

Calculadora de costo de viaje y vuelos

Análisis

Descripción: El programa funcionará como una calculadora que le debe permitir a Miguel (un viajero frecuente) poder explorar opciones para sus viajes en solitario o con amigos, y presupuestar el costo total de cada viaje, así mismo llevar una visión general de los últimos 10 viajes y el precio mayor calculador hasta el momento.

Entradas:

- **mainMenu** [int]: Receive the option chosen by the user from the menu.
- **travelerCompanionship** [int]: Receive the information if the user is traveling alone or accompanied.
- **luggageWeight** [Double]: Receive the approximate weight that the traveler will carry in kilograms.
- **chosenFare** [String]: Receive the fare desired by the traveler.
- **desiredSeat** [String]: Receive the desired seat choice from the person.
- **addService** [int]: Receive information on whether the user desires an additional service or not.
- **addLuggage10kg** [int]: Receive information if the user wishes to add extra luggage of 10 kg or not.
- **luggage10kg** [Int]: Receive information on how many additional 10 kg suitcases the user wants to add.
- **addLuggage23kg** [int]: Receive information if the user wishes to add extra luggage of 23 kg or not.
- **luggage23kg** [Int]: Receive information on how many additional 23 kg suitcases the user wants to add.
- **chooseAddSeat** [int]: Receive information from travelers with XS and S fares who wish to choose a seat prior to check-in, with an additional cost of \$50,000 COP.

Salidas:

- **array10lastPrices** [10] [Double]: This array stores the last 10 total travel prices. Each element is of type Double.
- **totalGroup** [Double]: This variable stores the total value of the entire group. It is of type Double.
- **total** [Double]: This variable stores the cumulative total of some concept or cost. It is of type Double.
- **maxPrice** [Double]: This variable stores the maximum price recorded so far. It is of type Double.

- **chosenFare** [String]: This variable stores the fare selected by the user. It is of type String.
- **ticketPrice** [Double]: This variable stores the value of the flight ticket. It is of type Double.
- **addServicesPrice** [Double]: This variable stores the total cost of additional services, such as seat selection. It is of type Double.

Ejemplos:

1) Al usuario se le muestran dos opciones de modo de viaje (solitario o con amigos)
- *-El usuario elige la opción solitario*

El programa le pregunta al usuario por el peso aproximado de su equipaje en kg
- *-El usuario dice que es 2*

El programa le sugiere la(s) tarifa(s) XS, S y M que van acorde al peso y le pregunta al usuario cuál tarifa desea escoger
- *-El usuario dice que XS*

El programa le muestra los servicios adicionales y pregunta si desea alguno
- *-El usuario dice que es si*

El programa le pregunta al usuario si desea adicionar una maleta de 10kg
- *-El usuario dice que es no*

El programa le pregunta al usuario si desea adicionar una maleta de 23kg
- *-El usuario dice que es no*

El programa le dice que al ser XS puede elegir asiento con un coste adicional y pregunta si desea tomar la opción
- *-El usuario dice que es si*

El programa le pregunta qué asiento desea elegir para su vuelo (pasillo, ventana o ninguno)
- *-El usuario dice que es ventana*

El programa le dice al usuario el total será de de \$225.000 COP

Contratos de los métodos

Método 1: menu

Descripción: Muestra tres opciones al usuario para que elija el tipo de daño que desea calcular

@param

@return int **mainMenu**, será el número de la opción elegida

Método 2: suggest Fares

Descripción: Permite calcular las tarifas sugeridas según los requerimientos de peso del equipaje.

@param double **luggageWeight**, dato para calcular las tarifas que se ajustan al equipaje del viajero.

@return string **chosenFare**, dato que representa la tarifa escogida por el usuario.

Método 3: fare Price

Descripción: Permite calcular el precio de la tarifa teniendo en cuenta porcentajes de comodidad y el precio de la tarifa anterior a la elegida

@param string **chosenFare**, dato de otro método necesario para calcular el precio de la tarifa anterior

@return double **ticketPrice**, el precio final de la tarifa resultado de la implementación de la fórmula del precio base.

Método 4: additional Services

Descripción: Permite calcular el valor total de servicios adicionales escogidos por el viajero

@param int **addService**, dato que indica si el usuario desea o no algún servicio extra

@param String **chosenFare**, dato que indica que servicios adicionales no ofrecer de acuerdo a la tarifa.

@return double **addServicesPrice**, el precio final de la tarifa resultado de la implementación de la fórmula del precio base.

Método 5: total Fare (for Each traveller)

Descripción: Permite calcular el precio total del vuelo por persona

@param double **ticketPrice**, dato de otro método para calcular el valor por servicios adicionales

@param double **addServicesPrice**, dato de otro método para calcular el valor por servicios adicionales

@return total, es la suma de los parámetros ingresados que representa el valor total del tiquete por persona.

Hecho por: Maria Juliana Marin Shek – A00405603