DESIGN OF RAILWAY BOOKING SYSTEM

Name: Parth Tusham
Roll No: 19CS30034

This is the design document stating the design details of (especially) LLD,HLD with principles guidelines followed, making use of specific constructs and idioms.

The following are the Design and implementation overview of the classes in the system.

Date:

- It has the following attributes date_, month_ and year_.
- The following are the overloaded operators
    - **Outstream operator** (<<): To easily output date.
    - **Instream operator** (>>): To input Date easily.
    - **Equality operator** (==): To check if two dates are equal.
    - **Difference operator** (-): To calculate age of a person through DOB and current date.
- It has the following methods:
    - **GetDay():** returns the day of the given date
    - **GetMonth()**: returns the month of the given date
    - **GetYear():** returns the year of the given date

Railways:

- It is a Meyer singleton class.
- It has two maps :
    - **sStations:** Contains list of the station in master data
    - **DistStations:** Contains the distance between any two given valid station
- It has a method GetDistance() to fetch the distance between any two stations provided as string arguments.

Station:

- It contains a single attribute which is the name of the station as a string data type.
- It has the following overloaded operators:
    - **Equality operator** (==): to check if two given stations are the same or not.
    - **Ostream operator** (<<): To easily output the station details.
- It has the following methods :

- **GetName()**: returns the name of the given station as a string
- **GetDistance(): r**eturns the distance between the current and given station as int data type

Passenger:

- This class stores the info about the the passenger and has the following attributes:
  - **firstName**: First name of the passenger stored as string.
  - **middleName:** Middle name of the passenger stored as string.
  - **lastName:** Last name of the passenger stored as string.
  - **dateOfBirth:** DOB of the passenger stored as a Date data type.
  - **gender:** gender of the passenger stored as Gender data type
  - **aadhaar**: aadhar card id stored as a string.
  - **mobile:** mobile number stored as a string.
  - **disabilityType**: disability type stored as a string.
  - **disabilityId**: disability id according to master data
  - **Category:** tells which type of concession category person belongs to.
- createPassenger() method is used to construct the passenger instance in which error handling is also done to check whether the last or first name exists or not and if the given date of birth is valid or not .If the given data is valid we'll create the passenger instance else throw error of bad_passenger.
- It also contains the following methods:
  - **getAge():** returns the age.
  - **getDisType()**: returns the disability type as a string .
  - **getCategory(): r**eturns the concession category person belongs to.

BookingClasses:

- It's an abstract class because derived classes need class specific functions hence methods are declared virtual here which are to be overloaded by the respective derived children classes.
- I have implemented the following pure virtual methods and are publicly declared.

    - **GetLoadFactor()**
    - **GetName()**
    - **IsAC()**
    - **IsSitting()**
    - **IsLuxury()**
    - **GetNumberOfTiers()**
- I have also provided the class with a virtual destructor because we have kept pointers to BookingClasses in the booking file .
- The constructor is declared as protected so that only derived children classes can only access it.
- It has the following attributes:
    - **loadFactor:** stores the load factor
    - **name**: name of the sub(child) class
    - **tatkalLoadfactor:** tatkal load factor
    - **minTatkalCharge**: minimum tatkal charge
    - **maxTatkalCharge**: maximum tatkal charge
    - **minTatkalDist:** minimum tatkal distance
- It has the following Methods:
    - **getLoadFactor():** double
    - **GetName():** returns the name of the booking class.
    - **IsStiting():** tells if there is sitting or not
    - **IsAC():** tells if the booking class contains AC or not.
    - **GetNumberOfTiers()**: returns number of tiers
    - **isLuxury()**: tells if the current booking child class is considered luxury or not.
    - **operator<<**(operator, BookingClasses): ostream (overloaded operator)
    - It has 8 child classes(flat hierarchy) which are Meyer singleton classes in which most of the above given methods are overloaded.

- Below are the 8 child (leaf) classes of BookingClass
    1. **Sleeper**
    2. **ExecutiveChairCar**
    3. **FirstClass**
    4. **AC3Tier**
    5. **ACFirstClass**
    6. **AC2Tier**
    7. **ACChairCar**
    8. **SecondSitting**
- In the above given 8 classes contains the following attributes and methods:
    - **name**: name of the class.

- **loadFactor**: stores the load factor
- **Type()**: returns the type of the class.
- These are the overridden methods **IsLuxury(), IsAC(), GetLoadFactor(),GetName().**

Booking

- It is a hierarchy less class designed for ease of maintainability and simplicity also reduces repetitive code.
- It has the following attributes
  - **PNRserial:** stores the PNR number as int(static to assign every booking respectively)
  - **BaseFare**: stores the base fare of travel as double
  - **sBaseFarePerKM:** stores the base fare per Km of travel as double
  - **ACSurcharge**: stores the AC surge charge as double
  - **LuxuryTaxPercent:** stores the tax on luxury as double
  - **BookingStatus_**: stores the booking status as a string
  - f**romStation:** it stores the station to travel from as a user defined station data type
  - **toStation:** it stores the station to travel to as a user defined station data type.
  - **date:** Date of travel stored as a user defined Date data type
  - **bookingClass**: The class of travel stored as reference to the object
  - **bookingCategory:** The class of concession category stored as reference to its object
  - **fare_:** the total fare stored as double
  - **selfPNRnumbe**r_: stores the current PNR number as a int
  - **sBookings**: vector<Booking* >: contains the all previously done booking
- The ostream operator is overloaded for ease of debugging.
- The following are the methods :
  - **NextPNR()**: increases the static PNR by one.
  - **GetFromStation()**: getter function for FromStation attribute.
  - **GetToStation()**:getter function for ToStation attribute.
  - **GetDate()**:getter function for date attribute.
  - **GetFare()**: getter function for fare attribute.
  - **GetSelfPNR()**: getter function for selfPNR attribute.
  - **GetBookingStatus()**:getter function for BookingStatus attribute.
  - **GetBookingClassName()**:getter function for name attribute.

- **ComputeFare()**: contains the business logic for computing fare.
- **CheckBooking()**:checks if the booking is valid (as per given logic)
- **DoBooking()**: adds the valid booking to the vector

Exceptions

- Multiple exceptions classes are inherited from std::exception: Bad_Passenger, Bad_Date, Bad_Railways and Bad_Booking.
- These are thrown when validity checks are failed, errors are thrown. These don't allow the system to go into an inconsistent state.
- Some exceptions classes are further specialised into subclasses to be more indicative of the errors they represent. Bad_Passenger has subclasses to represent whether there was an error in data for: name, age, aadhaar, mobile,etc

Concession:

- It is a **simple singleton** set which has a **hashmap** of all the concessionFactors related to Divyang in it.
- The attributes are:
  - **Concession_mat** : (map < pair< string, string>, float>) first of pair corresponds to the bookingClass name and second of the pair corresponds to Divyang Class name.
- The method of this class is:
  - **getConcessionFactor()** : it takes two strings as an input which are the name of the BookingClass and name of the DivyangCategory and returns the corresponding float value related to it.

Booking Category:

- It is abstract base class having attribute
  - **name** : (string) representing name of the Booking Category.

- It has the following methods
  - **getName()** : returns the string representing the name of the booking category
  - **getConcessionFactor()** : returns the float value corresponding to concessionFactor.

- Now the eligibility of the passenger to be valid for this booking category is contained in the Booking as method as **CheckBooking().**

- The business logic is implemented by getting the concessionFactor from BookingCategory and then it is used in the Booking.

The Hierarchy of BookingCategory is divided into 6 subclasses which are as follows:

- **SeniorCitizenBooking**
    - It's name is set as “SeniorCitizenBooking”.
    - It has two static parameters of maleConcession and femaleConcession.
    - It's getConcessionFactor returns to either the value of maleConcession or femaleConcession by getting the gender of the Passenger.

- **Ladies Booking**
    - Its name is set as “LadiesBooking”.
    - Its current getConcessionFactor returns 0 by default.
    - Its eligibility is checked earlier only.

- **TatkalBooking**
    - It's name is set as “TatkalBooking”.
    - Its current getConcessionFactor returned is 0 and in Booking the fare is calculated in a different way for it.
    - It is not checked for eligibility.

- **PremiumTatkalBooking**
    - It's name is set as “PremiumTatkalBooking”.
    - Its current getConcessionFactor returned is 0 and in Booking the fare is calculated in a different way for it similar to Tatkal computation.
    - It is also not checked for eligibility.

- **DivyangCateory:**
    - Its concessionFactor is returned by using the getConcessionFactor() method of class Concession and getting the corresponding value of concessionFactor.
    - It is an abstract subclass of the BookingCategory consisting of 4 sub classes.
        - **BlindBooking**
            - Its name is set as BlindBooking
            - The getConcessionFactor takes BookingClass as an input and returns the concessionFactor from the Concession Class.
        - **TBBooking**

- Its name is set as TBBooking
- The getConcessionFactor takes BookingClass as an input and returns the concessionFactor from the Concession Class.

- **OrthoBooking**
  - Its name is set as OrthoBooking
  - The getConcessionFactor takes BookingClass as an input and returns the concessionFactor from the Concession Class.
- **CancerBooking**
  - Its name is set as CancerBooking
  - The getConcessionFactor takes BookingClass as an input and returns the concessionFactor from the Concession Class.