Agregacja obiektów

Agregacja

Agregacja obiektów ma miejsce wtedy, gdy instancja jednej klasy zawiera instancje innych klas.

Agregacja

Rozróżniamy dwa rodzaje agregacji:

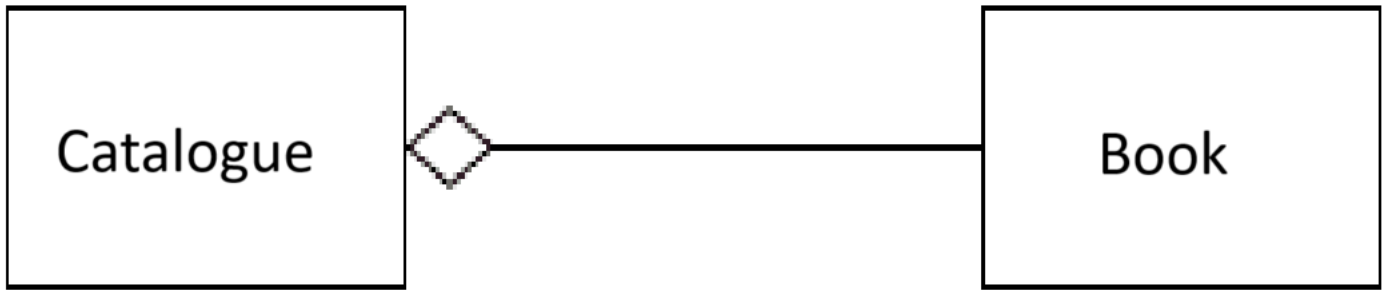
- Agregacja częściowa (ang. aggregation)
- Agregacja całkowita (ang. composition)

Agregacja częściowa

- W **agregacji częściowej** element częściowy należy do elementu głównego, jednak nie jest od niego zależny. Usunięcie elementu głównego nie wpływa na usunięcie elementu częściowego. Element częściowy może także należeć do wielu elementów głównych.
  (wg https://www.p-programowanie.pl)
- Agregacja jest często określana jako relacja typu **„zawiera”** (wg Wikipedii)

Agregacja częściowa – przykład

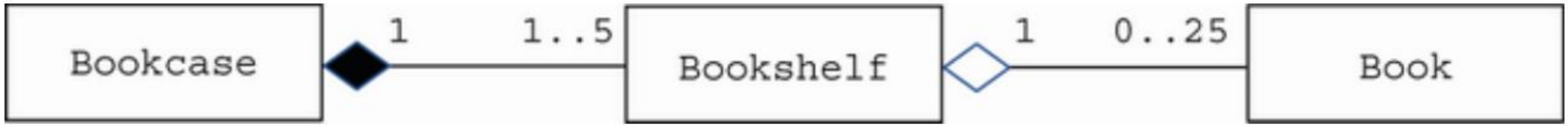
Diagram UML:



Agregacja całkowita

- Szczególnym przypadkiem agregacji jest **kompozycja**, która oznacza składanie się obiektu z obiektów składowych, które nie mogą istnieć bez obiektu głównego. Kompozycja jest relacją typu **„posiada”**. (wg Wikipedii)
- Kompozycja oznacza, że dana część może należeć tylko do jednej całości. Oznacza również, że część nie może istnieć bez całości, a usunięcie całości powoduje automatyczne usunięcie wszystkich jej części, związanych z nią związkiem kompozycji. (wg Wikipedii)

Agregacja całkowita – przykład (wg R.Mak)

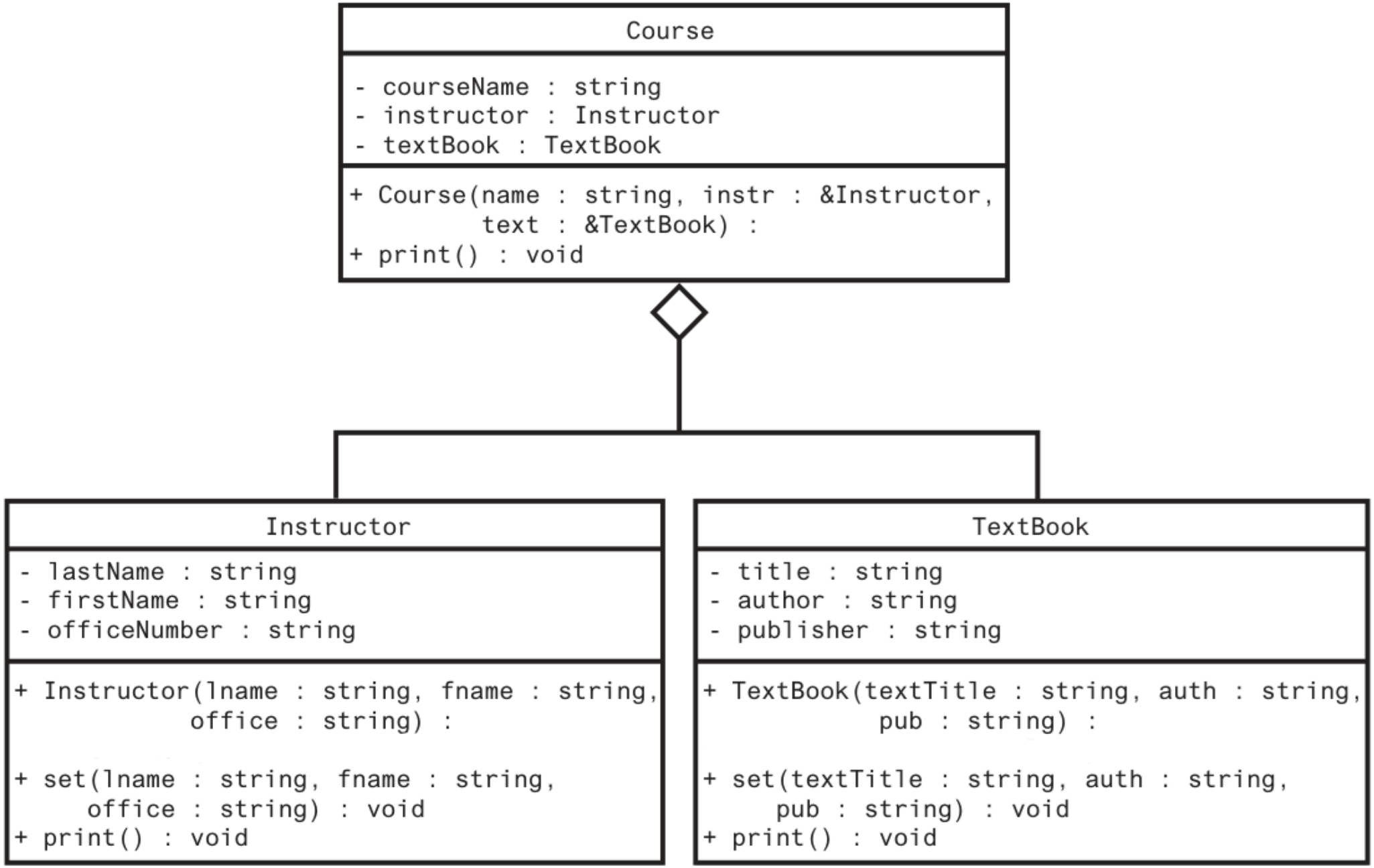


- Na tym diagramie każdy obiekt Bookshelf może zawierać od 0 do 25 obiektów Book, a obiekty Book mogą istnieć poza obiektami Bookshelf.
- Każdy obiekt Bookcase składa się z od 1 do 5 obiektów Bookshelf, ale obiekt Bookshelf nie może istnieć bez bycia częścią obiektu Bookcase.

Uwagi

- Rozróżnienie agregacji częściowej od kompozycji może być w praktyce trudne
- Nie zawsze mówimy o agregacji, gdy klasa A zawiera obiekty klasy B. Przykład: klasa Osoba zawiera obiekt klasy string nazwisko. W tym przypadku raczej myślimy, że nazwisko jest po prostu atrybutem osoby.

Zadanie: dany jest diagram klas, zaimplementuj te klasy



```
class Instructor {
private:
   string lastName, firstName, officeNumber;
public:
   Instructor() { set("", "", ""); }
   Instructor(string lname, string fname, string office) { set(lname, fname, office); }
   void set(string lname, string fname, string office) {
      lastName = lname;  firstName = fname; officeNumber = office;
   }
   void print() const {
      cout << "Last name: " << lastName << endl;
      cout << "First name: " << firstName << endl;
      cout << "Office number: " << officeNumber << endl;
   }
};
```

```
class TextBook {
private:
   string title, author, publisher;
public:
   TextBook()  { }
   TextBook(string textTitle, string auth, string pub)  {
      set(textTitle, auth, pub);
   }
   void set(string textTitle, string auth, string pub)  {
      title = textTitle; author = auth; publisher = pub;
   }
   void print() const  {
      cout << "Title: " << title << endl;
      cout << "Author: " << author << endl;
      cout << "Publisher: " << publisher << endl;
   }
};
```

```
class Course {
private:
   string courseName;
   Instructor instructor;
   TextBook textbook;
public:
   Course(string course,
            string instrLastName, string instrFirstName, string instrOffice,
            string textTitle, string author, string publisher) {
      courseName = course;
      instructor.set(instrLastName, instrFirstName, instrOffice);
      textbook.set(textTitle, author, publisher);
   }
   void print() const  {
      cout << "Course name: " << courseName << endl << endl;
      cout << "Instructor Information:\n"; instructor.print();
      cout << "\nTextbook Information:\n"; textbook.print();
      cout << endl; }
};
```

Przykład użycia

```
int main()
{
   Course myCourse(
      "Intro to Computer Science",          // Course name
      "Kramer", "Shawn", "RH3010",          // Instructor info
      "Starting Out with C++", "Gaddis",    // Textbook title and author
      "Pearson");                           // Textbook publisher

   myCourse.print();
}
```

Uwagi

- „Lepsza” wersja konstruktora

```
Course(string course,
        string instrLastName, string instrFirstName, string instrOffice,
        string textTitle, string author, string publisher)
        : instructor(instrLastName, instrFirstName, instrOffice),
          textbook(textTitle, author, publisher),
          courseName(course)
    {}
```

- Zamiast metod print można przeciążyć `operator<<`

Zadanie: Symulator parkomatu (Gaddis, Rozdział 14, Zad. 14)

W tym zadaniu utworzysz klasy, które będą współpracowały ze sobą i symulowały policjanta wypisującego bilet parkingowy. Utwórz następujące klasy:

- `ParkedCar` reprezentującą parkowany samochód,
- `ParkingMeter` reprezentującą parkometr,
- `ParkingTicket` reprezentującą bilet parkingowy,
- `PoliceOfficer` reprezentującą policjanta kontrolującego bilety parkingowe.

Napisz program demonstrujący współpracę powyższych klas.

ParkedCar

Przechowuje informacje o:

- marce
- modelu
- kolorze
- numerach rejestracyjnych
- czasie parkowania w minutach

`ParkingMeter`

Przechowuje informacje o:

- wykupionym czasie parkowania w minutach

ParkingTicket

- Przechowuje informacje o marce, modelu, kolorze i numerach rejestracyjnych nielegalnie zaparkowanego samochodu
- Wylicza mandat karny w wysokości 100 zł za pierwszą godzinę (lub część godziny) nielegalnego parkowania plus 40 zł za każdą kolejną rozpoczętą godzinę
- Podaje informację o imieniu, nazwisku i identyfikatorze policjanta, który wystawił mandat

`PoliceOfficer`

- Przechowuje informacje imieniu, nazwisku i identyfikatorze policjanta
- Weryfikuje zawartość obiektów typu `ParkedCar` i `ParkingMeter` oraz sprawdza czy upłynął czas parkowania
- Wypisuje mandat (tworzy obiekt typu `ParkingTicket`), jeśli upłynął czas parkowania

Przykładowe uruchomienie programu

(wg chegg.com)

```
Enter car make: Toyota
Enter car model: Camry
Enter car color: Blue
Enter car license number: LIC4356
Enter number of minutes parked: 45
Enter number of minutes purchased: 30
Enter officer name: Joe Friday
Enter badge number: BDG123
Police Officer Information:
        Name: Joe Friday
        Badge Number: BDG123
Car Information:
        Make: Toyota
        model: Camry
        Color: Blue
        License Number: LIC4356
        Minutes Parked: 45
Ticket Information:
        Minutes in violation: 15
        Fine: $ 35.00
```