AeroDescuentos

Se está desarrollando para una aerolínea su módulo de liquidación de tiquetes aéreos. Para el mismo, se tiene una función que aplica descuentos a la tarifa base del vuelo dependiendo del tiempo de antelación de la reserva y la edad del pasajero. Los descuentos SON ACUMULABLES.

*Normativa 005, sobre los descuentos:*

* ***15%*** *de descuento sacando el billete con antelación superior a 20 días.*
* ***5%*** *a los pasajeros con edad inferior a 18 años y* ***8%*** *a los pasajeros con edad superior a 65 años.*

La siguiente es la especificación de la función que se usará en el módulo del cálculo de los descuentos:

/\*\*

calcular la tarifa de cada billete según el trayecto, la antelación en la que se obtiene el billete y la edad del pasajero, de acuerdo con la normativa 005.

@param tarifaBase valor base del vuelo

@param diasAntelacion dias de antelación del vuelo @param edad - edad del pasajero

@throws ExcepcionParametrosInvalidos [XXXXXXXXXXXXXXXXXXXXXXXXXXX]

\*\*/

public long calculoTarifa(long tarifaBase, int diasAntelacion, int edad)

1. De acuerdo con lo indicado, y teniendo en cuenta que NO hay precondiciones, en qué casos se debería arrojar una excepción de tipo ExcepcionParametrosInvalidos?. Agregue esto a la especificación.
   * Debería mandar error si:
     + La tarifa base, días de antelación o la edad del pasajero son negativos.
     + Si la edad del pasajero es mayor a 102 años.
2. En la siguiente tabla enumere un conjunto de clases de equivalencia que -según usted- creen una buena división del conjunto de datos de entrada de la función anterior:

|  |  |  |
| --- | --- | --- |
| Número | Clase de equivalencia (en lenguaje natural o matemático). | Resultado correcto /  incorrecto. |
| 1 | Tarifa base negativa | Incorrecto |
| 2 | Tarifa base positiva > 0 | Correcto |
| 3 | Días de antelación negativos | Incorrecto |
| 4 | Días de antelación > 0 | Correcto |
| 5 | Edad positiva > 0 y < 102 | Correcto |
| 6 | Edad <= 0 | Incorrecto |
| 7 | Edad > 102 años | Incorrecto |

1. Para cada clase de equivalencia, defina un caso de prueba específico, definiendo: parámetros de entrada y resultados esperados.
2. calculoTarifa(long tarifaBase, int diasAntelacion, int edad)

|  |  |  |
| --- | --- | --- |
| Número | Ejemplos de las clases de equivalencia. | Resultado: correcto / incorrecto. |
| 1 | Tarifa base negativa  calculoTarifa( -1000000, 10, 21) | Incorrecto |
| 2 | Tarifa base positiva > 0  calculoTarifa(1000000, 10, 21) | Correcto |
| 3 | Días de antelación negativos  calculoTarifa(1000000, -20, 21) | Incorrecto |
| 4 | Días de antelación > 0  calculoTarifa(1000000, 10, 21) | Correcto |
| 5 | Edad positiva > 0 y < 102  calculoTarifa(1000000, 10, 21) | Correcto |
| 6 | Edad <= 0  calculoTarifa(1000000, 10, -23) | Incorrecto |
| 7 | Edad > 102 años  calculoTarifa(1000000, 10, 104) | Incorrecto |

1. A partir de las clases de equivalencia identificadas en el punto 2, identifique las condiciones límite o de frontera de las mismas.

Frontera días antelación: [1, 365]

Frontera tarifa base: [100000, 25000000]

Frontera edad: [1, 102]

1. Para cada una de las condiciones de frontera anteriores, defina casos de prueba específicos.

|  |  |  |
| --- | --- | --- |
| Número | Ejemplos de las clases de equivalencia. | Resultado: correcto / incorrecto. |
| 1 | calculoTarifa( -2000000, -10, 21) | Incorrecto |
| 2 | calculoTarifa(1000000, 10, 1) | Correcto |
| 3 | calculoTarifa(5000000, -20, -25) | Incorrecto |
| 4 | calculoTarifa(1000000, 45, -21) | Incorrecto |
| 5 | calculoTarifa(1000000, 366, 21) | Incorrecto |
| 6 | calculoTarifa(1250000, 367, -23) | Incorrecto |
| 7 | calculoTarifa(1500000, 10, 104) | Incorrecto |