

Diagramy klas.

Diagram klas określa statyczną strukturę modelu. Zawiera zestaw pakietów, klas, interfejsów oraz ich związków. Obiekty te tworzą razem strukturę stanowiącą projekt logiczny całości, bądź części systemu oprogramowania.

Diagram klas, jako część modelu zorientowanego obiektowo (**OOM – Object Oriented Model**) jest jednym z diagramów języka UML (ang. Unified Modeling Language). Jego zadaniem jest zdefiniowanie rzeczywistych obiektów systemu, takich jak: klasy, pakiety i związki.

Celem tworzenia diagramu klas jest uproszczony, abstrakcyjny opis możliwych **interakcji pomiędzy obiektami modelowanego systemu**. Klasa w takim diagramie opisuje zbiór obiektów o identycznych cechach (atrybuty) i operacji (metody). Obiekty, to konkretne wystąpienia (instancje) klas.

Definicja klasy w diagramie klas.

Klasa to opis zbioru obiektów o podobnej strukturze i zachowaniu, które posiadają wspólne atrybuty, operacje oraz związki. Ich struktura opisana jest poprzez atrybuty i związki, natomiast zachowanie opisane jest przez operacje.

Klasy to główne składowe modelu OOM, które tworzą jego podstawową strukturę. Klasa definiuje pewną koncepcję w ramach modelowanej aplikacji, jak np. obiekt fizyczny, biznesowy, logiczny, bądź związany z aplikacją czy zadaniem.

Poniższy przykład przedstawia samolot (klasa aircraft) wraz z atrybutami (zasięg i długość) oraz operacjami (start_engines):

aircraft
+ range : int
+ length : int
+ startengines() : void

Najważniejsze właściwości, jakie można ustalić dla klasy, to:

- Attributes – atrybuty, czyli charakterystyka cech,
- Identifiers – atrybuty bądź ich kombinacje, których wartości jednoznacznie identyfikują każde wystąpienie klasy,
- Operations – możliwe zapytania, jakich wykonanie można zlecić interfejsowi (metody),

Klasa – reprezentacja obiektu świata rzeczywistego, określają ją:

- Unikalna nazwa,
- Atrybuty,
- Metody

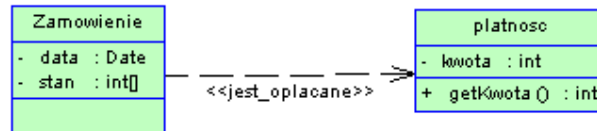
Klasa może implementować działanie elementów innych diagramów (np. przypadki użycia).

Związki – ilustrują relację znaczeniową pomiędzy klasami, określają je:

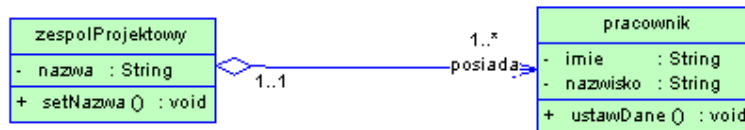
- Nazwa (nieobowiązkowa),
- Kierunek,
- Liczebność (udział, cardinality).

Rodzaje związków:

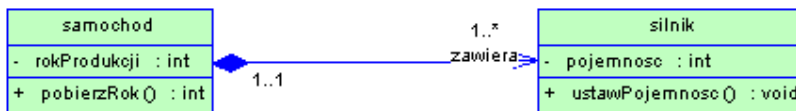
- Zależność - **Dependency**: zachowanie jednej klasy wpływa na zachowanie innej. Oznacza to, że jeden z obiektów diagramu korzysta z usług lub składowych innego obiektu.



- Agregacja - **Aggregation**: dana klasa składa się z innych, tzn. obiekty jednej klasy zawierają obiekty innej:



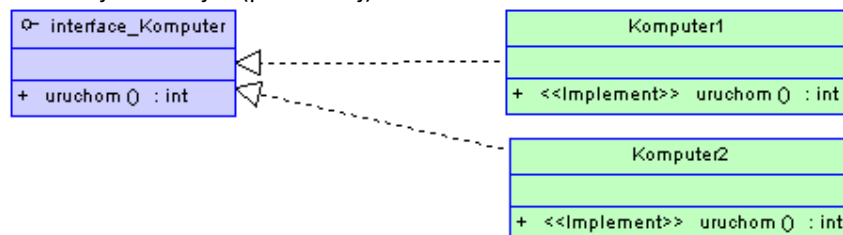
- Kompozycja - **Composition**: jeden komponent może należeć tylko do jednej całości, czas życia komponentu zdeterminowany jest przez czas życia całości.



- Dziedziczenie - **Generalization**: dana klasa zachowuje się podobnie do innej i posiada dodatkowe cechy (atrybuty i funkcjonalność).



- Implementacja interfejsu (patrz niżej) - **Realization**



Po dodaniu związku, w zakładkach Dependencies / Associations właściwości klasy pojawia się odpowiedni wpis.

Power Designer

Właściwości klasy

General: ustawić można następujące parametry:

- Name,
- Code,
- Comment,
- Visibility (Private - Only to the class itself; Protected - Only to the class and its derived objects; Package - To all objects contained within the same package; Public - To all objects in the model)
- Cardinality – liczebność, liczba instancji obiektów, jakie może mieć klasa,
- Generate - The class is automatically included among the objects generated from the model when you launch the generation process

Attributes: atrybuty (zmiennne) klasy i ich typy

Operations: nagłówki metod zdefiniowanych dla klas oraz zwracane przez nie typy,

Atrybuty i operacje danej klasy są osobnymi obiektami w drzewie obiektów i można je np. kopiować do innych klas.

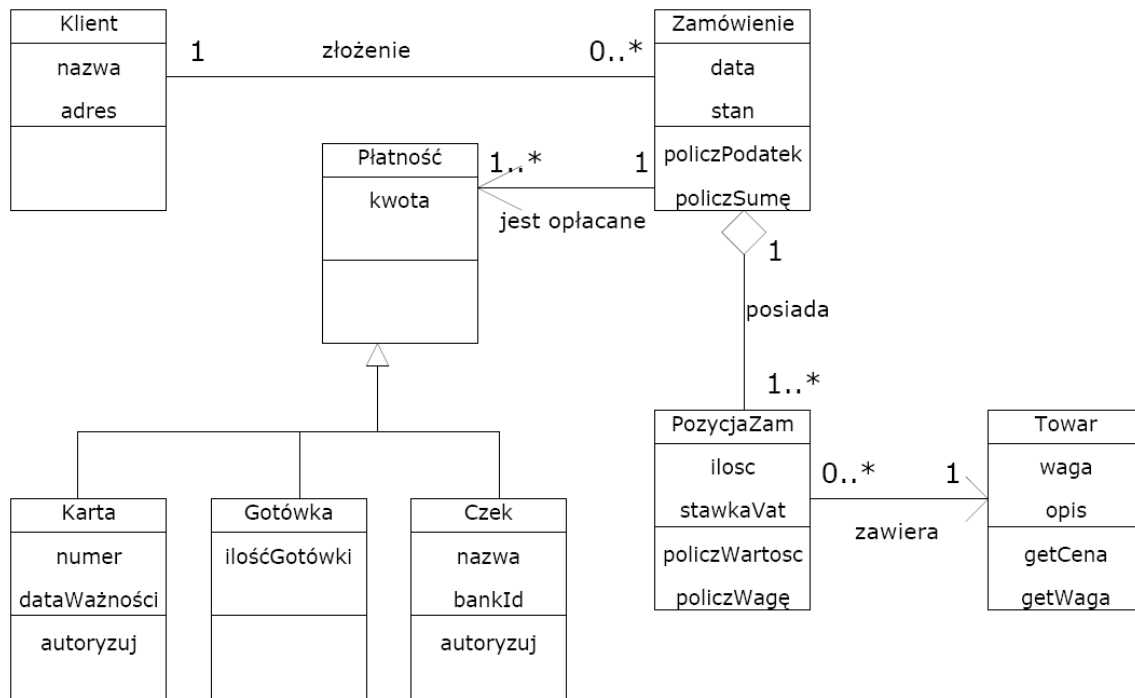
Generowanie pliku z kodem klasy na podstawie diagram Language -> generate java code, albo zakładka **preview**.

Tworzenie klasy

- Z diagramu: wybierz narzędzie klasy z przybornika, kliknij na polu roboczym, wejdź do właściwości klasy i ustaw jej nazwę oraz inne parametry.
- Na podstawie interfejsu, co pozwala na dziedziczenie wszystkich operacji określonych w interfejsie. Automatycznie tworzone jest połączenie realizacji pomiędzy klasą a interfejsem, wymienione również w zakładce Dependencies we właściwościach nowej klasy: menu kontekstowe interfejsu -> create class. [można również stworzyć interfejs na podstawie klasy]

Ćwiczenia

1. Stwórz w programie Power Designer następujący model:



2. Zamodeluj diagram klas dla następującego problemu:

Zespół roboczy realizuje projekty informatyczne. Projekty te dzielą się na zlecenia programistyczne oraz zadania Quality Assurance (zapewnianie jakości). Każdy z projektów składa się z różnych komponentów. Z zespołem związani są pracownicy.

3. Zamodeluj diagram klas dla następującego problemu:

W każdej klasie szkoły uczy się grupa uczniów. Cała klasa uczy się tych samych przedmiotów (w określonym wymiarze tygodniowym): przedmiotów zwykłych oraz języków obcych (niezależnie od klas istnieją grupy językowe: część dzieci uczy się angielskiego a część francuskiego). Nauczyciel może uczyć kilku przedmiotów oraz klasa ma wychowawcę. Wystawiane są oceny. Oprócz ocen cząstkowych za przedmioty, uczeń otrzymuje ocenę końcową. Dodatkowo uczeń otrzymuje końcową ocenę z zachowania.

4. Zaproponuj diagram klas dla gabinetu medycznego, realizowanego (diagram przypadków użycia) na poprzednich zajęciach.