

Laboratorium 4

Identyfikacja klas reprezentujących logikę biznesową projektowanego oprogramowania, definicja atrybutów i operacji klas oraz związków między klasami - na podstawie analizy scenariuszy przypadków użycia. Opracowanie diagramów klas i pakietów.

Cel laboratorium

Budowa diagramu klas reprezentujących logikę biznesową projektowanego oprogramowania. W oparciu o zdefiniowane elementy z poprzednich laboratorium należy zaproponować diagram klas dla projektowanego systemu.

Diagram klas jest jednym z podstawowych diagramów w notacji UML (Unified Modeling Language). Służy do modelowania struktury systemu poprzez przedstawienie jego klas, a także ich atrybutów, metod oraz relacji między nimi. Diagram klas opisuje statyczny aspekt systemu, czyli jak wygląda struktura danych, a nie jak te dane są przetwarzane.

Zastosowanie diagramu klas

Diagram klas znajduje szerokie zastosowanie, szczególnie w:

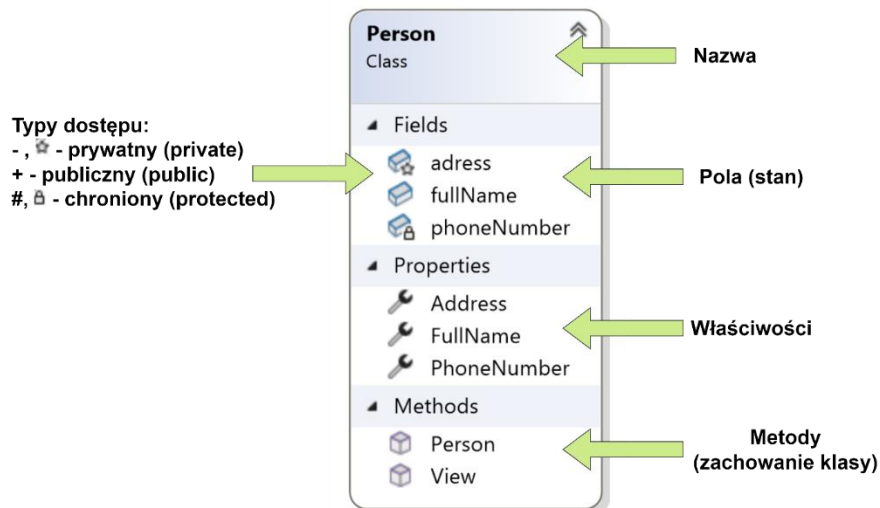
- Projektowaniu oprogramowania – opisuje strukturę systemu przed jego implementacją, co pozwala programistom lepiej zrozumieć model danych.
- Analizie wymagań – służy do zrozumienia wymagań i związków między różnymi elementami systemu.
- Modelowaniu baz danych – może być użyty jako narzędzie do projektowania schematów relacyjnych baz danych.
- Dokumentacji systemu – przedstawia szczegóły dotyczące klas, ich właściwości oraz relacji, co jest pomocne w dokumentacji technicznej.
- Przypadkach użycia – diagram klas może wspierać modelowanie przypadków użycia, ilustrując powiązane obiekty i klasy.

Diagram klas – notacja i semantyka

Klasa - to opis zbioru obiektów o takich samych atrybutach i operacjach (patrz Rysunek 1). Kluczowe aspekty klasy:

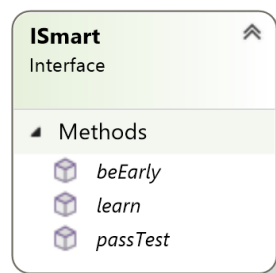
- Atrybuty: Nazwane właściwości klasy, określające zestaw możliwych wartości dla instancji.
- Operacje: Funkcje dostarczane przez obiekt, manifestujące się przez jego zachowanie.

Inżynieria Oprogramowania – Laboratorium



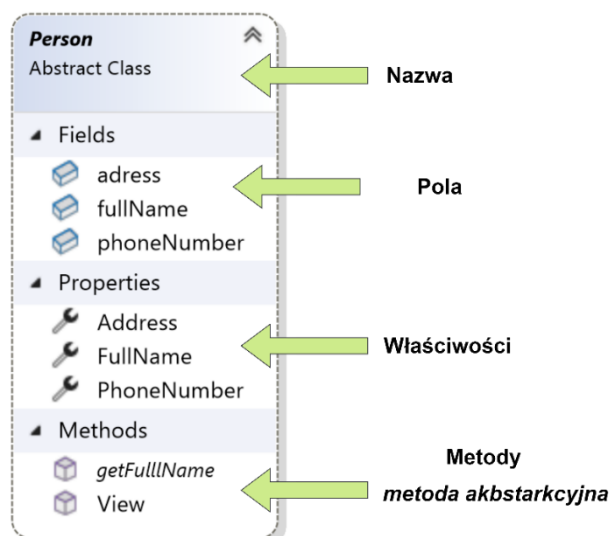
Rysunek 1. Przykład klasy Person

Interfejs w diagramie klas UML tworzymy przez dodanie słowa `<<interface>>` nad jego nazwą. Spójrz na przykład interfejsu:



Rysunek 2. Przykład interfejsu

Klasy abstrakcyjne tworzymy dodając `<<Abstract>>` nad nazwą klasy. Jak pokazano na przykładzie poniżej, oprócz metod mogą one posiadać również pola i właściwości.



Rysunek 3. Przykład klasy abstrakcyjnej

Relacje/powiązania pomiędzy klasami

Obecnie diagram klas przeważnie wykorzystywany jest do budowania diagramów mających na celu opisanie struktury przetwarzanych danych. Na Rysunku 4 przedstawiono fragment diagramu klas UML, pokazujący relacje między czterema klasami: A, B, C i D. Poniżej przedstawiono opis diagramu wraz z definicjami poszczególnych typów relacji:

1. Asocjacja (między Klasą D i Klasą A):

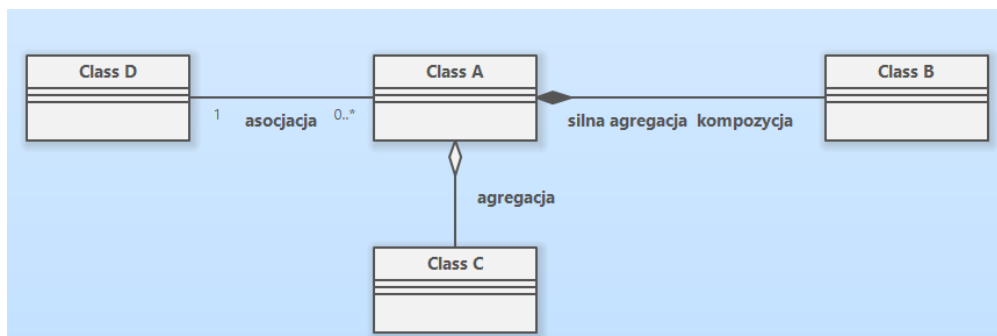
- Oznaczenie: prosta linia
- Krotność: 1 (Klasa D) do 0..* (Klasa A)
- Definicja: Asocjacja to ogólny związek między klasami, oznaczający, że obiekty tych klas są w jakiś sposób powiązane ze sobą. W tym przypadku jeden obiekt Klasy D może być powiązany z wieloma obiektami Klasy A (lub z żadnym).

2. Silna agregacja – kompozycja (między Klasą A i Klasą B):

- Oznaczenie: linia z wypełnionym rombem przy klasie zawierającej
- Definicja: Kompozycja to specjalny rodzaj agregacji, gdzie część (Klasa B) nie może istnieć bez całości (Klasa A). Obiekty klasy składowej są tworzone i niszczone wraz z obiektem klasy zawierającej.

3. Agregacja (między Klasą A i Klasą C):

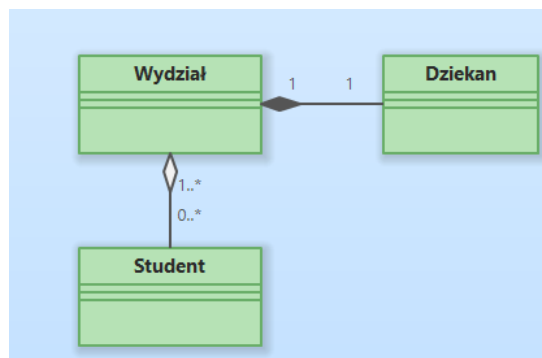
- Oznaczenie: linia z pustym rombem przy klasie zawierającej
- Definicja: Agregacja to relacja typu „całość-część”, gdzie część (Klasa C) może istnieć niezależnie od całości (Klasa A). Wskazuje, że obiekt jednej klasy zawiera obiekty innej klasy, ale nie są one ściśle zależne od siebie.



Rysunek 4. Podstawowe relacje stosowane na diagramie klas

Klasy mogą być powiązane w sposób szczególny tworząc Agregacje i Kompozycje (patrz Rysunek 5.).

- Agregacja jest wtedy, kiedy klasa może istnieć bez przypisania drugiej strony,
- Kompozycja jest wtedy, gdy klasa nie istnieje, jeśli nie występuje chociaż jedna instancja drugiej klasy wchodzącej w powiązanie.



Rysunek 5. Rodzaje agregacji

Jak widać powiązania opisują także liczebności i mogą przybierać wartości:

- 1 – dokładnie jeden,
- 1..* – od jednego do wielu,
- 0..1 – zero albo jeden,
- – dowolna ilość,
- 0..* – od zera do wielu,
- n – dokładnie n (gdzie $n > 1$),
- 1..n – od jednego do n,
- 0..n – od zera do n,
- n..m – od n do m (gdzie $n, m > 1$),
- n..* – od n do wielu (gdzie $n > 1$),
- n, m, o..p, q – o wartościach n, m, od o do p i q.

Dodatkowo, warto wspomnieć o innych typach relacji UML, które przedstawiono na rysunku 6.

1. Dziedziczenie (generalizacja):

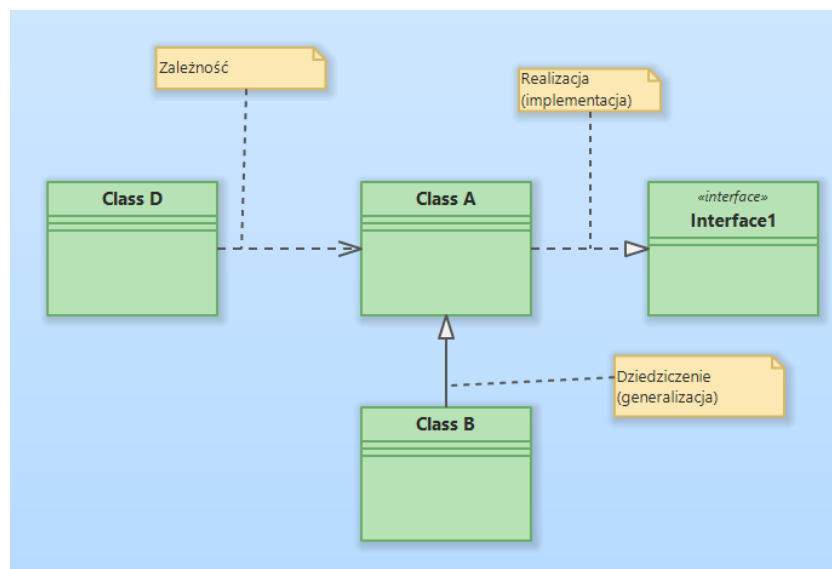
- Oznaczenie: strzałka z pustym grotem wskazująca na klasę nadrzędną
- Definicja: Relacja między klasą ogólną (nadrzędną) a klasą bardziej szczegółową (podrzędną). Klasa podrzędna dziedziczy atrybuty i metody klasy nadrzędnej.

2. Realizacja (implementacja):

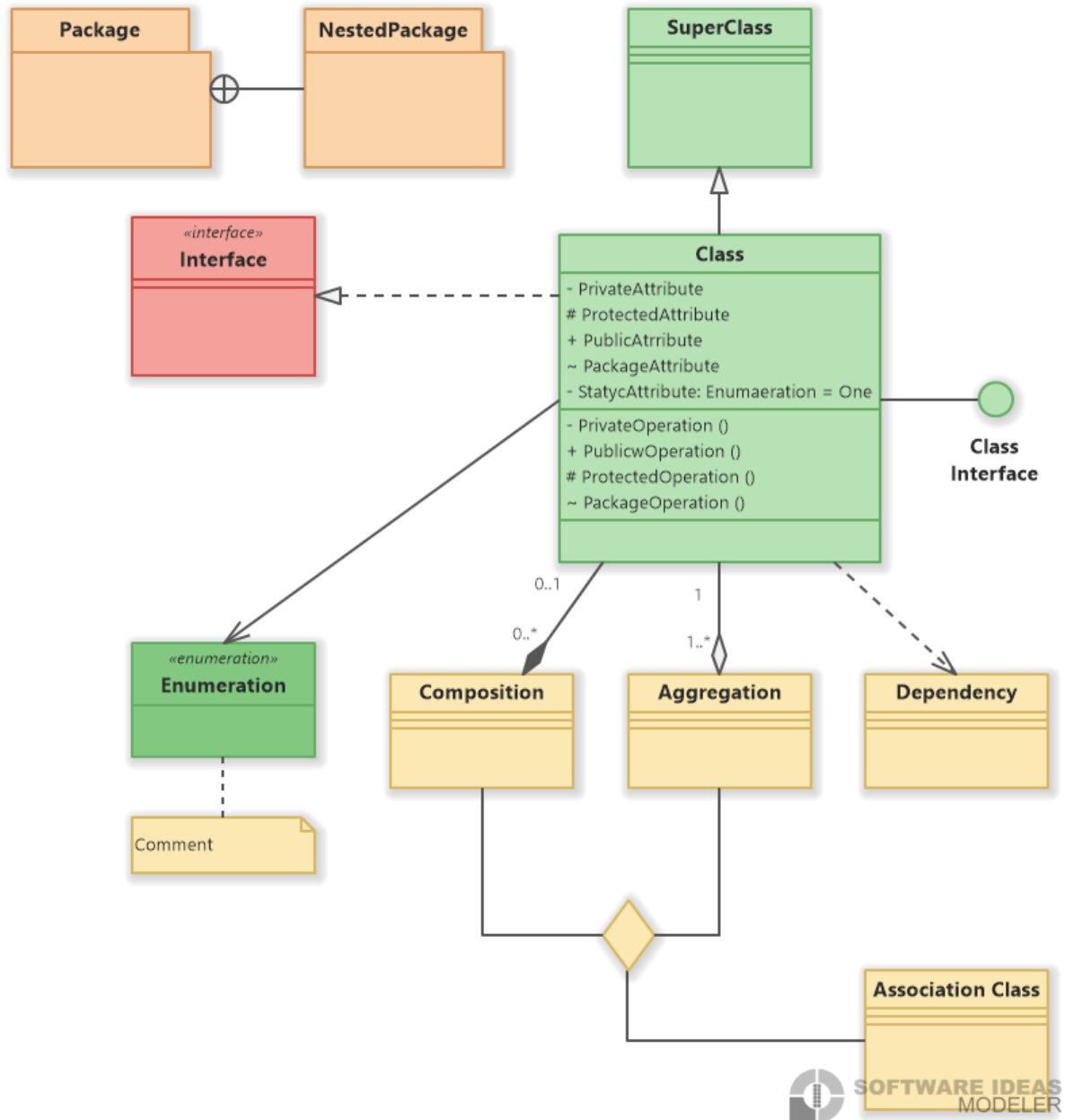
- Oznaczenie: przerywana linia ze strzałką wskazującą na interfejs
- Definicja: Wskazuje, że klasa implementuje określony interfejs, zobowiązując się do dostarczenia wszystkich metod zdefiniowanych w tym interfejsie.

3. Zależność:

- Oznaczenie: przerywana linia ze strzałką
- Definicja: Luźna relacja między klasami, gdzie zmiana w jednej klasie może wpłynąć na drugą, ale nie ma między nimi bezpośredniego powiązania strukturalnego.



Rysunek 6. Relacje w diagramie klas.



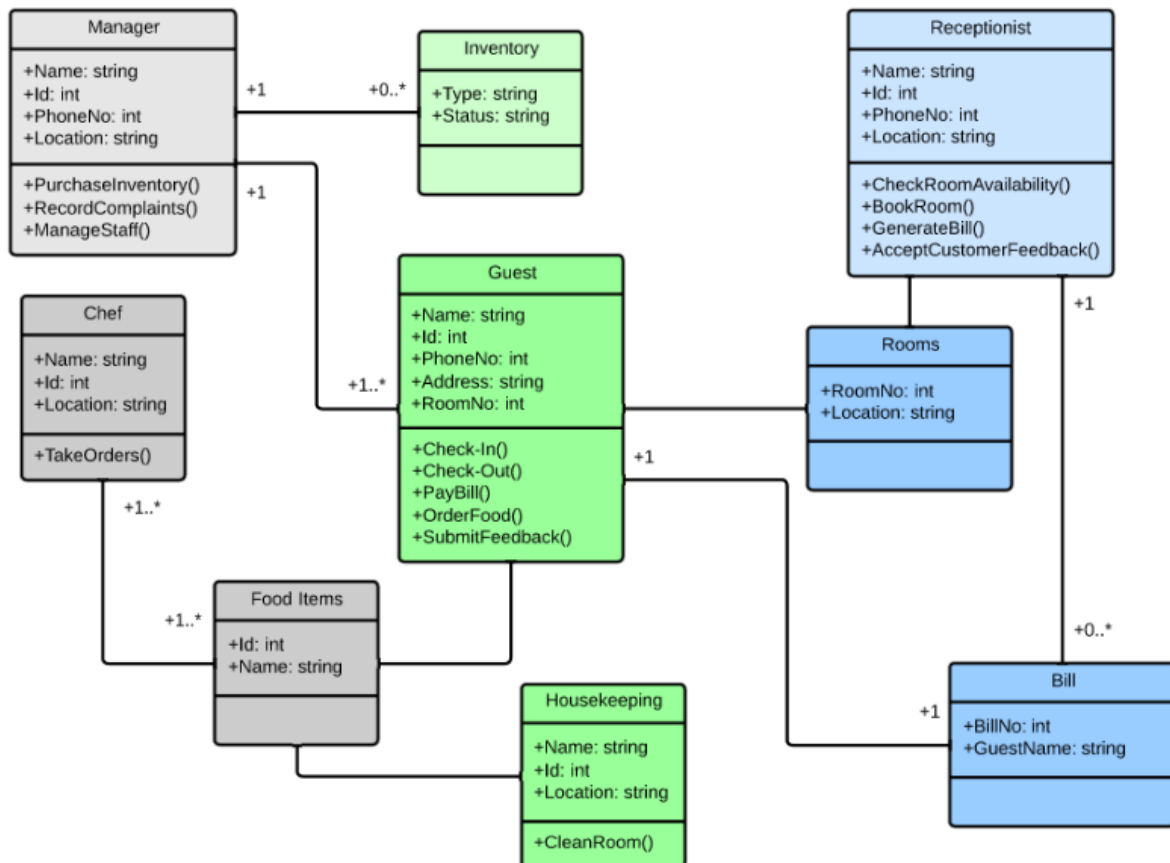
Rysunek 7. Relacje w diagramie klas.

Diagramy klas są nadzbiorem diagramów ERD (encja-związek). Rozszerzają ERD (modelowanie danych) o modelowanie zachowania.

Przykłady

Diagram klas dla systemu zarządzania hotelem

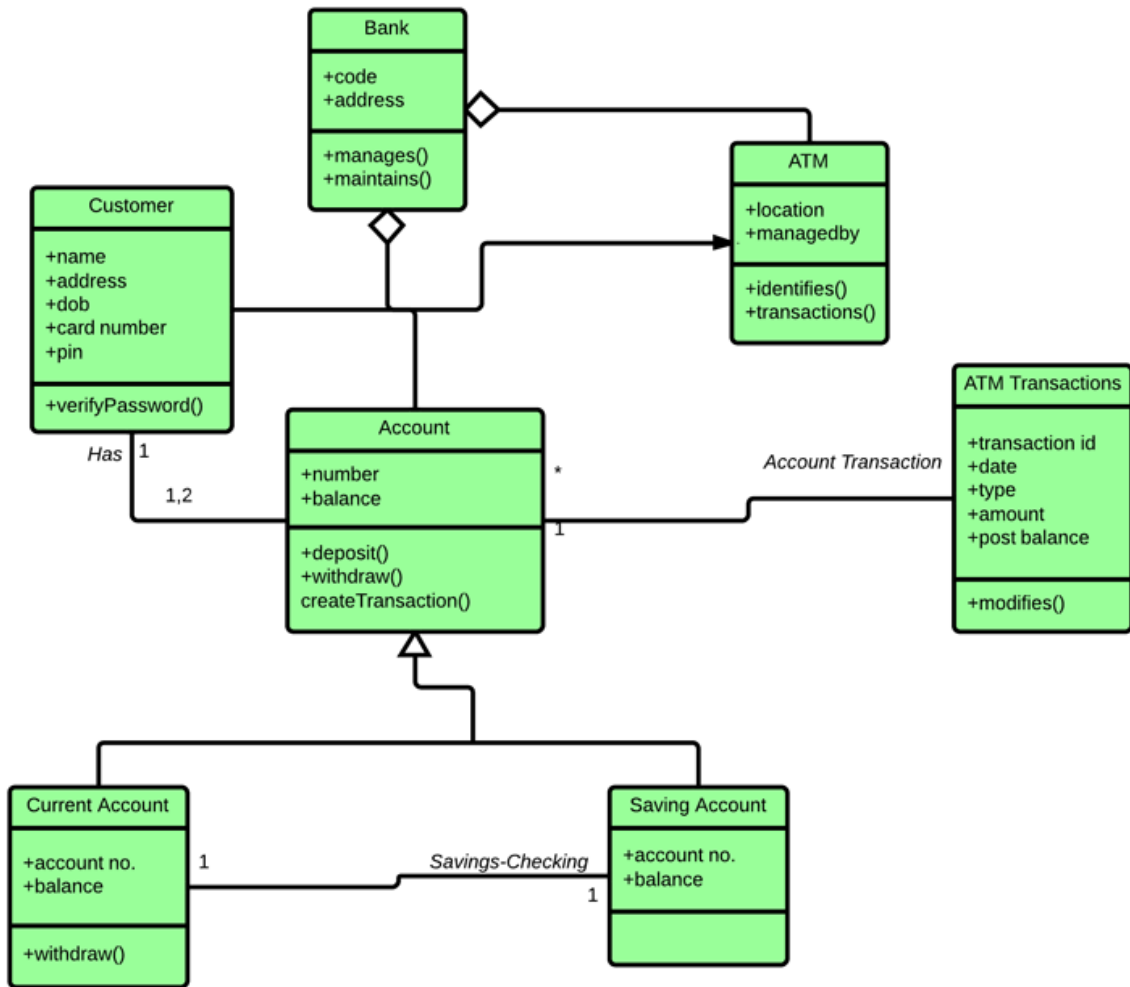
Diagram klas może pokazywać relacje między każdym obiektem w systemie zarządzania hotelem, w tym informacje o gościach, obowiązki personelu i obłożenie pokoi. Poniższy przykład stanowi przydatny przegląd systemu zarządzania hotelem.



Rysunek 8. Diagram klas dla systemu zarządzania hotelem

Diagram klas dla systemu bankomatu

Bankomaty są zwodniczo proste: chociaż klienci muszą tylko nacisnąć kilka przycisków, aby otrzymać gotówkę, istnieje wiele warstw zabezpieczeń, przez które musi przejść bezpieczny i skuteczny bankomat, aby zapobiec oszustwom i zapewnić wartość dla klientów bankowych. Na tym łatwym do odczytania diagramie zilustrowano różne ludzkie i nieożywione części systemu bankomatowego — każda klasa ma swój tytuł, a atrybuty są wymienione poniżej.



Rysunek 9. Diagram klas systemu bankomatu

Diagram klas dla systemu rezerwacji wypożyczalni samochodów.

Na rysunku 9 przedstawiono fragment diagramu klas dla systemu rezerwacji dla wypożyczalni samochodów. Poniżej szczegółowy opis diagramu:

1. Klasa Klient:

- Atrybuty: imię (String), nazwisko (String), numerPrawaJazdy (String)
- Relacja: agregacja z klasą Adres (biały romb)
- Relacja: asocjacja z klasą Rezerwacja (linia prosta), krotność 1 do 0..* (jeden klient może mieć wiele rezerwacji lub żadnej)

2. Klasa Adres:

- Atrybuty: miasto (String), ulica (String)
- Jest agregowana przez klasę Klient

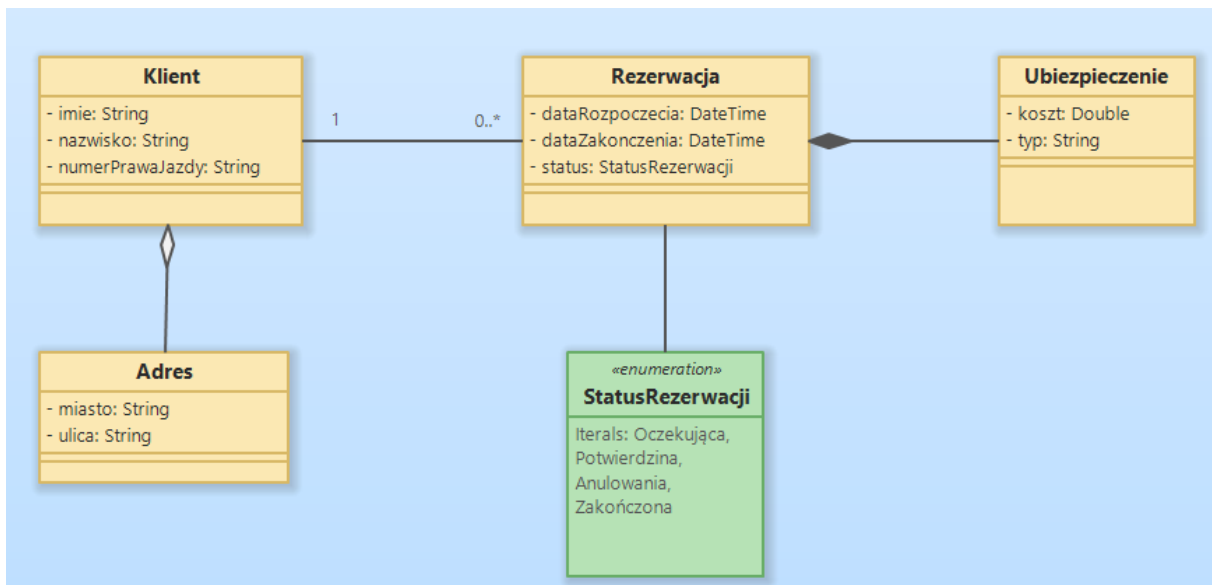
3. Klasa Rezerwacja:

- Atrybuty: dataRozpoczecia (Date), dataZakonczenia (Date), status (StatusRezerwacji)

- Relacja: kompozycja z klasą Ubezpieczenie (czarny romb), krotność 1 do 1 (każda rezerwacja ma dokładnie jedno ubezpieczenie)
 - Relacja: asocjacja z enumeracją StatusRezerwacji, krotność 1 do 1
4. Klasa Ubezpieczenie:
- Atrybuty: koszt (Double), typ (String)
 - Jest częścią kompozycji z klasą Rezerwacja
5. Enumeracja StatusRezerwacji:
- Wartości: OCZEKUJĄCA, POTWIERDZONA, ANULOWANA, ZAKOŃCZONA

Relacje na diagramie:

- Agregacja między Klient a Adres: sugeruje, że klient ma adres, ale adres może istnieć niezależnie od klienta.
- Asocjacja między Klient a Rezerwacja: pokazuje, że klient może dokonywać rezerwacji.
- Kompozycja między Rezerwacja a Ubezpieczenie: wskazuje, że ubezpieczenie jest integralną częścią rezerwacji i nie może istnieć bez niej.
- Asocjacja między Rezerwacja a StatusRezerwacji: określa, że każda rezerwacja ma przypisany status.



Rysunek 10. Diagram klas (fragment) dla systemu rezerwacji wypożyczalni samochodów.

Diagram ten przedstawia kluczowe elementy systemu rezerwacji, pokazując jak klient, jego adres, rezerwacja i ubezpieczenie są ze sobą powiązane, oraz jak status rezerwacji jest reprezentowany w systemie.