



Tarea 1. Valor 1%

Temas de Estudio

1. Tema 1 - Java y la programación orientada a objetos
2. Tema 2 - Tema 2 - Programación orientada a objetos: Métodos Subtemas

Objetivo

Desarrollar una aplicación en java que permita demostrar el conocimiento adquirido sobre la definición de clases y su utilización en un programa ejecutable con iteración mediante la unidad de interface gráfica.

Software de Desarrollo

NetBeans

Desarrollo

La empresa bancaria, "CRUnion" desea contratarlo para programar una aplicación que permita calcular el cambio de moneda en aeropuertos para turistas que ingresan al país. La regla principal es que cualquier conversión de moneda debe ser realizada a dólares Estado Unidoses y luego a la moneda destino según la siguiente tabla de conversión:

	Precio compra dólares por parte del CRUnion	Precio venta dólares por parte de CRUnion
1 dólar estadounidense	0.85 euros	0.91 euros
1 dólar estadounidense	660 colones	670 colones
1 dólar estadounidense	20 pesos mexicanos	20.5 pesos mexicanos
1 dólar estadounidense	33 córdobas oro	36 córdobas oro
1 dólar estadounidense	110 pesos argentinos	112.5 pesos argentinos
1 dólar estadounidense	0.70 libra esterlina	0.76 libra esterlina
1 dólar estadounidense	119 yen	122.5 yen

De esta forma, una persona no solo puede traer dólares al ingresar al país, si no que puede traer monedas de los otros países incluidos en la tabla anterior.

De esta forma, una persona puede venir con 100 dólares Estado Unidenses y desea cambiarlos a colones costarricenses, por lo que el banco le estará entregando un total de 660 colones, por qué el CRUnion realiza la compra de dólares a 660 colones.

Otro ejemplo, es que el turista venga con euros, por ejemplo 150 euros y desea convertirlos a Córdobas, en este caso, el banco primero le vendería dólares (para pasar los euros a dólares) y con el resultado de esta conversión compraría los dólares al tipo de cambio respectivo para cambiarlos a córdobas. De esta forma 150 euros corresponden a 164.83 dólares estadounidenses (venta de dólares que corresponde a $150/0.91$) y finalmente entregaría al turista 5439.50 córdobas (producto de la compra de dólares que corresponde a $164.83 * 33$).

Para ello se debe desarrollar tres diferentes clases dentro del proyecto:

1. Una clase **equivalencias** que maneje los precios del dólar (compra y venta de dólares).
2. Una segunda clase llamada **conversion** con las reglas del definidas para convertir de una moneda origen a otra moneda destino.
3. La tercera clase llamada **CRUnion** (clase principal del proyecto), que es la encargada de desarrollar la interface gráfica que permite la iteración del usuario.

La clase equivalencias

Como se mencionó, esta clase maneja los precios de compra y venta de dólares, pero no es la clase encargada de realizar el cálculo de una conversión. Su objetivo es solo poder administrar los precios bases del dólar hacia las diferentes monedas con que se está manejando en el presente proyecto.

La clase “equivalencias” debe contener como mínimo los siguientes atributos:

1. `Descripciones_monedas`: corresponde a un arreglo de hileras de caracteres con la descripción del nombre de la moneda. Este atributo debe ser declarado como atributo private y su dimensión debe ser definida o inicializada en el constructor. Así mismo en el método constructor deben ser llenada lo siguientes valores que fueron tomados de la tabla mostrada anteriormente en la descripción del proyecto.

Dólar Estado Unidense
Euros
Colones
Pesos Mexicanos
Córdobas Oro

Pesos Argentinos
Libras Esterlina
Yen

2. `Tabla_conversiones`: corresponde a un arreglo bidimensional de valores flotantes con los valores de las respectivas conversiones. Este atributo debe ser declarado como atributo `private` y sus dimensiones deben ser definidas o inicializadas en el constructor. Así mismo en el método constructor deben ser llenado los campos de la tabla con los siguientes valores que fueron tomados de la tabla mostrada anteriormente en la descripción del proyecto (nótese que la fila del arreglo `Descripciones_monedas` corresponde a la fila de la presente `tabla_conversiones`)

0.85	0.91
660	670
20	20.5
33	36
110	112.5
0.70	0.76
119	122.5

Además de estos atributos, usted puede definir aquellos atributos que considere necesario para desarrollar su solución.

La clase `equivalencias` debe contener como mínimos las siguientes funciones:

1. `PrecioCompra(moneda_origen)`: recibe el tipo de moneda origen (por ejemplo Euros) y devuelve el valor del precio de compra según la tabla `Tabla_conversiones`
2. `PrecioVenta(moneda_origen)`: recibe el tipo de moneda origen (por ejemplo Euros) y devuelve el valor del precio de venta según la tabla `Tabla_conversiones`
3. `PosicionMoneda(moneda_origen)`: recibe el tipo de moneda origen y devuelve la fila en la que se encuentra la misma, esto para poder conocer la fila donde se encuentra el valor de compra y venta de la moneda en la tabla `Tabla_conversiones`.

Se debe crear dos métodos constructor

- Constructor sin parámetros: inicia los atributos con los valores básicos indicados anteriormente
- Constructor con parámetros: recibe por parámetro la cantidad de países con los que CRUnion desea trabajar. Como mínimo el valor debe ser 7 para cubrir los países incluidos en la descripción del proyecto; y un máximo de 25, donde los espacios del 8º al 25º fila debe quedar en blanco para ser usado en un futuro proyecto.

Para ambos constructores, usted debe analizar si los otros atributos que usted defina requieren un valor inicial por defecto o calculado.

La clase conversion

Como se mencionó, esta clase maneja las reglas definidas en el proyecto para convertir dinero de una moneda origen a una moneda destino. Su objetivo es solo poder realizar cálculos, donde no interviene iteración alguna con el usuario. Es decir, todos sus cálculos se realizan con base en los valores que las funciones reciben mediante parámetros.

La clase “equivalencias” debe contener como mínimo los siguientes atributos:

1. equivalencia: objeto del tipo **Equivalencias** que permite consumir las funciones definidas en la clase y con ello poder obtener los valores base de conversión del dólar.

Además de estos atributos, usted puede definir aquellos atributos que considere necesario para desarrollar su solución.

La clase **conversión** debe contener como mínimos las siguientes funciones:

1. CompraDolares: función que recibe como parámetros moneda_destino y el monto de dólares a comprar, devuelve el monto de dinero a entregar al cliente.
2. VentaDolares: función que recibe como parámetros moneda_origen y el monto de dinero a recibir, devuelve la cantidad de dólares que el banco va a entregar al cliente
3. Conversion_moneda: función que recibe como parámetros la moneda_origen, la moneda_destino y el monto de dinero inicial. Su objetivo es tomar el monto de dinero que viene en la moneda origen y devolver el monto correspondiente en la moneda destino.

La clase principal o CRUnion

Como se mencionó, esta clase es la encargada de desarrollar la interface gráfica que permite la iteración del usuario. En esta clase se debe declarar un objeto de tipo conversión y según valores expuestos por el usuario en la interface gráfica (desarrollada con elementos de la GUI) se debe desplegar el valor de la conversión.

Esta clase es la clase donde se encuentra el método main y se ejecuta la aplicación como tal.

La ejecución de la aplicación no debe estar limitada a solo 1 conversión, si no que el usuario pueda usar esta aplicación como un tipo de calculadora donde pueda realizar la cantidad de conversiones necesarias.

Al finalizar, el programa debe mostrar la cantidad de dólares comprados y vendidos durante la ejecución del proyecto.

Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra < tarea1>.
- La <Tarea1> debe estar desarrollado en **NetBeans IDE - Java** que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-tarea1.**
- La entrega de la <Tarea1> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.

- Si no concluyó a tiempo la tarea, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

Rúbrica de Evaluación

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato pero los aportes no son significantes	No cumple o no presenta lo solicitado
Presentación del código: Correcto uso de variables, nombres de métodos significativos, no presenta métodos vacíos	5	3	2	0
Comentarios en código: No comete errores gramaticales u ortográficos dentro de los comentarios del código y son significativos para cada método y atributo	5	3	2	0
Correcta creación de clase "equivalencias"	10	7	4	0
Correcta creación de clase "Conversión"	10	7	4	0
Correcta creación de clase principal (Debe utilizar por lo menos 3 diferentes elementos de la GUI)	15	10	5	0
Desarrollo de los diferentes métodos solicitados	15	10	5	0
Correcta iteración con el usuario	10	7	4	0
Permite tener dos o más conversiones en una misma ejecución	10	7	4	0
Al finalizar muestra la cantidad total de dólares comprados y vendidos en todas las conversiones	10	7	4	0
Correcto uso del programa para que no se interrumpa la ejecución de este (Manejo de excepciones)	10	7	4	0
TOTAL	100			

