

Design

By CHAPPIDI YOGA SATWIK (19CS30013)

General Conventions

- All classes have friend operator << () functions to print the state of the object for debugging.
- All classes have static UnitTest() functions
- All static Methods are named as sNameOfObject
- All non-static Methods are names nameOfObject_
- Every class has a destructor but is not mentioned in the document.
- Every class mentioned as leaf class has all virtual functions implemented in the class higher in the hierarchy

1 Date

Attributes

- sMonthNames: Static const private vector contains all the names of the months.
- Static const private vector sDayNames: Static const private vector contains all the names of days
- date_: Private unsigned int stores the date.
- month_: Private Month stores the month as an enum.
- year_: Private unsigned int stores the year

Methods

- Date(): Public constructor takes in a day, month, and year.
- CheckYearSpan(): Static public function to check that the date is within a year from today.
- GetAge(): Static public function that returns the age as of today given the birthday.
- ValidDate(): Static public function that checks if date is valid (unlike, 32/3/1)

2 Railways

Attributes

- sStations: Static private vector contains all Station objects belonging to the railways
- sDistStations: Static private map<string, map<string, int>> that stores the distances between two stations

- sName: Static public const string stores the name of the Railways

Methods

- Railways(): Private constructor to ensure singleton nature.
- SetStations(): Private function sets the values in sStations
- SetDistStations(): Private function set the values in sDistStations
- GetDistances(): Public function that returns the distance between two stations.
- IndianRailways(): Static Public function that enables the access of the singleton Railways object.

3 Station

Attributes

- name_: private const string which stores the name of Station

Methods

- Station(): public constructor
- GetName(): public function that returns the name of the station.
- GetDistance(): public function that returns the distance from a station (input)

4 Gender

This is the abstract base class for the flat hierarchy of templated GenderType.

Methods

- Gender(): protected constructor so it can be called by GenderType

5 GenderType

This is the template class for Male and Female. They are represented as GenderType<MaleType> and GenderType<FemaleType> respectively.

Attributes

- sName: static private const string storing the name of the gender

Methods

- GenderType(): Private constructor to enable singleton nature.
- Type(): Public Static function that returns the singleton GenderType<T> object.

6 Passenger

Attributes

- firstName_: private const string that stores the first name of the passenger
- middleName_: private const string that stores the middle name of the passenger
- lastName_: private const string that stores the last name of the passenger
- DOB_: private Date that stores the Date of Birth
- gender_: private GenderType stores the gender of the passenger
- aadhar_: private string stores unique Aadhar ID
- mobile_: private string stores phone number
- disabilityType_: private DivyaangType stores the type of disability of the passenger
- disabilityID: private string stores the disability ID of the type of disability

Methods

- CheckFields(): public function that checks all required fields of the passenger are available
- Passenger(): private constructor can only be called by Get()
- Get(): public static function that performs checks before the construction of passenger and returns the object

7 BookingClass

This is the abstract base class for the flat hierarchy of templated BookingClassType.

Attributes

- name_: protected const string that stores the name of the BookingClass

Methods

- BookingClass(): protected constructor so that only the children of this class can construct this.
- GetLoadFactor(): public virtual function that returns the load factor for the booking class.
- GetName(): public virtual function that returns the name of the booking class.
- IsSitting(): public virtual function that returns if the booking class is sitting or not.

- IsAC(): public virtual function that returns if the booking class is AC or not.
- GetNumberOfTiers(): public virtual function that returns the number of tiers in the booking class.
- IsLuxury(): public virtual function that returns if the booking class is Luxury or not.
- GetReservationCharge(): public virtual function that returns the reservation charge.
- GetTatkalPercent(): public virtual function returns the percent charge for Tatkal Booking
- GetMinTatkalCharge(): public virtual function returns the minimum charge for Tatkal Booking
- GetMaxTatkalCharge(): public virtual function returns the maximum charge for Tatkal Booking

8 BookingClassType

This is the class template for each of BookingClass in the railways system. They can be accessed with BookingClassType<Name>. All the virtual functions from the above abstract base class are implemented.

Attributes

- sSitting: private static bool that is true if the booking class is sitting.
- sAC: private static bool that is true if the booking class is AC.
- sNumberOfTiers: private static unsigned int that stores the number of tiers in the booking class.
- sLuxury: private static bool that is true if the booking class is luxury class.
- sReservationCharge: private static unsigned int that stores value of reservation charge for the booking class.
- sTatkalPercent: private static double the percent rate of extra charge for tatkal booking
- sMinTatkalCharge: private static unsigned int the minimum levied charge for tatkal booking.
- sMaxTatkalCharge: private static unsigned int the maximum levied charge for tatkal booking.

Methods

- BookingClassType(): private constructor that can be called by Type()
- Type(): public static function to return a singleton BookingClassType object.

9 BookingCategory

This is an abstract base class that acts as the root for the entire BookingCategory hierarchy. This hierarchy involves three main categories: GeneralCategory, ConcessionalCategory, and PriorityCategory.

Attributes

- name_: protected const string that stores the name of the BookingCategory.

Methods

- BookingCategory(): protected constructor so it can be constructed only by its children.
- CheckEligibility(): public virtual function that checks if a passenger eligible for the booking category.
- GetName(): returns the name of the booking category
- GetNewBooking(): public virtual function that calls the construction of the appropriate Booking class.

10 GeneralCategory

This is a leaf class in the hierarchy with CheckEligibility implemented. This is available for every passenger.

Methods

- GeneralCategory(): private constructor so it can be called only by Type()
- Type(): public static function that maintains the singleton nature of GeneralCategory
- CreateNewBooking(): public function (implementation of the virtual function above) that calls for the construction of GeneralBooking in Booking.

11 PriorityCategory

This is an abstract base class for a flat hierarchy of Tatkal and PremiumTatkal.

Methods

- PriorityCategory(): protect constructor so it can be called by its children.
- GetPriorityCharge(): public virtual function that returns the additional charge for tatkal or premium tatkal booking.
- CreateNewBooking(): public function (implementation of the virtual function above) that calls for the construction of PriorityBooking in Booking.

12 Tatkal

This is a leaf class that has GetPriorityCharge, CheckEligibility implemented.

Methods

- Tatkal(): private constructor that can be called by Type()
- Type(); public static function that maintains the singleton nature of Tatkal

13 PremiumTatkal

This is a leaf class that has GetPriorityCharge, CheckEligibility implemented.

Methods

- PremiumTatkal(): private constructor that can be called by Type()
- Type(); public static function that maintains the singleton nature of PremiumTatkal

14 ConcessionalCategory

This is the abstract base class and acts as the root for the SeniorCitizen and Divyaang.

Methods

- ConcessionalCategory(): protected constructor so it can be constructed by its children.
- GetConcession(): public virtual function that returns the concession rate for the booking category.
- CreateNewBooking(): public function (implementation of the virtual function above) that calls the construction of ConcessionBooking in Booking.

15 SeniorCitizen

This is a leaf class that has GetConcession, CheckEligibility implemented.

Attributes

- passengerGender_: private GenderType that is the gender of the passenger received when CheckEligibility is called.
- sConcessionMale: private static double that is the concession for Senior Citizen males
- sConcessionFemale: private static double that is the concession for Senior Citizen females.

Methods

- SeniorCitizen: private constructor that can be called by Type()

- Type(); public static function that maintains the singleton nature of SeniorCitizen

16 Divyaang

This is an abstract base class for a flat templated hierarchy of DivyaangType.

Methods

- Divyaang(): protected constructor so that it can be constructed by its children
- GetDisabilityName(): private virtual function that returns name of the disability type

17 DivyaangType

This is the class template for each Disability type. Each of them can be accessed by DivyaangType<Name>. It is also a leaf class with CheckEligibility(), GetConcessional(), GetDisabilityName() implemented.

Attributes

- sConcessionACFirstClass: private static double that stores the concession rate for ACFirstClass.
- sConcessionAC2Tier: private static double that stores the concession rate for AC2Tier.
- sConcessionAC3Tier: private static double that stores the concession rate for AC3Tier.
- sConcessionACChairCar: private static double that stores the concession rate for ACChairCar.
- sConcessionExecutiveChairCar: private static double that stores the concession rate for ExecutiveChairCar.
- sConcessionFirstClass: private static double that stores the concession rate for FirstClass.
- sConcessionSleeper: private static double that stores the concession rate for ACSleeper.
- sConcessionSecondSitting: private static double that stores the concession rate for SecondSitting.
- sDisabilityName: private static string that stores the

Methods

- DivyaangType(): private constructor that can be called by Type()
- Type(): public static function that maintains the singleton nature of DivyaangType

18 Booking

This is the abstract base class for a flat hierarchy containing GeneralBooking, ConcessionBooking, and PriorityBooking.

Attributes

- sPNRSerial: private static unsigned int that remembers the next PNR for a booking.
- sNoOfBooking: private static unsigned int that stores the number of bookings in the system.
- sBaseRate: private static double that stores the base rate for each km of distance.
- PNR_: PNR of the booking
- dateOfReservation_: Date of Reservation of the booking
- dateOfBooking_: Date of Booking
- bookingStatus_: status of the booking
- bookingClass_: booking class of the booking
- bookingCategory_: booking category of the booking
- passenger_: passenger travelling in the train
- start_: start Station
- dest_: destination Station
- sBookings: public static vector of Bookings that stores every successful Booking.

Methods

- CreateBooking(): public static helper function that performs necessary checks and constructs the necessary child of booking based on BookingCategory
- Booking(): protected constructor so it can be called by its children.
- ComputeFare(): protected virtual function that computes the fare with different logic based on type.

19 GeneralBooking

This is the leaf class that has ComputeFare implemented.

Methods

- GeneralBooking(): private constructor that can be called by GeneralCategory::CreateNewBooking()

20 ConcessionBooking

This is the leaf class that has ComputeFare implemented.

Methods

- `ConcessionBooking()`: private constructor that can be called by `ConcessionalCategory::CreateNewBooking()`

21 PriorityBooking

This is the leaf class that has `ComputeFare` implemented.

Methods

- `PriorityBooking()`: private constructor that can be called by `PriorityCategory::CreateNewBooking()`