



# Podstawy inżynierii oprogramowania

DR INŻ. KRYSZTOF WOJTKIEWICZ



# Modelowanie i analiza systemów informatycznych

# Czym jest model?

## MODEL:

- konstrukcja, schemat lub opis ukazujący działanie, budowę, cechy, zależności jakiegoś zjawiska lub obiektu
- system założeń, pojęć i zależności między nimi pozwalający opisać (modelować) w przybliżony sposób jakiś aspekt rzeczywistości
- reprezentacja otaczającego świata w umyśle człowieka (może być niezgodna z rzeczywistością)
- uproszczenie rzeczywistości

# Typy modeli

## Aspekt dynamiczny

- Zmiana stanu systemu
- Zmiana danych wejściowych na wyjściowe

## Aspekt statyczny

- Składowe systemu
- Zależności między składowymi

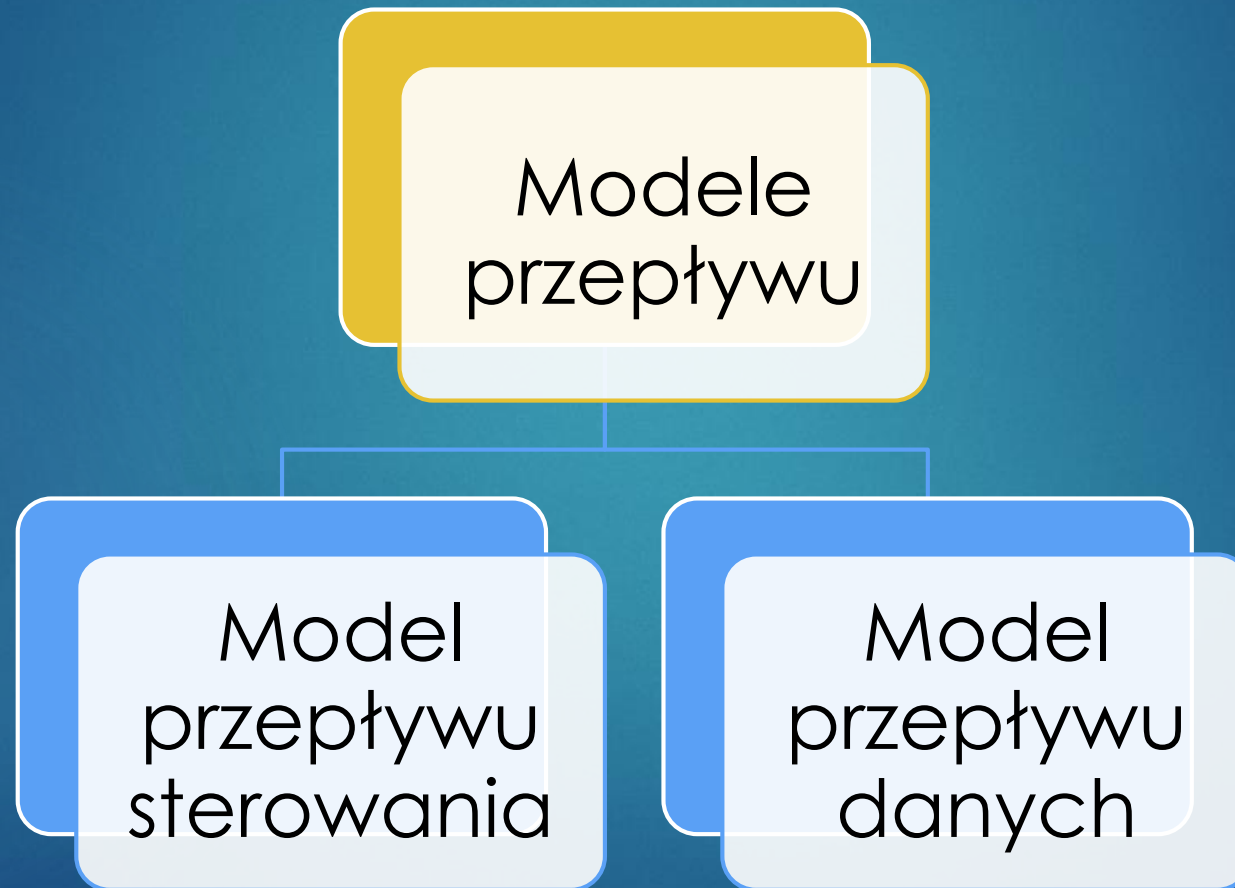
# Modele maszyn stanowych

Ukierunkowane na zmiany stanów systemu

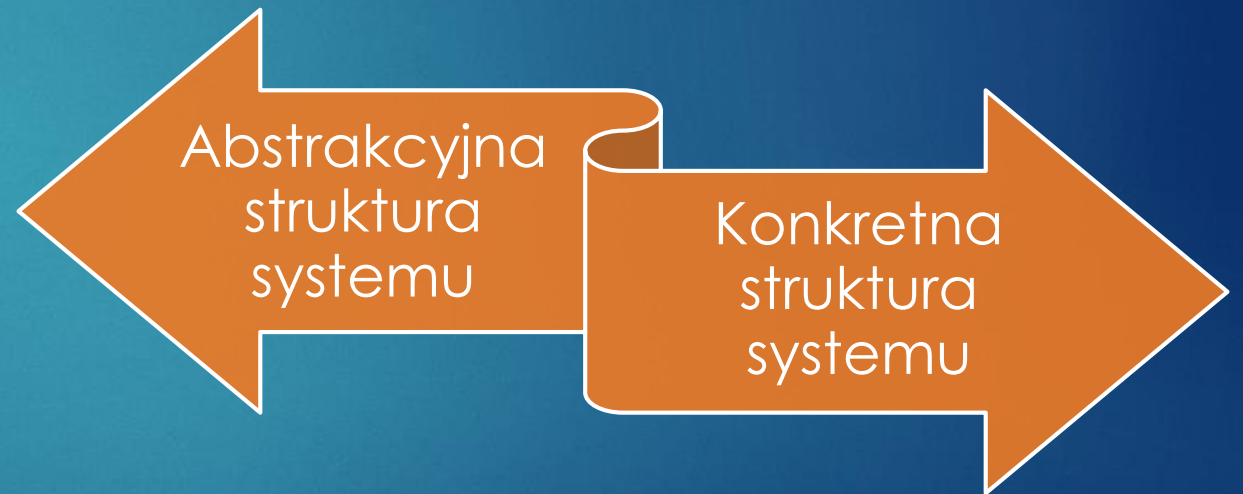
Pewna liczba stanów systemu

Przejścia między stanami odbywają się z małym udziałem przetwarzania danych

# Modele przepływu



# Modele struktury



# Modele danych



---

abstrakcyjny model, który organizuje elementy danych i standaryzuje ich wzajemne powiązania oraz właściwości obiektów w świecie rzeczywistym

---

jednoznacznie określa strukturę danych

---

abstrakcyjna formalizacja obiektów i relacji znalezionych w określonej domenie aplikacji

---

zbiór pojęć używanych przy definiowaniu takich formalizacji obiektów i relacji znalezionych w określonej domenie aplikacji



# Narzędzia CASE

(ang. Computer-Aided Software Engineering)

Edytory  
diagramów

Narzędzia  
sprawdzania  
poprawności  
modeli

Repozytoria  
danych

Słowniki danych

Narzędzia do  
raportowania

Narzędzia do  
tworzenia  
dokumentacji

Narzędzia do  
formularzy

Narzędzia  
wymiany  
danych

Generatory  
kodu

# UML



The Unified Modeling Language (UML) is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system

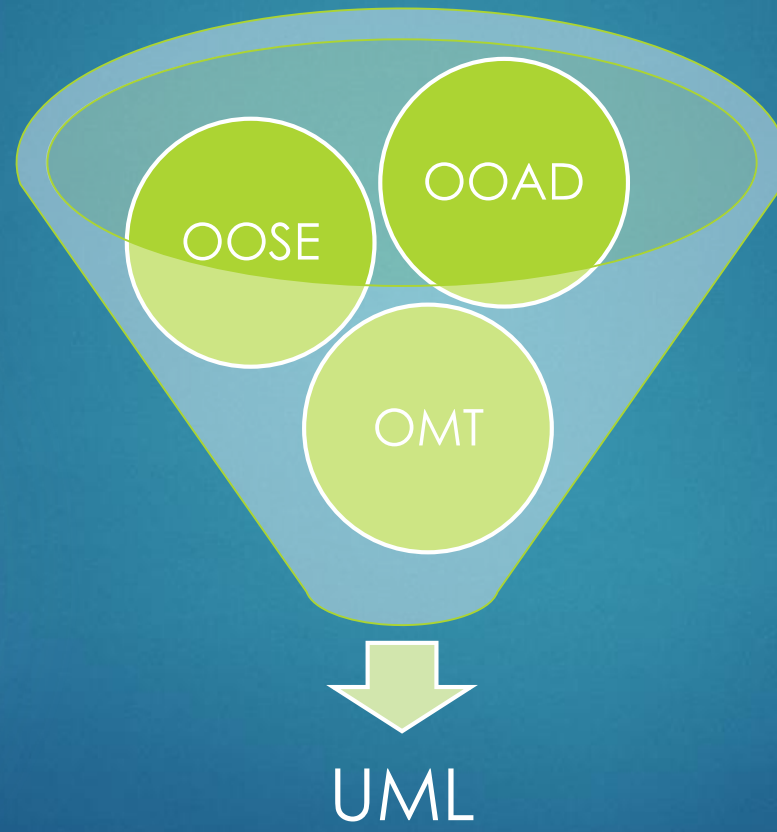
# UML

- ▶ został opracowany przez Grady Booch, Ivar Jacobson i James Rumbaugh w Rational Software w latach 1994–1995,
- ▶ w styczniu 1997 powstaje wersja 1.0 jako propozycja standardu OMG
- ▶ w sierpniu 2003 została zaprezentowana wersja 2.0
- ▶ najnowsza wersja (2.5.1) pochodzi z 2017 roku



2.5.1	December 2017
2.4.1	July 2011
2.3	May 2010
2.2	January 2009
2.1.2	October 2007
2.0	July 2005
1.5	March 2003
1.4	September 2001
1.3	February 2000
1.2	July 1999
1.1	December 1997

# UML



# UML

To nie język programowania

Narzędziem...

Metodyką projektowania

Sposobem analizy i projektowania systemów informatycznych

# UML - diagramy

## strukturalne

- Diagram pakietów
- Diagram klas
- Diagram obiektów
- Diagram struktur złożonych
- Diagram komponentów
- Diagram wdrożenia

## behawioralne

- Diagram przypadków użycia
- Diagram aktywności
- Diagram maszyny stanowej
- Diagram komunikacji
- Diagram sekwencji
- Diagram czasu
- Diagram interakcji

# UML – diagram pakietów

Pakiet jest uniwersalnym mechanizmem służącym do organizowania elementów w grupy

Grupuje elementy modelu i diagramy

Pakiet stanowi obszar nazw (*namespace*) i pozwala na oznaczenie przynależności

Używanie pakietów jest zalecane w celu utrzymania porządku w repozytorium projektu

# UML – model obiektowy

## Obiekt

- Struktura danych stanowiąca odwzorowanie bytu, posiadająca określone granice i własności, wyróżnialna w analizowanym fragmencie domeny
- Każdy obiekt posiada: tożsamość, stan, zachowanie, typ.



# UML - klasa

## Klasa

- Nazwany zbiór obiektów o podobnych własnościach: semantyce, atrybutach, zachowaniu, związkach z innymi obiektami itd.
- **Nazwany opis grupy obiektów o podobnych właściwościach.**

# UML - klasa

---

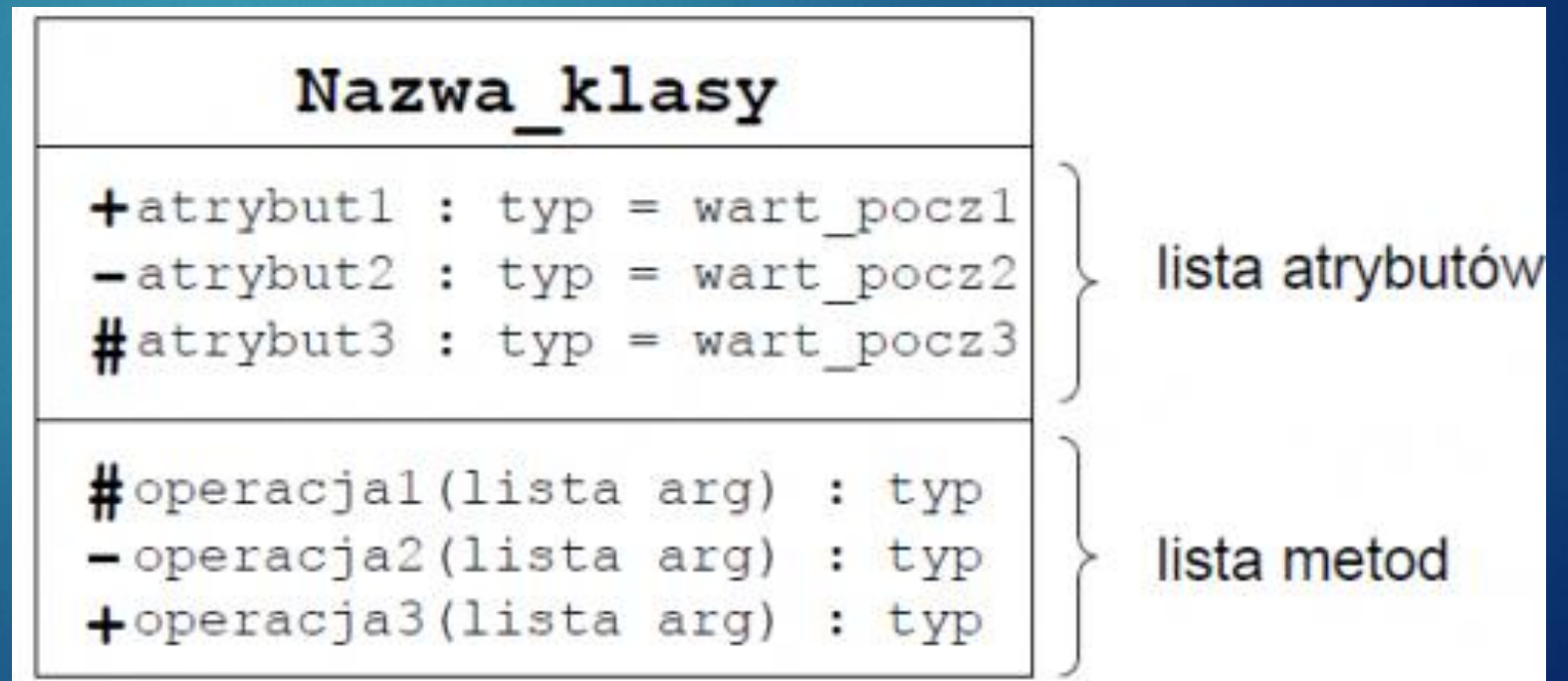
Nazwa klasy

---

Atrybuty

---

Metody



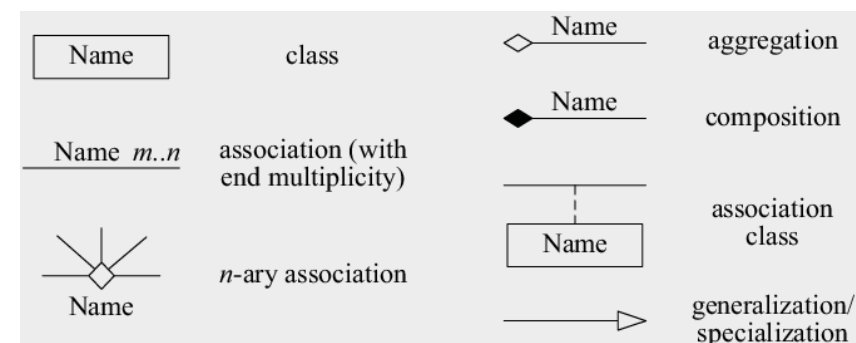
# UML – diagram klas

## Definicja klas

- Nazwa
- Atrybuty
- Metody

## Definicja związków

- Generalizacja-Specjalizacja
- Asocjacje
  - Kierunek
  - Krotność
  - Agregacja, kompozycja
  - Role
  - Nawigacja



# UML – diagram przypadków użycia

## Aktor

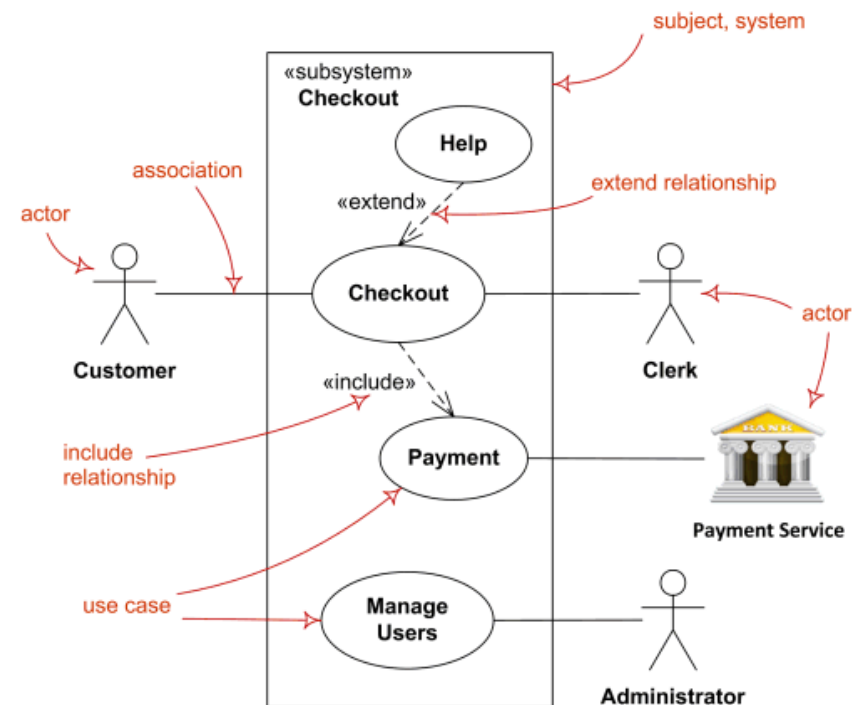
osoby, grupy osób lub elementy otoczenia systemu wchodzące w interakcje z systemem i odgrywający pewną rolę (jedna osoba może być reprezentowana przez kilku aktorów odgrywających różne role)

## Przypadek użycia

sekwencja możliwych czynności, które system może wykonać

## Powiązania

- połączenie między elementami systemu
- nie posiadają nazw
- asocjacje dwukierunkowe lub jednokierunkowe, uogólnienia, zależność, realizacja
- związki nie określają kierunku przepływu informacji
- generalizacja, jak w diagramie klas (rodzaje aktorów i przypadków użycia)



**WHAT IF I TOLD YOU**

**ITS NOT OVER YET**