dziesiętnym
  - Al-Chuwarizmi

## 4. Historia algorytmów (c.d.)

- XIX wiek – rozwój przemysłu potrzebuje automatyzacji procesów produkcyjnych
- 1801 – Joseph Jacquard projektuje i wdraża krosno tkackie sterowane za pomocą kart przedziurkowanych
  - Jest to pierwszy „sprzęt” programowany w historii ludzkości
  - Nie ma on nic wspólnego z obliczeniami matematycznymi (!)

## 4. Historia algorytmów (c.d.)



Krosno tkackie –  
pierwsze w dziejach  
urządzenie  
programowalne

Źródło: <http://www.grafull.mtvk.pl/>

## 4. Historia algorytmów (c.d.)

- 1833 – Charles Babbage projektuje „maszynę analityczną” do realizacji algorytmów obliczeniowych
  - Maszyna będzie posiadała pamięć, moduł obliczeniowy („procesor”), moduł wejścia-wyjścia (karty perforowane)
  - Całość ma działać w systemie dziesiętnym
  - Przewidywana wydajność - jedna operacja na sekundę
  - „Komputer” jest całkowicie mechanicznym urządzeniem

## 4. Historia algorytmów (c.d.)

- Charles Babbage (c.d.)
  - Napisaniem programów do nieistniejącej jeszcze maszyny zajmuje się Ada Lovelace (Byron)
    - Pierwszym programistą była kobieta!
    - W latach 70. XX wieku powstanie język programowania ADA
  - Maszyna Babbage'a nigdy nie została zbudowana ze względu na brak ówczesnych możliwości technicznych, jego koncepcje zostały zrealizowane dopiero po 70 latach
    - W roku 1944 firma IBM konstruuje pierwszy komputer MARK-1 wzorowany na idei Babbage

## 4. Historia algorytmów (c.d.)

- XX wiek, lata 30-40:
  - Fundamentalne matematyczne prace dot. teorii algorytmów
  - Formalne udowodnienia skończoności, wykonalności, ograniczeń stosowania algorytmów
- II Wojna Światowa
  - Błyskawiczny rozwój kryptografii
  - Urządzenia wspomagające szyfrowanie wiadomości (nie są to jednak komputery)

## 4. Historia algorytmów (c.d.)

- 1945 – John von Neumann opracował koncepcję komputera posiadającego:
  - Procesor
    - wykonanie obliczeń
  - Pamięć
    - Przechowanie danych oraz programu
  - Programowe sterowanie wykonaniem algorytmu
- Architektura von Neumanna jest podstawą funkcjonowania współczesnych komputerów
- Dalej zaczyna się historia komputerów



# 5. Sposoby zapisu algorytmu

- Algorytm wymaga formalnych sposobów zapisu
- Podstawowe formy zapisu algorytmów:
  - Lista kroków
  - Schemat blokowy
  - Kod źródłowy programu

# 5. Sposoby zapisu algorytmów (c.d.)

- Lista kroków
  - Forma zapisu zorientowana na wykonanie przez człowieka
  - Przykład algorytmu „Pobranie gotówki”:
    - 1) włóż kartę
    - 2) wprowadź PIN
    - 3) jeśli PIN niepoprawny
      - 3.1) jeśli ilość prób  $< 3$ , przejdź do pkt. 2)
      - 3.2) jeśli ilość prób  $\geq 3$ , przejdź do pkt. 6)
    - 4) wprowadź kwotę
    - 5) odbierz pieniądze
    - 6) odbierz kartę

# 5. Sposoby zapisu algorytmów (c.d.)

- Lista kroków (c.d.)
  - Zalety:
    - Łatwość napisania i zrozumienia algorytmu
  - Wady:
    - Mała czytelność złożonych algorytmów
    - Dowolność traktowania poszczególnych kroków poprzez różne osoby
  - Zastosowanie (w informatyce):
    - Przeważnie do zapisu ogólnych zasad funkcjonowania systemu na wczesnych etapach projektowania

# Zadanie

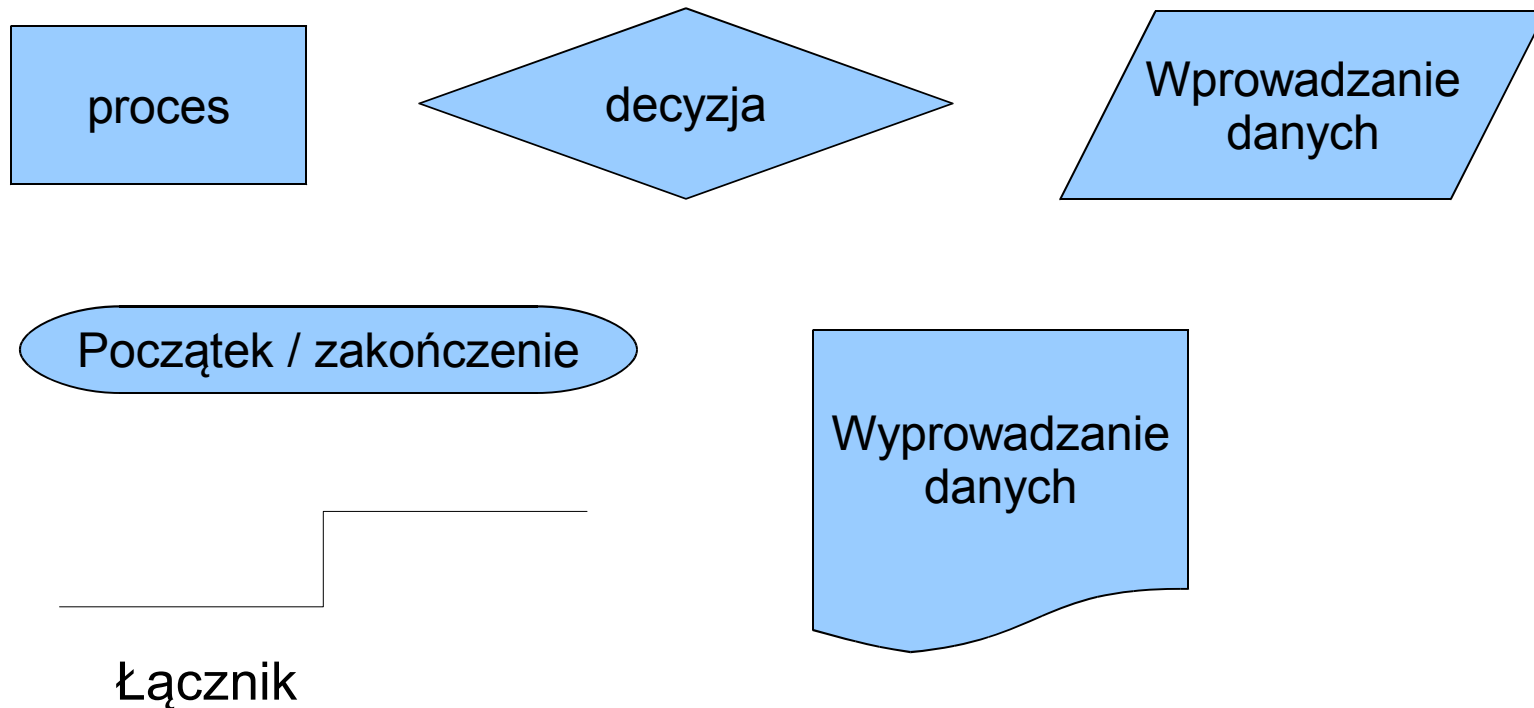
- Zmodyfikować zapis algorytmu „Pobranie gotówki” uwzględniając:
  - Wysokość pobranej kwoty nie może przekraczać środków zgromadzonych na koncie
  - Przy wydaniu gotówki bankomat najpierw zwraca kartę, dopiero potem wydaje gotówkę

## 5. Sposoby zapisu algorytmów (c.d.)

- Schematy blokowe
  - Graficzny sposób zapisu algorytmów
  - Szeroko rozpowszechniony i bardzo popularny
  - Brak ogólnego standardu spowodował pojawienie się wielu odmian różniących się w pewnych szczegółach

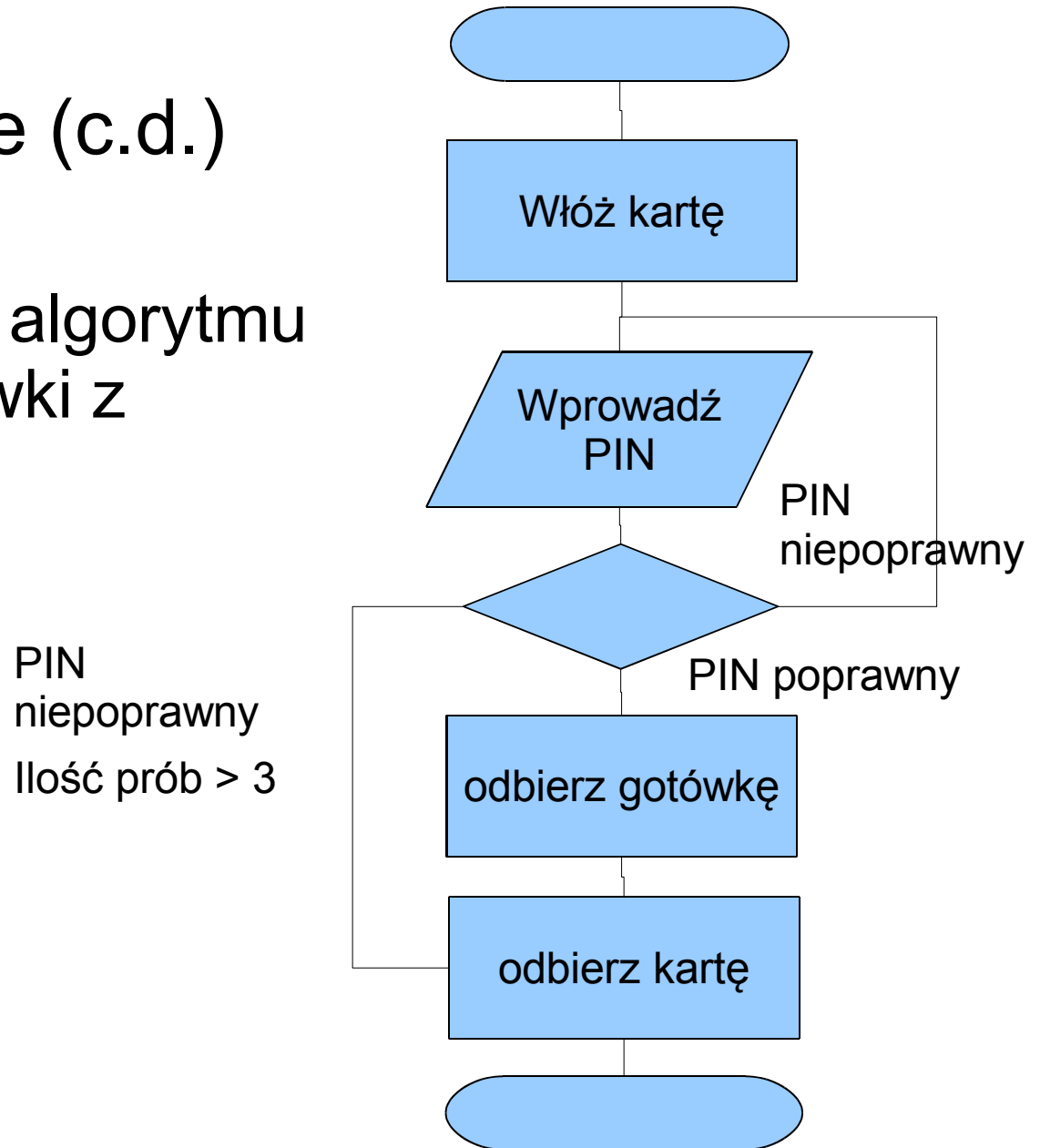
# 5. Sposoby zapisu algorytmów (c.d.)

- Schematy blokowe (c.d.)
  - Elementy podstawowe schematów blokowych



# 5. Sposoby zapisu algorytmów (c.d.)

- Schematy blokowe (c.d.)
  - Prezentacja uproszczonego algorytmu „Pobranie gotówki z bankomatu”



# 5. Sposoby zapisu algorytmów (c.d.)

## Schematy blokowe (c.d.)

- Zalety:
  - Czytelność informacji
  - Uniwersalność (niezależność od platformy sprzętowej i języka programowania)
- Wady:
  - Czasochłonność opracowania
  - Trudności reprezentacji skomplikowanych algorytmów
    - Zaleca się umieszczenie algorytmu na jednej stronie dokumentu



# 5. Sposoby zapisu algorytmów (c.d.)

- Kod źródłowy programu
  - Ciąg instrukcji w wybranym języku programowania, zapisanych w formie tekstowej
  - Jest formą zapisu zrozumiałą dla człowieka (pod warunkiem posiadania pewnych kwalifikacji)
  - Może być automatycznie i jednoznacznie konwertowany na zapis w formie wykonywalnej przez komputer
    - i odwrotnie: zapis algorytmu w formie komputerowej może być konwertowany do postaci zrozumiałej dla człowieka, lecz nie zawsze jest to jednoznaczna konwersja

## 5. Sposoby zapisu algorytmów (c.d.)

- Kod źródłowy programu (c.d.)
  - Przykład kodu źródłowego algorytmu obliczenia silni ( $N!$ ) w języku programowania PASCAL

```
procedure Silnia( n: integer );  
  var i : integer;  
begin  
  if n < 0 then  
    writeln('liczba nie moze byc ujemna!')  
  else if n = 0 then  
    w := 1  
  else  
    begin  
      w := 1;  
      for i:=1 to n do w:=w*i  
    end;  
    writeln('wynik:', w);  
end;
```

# 5. Sposoby zapisu algorytmów (c.d.)

- Kod źródłowy programu (c.d.)
  - Zalety:
    - Jednoznaczny i uniwersalny sposób zapisu algorytmu
    - Łatwo się konwertuje do postaci programu wykonywalnego na komputerze
  - Wady:
    - Wymaga znajomości wybranego języka programowania
    - Nie dla każdej platformy sprzętowej oraz programowej może być zrealizowany

# 6. Klasyfikacja algorytmów

- Nie istnieje ogólnej klasyfikacji algorytmów (!)
- Algorytmy komputerowe można warunkowo podzielić na kilka kategorii:
  - Algorytmy matematyczne
    - Obliczenia, funkcje, grafy itp.
  - Algorytmy wyszukiwania
    - Minimalna, maksymalna bądź wskazana wartość w tablicy, drzewie, tekście itp.
  - Algorytmy sortowania
  - Algorytmy kompresji danych
  - Algorytmy grafiki komputerowej

## 6. Klasyfikacja algorytmów (c.d.)

- (c.d.):
  - Algorytmy zarządzania w rozproszonych systemach komputerowych
  - Algorytmy sieci neuronowych
  - Algorytmy medyczne
  - Algorytmy genetyczne
  - itp.

# Podsumowanie

- Algorytm jest skończonym ciągiem jednoznacznych oraz wykonalnych instrukcji, mających za zadanie osiągnięcie określonego celu
- Algorytm może być wykonany przez człowieka bądź automat (komputer lub inne urządzenie)
- Algorytm wykonywany za pomocą komputera powinien być zapisany w formie programu komputerowego