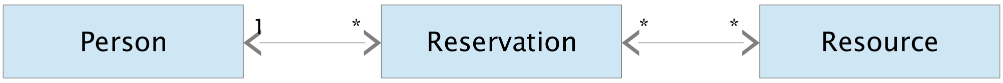
Prof. Dr. Eric Dubuis

Exercise on Abstract Factory

Introduction

Suppose your software is based on a certain number of inter-related concepts. However, due to different situations (different clients, different operating systems, different databases, etc.), your software should be configurable such that it supports *different implementations* of these concepts. In order to add the necessary flexibility of *creating instances* of the required implementation to your software, the *Abstract Factory* pattern comes into place.

Consider the following concept classes:



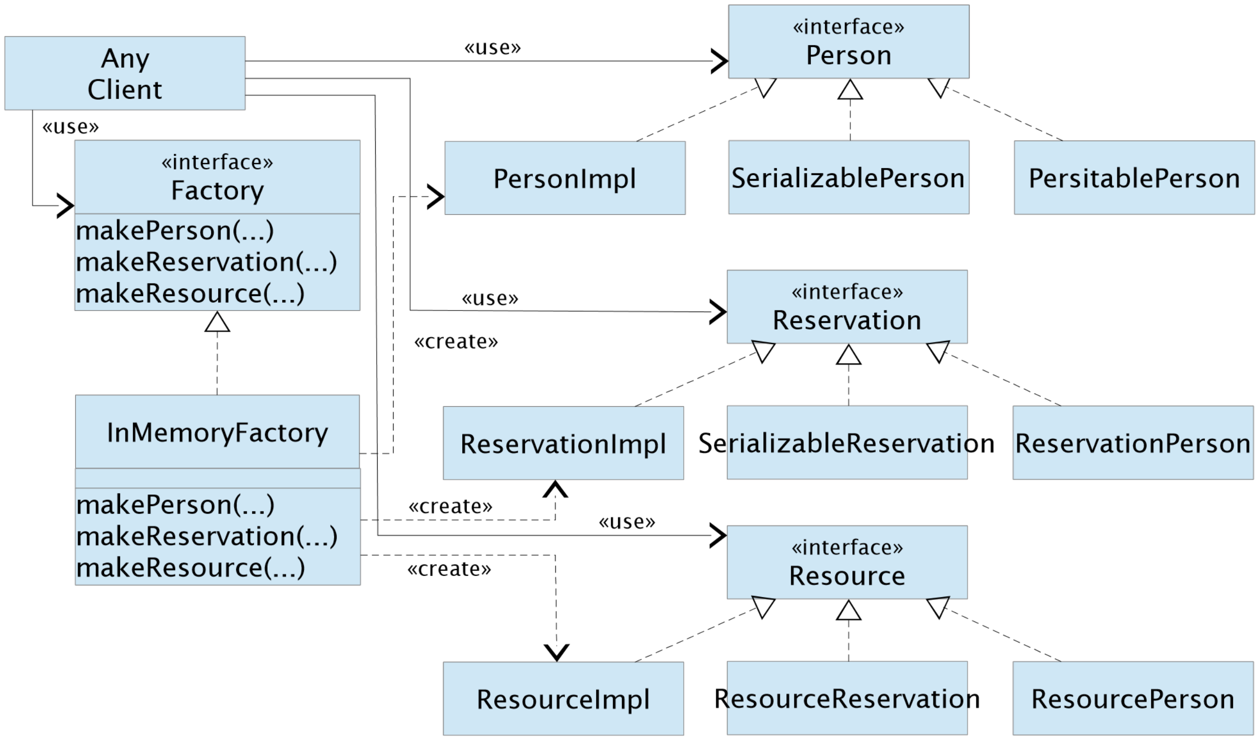
There are resources. Person can book (make a reservation) zero or more reservations. Each reservation books one or more resources. A resource can have many (timely distinct) reservations.

The controller class `ReservationManager` provides the feature for making reservations. It delegates the creation of a reservation to a concrete (e.g., configurable) factory class implementing the following interface:

```
public interface Factory {
       public PersistenceManager getPersistenceManager();
       public Person makePerson(String name);
       public Resource makeResource(String name);
       public Reservation makeReservation(Set<Resource> rs, Person p,
                                                              DateRange dr);
}
```

Note: The persistence manager's task is to store the newly created instances in the persistence store. (The in-memory persistence manager does not persist the instances of the concept classes, however.)

Among others (e.g., JUnit test classes), class `ReservationManager` is the client of a concrete factory class, as illustrated in the following UML class diagram:



Task

Complete the concrete factory classes you'll find in the source code. Watch the TODO markers. To test your implementation, run the JUnit tests.

Notice: The type of this project is Maven, i.e., this is a Maven project. Thus, import it as a Maven project into your IDE (Eclipse, IntelliJ, ...). It depends on “date-range”, version see `pom.xml`. Consult your course pages to learn how you can resolve this dependency.

Biel, November 20, 2019