

Aufgabe 3 Modell eines Reisebuchungssystems (Teil B)

Lernziele

1. Ein etwas umfangreicheres Modell implementieren.
2. Abstrakte Klassen und Interfaces verwenden.
3. Wissen über Collection-Klassen vertiefen
4. Wissen über Streams für Collections vertiefen.
5. Wissen über Enums anwenden.
6. Das Rechnen mit Währungsgrößen realisieren.

Hintergrund

Reisebuchungssysteme finden Sie im Internet vielfältig. Sie sollen mit dieser Aufgabe ein vereinfachtes Modell eines Reisebuchungssystems für Reisen, die aus Flüge, Hotelbuchungen und Mietwagen bestehen, umsetzen. Das Modell ist in der Struktur vorgegeben. Ihre Aufgabe wird es sein, das Modell zu verstehen und die abstrakten Methoden und Interfaces zu implementieren.

Aufbau des Projektes

Das Projekt enthält 6 Packages:

1. **de.hawhh.akteure**: Natürliche und juristische Personen, die bei einer Reisebuchung auftreten. (gegeben)
2. **de.hawhh.kosten**: Typen zur Repräsentation von Preisen und Rabatten.
3. **de.hawhh.reisebuchung**: Allgemeine Typen für eine Reisebuchung
4. **de.hawhh.reisebuchung.flug**: Typen für das Buchen von Flügen
5. **de.hawhh.reisebuchung.hotel**: Typen für das Buchen von Hotels
6. **de.hawhh.reisebuchung.mietwagen**: Typen für das Buchen von Mietwagen

Das Modell im Überblick

Package: *de.hawhh.reisebuchung*

Enthaltene Klassen und Interfaces (zur Beschreibung siehe Aufgabenteil A)

1. **ReiseBuchung**
2. **Buchung**
3. **Reise**
4. **AbstractReise**
5. **ReiseBaustein**
6. **EinfachReise**
7. **RundReise**
8. **Ort**

Die Interfaces Buchung und Reise

```
public interface Buchung {  
    public LocalDateTime getBeginn();  
    public LocalDateTime getEnde();  
    public GeldBetrag getPreis();  
    public Duration getDauer();  
    public Ort getEndOrt();  
    public Ort getStartOrt();  
}
```

Beginn und Ende einer Reise werden durch den Javatyp *LocalDateTime* repräsentiert. Dieser enthält Angaben zu Datum und Uhrzeit. Die Dauer einer Reise durch den Javatyp *Duration*. Mit der statischen Methode *Duration.between* lässt sich die Dauer zwischen zwei *LocalDateTime*-Objekten berechnen.

Package: *de.hawhh.reisebuchung.hotel*

Klassen / Interfaces / Enums:

1. *HotelBaustein*
2. *Hotel*
3. *Stern*
4. *Verpflegung*
5. *ZimmerTyp*

Ein *HotelBaustein* ist ein *ReiseBaustein*. Ein *HotelBaustein* hält eine Referenz auf das gebuchte *Hotel*, das Ankunfts- und Abreisedatum (*LocalDateTime*). Des Weiteren kennt ein *HotelBaustein* den gebuchten *ZimmerTyp* und die Art der *Verpflegung*. Die Objekte für diese Eigenschaften werden dem *HotelBaustein* bei der Erzeugung übergeben.

Das *Hotel* hat einen *Anbieter*, einen Namen (*String*), eine Kategorie (*Stern*), einen *Ort*. Ein *Hotel* verwaltet die Preisinformation für die Zimmer in einer Tabelle, in der jedem *ZimmerTyp* ein Tagespreis (*GeldBetrag*) zugeordnet ist. Ebenso verwaltet das *Hotel* eine Tabelle, in der jedem *Verpflegungstyp* (*Verpflegung*) ein Tagespreis (*GeldBetrag*) zugeordnet ist.

Um die Tabellen für Zimmerpreise und Verpflegungspreise initialisieren zu können, werden dem *Hotel* bei der Erzeugung für Zimmerpreise ein Array von *Zimmertypen* und ein Array für die Preise und für die Verpflegungspreise ein Array für die *Verpflegungstypen* und ein Array für die Preise übergeben.

ZimmerTyp, *Stern* und *Verpflegung* sind Enums. *ZimmerTyp* ist EINZEL, DOPPEL, APPARTMENT, EINZELLUXUS, DOPPELLUXUS, APPARTMENTLUXUS. *Stern* ist EINS, ZWEI, DREI, VIER, FUEF. *Verpflegung* ist KEINE, HALBPENSION, VOLLPENSION, ALL_IN. Bei der Darstellung der Enums sollen korrekte deutsche Begriffe in korrekter Groß- und Kleinschreibung ausgegeben werden. Präparieren Sie die Enums entsprechend.

Zur Preisgestaltung: Der Preis für einen Hotelaufenthalt berechnet sich aus der Summe der Tagespreise für den Zimmertyp und die Verpflegung multipliziert mit der Anzahl der Tage, multipliziert mit dem Rabatt, den der **Anbieter** für eine Wochenanzahl gewährt. Die Methode **getPreis(ZimmerTyp,Verpflegung,Duration)** der Klasse **Hotel** führt diese Berechnung durch. Die Methode **getPreis** der Klasse **HotelBaustein** nutzt die Methode von **Hotel** zur Berechnung.

Die Klasse **Duration** hat eine Objekt-Methode für die Umwandlung in Tage. Die Klasse **Anbieter** hat die Methode **rabatt(long wochen)**, die den Rabatt als double im Intervall (0..1] liefert. (Implementierung der Klasse **Anbieter** ist gegeben).

Package: de.hawhh.reisebuchung.mietwagen

Klassen / Interfaces / Enums:

1. **MietwagenBaustein**
2. **Mietwagen**
3. **Station**
4. **Uebergabe**
5. **WagenTyp**

Ein **MietwagenBaustein** ist ein **ReiseBaustein**. Ein Mietwagenbaustein hält eine Referenz auf den gebuchten Mietwagen, die Abholung und die Abgabe (**Uebergabe**). Die Objekte für diese Eigenschaften werden dem Mietwagenbaustein bei der Erzeugung übergeben. Die **Uebergabe** hält eine Referenz auf die **Station** und das Übergabedatum (**LocalDateTime**). Die Objekte für diese Eigenschaften werden bei der Erzeugung übergeben.

Der **Mietwagen** hält eine Referenz auf den Anbieter (**MietwagenAnbieter**) und den Wagentyp (**WagenTyp**). Die Objekte für diese Eigenschaften werden bei der Erzeugung übergeben.

Der **MietwagenAnbieter** ist ein Anbieter mit einem Namen, einer Referenz auf das **RabattModell** und einer Tabelle für die Tagespreise der Wagentypen. Bei der Erzeugung werden Name und Rabatt-Modell sowie ein Array für die Wagentypen und ein Array für die Preise der Wagentypen (**GeldBetrag**) übergeben. Der Mietwagen-Anbieter wandelt die beiden Arrays intern in eine Tabelle um, die einem Wagentyp einen Geldbetrag zuordnet.

WagenTyp ist eine Klasse (kein Enum, da sich die Mietwagen-Anbieter der realen Welt durch individuelle Bezeichnungen für ihre Wagenkategorien signifikant unterscheiden.), deren Objekte mit einem Namen erzeugt werden.

Zur Preisgestaltung

Der Preis für einen Mietwagen berechnet sich aus dem Tagespreis für einen Wagentyp multipliziert mit der Anzahl der Tage, multipliziert mit einem Rabatt, der von der Anzahl der Wochen abhängt. Die Methode **getPreis(WagenTyp,Duration)** der Klasse **MietwagenAnbieter**, nimmt die Berechnung vor. Die Methode **getPreis(Duration)** der Klasse **Mietwagen** liefert den Preis für die gesamte Mietdauer.

TODO

Implementieren Sie alle Klassen / Enums / Interfaces, die noch nicht in der Projektvorlage enthalten sind. Generieren Sie für alle Klassen und Enums geeignete *toString()* Methoden sowie für die Klassen Methoden für Gleichheit und Hash Code.

Denken Sie sich 3 Hotels und 2 Mietwagenanbieter aus und legen Sie für diese die Preise für die einzelnen Komponenten fest. Legen Sie für die Mietwagenanbieter die zulässigen Wagentypen fest und definieren Sie für jeden Mietwagenanbieter je 10 Übergabestationen.

Überlegen Sie wie Sie die Objekte für *Hotel*, *MietwagenAnbieter* und *Station* mittels Generatoren einfach erzeugen können.