

Wydział Informatyki Politechnika Białostocka Zaawansowane techniki programistyczne	Data 13.12.2022
Dokumentacja projektu Temat: Przepływ dokumentów w firmie zajmującej się sprzedażą. 1.Michał Wołosewicz 2.Patryk Wójtowicz 3.Magda Zaborowska	Prowadzący: mgr inż. Daniel Reska

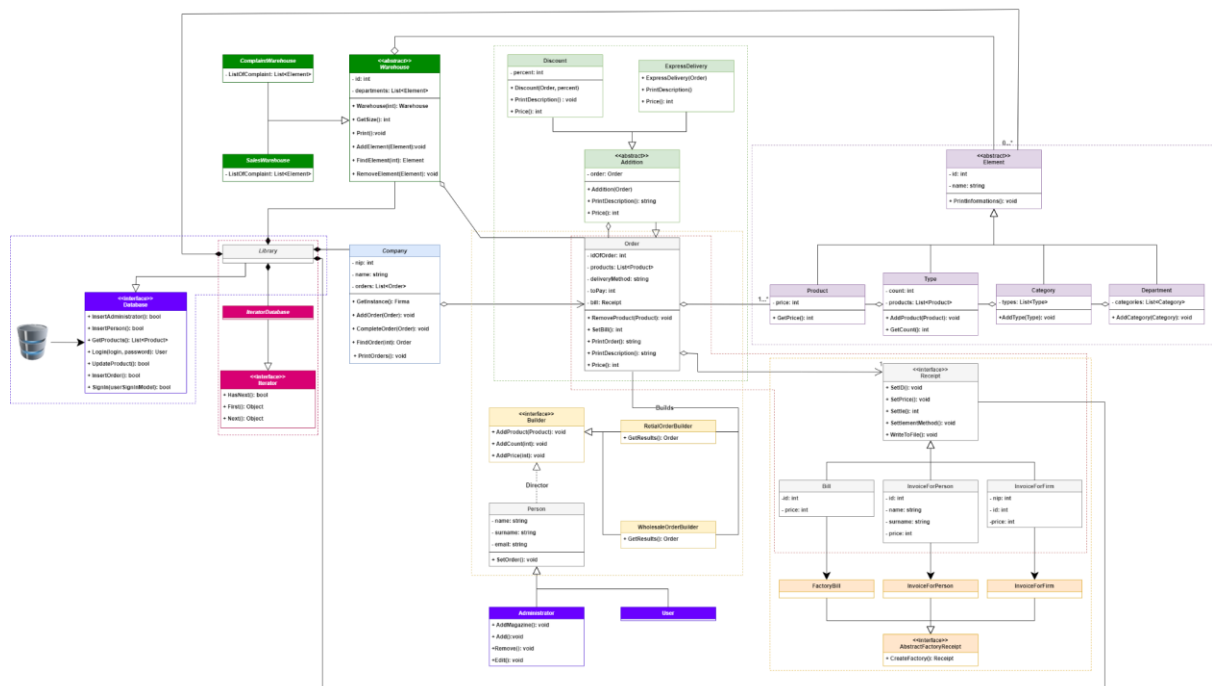
1. Opis i cel projektu

Z przykładowych tematów projektów wybrano temat 3 "Przepływ dokumentów w firmie zajmującej się sprzedażą". Aplikacja pozwala na zarządzanie stanem magazynowym oraz obsługą zamówień tj. realizacja zamówień (wystawianie faktur).

2. Funkcjonalności programu

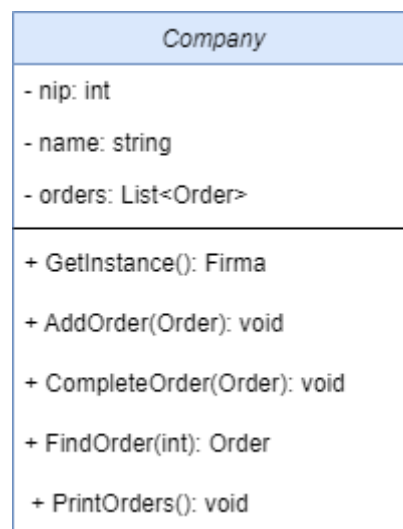
- wprowadzanie towaru do magazynu (nazwa, ilość, cena, stawka VAT)
- składanie zamówień (lista towarów ich ilość, kto składa zamówienie)
- realizacja zamówień
- wystawianie faktur
- wydawanie towarów (zmniejszenie ilości magazynowych)
- podgląd szczegółów zamówienia
- wybór uprawnień konta użytkownika
- rejestracja i logowanie
- dodawanie nowych kategorii przedmiotów
- sprawdzanie stanów magazynowych w poszczególnych magazynach
- wyszukiwanie towarów
- eksportowanie faktur do plików tekstowych
- zarządzanie bazą danych z poziomu administratora

3. Diagram Klas



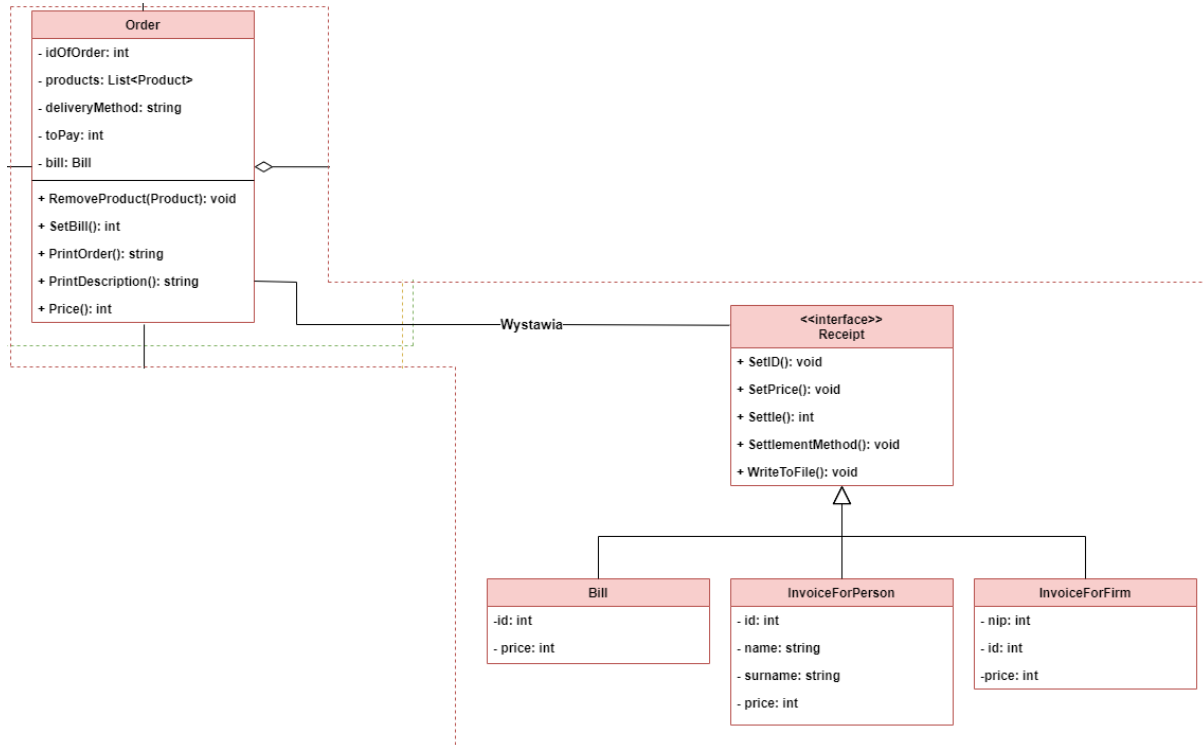
4. Opisy wzorców projektowych

- Singleton



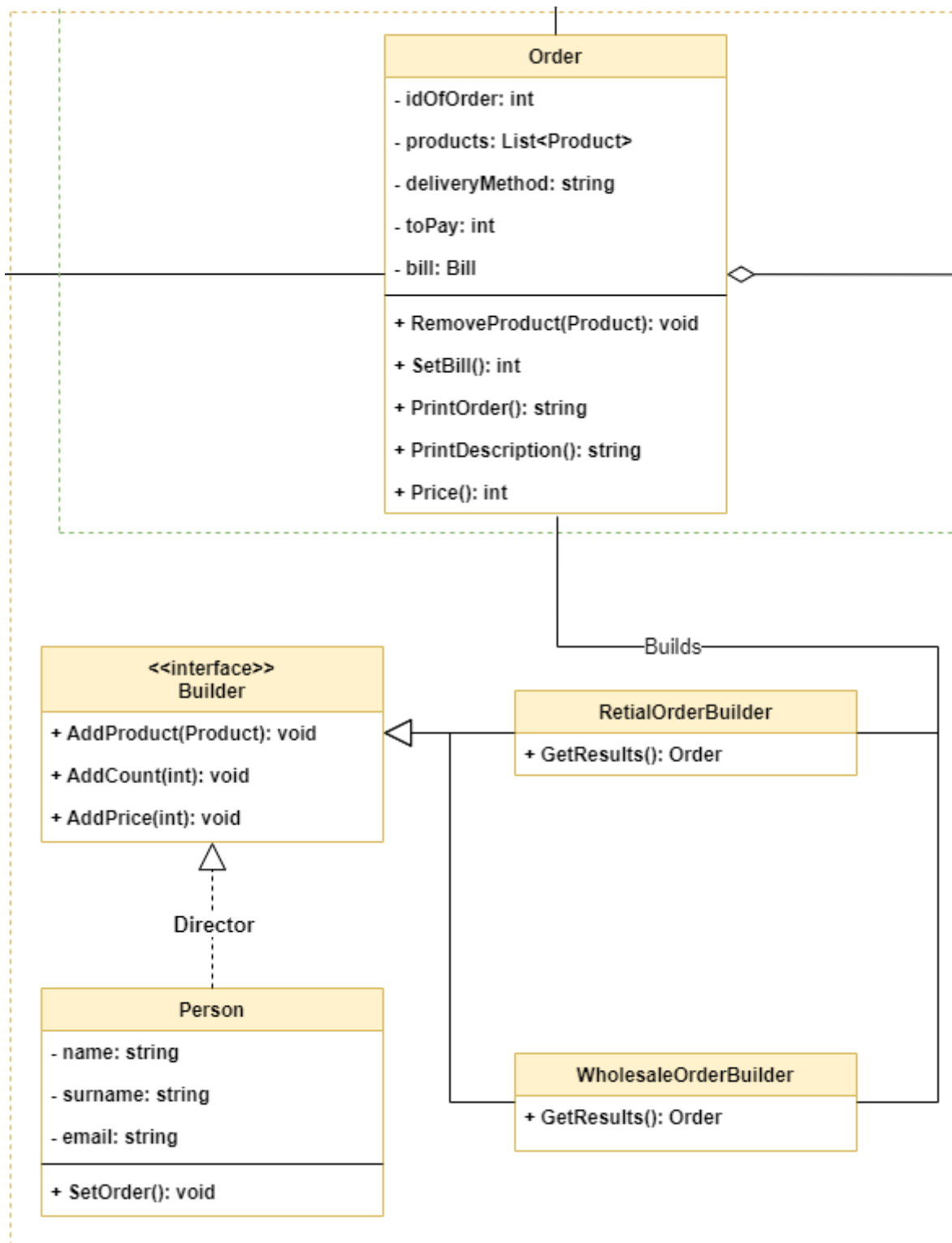
Zastosowano wzorec Singleton dla klasy Company, aby istniała tylko jedna jej instancja. Dodatkowo dzięki temu wzorcowi dostęp do tej instancji jest globalny.

• Strategia



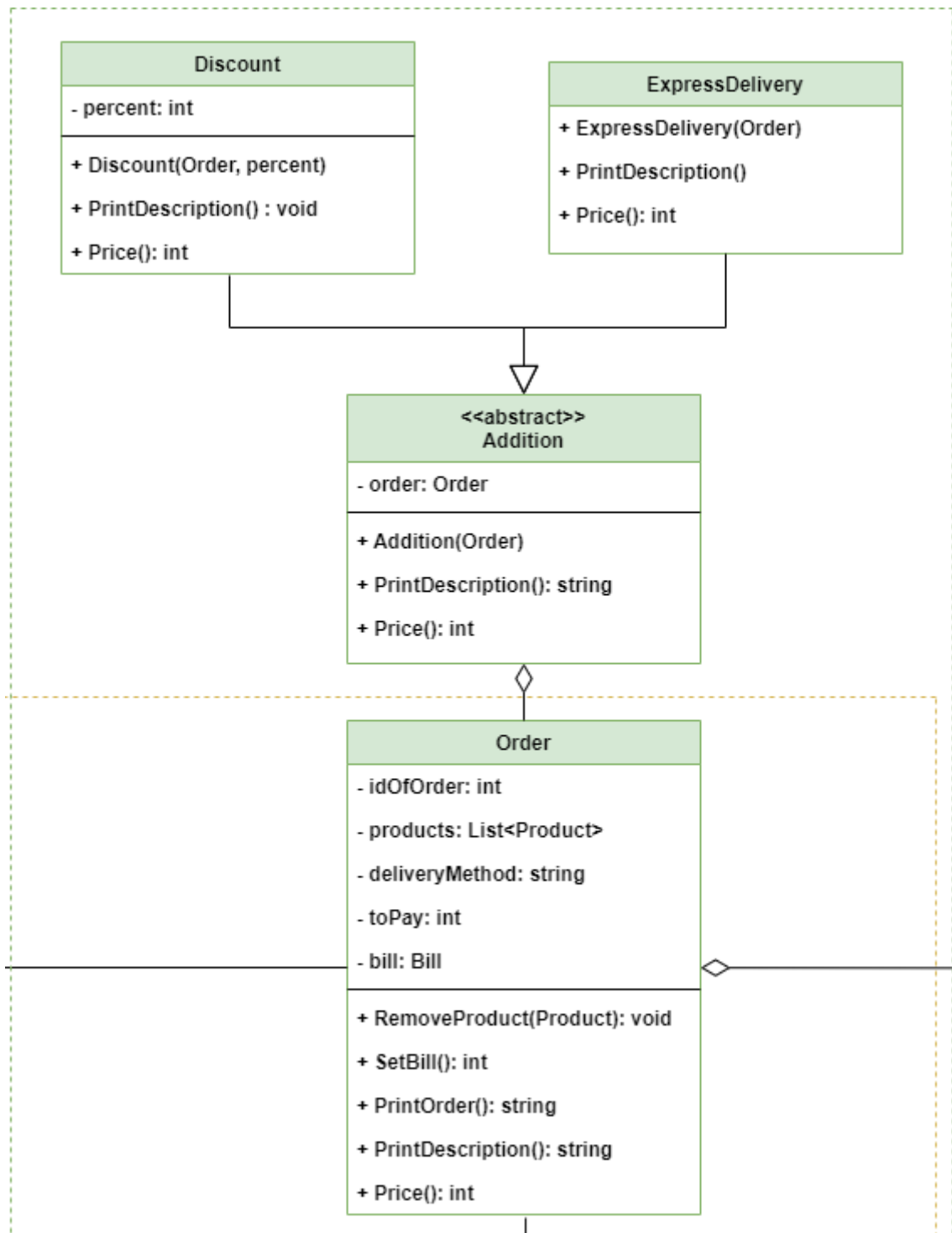
Wzorzec strategia wykorzystamy do realizacji zamówień. Ułatwi on wybór sposobu rozliczenia zamówienia, dzięki czemu będą one wymienne. Kontekstem będzie **Order** oraz metody jego rozliczenia

- Builder



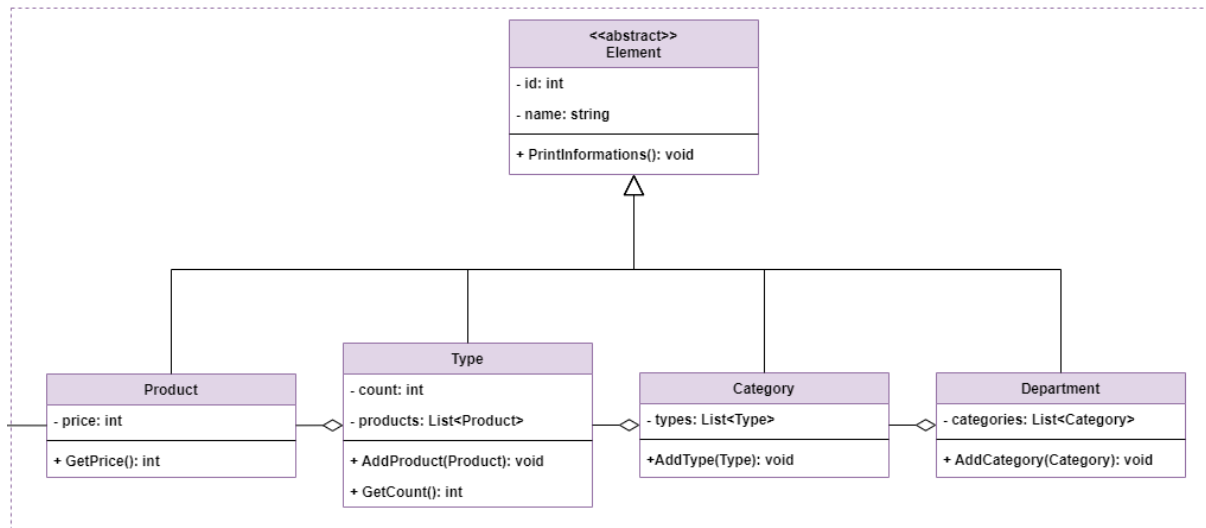
Wzorzec Builder wykorzystamy do tworzenia obiektu **Order**. Ułatwi to tworzenie własnego, spersonalizowanego zamówienia. Użytkownik aplikacji będzie generował scenariusz tworzenia zamówienia. Proces tworzenia zamówienia będzie podzielony na mniejsze etapy, a każdy z nich może być implementowany na wiele sposobów.

- Dekorator



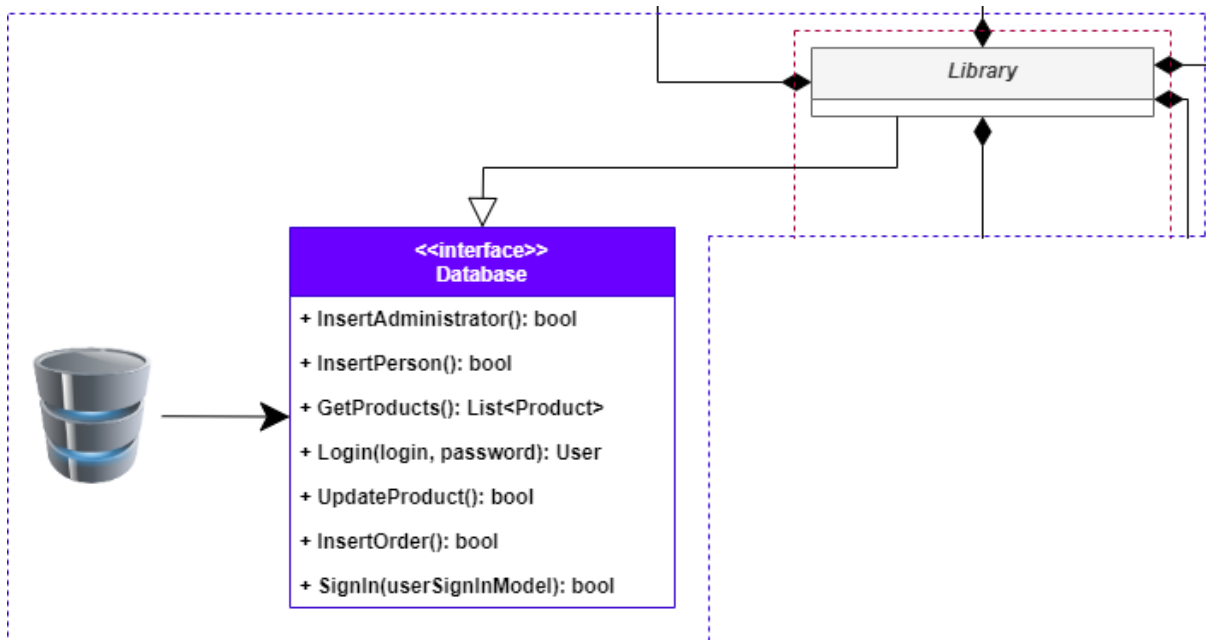
Wzorzec dekorator pozwoli na łatwe zmiany w cenie np. uwzględnienie zniżek lub wyboru sposobu dostawy co odpowiednio zmniejszy lub zwiększy finalną kwotę. Dodatkowo wzorzec ten pozwala na łatwą rozbudowę projektu o kolejne modyfikatory finalnych kosztów.

- Kompozyt



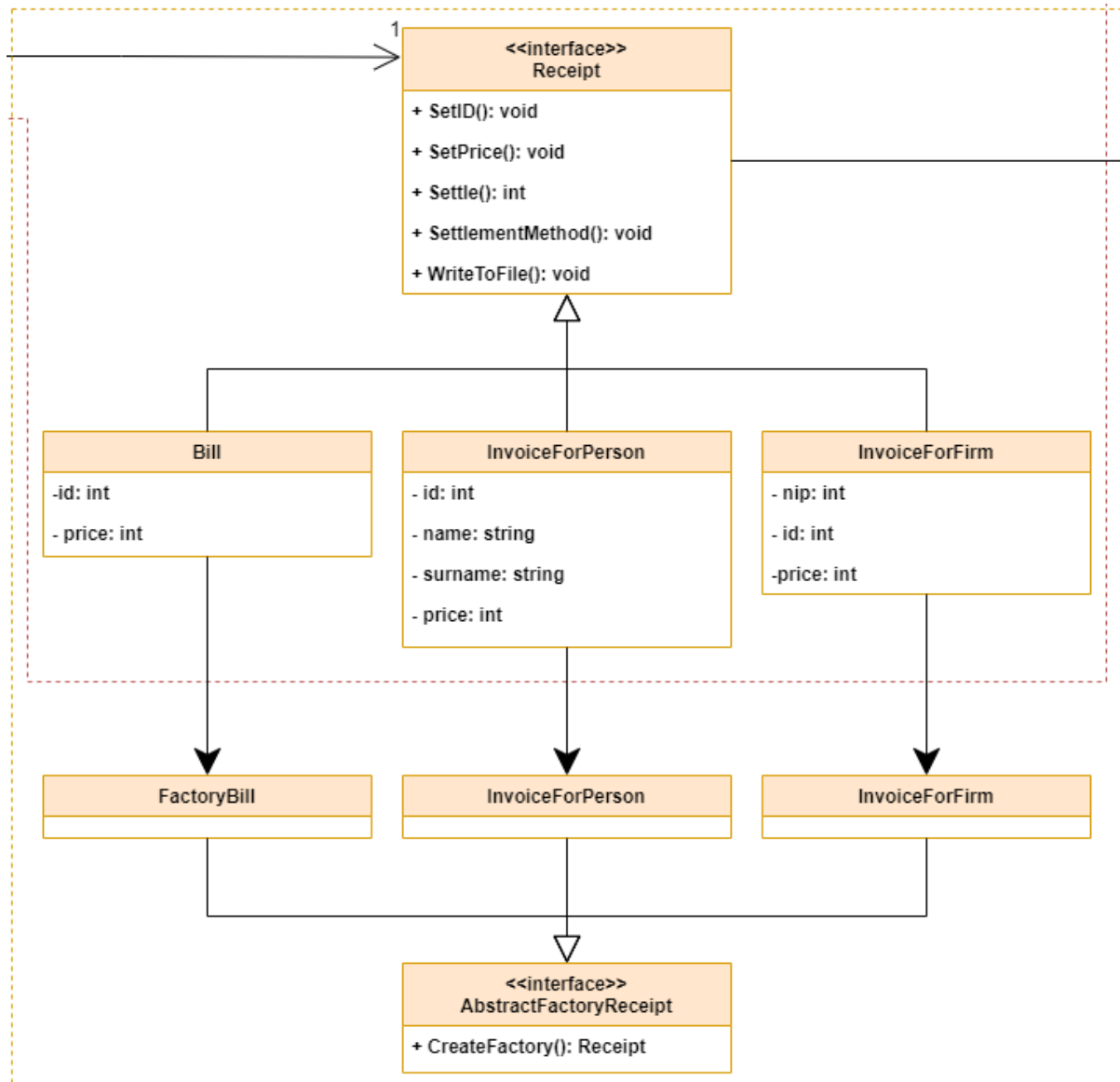
Wzorzec kompozyt został użyty do stworzenia drzewiastej struktury obiektów co pozwala grupować obiektów na działy, typy oraz kategorie. Do tworzenia nowych działów, typów lub kategorii nie wymagane jest utworzenie nowej klasy, a należy stworzyć tylko obiekt danej klasy.

- Proxy



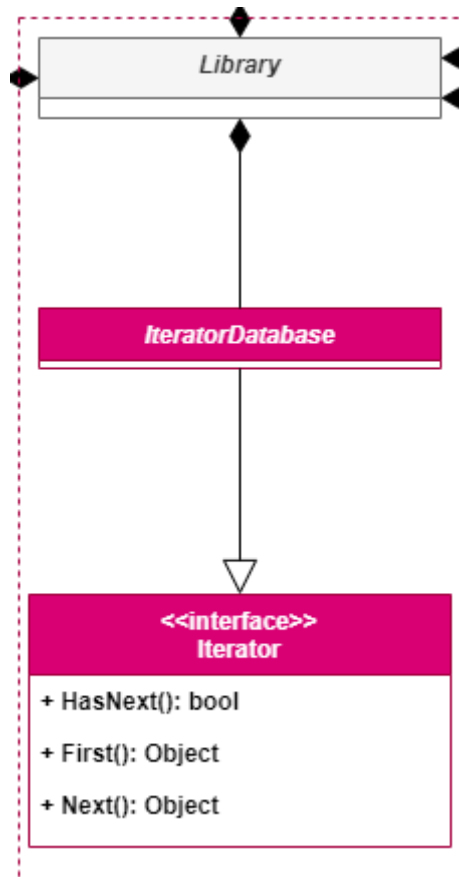
W projekcie zostanie stworzona baza danych poprzez interfejs Database i klasę implementującą - Library. Klasa Library będzie pełnić rolę pośrednika przekazując wszystkie wywołania do docelowego obiektu. Ma to ułatwić wykonywanie zapytań do bazy danych.

- Fabryka Abstrakcyjna



Wzorzec fabryka abstrakcyjna pozwala na tworzenie różnych obiektów należących do tej samej rodziny. Obiekty różnych rodzajów implementują jeden interfejs (Receipt), ale są różne. Metoda CreateFactory() jest metodą bezparametrową i w zależności od rodzaju fabryki tworzy odpowiedni rodzaj rachunku dzięki czemu nie trzeba robić tego samodzielnie.

- Iterator



Wzorzec iterator odpowiada za poruszanie się po bazie danych oraz za odczytywanie jej poszczególnych rekordów. Metoda **Next()** pobiera element i przechodzi do kolejnego, natomiast **HasNext()** sprawdza czy dany element jest ostatni.