In-class activity: Application 3 - detailed design

LWG3 In-class activity Application 3 conceptual design and test plan

Christian Sjoberg

venkatesh perumal

sai lalith kumar reddy pati

Task 1

Derive a high-level conceptual design/model for the back-end module that supports the creation of a vacation package according to the user’s choices and calculation of costs. Do not worry about anything that is unrelated to this overall goal.

Conceptual design:

List of nouns, with notes about which ones become classes/interfaces:

Package(Interface)
BasePackage(base/mandatory)
Decorator(Decorator abstract class)
HotelDecorator(Decoration)
RentalCarDecorator(Decoration)
RoundTripFLight(class)
FLightValidate(class)
Hotel(class)
HotelValidate (class)
RentalCar(class)
RentalValidate(class)

Interactions/relationships between interfaces/classes:

BasePackage implements Package
Decorator implements Package
BasePackage uses RoundTripFlight
HotelDecorator uses Hotel
RentalCarDecorator uses RentalCar
FlightValidate uses RoundTripFlight
HotelValidate uses Hotel
RentalValidate uses RentalCar

Responsibilities of Interfaces/classes:

Package(Interface)

- Getter for description.
- Method to calculate cost.
- Method to calculate total cost of package.

FlightValidate(concrete class)

- Has parameterized constructor FlightValidate.
- Method to validate inputs for flight.
- Returns RoundTripFlight.

HotelValidate(concrete class)

- Has parameterized constructor HotelValidate.
- Method to validate inputs for Hotel.
- Returns Hotel.

RentalValidate(concrete class)

- Has parameterized constructor RentalValidate.
- Method to validate inputs for RentalCar.
- Returns RentalCar.

RoundTripFlight(Concrete Class)

- Getters and setters for origin and destination airport.
- Getters and setters for departure and return date.
- Getters and setters for adult and child passengers.

Hotel(Concrete Class)

- Getters and setters for CheckIn and CheckOut date.
- Getters and setters for number of occupants in each room(tuple:pairs).
- Getters and setters for adult and child passengers.

RentalCar(Concrete Class)

- Getters and setters for pickup location and drop-off location.
- Getters and setters for pickup time and drop-off time.

BasePackage

- Takes RoundTripFLight instance.
- Has method to get description.
- Has method to calculate cost of base package component.

- Has method to calculate cost of total package(if any add-ons).

Decorator(Abstract Class)

- Decorator takes instance of BasePackage.
- getter for description.
- Method to calculate cost.

HotelDecorator(Concrete Decorator)

- Takes BasePackage instance.
- Has method to get description of package.
- Has method to calculate cost of Hotel component.
- Has method to calculate cost of total package.

RentalCarDecorator(Concrete Decorator)

- Takes BasePackage instance.
- Has method to get description of package.
- Has method to calculate cost of RentalCar component.
- Has method to calculate cost of total package.

Design Pattern: Decorator

Here the BasePackage is the concrete base package with RentalCar and Hotel as the add-ons(decorators) and any combination of base package with add-ons will give a different cost.

Test Plan:

BasePackage- This class will require a mock of the RoundTripFlight class.

HotelDecorator - This class will require a mock of the BasePackage and Hotel class to test its ability to decorate or modify its attributes. Testing will assert that the expected changes have been made to the mock BasePackage and that all conditions have the expected behavior.

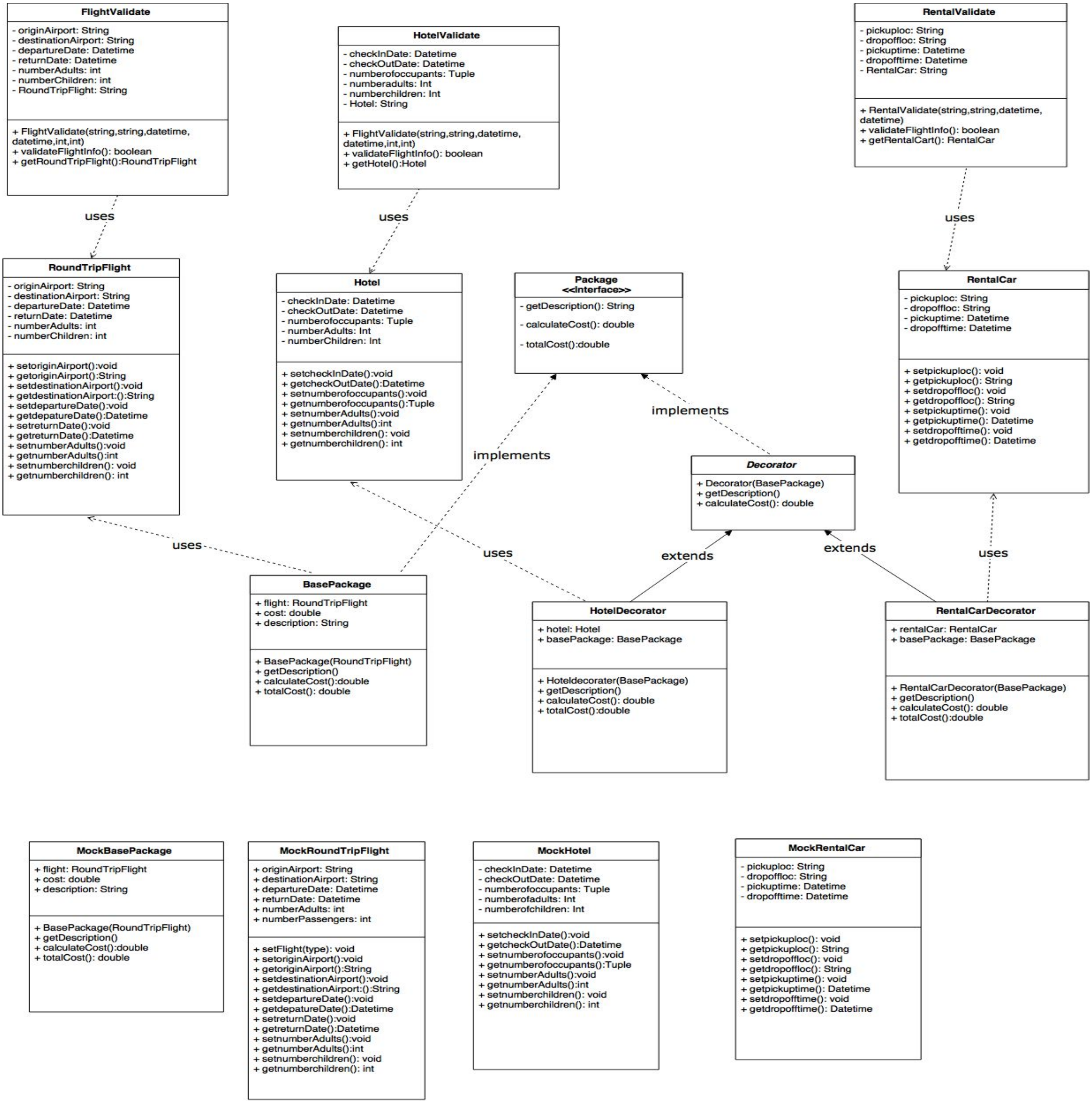
RentalCarDecorator - This class will require a mock of the BasePackage and RentalCar class to test its ability to decorate or modify its attributes. Testing will assert that the expected changes have been made to the mock BasePackage and that all conditions have the expected behavior.

Hotel, RentalCar, and RoundTripFlight - These classes will not require mocks for testing. They will be directly tested to ensure that all conditions have the expected behavior.

FlightValidate - Will require a mock of the RoundTripFlight class for testing.

HotelValidate - Will require a mock of the Hotel class for testing.

RentalCarValidate - Will require a mock of the RentalCar class to validate.


FlightValidate
- originAirport: String
- destinationAirport: String
- departureDate: Datetime
- returnDate: Datetime
- numberAdults: int
- numberChildren: int
- RoundTripFlight: String
+ FlightValidate(string,string,datetime, datetime,int,int)
+ validateFlightInfo(): boolean
+ getRoundTripFlight():RoundTripFlight
HotelValidate
- checkInDate: Datetime
- checkOutDate: Datetime
- numberofoccupants: Tuple
- numberadults: Int
- numberchildren: Int
- Hotel: String
+ FlightValidate(string,string,datetime, datetime,int,int)
+ validateFlightInfo(): boolean
+ getHotel():Hotel
RentalValidate
- pickuploc: String
- dropofffloc: String
- pickuptime: Datetime
- dropofftime: Datetime
- RentalCar: String
+ RentalValidate(string,string,datetime, datetime)
+ validateFlightInfo(): boolean
+ getRentalCart(): RentalCar
uses
uses
uses
RoundTripFlight
- originAirport: String
- destinationAirport: String
- departureDate: Datetime
- returnDate: Datetime
- numberAdults: int
- numberChildren: int
+ setoriginAirport():void
+ getoriginAirport():String
+ setdestinationAirport():void
+ getdestinationAirport:():String
+ setdepartureDate():void
+ getdepatureDate():Datetime
+ setreturnDate():void
+ getreturnDate():Datetime
+ setnumberAdults():void
+ getnumberAdults():int
+ setnumberchildren(): void
+ getnumberchildren(): int
Hotel
- checkInDate: Datetime
- checkOutDate: Datetime
- numberofoccupants: Tuple
- numberAdults: Int
- numberChildren: Int
+ setcheckInDate():void
+ getcheckOutDate():Datetime
+ setnumberofoccupants():void
+ getnumberofoccupants():Tuple
+ setnumberAdults():void
+ getnumberAdults():int
+ setnumberchildren(): void
+ getnumberchildren(): int
Package
<<Interface>>
- getDescription(): String
- calculateCost(): double
- totalCost():double
RentalCar
- pickuploc: String
- dropoffloc: String
- pickuptime: Datetime
- dropofftime: Datetime
+ setpickuploc(): void
+ getpickuploc(): String
+ setdropoffloc(): void
+ getdropoffloc(): String
+ setpickuptime(): void
+ getpickuptime(): Datetime
+ setdropofftime(): void
+ getdropofftime(): Datetime
implements
implements
Decorator
+ Decorator(BasePackage)
+ getDescription()
+ calculateCost(): double
uses
uses
extends
extends
uses
BasePackage
+ flight: RoundTripFlight
+ cost: double
+ description: String
+ BasePackage(RoundTripFlight)
+ getDescription()
+ calculateCost():double
+ totalCost(): double
HotelDecorator
+ hotel: Hotel
+ basePackage: BasePackage
+ Hoteldecorater(BasePackage)
+ getDescription()
+ calculateCost(): double
+ totalCost():double
RentalCarDecorator
+ rentalCar: RentalCar
+ basePackage: BasePackage
+ RentalCarDecorator(BasePackage)
+ getDescription()
+ calculateCost(): double
+ totalCost():double
MockBasePackage
+ flight: RoundTripFlight
+ cost: double
+ description: String
+ BasePackage(RoundTripFlight)
+ getDescription()
+ calculateCost():double
+ totalCost(): double
MockRoundTripFlight
+ originAirport: String
+ destinationAirport: String
+ departureDate: Datetime
+ returnDate: Datetime
+ numberAdults: int
+ numberPassengers: int
+ setFlight(type): void
+ setoriginAirport():void
+ getoriginAirport():String
+ setdestinationAirport():void
+ getdestinationAirport:():String
+ setdepartureDate():void
+ getdepatureDate():Datetime
+ setreturnDate():void
+ getreturnDate():Datetime
+ setnumberAdults():void
+ getnumberAdults():int
+ setnumberchildren(): void
+ getnumberchildren(): int
MockHotel
- checkInDate: Datetime
- checkOutDate: Datetime
- numberofoccupants: Tuple
- numberofadults: Int
- numberofchildren: Int
+ setcheckInDate():void
+ getcheckOutDate():Datetime
+ setnumberofoccupants():void
+ getnumberofoccupants():Tuple
+ setnumberAdults():void
+ getnumberAdults():int
+ setnumberchildren(): void
+ getnumberchildren(): int
MockRentalCar
- pickuploc: String
- dropoffloc: String
- pickuptime: Datetime
- dropofftime: Datetime
+ setpickuploc(): void
+ getpickuploc(): String
+ setdropoffloc(): void
+ getdropoffloc(): String
+ setpickuptime(): void
+ getpickuptime(): Datetime
+ setdropofftime(): void
+ getdropofftime(): Datetime