



Podstawy programowania III

WYKŁAD 4

Jan Kazimirski



Podstawy UML-a



UML

- UML – Unified Modeling Language – formalny język modelowania systemu informatycznego.
- Aktualna wersja 2.3
- Stosuje paradygmat obiektowy.
- Graficzna reprezentacja systemu – diagramy.
- Różne diagramy koncentrują się na różnych aspektach modelowanego systemu.



Wybrane diagramy UML

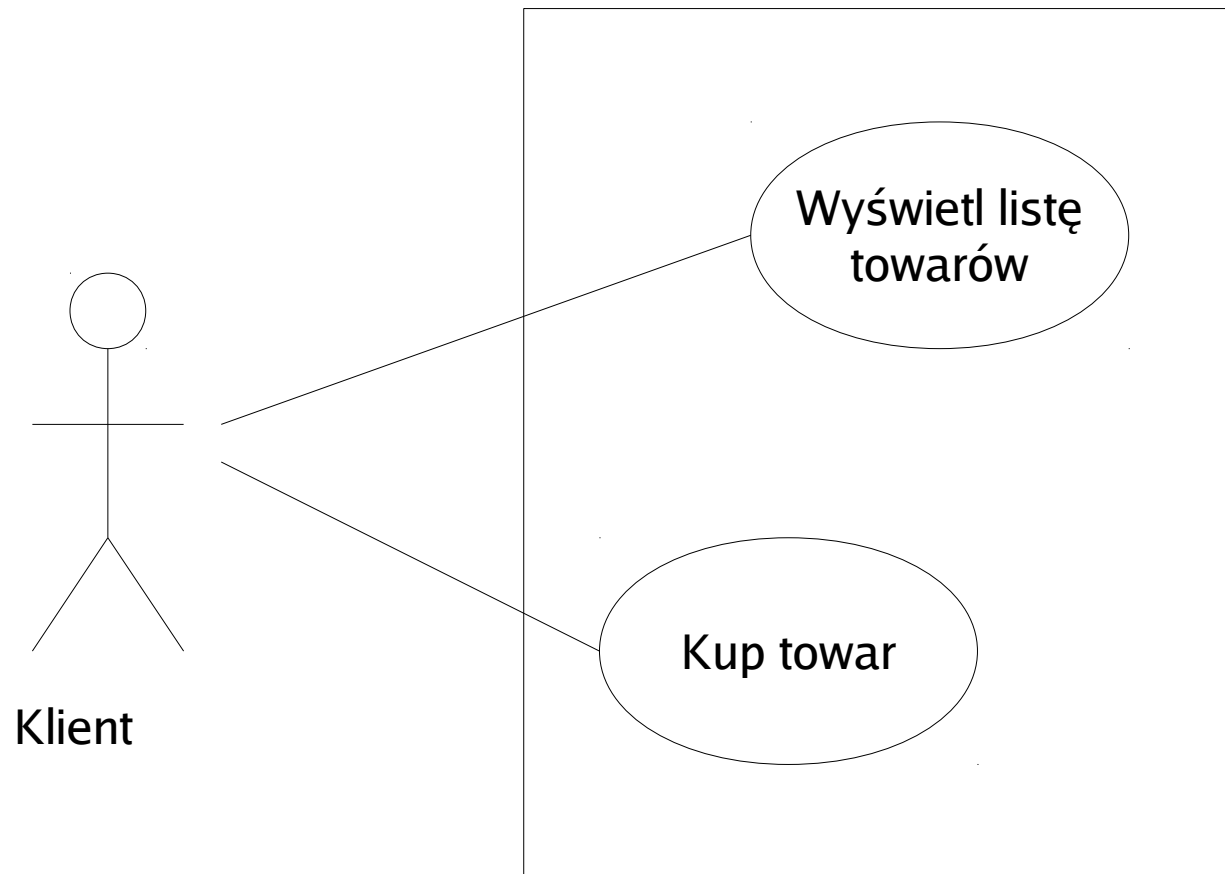
- Diagramy struktury
 - Diagramy klas, obiektów, pakietów
 - Diagramy komponentów i rozlokowania
- Diagramy dynamiki (zachowań)
 - Diagram przypadków użycia
 - Diagram czynności
 - Diagram maszyny stanowej
 - Diagram sekwencji



Diagram przypadków użycia

- Określa jakie usługi system udostępnia swoim użytkownikom.
- Podstawowe pojęcia:
 - Aktor – Kto korzysta z systemu
 - Przypadek użycia – Co system może zrobić
 - Związki – powiązania między elementami modelu.

Diagram przypadków użycia c.d.





Scenariusze

- Diagram przypadków użycia są bardzo ogólne
- Dokładniejszą dokumentację przypadku użycia stanowią „scenariusze”
- Scenariusz:
 - szczegółowy przebieg zdarzeń związanych z realizacją danego przypadku użycia,
 - warunki wstępne i końcowe
 - inne informacje związane z danym przypadkiem użycia



Diagram klas

- Statyczny obraz struktury systemu – klasy i powiązania między klasami
- Jeden z najczęściej stosowanych diagramów
- Podobnym diagramem, ale reprezentującym strukturę systemu w określonej chwili jest **diagram obiektów**

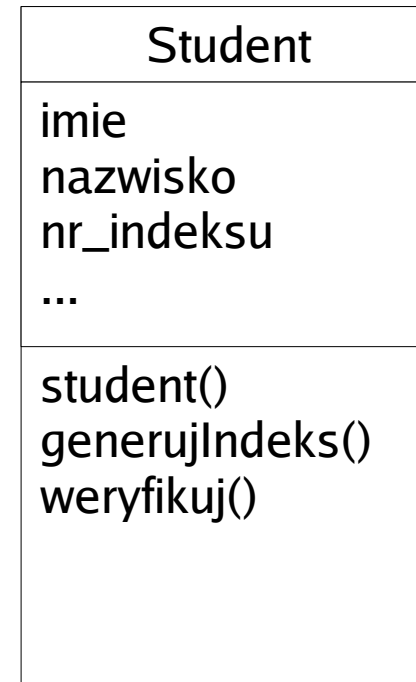


Klasa

Nazwa klasy →

Atrybuty →

Metody →





Notacja atrybutów i metod

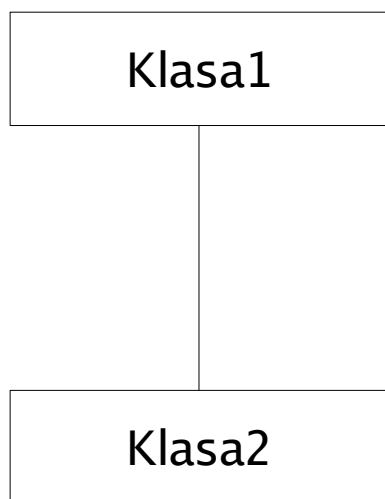
- Widoczność: prywatne (-), publiczne (+), chronione (#)
- Statyczne – podkreślone
- Atrybuty – wyszczególnienie typu, wartości początkowe
- Metody – wyszczególnienie parametrów i ich typów, wartości domyślnych parametrów



Notacja atrybutów i metod c.d.

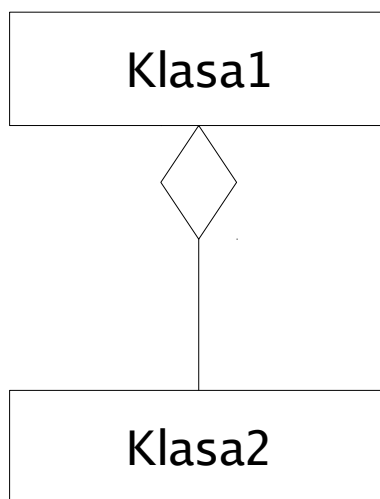
Student
- imie : String - nazwisko : String - nr_indeksu : Integer - grupa : Integer = 0
+ student() + generujIndeks() : Integer # weryfikuj(nr_ind : Integer) : Boolean

Związki między klasami

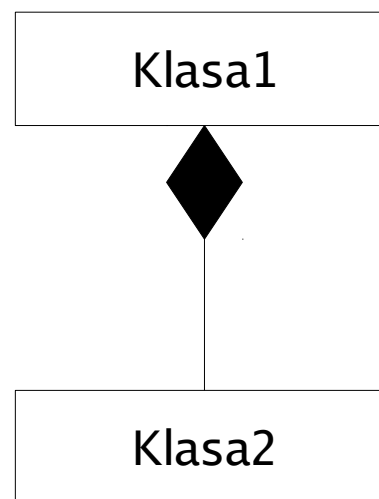


Asocjacja

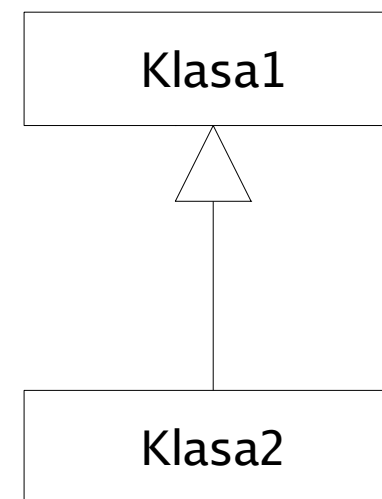
Dodatkowe symbole mogą dokładniej specyfikować rodzaj



Agregacja



Kompozycja
(Agregacja
zupełna)



Uogólnienie



Diagram czynności

- Diagram opisujący dynamikę systemu
- Przedstawia przepływy sterowania oraz danych pomiędzy uporządkowanymi ciągami czynności
- Zastosowania:
 - modelowanie procesów biznesowych
 - przedstawianie scenariuszy przypadków użycia
 - przedstawianie algorytmów i sekwencji operacji

Diagram czynności c.d.

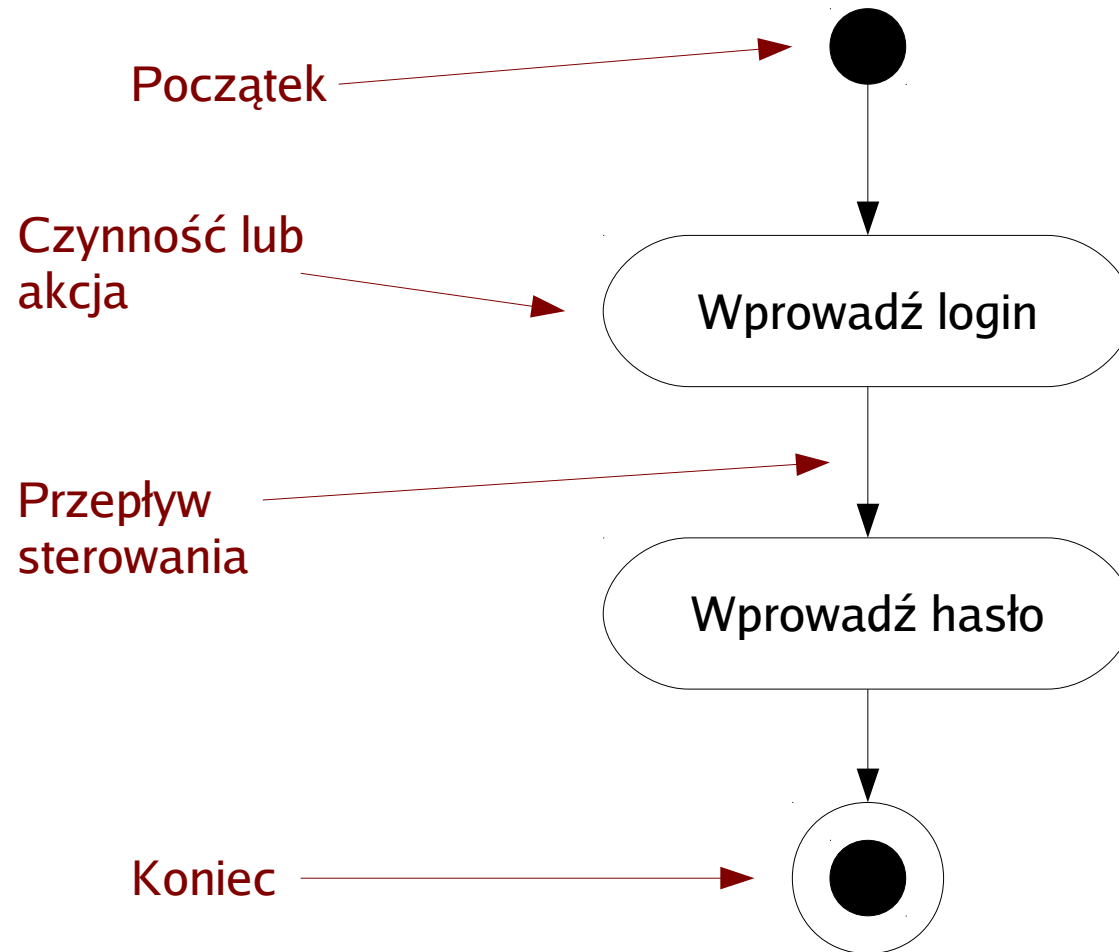




Diagram maszyny stanowej

- Ilustruje dopuszczalne stany i przejścia między stanami określonego obiektu.
- Model maszyny stanowej
 - modelowany układ znajduje się w jednym z dopuszczalnych stanów
 - pomiędzy stanami dopuszczalne są określone przejścia (zmiany stanów)

Diagram maszyny stanowej c.d.

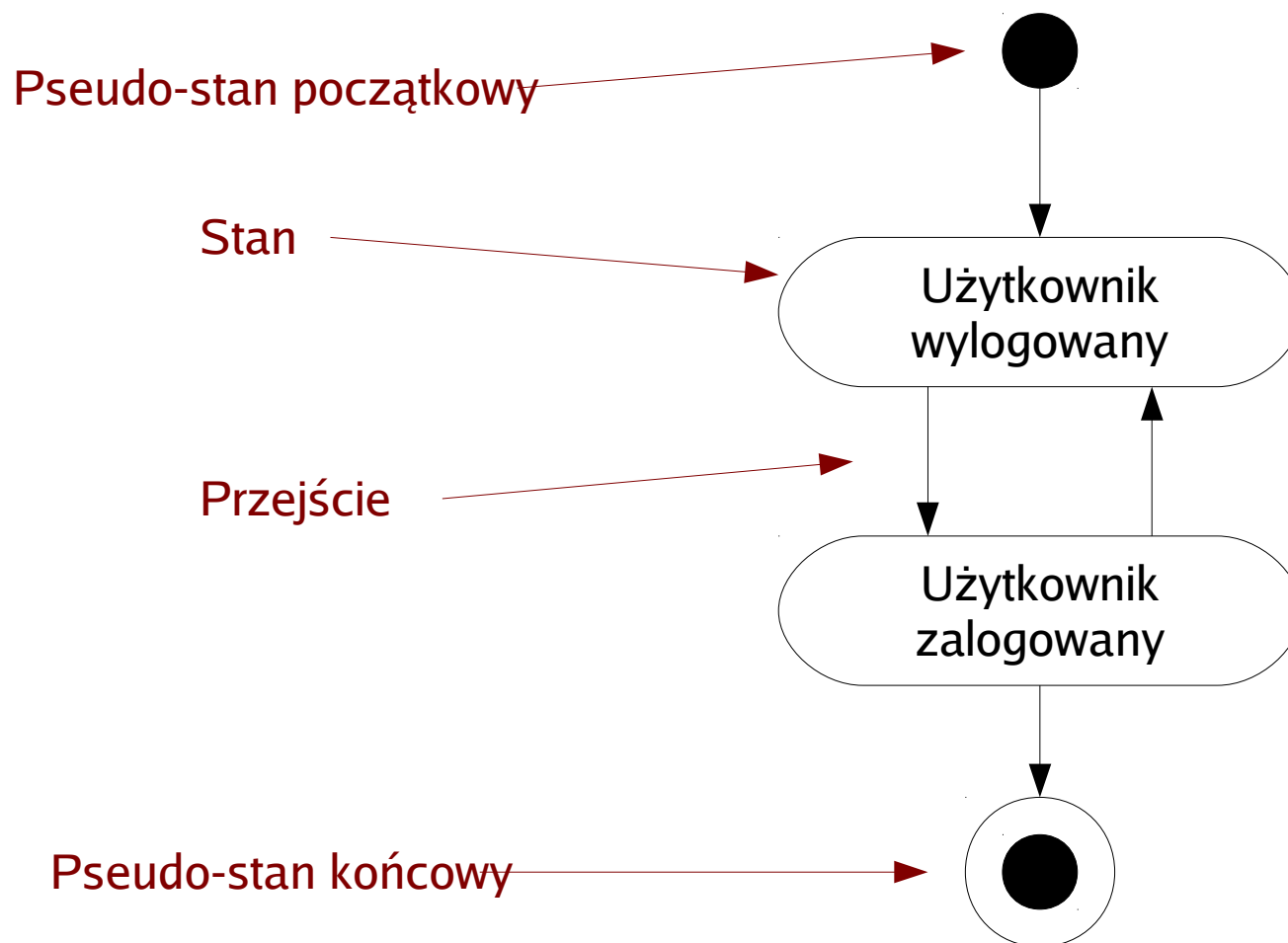
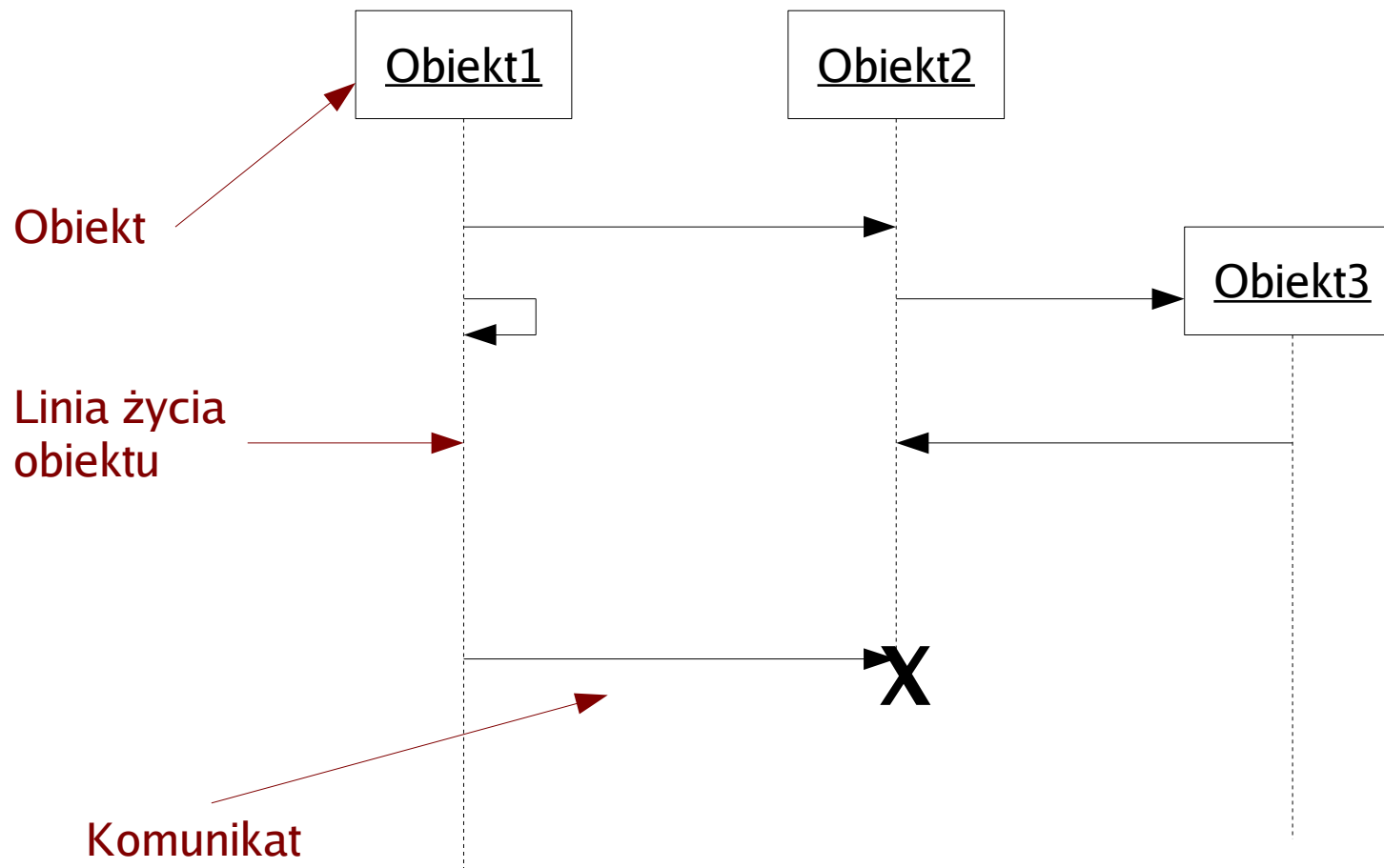




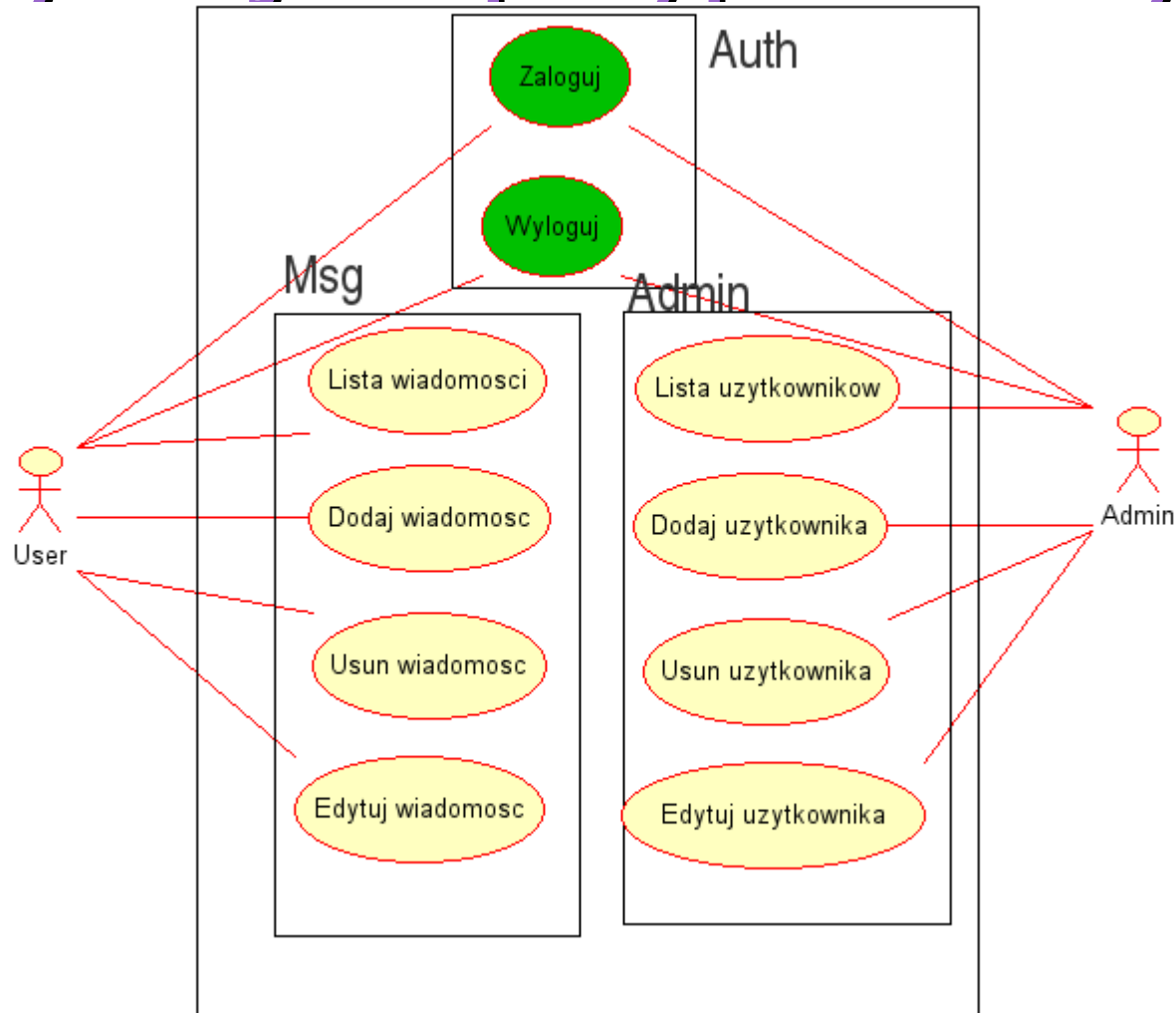
Diagram sekwencji

- Przedstawia interakcje między elementami systemu
- Zgodnie z modelem obiektowym system reprezentowany jest przez zbiór obiektów i wymienianych między nimi komunikatów

Diagram sekwencji c.d.



Autoryzacja – przypadki użycia





Scenariusz - Zaloguj

- Przypadek użycia: Zaloguj
- Aktorzy: User, Admin
- Warunki wstępne: użytkownik niezalogowany
- Warunki końcowe: użytkownik zalogowany
- Główny przepływ zdarzeń:
 1. System wyświetla formularz logowania
 2. Użytkownik wpisuje login i hasło
 3. Użytkownik zatwierdza dane
 4. System wyświetla informację o zalogowanym użytkowniku
- Alternatywny przebieg zdarzeń:
 - 4a. W przypadku niepoprawnych danych system ponownie wyświetla formularz logowania



Scenariusz - Wyloguj

- Przypadek użycia: Wyloguj
- Aktorzy: User, Admin
- Warunki wstępne: użytkownik zalogowany
- Warunki końcowe: użytkownik niezalogowany
- Główny przepływ zdarzeń:
 1. Użytkownik wybiera z panelu operację wylogowania
 2. System wyświetla formularz logowania
- Alternatywny przebieg zdarzeń:

Diagram czynności - Zaloguj

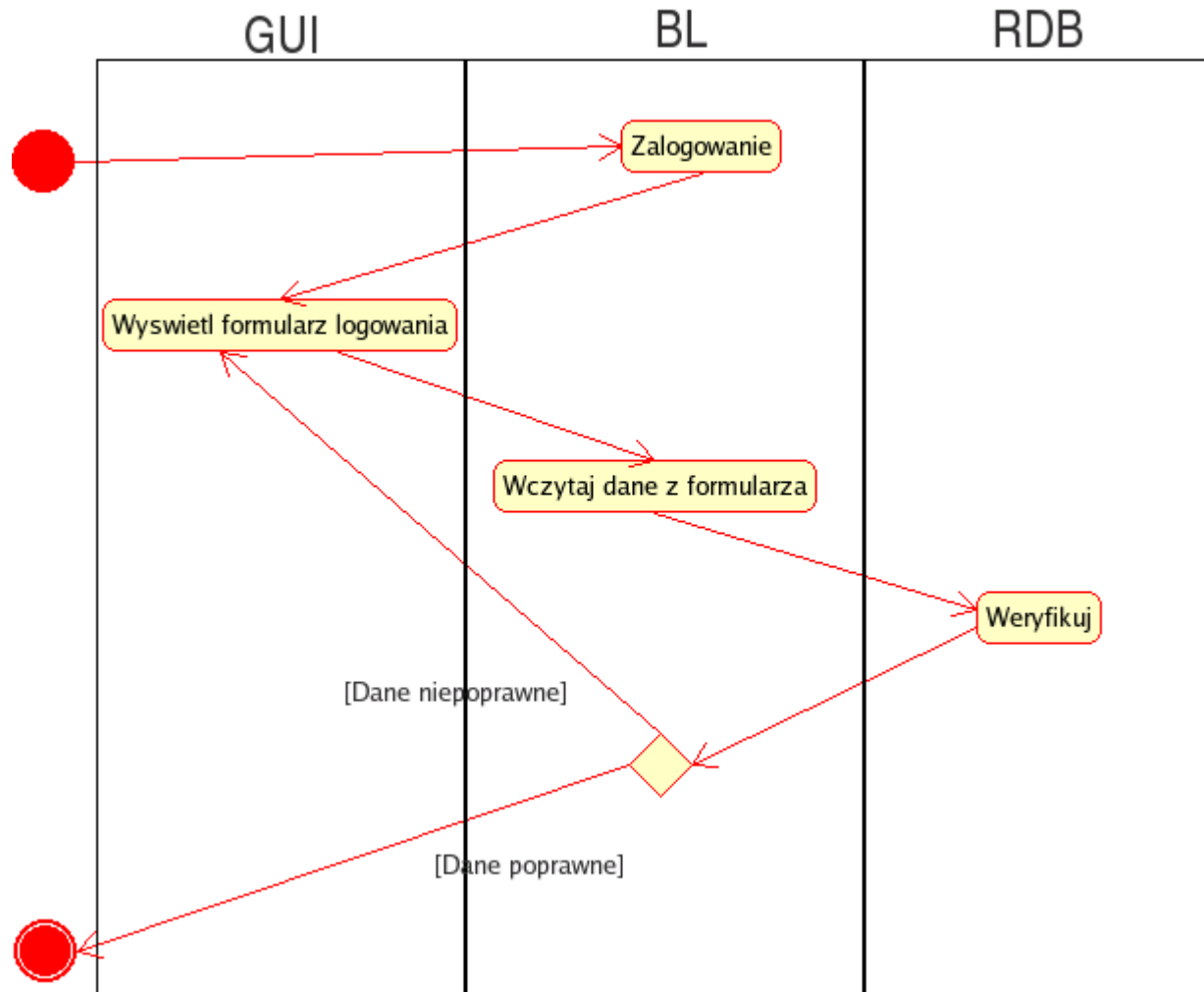


Diagram czynności - Wyloguj

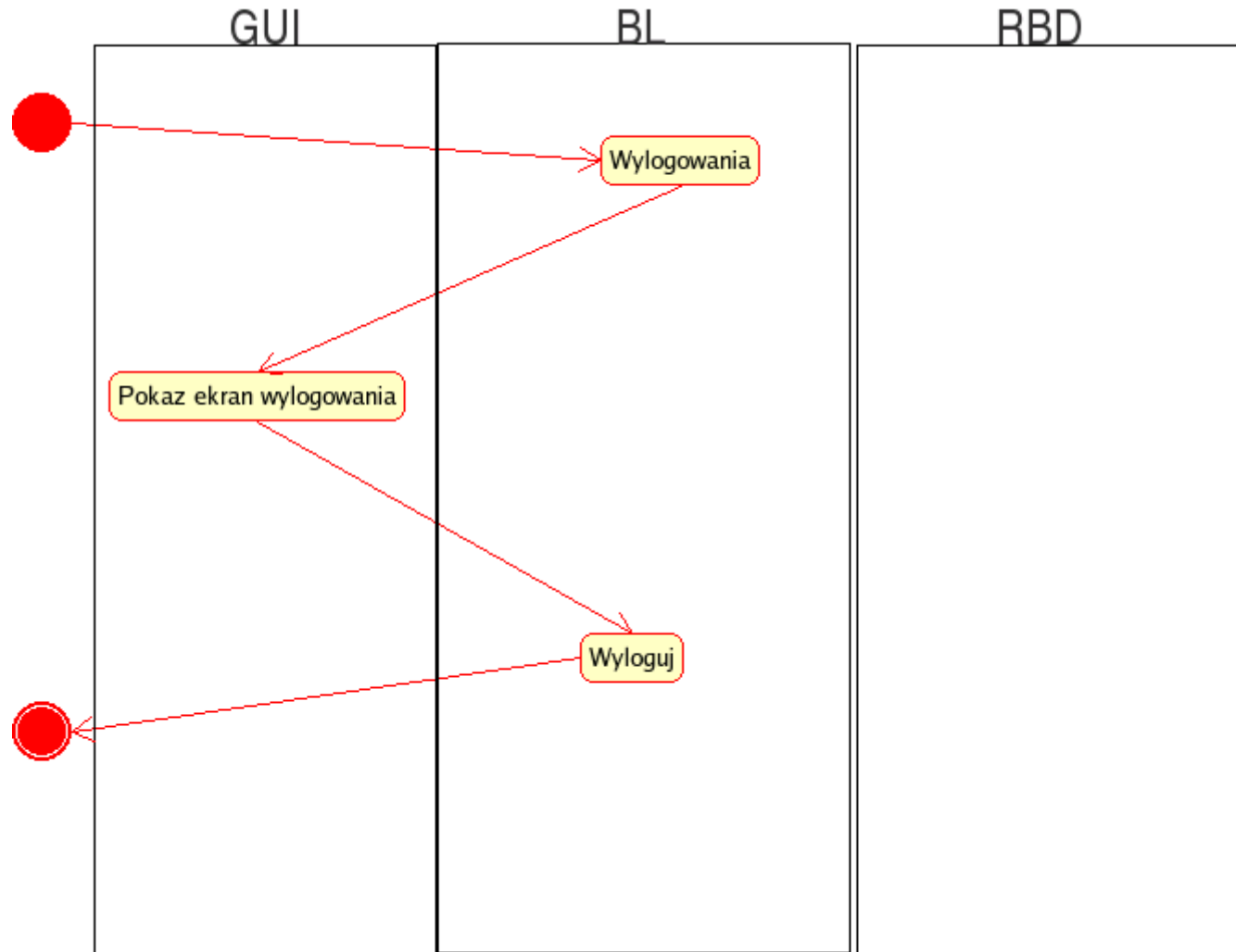


Diagram klas

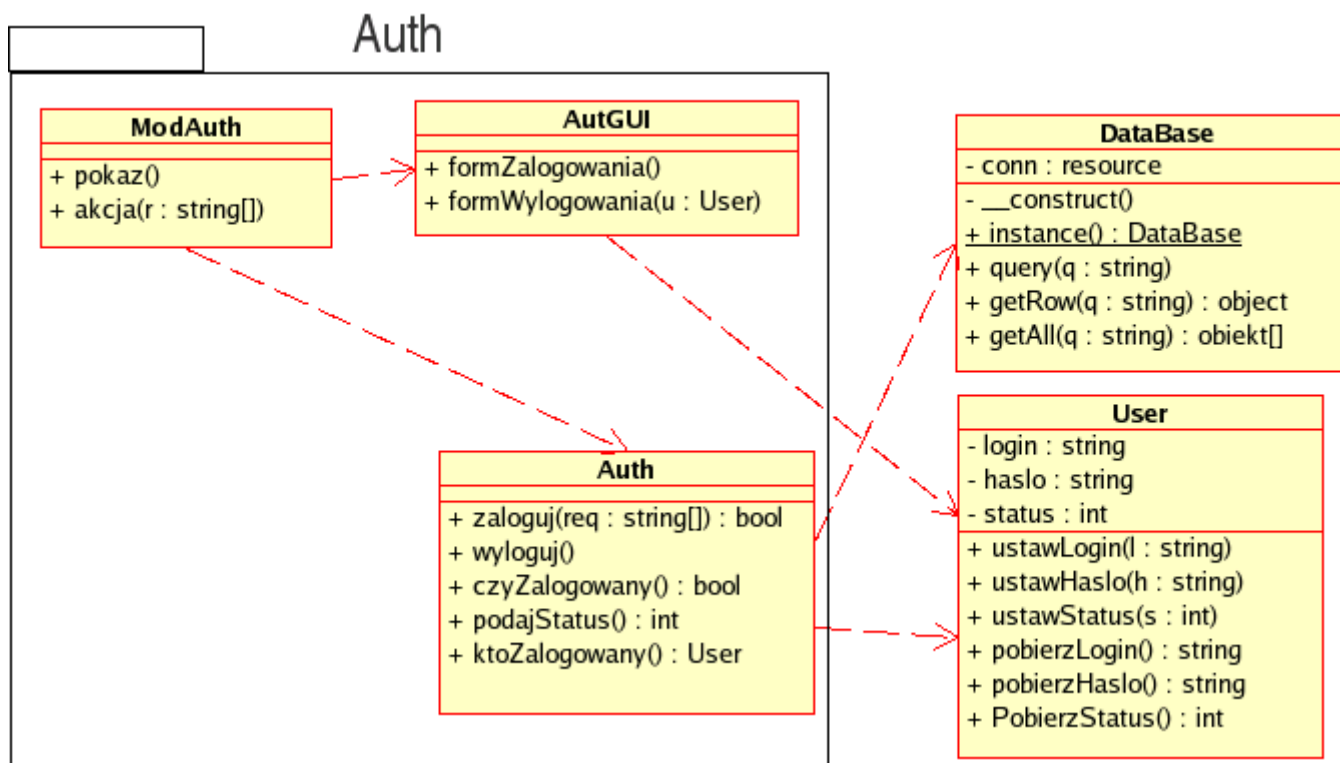


Diagram sekwencji - Zaloguj

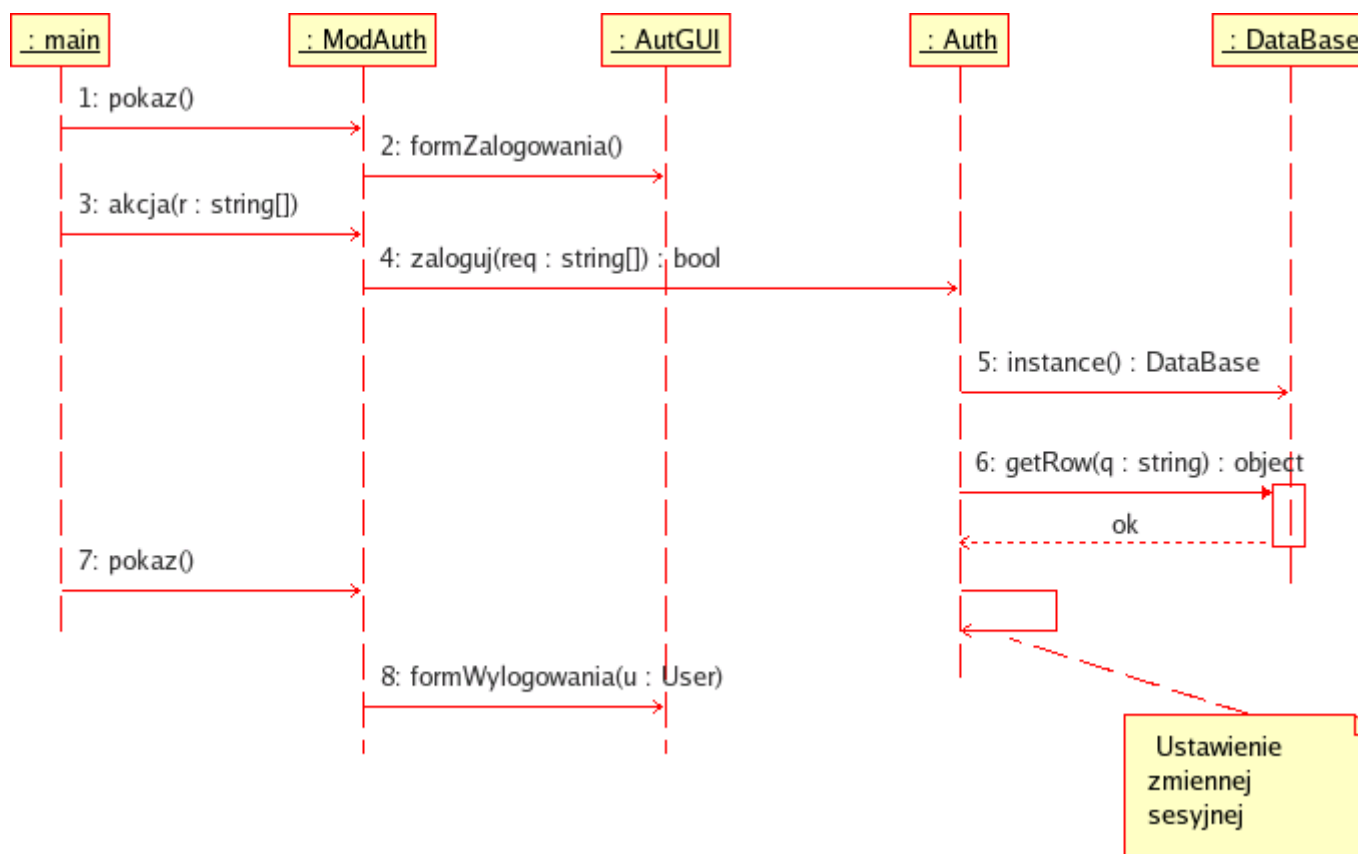
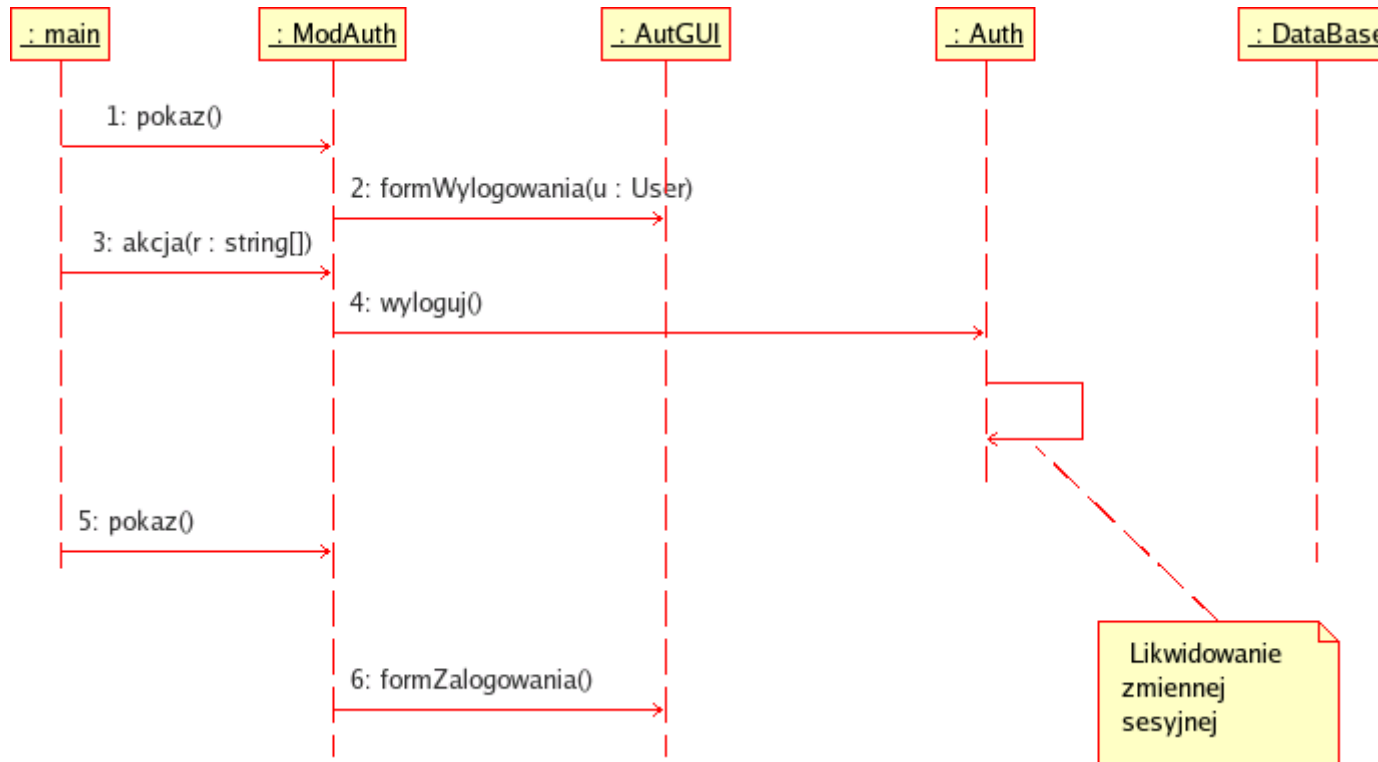


Diagram sekwencji - Wyloguj





Projektowanie aplikacji

- Analiza wymagań – identyfikacja aktorów i przypadków użycia
- Tworzenie opisów (scenariuszy) przypadków użycia.
- Identyfikowanie klas, atrybutów i operacji
- Identyfikowanie zależności między klasami
- Analiza dynamiki systemu – diagramy czynności, sekwencji, stanów



Scenariusze

- Tworzone na bazie „opowieści użytkownika” - tzn. opisów oczekiwanego zachowania systemu.
- System „punktowy” - 1 punkt to np. ~0.5 dnia kodowania.
- Koszt scenariuszy: - 1,2 lub 3 punkty.
Scenariusze łatwiejsze powinny być scalane, trudniejsze – podzielone na mniejsze.



Identyfikowanie klas

- Metoda rzeczowników i czasowników:
 - Przeglądamy opisy użytkownika i scenariusze
 - Rzeczowniki mogą być dobrymi kandydatami na klasy lub atrybuty (formularz, student, grupa)
 - Czasowniki i opisy czynności – kandydaci na metody (ustaw, pobierz, zmień, uruchom)
- Klasy zbyt małe łączymy, zbyt złożone – dzielimy na mniejsze.



Modele tworzenia oprogramowania

- Model kaskadowy.
 - Poszczególne etapy wykonywane kolejno: planowanie, projektowanie, implementacja, testowanie, wdrożenie
- Model iteracyjny (przyrostowy).
 - Ogólny projekt systemu
 - Wybór części systemu i jego realizacja
 - Wybór kolejnej części systemu ...



Model iteracyjny

- Określenie zbioru scenariuszy realizowanego w ramach iteracji.
- Koszt zbioru scenariuszy (punkty).
- Weryfikacja oszacowań kosztu scenariuszy po każdej iteracji.
- Weryfikacja wymagań klienta po każdej iteracji
- Krótkie iteracje – np. 2 tygodnie



Implementacja

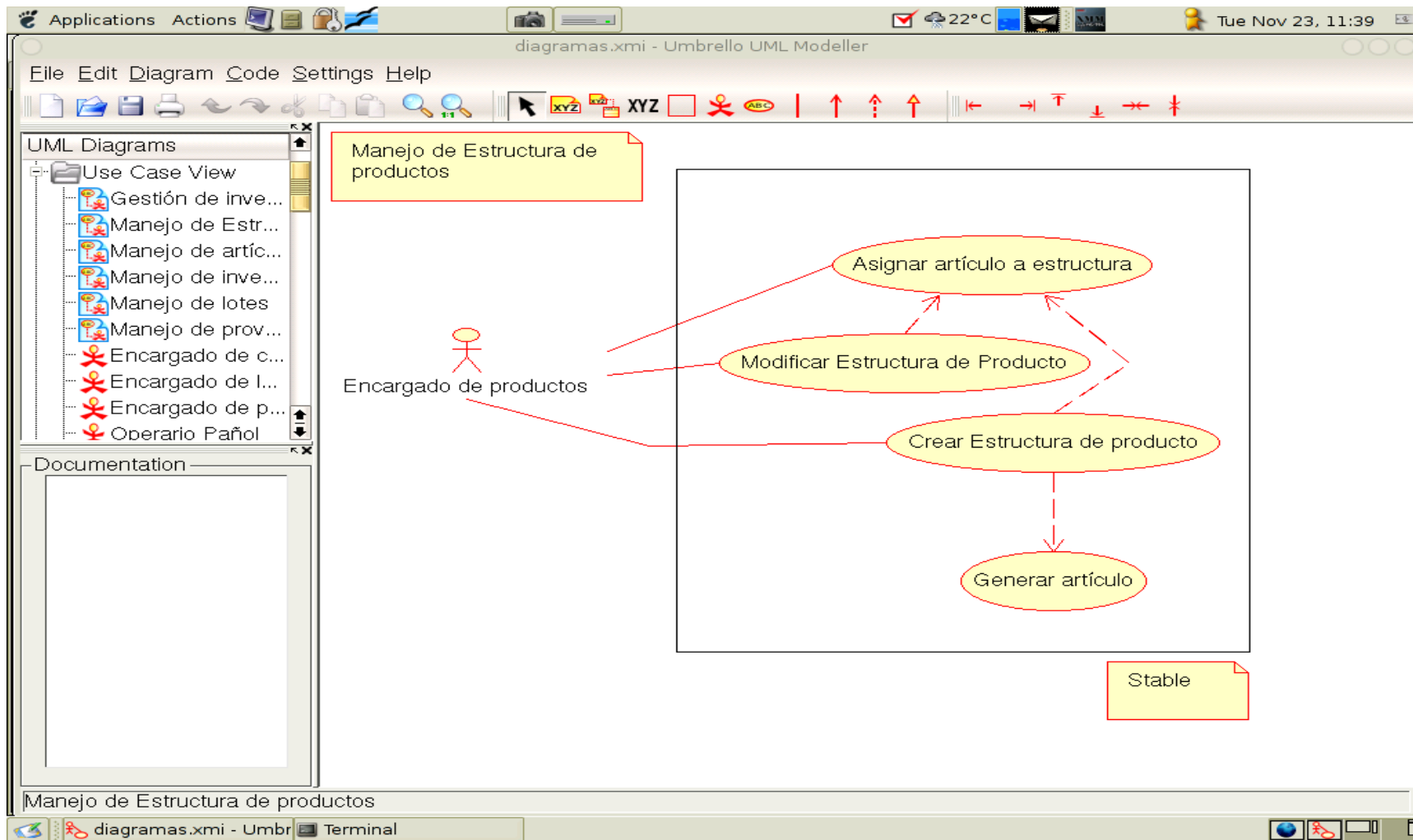
- Wytwarzanie sterowane testami (TDD)
- Refaktoryzacja
- Stosowanie wzorców projektowych
- Automatyzacja procesu budowania i testowania systemu
- Kontrola wersji



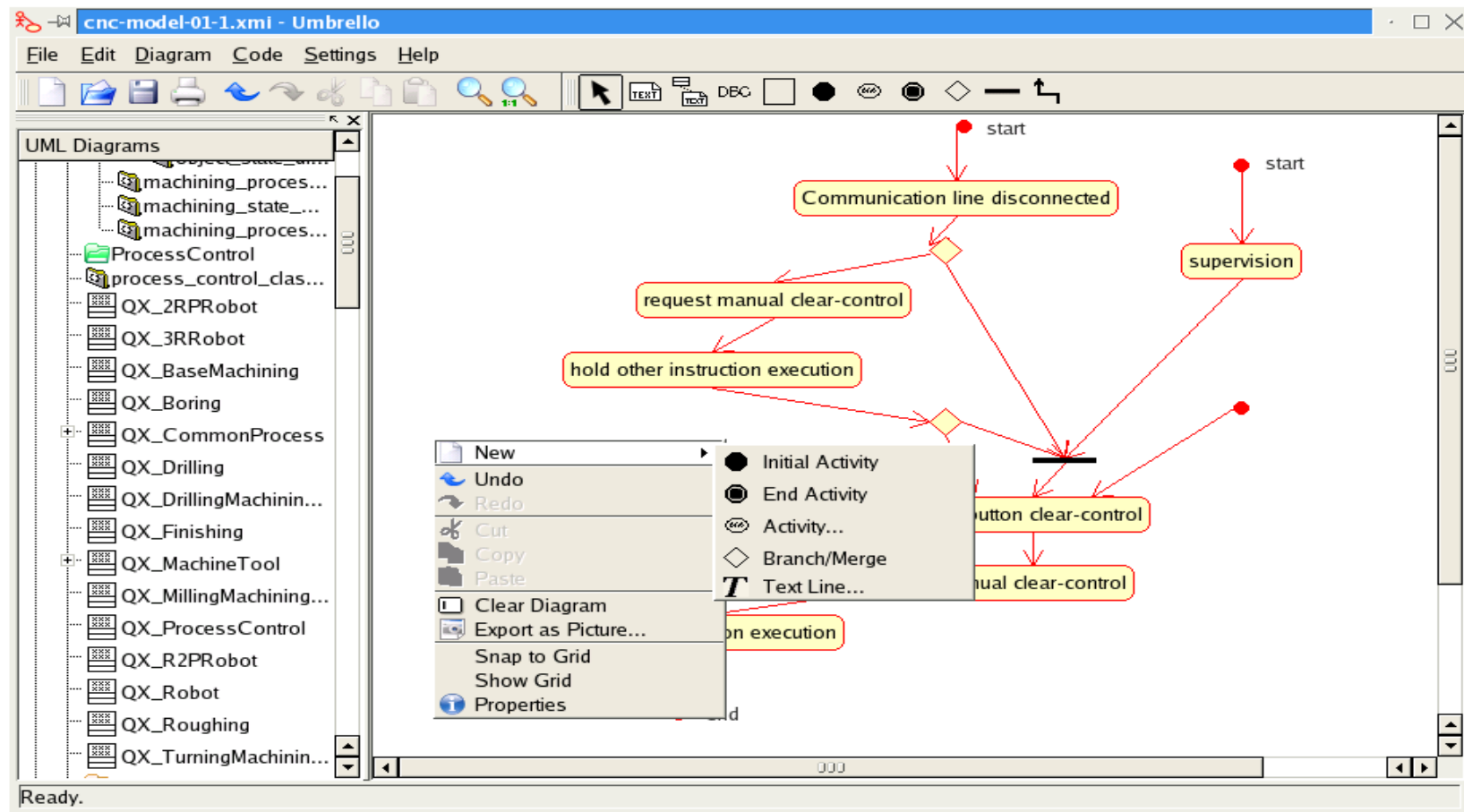
Modelowanie UML - Umbrello

- WWW: <http://uml.sourceforge.net/>
- Program do tworzenia diagramów UML
- Dostępny jako oprogramowanie Open Source
- Wersje dla Windows i Linux-a
- Możliwość eksportowania i importowania (C++) kodu.

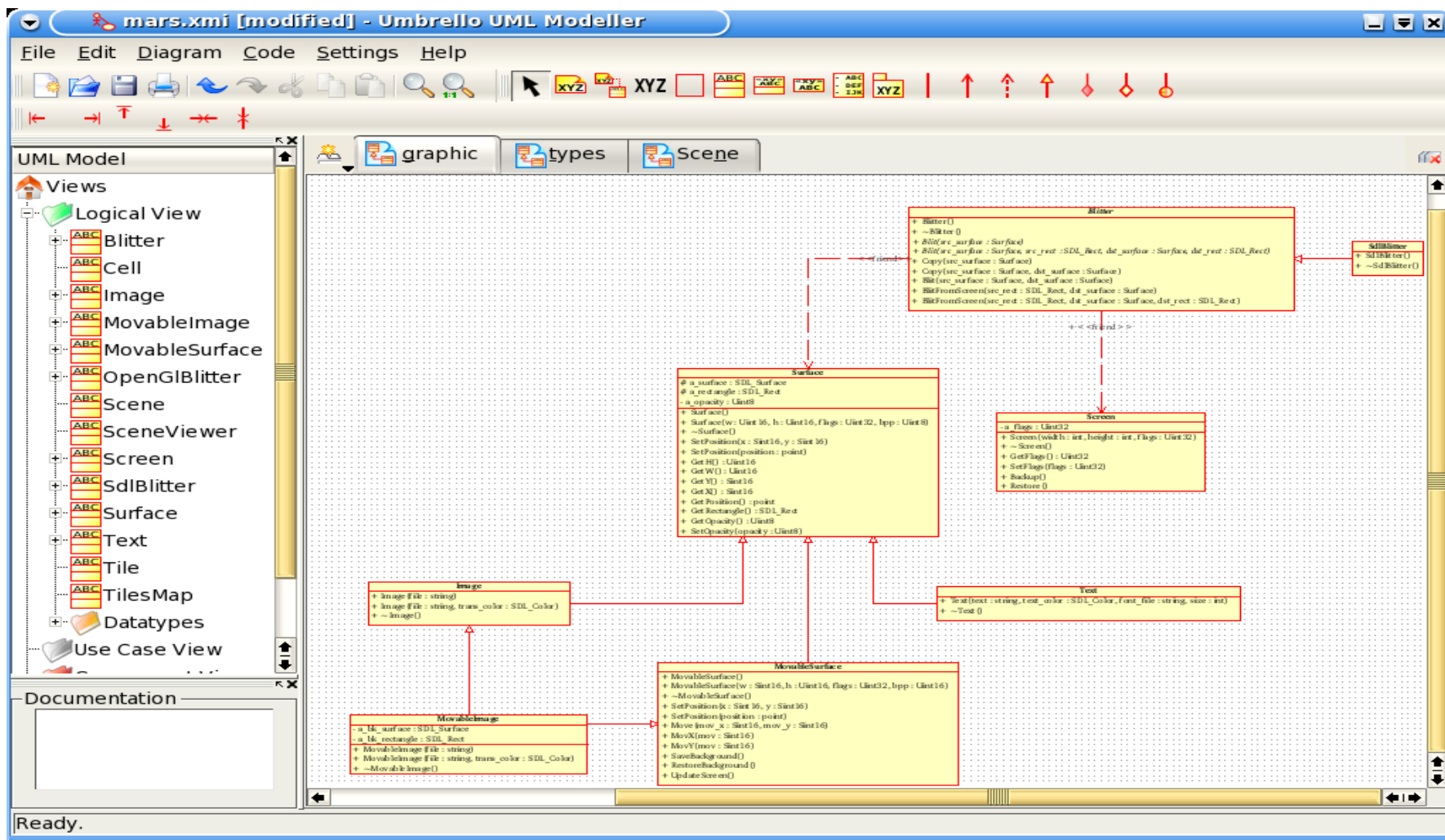
Umbrello c.d.



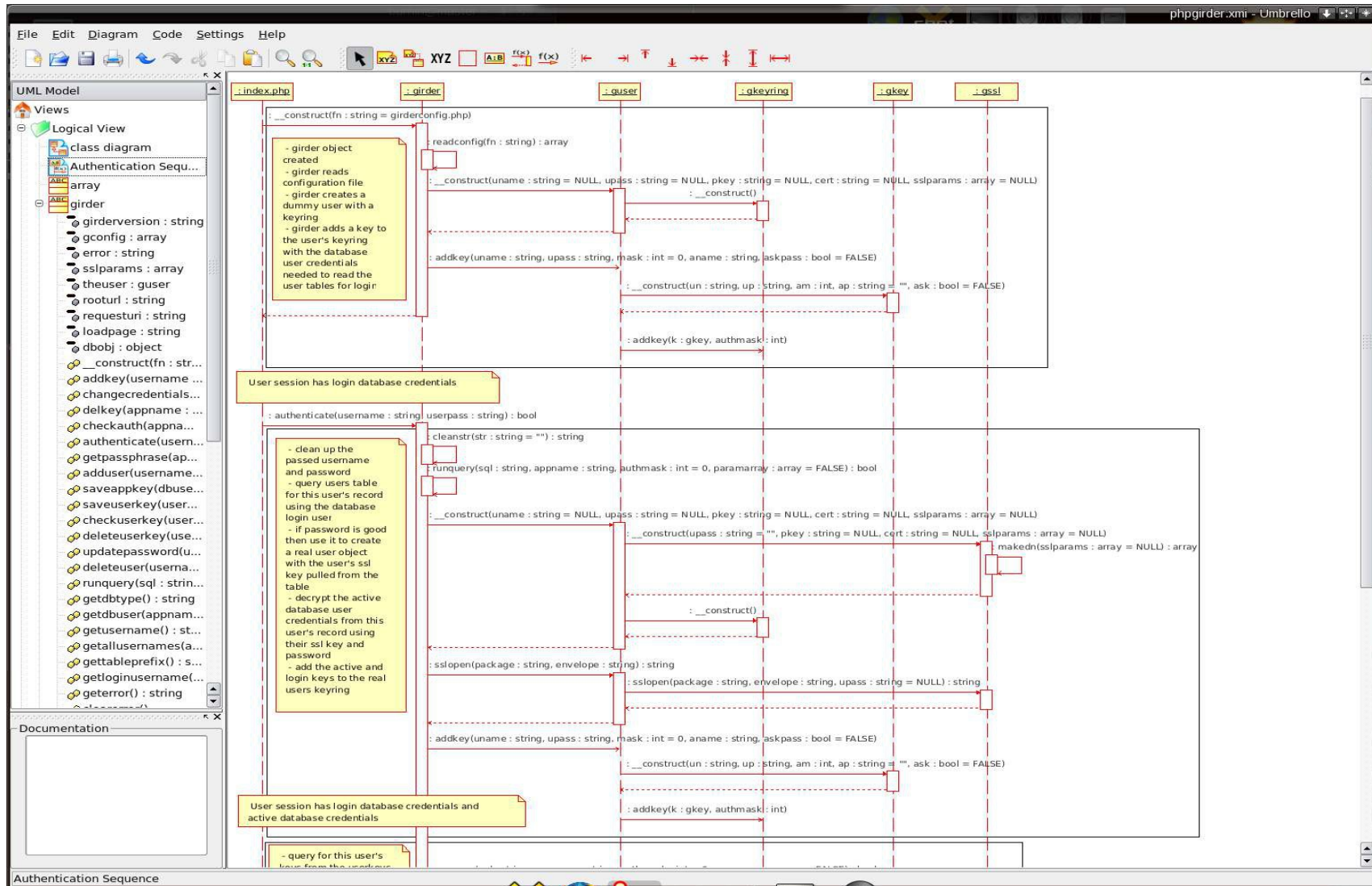
Umbrello c.d.



Umbrello c.d.



Umbrello c.d.





Podsumowanie

- Podstawowe diagramy UML-a
- Przykład wykorzystania diagramów UML-a
- Elementy inżynierii oprogramowania
- Umbrello – UML Modeller