

## Wprowadzenie do modelowania obiektowego

dr hab. inż. Michał Śmiałek, prof. uczelni  
dr inż. Kamil Rybiński



**Wydział  
Elektryczny**  
POLITECHNIKA WARSZAWSKA

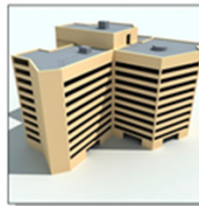
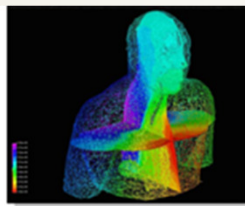
Inżynieria oprogramowania



1

### Co to jest model?

- Reprezentacja rzeczywistości, która pomaga w jej zrozumieniu



2

## Cele modelowania

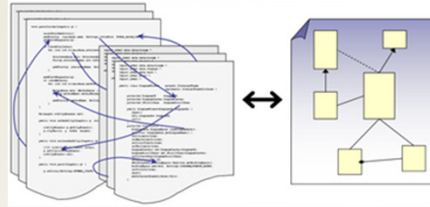
- Testowanie właściwości fizycznych obiektów
  - Możliwość udoskonalenia konstrukcji bez ponoszenia kosztów wykonania prototypu
- Walidacja i komunikacja z klientem
  - Prezentacja możliwych efektów pracy przed wykonaniem właściwego systemu
- Redukcja złożoności
  - Uproszczona reprezentacja rzeczywistości
  - Pominięcie elementów nieistotnych w danym kontekście
  - Lepsze zrozumienie złożonych zagadnień

## Zasada abstrakcji



- Koncentracja na rzeczach ważnych
- Klasyfikacja
  - Grupowanie elementów w kategorie
- Generalizacja
  - Tworzenie pojęć ogólnych

## Modele dla kodu



- Zrozumienie celu budowy systemu oraz sposobu jego realizacji
- Ułatwienie komunikacji pomiędzy twórcami systemu oraz zamawiającym
- Ułatwienie zarządzania realizacją systemu oraz zarządzanie ryzykiem
- Ułatwienie dokumentowania systemu

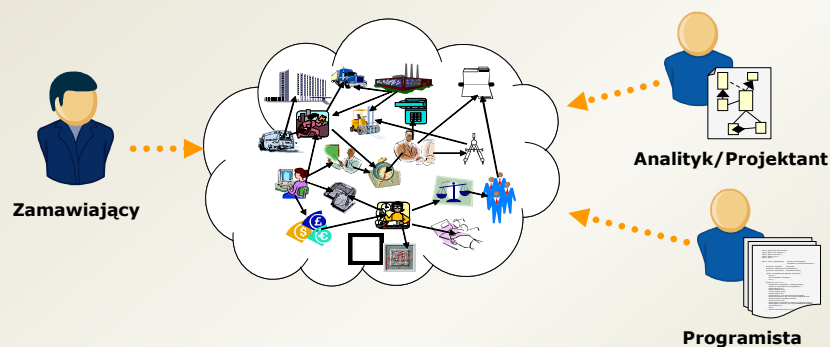
## Język UML

- Programowanie obiektowe
  - Języki obiektowe: C#, Java, Python, TypeScript, ...
- Modelowane obiektowe
  - W latach 90. pojawiło się wiele języków modelowania
  - Język UML (Unified Modeling Language) unifikuje różne języki
  - Twórcy: Grady Booch, James Rumbaugh, Ivar Jacobson
  - Obecnie jest to standard zarządzany przez Object Management Group (m.in. ISO/IEC 19505)

## Zasady dobrego modelowania

- Dobre praktyki modelowania promowane przez twórców UMLa
  - „Podjęcie decyzji, jakie modele tworzyć, ma wielki wpływ na to, w jaki sposób zaatakujemy problem i jaki kształt przyjmie rozwiązanie.”
  - „Każdy model może być opracowany na różnych poziomach szczegółowości.”
  - „Najlepsze modele odpowiadają rzeczywistości.”
  - „Żaden jeden model nie jest wystarczający. Niewielka liczba niemal niezależnych modeli to najlepsze rozwiązanie w wypadku każdego niebanalnego systemu.”

## Obiekt jako podstawa modelowania

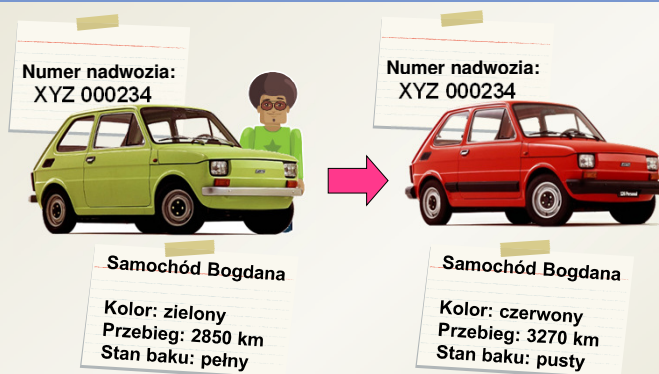


- Wyodrębnienie istotnych elementów rzeczywistości
- Dobra płaszczyzna porozumienia w projekcie
- Naturalny sposób rozumienia złożonych zagadnień

## Modelowanie obiektowe

- Modele odzwierciedlają różne aspekty modelowanej dziedziny
- Modelowanie obiektowe polega na:
  - znajdowaniu interesujących nas konkretnych obiektów w danej dziedzinie,
  - opisywaniu struktury i sposobu działania tych obiektów,
  - klasyfikacji i generalizacji obiektów,
  - znajdowaniu powiązań między nimi,
  - opisywaniu dynamicznych aspektów współpracy pomiędzy obiektami.

## Co to jest obiekt?



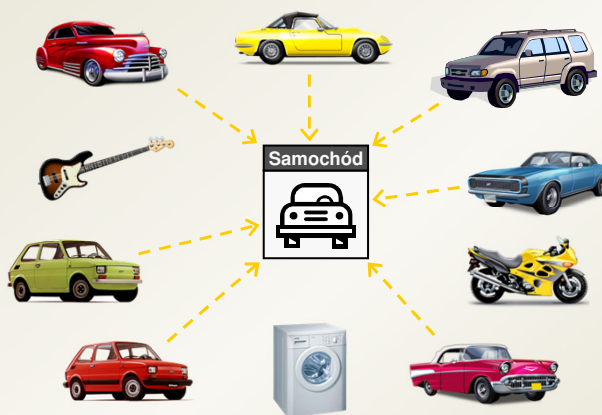
- Obiekty posiadają: tożsamość, stan i zachowanie
  - Tożsamość - unikalna cecha obiektu
  - Stan - aktualne wartości właściwości obiektu
  - Zachowanie - zbiór usług wykonywanych przez obiekt

## Notacja dla obiektów

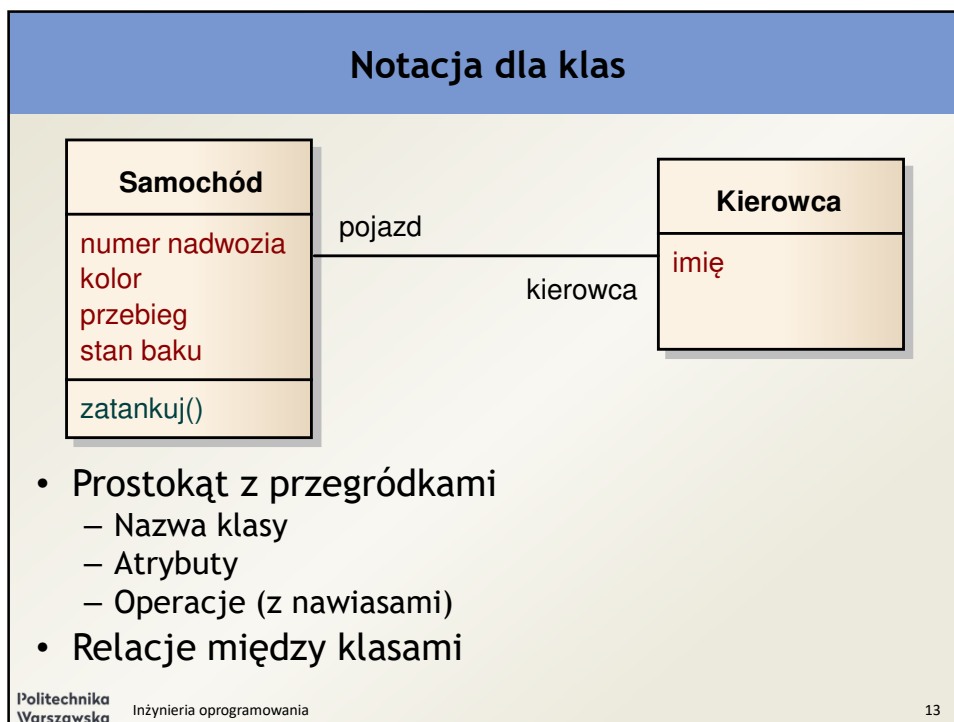


- Ikona: prostokąt z podkreśloną nazwą i ew. typem
- Określenie stanu (wartości właściwości)
- Łączniki między obiektami określające relacje

## Klasy obiektów



- Klasa to opis grupy obiektów, które mają taki sam zestaw właściwości oraz sposób zachowania

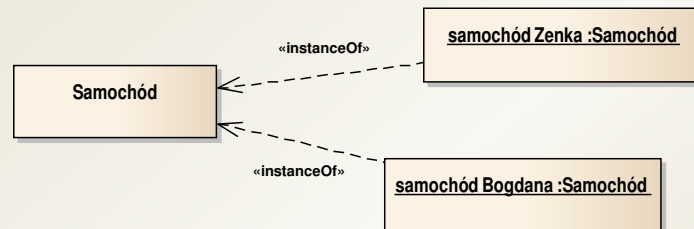


13



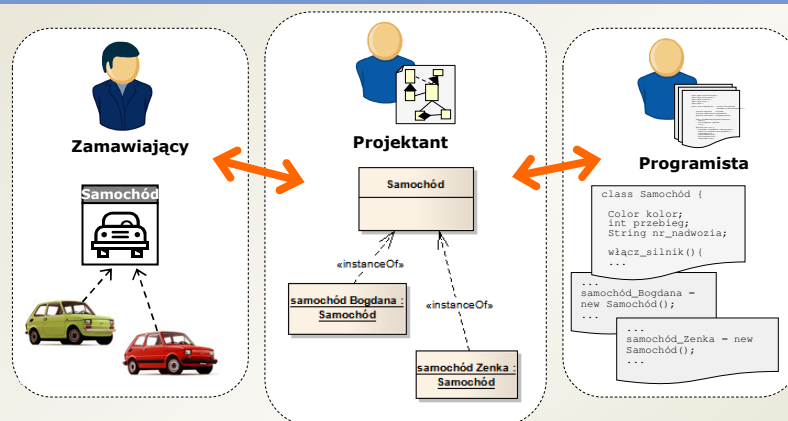
14

## Obiekty jako instancje klasy



- Obiekt może posiadać typ zgodny z określoną klasą
- Na diagramie możemy zaznaczyć klasę i jej instancje

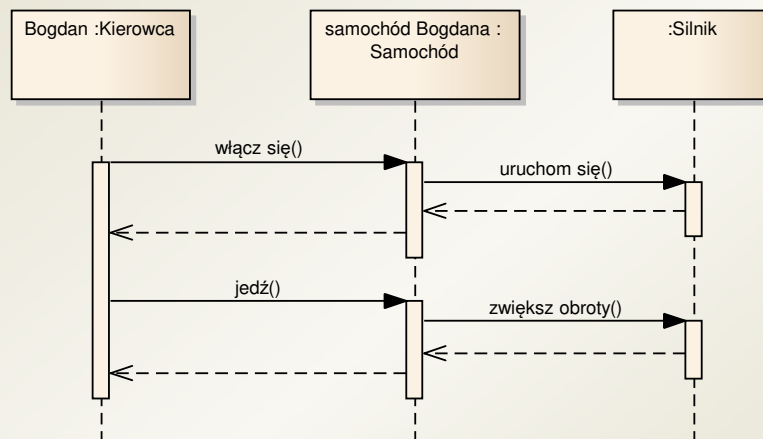
## Klasy w różnych perspektywach



- Różne poziomy szczegółowości modeli w IO
  - Perspektywa pojęciowa, perspektywa specyfikacyjna, perspektywa implementacyjna



## Współpraca obiektów



- Diagram interakcji: przesyłanie komunikatów
  - Komunikat: prośba o wykonanie usługi lub odpowiedź

## Warunki wykonania usług przez obiekty

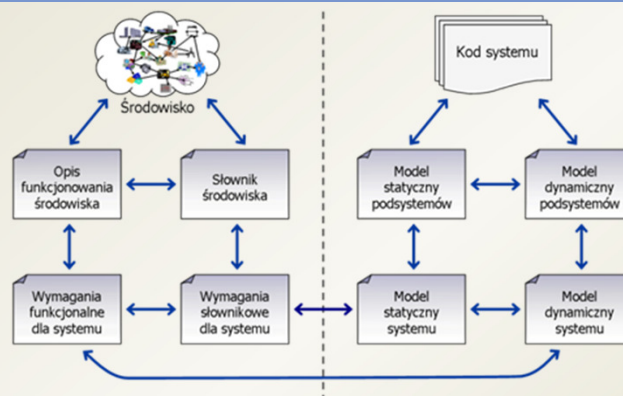
- Na wykonanie usługi przez obiekt mają wpływ trzy czynniki
  - Aktualny stan obiektu (w momencie odebrania komunikatu od innego obiektu)
  - Parametry komunikatu (lista wartości lub obiektów przekazywanych adresatowi komunikatu)
  - Stan innych obiektów (gdy konieczne skorzystanie z usług innych obiektów)

## Obiekt jako „czarna skrzynka”



- Obiekty nie wiedzą jak działają inne obiekty
- Ważny jest tylko oczekiwany rezultat wykonania usługi
  - Nie jest istotne jak jest wykonana usługa
  - Istotne jest tylko wejście i oczekiwane wyjście

## Modele w procesie inżynierii oprogramowania



- Większość dyscyplin korzysta z modeli
  - Modele struktury (słowniki, modele statyczne)
  - Modele działania (opisy procesów, funkcje, modele dynamiki)