

Asocjacje

Dokument ten zawiera wytyczne dotyczące drugiego mini-projektu. Jego zakres obejmuje asocjacje.

„Zwykła”

Jest to nic innego jak najprostsza relacja między dwoma klasami. Każda asocjacja może mieć swoją licznosc. Licznosc określa ile elementów może/musi być z każdej ze stron asocjacji. Od strony programistycznej nie wymaga ona żadnych dodatkowych zabiegów.

Założmy, że mamy klasę **Klient** oraz **Zamówienie**. Jeden klient może mieć wiele zamówień.

```
class Client {  
    private List<Order> orders = new ArrayList<>();  
}
```

```
class Order {  
    private Client client;  
}
```

Asocjacja z Atrybutem

Ten typ asocjacji to specjalna odmiana typowej relacji wiele-do-wielu. Różnica polega na tym, że tabela asocjacyjna niezbędna do utrzymania relacji zamienia się w klasę asocjacyjną, która jest pełnoprawną klasą biznesową. Klasa asocjacyjna musi posiadać odniesienie do obu końców relacji oraz przechowywać odpowiednie atrybuty.

Przykład: Chcemy przechować historię zatrudnienia osoby w firmach. Jest to przykład asocjacji z atrybutem między Pracownik - Firma, gdzie klasą asocjacyjną jest Historia trzymająca datę zatrudnienia od do.

```
class Employee {  
    private List<History> history;  
}
```

```
class Company {  
    private List<History> history;  
}
```

```
class History {  
    private Employee employee;  
    private Company company;  
    private Date startDate;  
    private Date endDate; // atrybut opcjonalny  
}
```

Asocjacja kwalifikowana

Wyróżnia się ona kwalifikatorem, czyli atrybutem, który pozwala na jednoznaczne zidentyfikowanie elementu relacji. Wykonuje się to za pomocą mapy lub słownika.

Kompozycja

Kluczowe jest trzymanie się trzech zasad kompozycji:

1. nie ma części bez całości
2. Część nie może być współdzielona
3. Usunięcie całości powoduje usunięcie wszystkich części

Doskonałym przykładem kompozycji jest budynek i sale które w nim się znajdują. Budynek może istnieć bez sal. Jest to pusta bryła w środku, ale jest to pełnoprawny budynek. Sala z kolei nie może istnieć bez budynku bo jest jego integralną całością.

Przedstawię najpopularniejszą formę tworzenia kompozycji.

Ad.1

Tworzymy prywatny konstruktor tak, aby nie można było stworzyć obiektu samodzielnie. Mówimy tu oczywiście o konstruktorze części czyli w naszym przykładzie dla klasy Sala.

Następnie tworzymy metodę tworzącą część. Powinna być statyczna, publiczna i umieszczona w klasie sali. Metoda przyjmuje jako parametr obiekt całości (budynek) oraz parametry obowiązkowe.

W tej metodzie następuje utworzenie obiektu, powiązanie go z całością i zwrócenie. Również w tej metodzie należy dodać część do odpowiedniej kolekcji w ramach klasy całości.

Ad. 2

Przed dodaniem części do całości należy upewnić się, że taka część nie istnieje już w ekstensji.

Ad. 3

Gdy usuwamy całość należy pierw przeliterować części aby je usunąć, a potem dopiero usunąć całość.

Obrona projektu

Kod musi się kompilować, ale nie wymagam żadnego wyświetlania informacji na konsoli. W przypadku MP2 skupiamy się w 100% na kodzie i sposobie implementacji. Oceniana jest jakość kodu i sposób implementacji.