

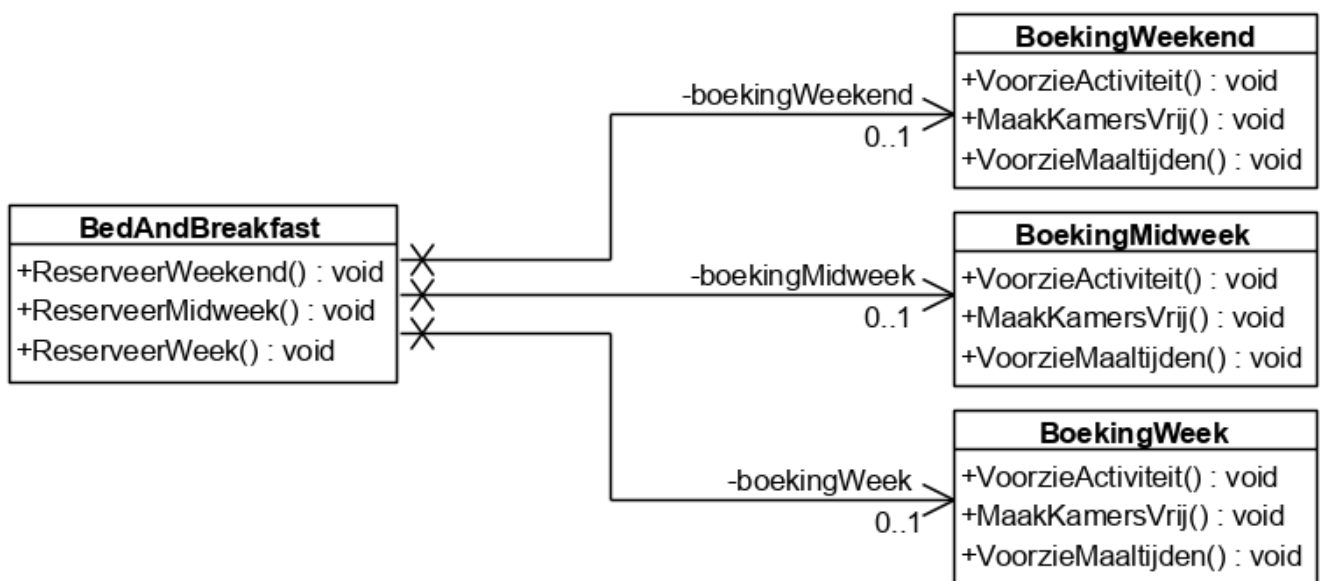
Ontwerp - Design patterns: Simple Factory pattern: Opdracht

1. Deadline

8 mei 2022

2. Opdracht: BedAndBreakfast

Een software bureau zit vast met volgend ontwerp:



Men zou dit ontwerp graag aanpassen met volgende functionaliteit in gedachte:

- Boekingens zouden inwisselbaar moeten zijn, ook tijdens het uitvoeren van de applicatie.
- De gebruiker van de klasse **BedAndBreakfast** zou een reservatie moeten kunnen maken a.d.h.v. een enumeratietype, zodat de gebruiker niet zelf het boekingsobject dient aan te maken.
- Nieuwe boekingstypes dienen mogelijk aangemaakt te worden, maar men wenst niet steeds de klasse **BedAndBreakfast** te moeten aanpassen.
- Momenteel laat het UML toe een **BedAndBreakfast** object aan te maken zonder boeking. Dit wenst men te voorkomen: een **BedAndBreakfast** object zou steeds over exact één boeking moeten beschikken.

Pas het ontwerp aan en geef een voorbeeld implementatie? Maak gebruik van het 'strategy pattern' en 'simple factory' pattern:

- Klasse diagram
- Maak als voorbeeld (= een implementatie in C#) een **BedAndBreakfastApp**:
 - implementeer alle klassen/interfaces uit je nieuwe ontwerp. Het eigenlijke gedrag voor

`VoorzieActiviteit()`, `MaakKamersVrij()` en `VoorzieMaaltijden` dien je niet te implementeren. Hier kan je een string naar de console schrijven die aangeeft om welke functionaliteit het gaat.

- Maak een `BedAndBreakfastApp` klasse:
 - een instantie van de klasse `BedAndBreakfast` met een initiële boeking dat je kiest a.d.h.v. een enum type
 - wissel de boeking naar een ander type boeking

3. Indienen - Opdracht opladen

- Comprimeer je Visual Studio project folder in een `zip` bestand, en dit zonder de subfolders `bin` en `obj`.
- Voeg aan dit zip bestand ook je Visual Paradigm bestand toe met daarin je klassediagram, en dit in de subfolder `vpproject`.

Hernoem dit zip bestand als volgt:

Naam_Voornaam_Ontwerp_SimpleFactory.zip

Laad dit bestand op in Chamilo onder de juiste opdracht.