Calculator of cost travel and flights

Samuel J. Rengifo A00404150

Analysis

Description: The program will work as a calculator of travel cost and flights which will allow **Miguel** (passionate traveler) to discover new destinations and explore options for his flights alone or with his friends.

Input:

1. **int** menuSelectOption **(1, 2 , 3, 4)**
   -The user will input a number from 1 to 4, in that order he will select an option from the main menu.

2. **double** passengerW **(x>0, x<=109)**
   -The user will indicate his estimated weight that he is carrying to flight with.

3. **String** chooseRate **(XS - S - M)**
   -The user will choose a rate, the valid rates will be XS, S and M.

4. **String** ynSeatSelection **(Yes/No)**
   -If the user did not select the rate M, he will decide to take or not the seat selection service.
   4.1 **String** choosePositionSeat **(Aisle - Middle - Window)**
   -If the user selects this service or if he is traveling at M rate, they will have to choose the seat position to fly with.

5. **String** ynChooseExtraTenSuitcase **(Yes/No)**
   -No matter the user, he will have the option to select to take the extra service of adding extra 10kg suitcases.
   5.1 **int** amountExtraTenSuitcase **(x>0, x<=3)**
   -If the user selects YES to select the service, he will have to indicate how many extra suitcases he want, it must be a valid value between 1 to 3.

6. **String** ynChooseExtraTwentyTSuitcase **(Yes/No)**
   -No matter the user, he will have the option to select to take the extra service of adding extra 23kg suitcases.
   6.1 **int** amountExtraTwentyTSuitcase **(x>0, x<=2)**
   -If the user selects YES to select the service, he will have to indicate how many extra suitcases he want, it must be a valid value between 1 to 3.

Output:

1. **double** totalPassengerToPay
   -This variable will be the individual total for a passenger, it will change depending if he is flying alone or accompanied.

2. **double** travelAccompaniedToPay
   -This variable will be the sum of passenger tickets if there's more than one passenger traveling.

3. **double[]** lastTenPrices
   -This array will contain the latest 10 flight prices and will be shown selecting option 3 of the main menu.

Example:
The user is asked to select an option from the main menu [1, 2 ,3 ,4]
*- -The user chooses the option **1***
The program asks for an approximate weight in kg
*- -The user says **9***
The program asks for the flight rate to choose
*- -The user choose the **S** rate*
The program asks to the user if he wants to select the seat for his flight
*- -The user says **yes***
The program show to user 3 seat positions to choose
*- -The user choose **window** as seat*
The program asks to the user if he wants to add extra 10kg suitcases
*- -The user says **no***
The program asks to the user if he wants to add extra 23kg suitcases
*- -The user says **yes***
The program asks to the user how many 23kg suitcases he need
*- -The user says **1***
The program says to the user that the total value of the ticket will be: $*383750*

Contract methods

Method 1: Welcome | METHOD int
Description: This method will welcome the user and it will ask him for a menu selection.
@return menuSelectOption

Method 2.1: Rate_XS_Includes | METHOD void
Description: This method will just display information about what the XS rate brings.

Method 2.2: Rate_S_Includes | METHOD void
Description: This method will just display information about what the S rate brings.

Method 2.3: Rate_M_Includes | METHOD void
Description: This method will just display information about what the M rate brings.

Method 2.4: Rate_Recommendation | METHOD void
Description: This method will manage the three possible rates to choose and will display the best one to choose.

Method 3: Choose-Flight-Rate | METHOD String
Description: This method will show to the user a suggested rate depending on the imputed weight.
@return chooseRate

Method 4.1: Seat-Selection-XS-S | METHOD Double
Description: This method will display the option to select a seat for the XS and S rates, returning the sum value to the check.
@param **double** totalPassengerToPay
@return totalPassengerToPay

Method 4.2: Seat-Selection-M | METHOD Double
Description: This method will allow the rate M user to select their seat, returning the sum value to the check.
@param **double** totalPassengerToPay
@return totalPassengerToPay

Method 5: Extra-10KG-Checked-Suitcase | METHOD Double
Description: This method will allow the user to add extra 10kg suitcases, returning the sum value to the check.

@param **double** totalPassengerToPay
@return totalPassengerToPay

Method 6: Extra-23KG-Checked-Suitcase | METHOD Double
Description: This method will allow the user to add extra 23kg suitcases, returning the sum value to the check.
@param **double** totalPassengerToPay
@return totalPassengerToPay

Method 7: Rate-Selection-Menu | METHOD Double
Description: This method will allow you to call a menu for selecting the rate flight, allowing to call it repetitive times when the user is traveling accompanied.
@param **String** chooseRate
@param **double** totalPassengerToPay
@return totalPassengerToPay

Method 8.1: Set_Array_Prices_Accompanied | METHOD int
Description: This method will check if i is equal to 1 or higher, if it is so, it will assign to that array index the corresponding value.
@param **int** i
@param **int** z
@param **double** totalPassengerToPay
@param **double[]** lastTenPrices
@return z

Method 8.2: Set_Array_Prices_Alone | METHOD int
Description: This method will assign to the array index the corresponding value and will sum 1 to z array index counter.
@param **int** z
@param **double** totalPassengerToPay
@param **double[]** lastTenPrices
@return z

Method 8.3: Display_Accompanied_Total_Amount_To_Pay | METHOD double
Description: This method will display the total amount to pay for the accompanied users and also will set the variable to 0.
@param **int** i
@param **double** travelAccompaniedToPay
@return travelAccompaniedToPay

Method 9: Welcome_Accompanied_User | METHOD void
Description: This method will display a message to welcome the users when selecting option 3 in the main menu.
@param **int** i

Method 10: Array_Prices_Printer | METHOD void
Description: This method will display the array with the prices by a FOR cycle and will indicate the highest price between them.
@param **int** k
@param **double** maxPrice
@param **double[]** lastTenPrices