**TEMA DESARROLLO 1**

**Observe el código a continuación**

|  |
| --- |
| class Receta {  private String nombre;  private String dispositivo;  public Receta(String nombre, String dispositivo) {  this.nombre = nombre;  this.dispositivo = dispositivo;  }  public String getNombre(){  return nombre;  }  public String getDispotivo(){  return dispositivo;  }  }    class DispositivoInvalidoException \_\_\_\_\_\_\_\_\_{  //completar  }  public class RecetaServicio {  public void cocinar(Receta receta, int tiempo, String dispositivo) \_\_\_\_\_\_\_\_\_\_\_ {  //completar  }    public static void main(String[] args){  RecetaServicio recetaservicio = new RecetaServicio();  List<Receta> recetas = new ArrayList<>();  recetas.add(new Receta("crema batida","batidora"));  recetas.add(new Receta("huevo frito","sarten"));  Scanner sc = new Scanner(System.in);  for(Receta r : recetas){  System.out.println("Receta:"+r.getNombre());  System.out.println("Ingrese tiempo de coccion en segundos");//en segundos  int tiempo = sc.nextInt();  sc.nextLine();  System.out.println("Ingrese dispositivo");  String dispositvo = sc.nextLine();    //completar  }  }  } |

1. Implemente la excepción **DispositivoInvalidoException** para que sea una excepción **tipo** **checked** y tenga un constructor que reciba el mensaje de la excepción como parámetro **(5 puntos)**
2. En la clase **RecetaServicio** complete el método **cocinar** para que haga lo siguiente
   1. Si el dispositivo enviado como parámetro **no es igual al requerido** lanza una excepción de tipo **DispositivoInvalidoException** con el mensaje - “Por favor, use el dispositivo correcto para la receta”. (**5 puntos**)
   2. A continuación, cree y ejecute un Thread que realice lo siguiente **(15 puntos)**
      1. Cree un lazo que imprima en pantalla “Cocinando **NombreReceta**” (por ejemplo: Cocinando huevo frito), cada 5 segundos por el tiempo recibido como parámetro.
3. **(5 puntos)** En la clase **RecetaServicio** complete el método **main** para que, por cada receta en la lista, llame al método **cocinar** usando los parámetros ingresados por el usuario. Debe manejar las excepciones que el método **cocinar** puede lanzar mostrando el mensaje de la excepción y el tipo de error producido.

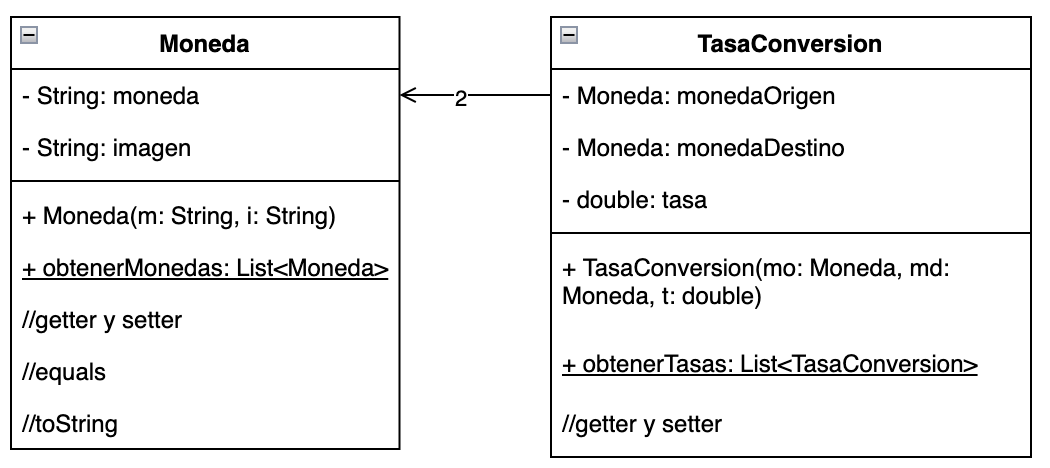
Ejemplo de salida de ejecutar el método main

|  |
| --- |
| Receta:crema batida  **Ingrese tiempo de coccion en segundos**  **20**  Ingrese dispositivo  batidora  Receta:huevo frito  **Ingrese tiempo de cocción en segundos**  Cocinando crema batida  Cocinando crema batida  **15**  Ingrese dispositivo  sarten  Cocinando huevo frito  Cocinando crema batida  Cocinando huevo frito  Cocinando crema batida  Cocinando huevo frito |

|  |
| --- |
| Receta:crema batida  Ingrese tiempo de cocción en segundos  20  Ingrese dispositivo  batidora  Receta:huevo frito  Ingrese tiempo de cocción en segundos  Cocinando crema batida  15  Ingrese dispositivo  Cocinando crema batida  olla  class temaDesarrollo1.DispositivoInvalidoException  Por favor use el dispositivo correcto para la receta  Cocinando crema batida  Cocinando crema batida |

**Tema Desarrollo 2:**

**Se necesita implementar un Conversor de monedas que permita saber dado x cantidad de dinero en un moneda (por ejemplo dólar) cuál es su equivalente en otra moneda (por ejemplo euro). Para esto a usted se le da un proyecto base. El diagrama de clases del proyecto es el siguiente:**



**Realice lo siguiente:**

**En la clase Moneda:**

**(5 puntos)** Implemente la interfaz Comparable para que el orden natural sea por el nombre de la moneda.

**En la clase TasaConversion:**

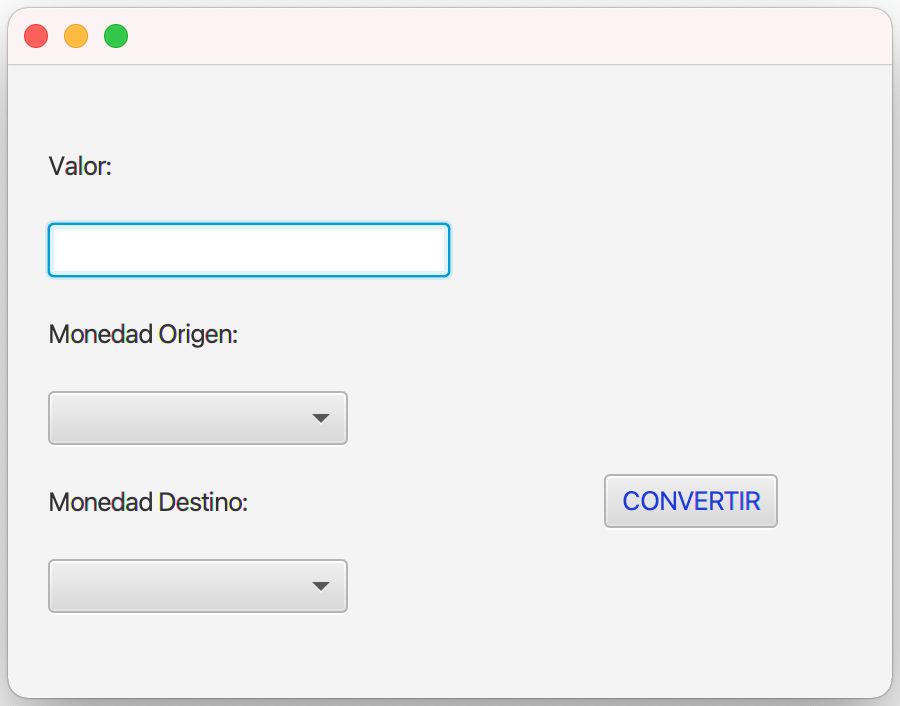
**(10 puntos)** Usted deberá implementar el método estático **obtenerTasas** que procede a leer el archivo serializado **tasas.dat** que tiene un objeto de tipo ArrayList<TasaConversion>. El método debe retornar la lista que extrae del archivo serializado. Considere implementar todo lo necesario para que pueda deserializar correctamente desde esta clase.

NOTA: Una vez haya hecho los cambios requeridos en la clase e implementado el método corra la clase **TestTasasConversion** para generar el archivo serializado

**(41 puntos) Interfaz Gráfica**

Al iniciar la aplicación debe aparecer una ventana como la que se ve en la Imagen1.

Edite el archivo primary.fxml y la clase PrimaryController.java para crear la vista. En la ventana debe llenar los valores de los combobox con las distintas monedas que existen para que cuando se cargue la ventana se pueda seleccionar de estos las monedas. Las monedas existentes las podrá obtener del método estático **obtenerMonedas** que está en la clase Moneda.

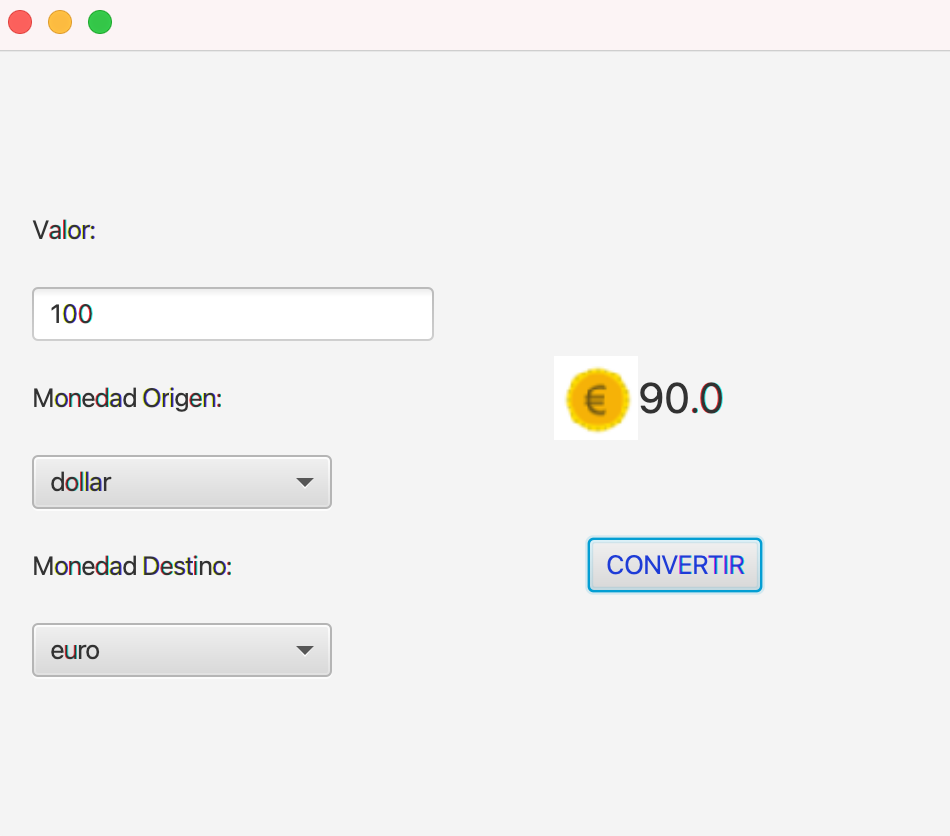


**Imagen1.- Imagen que aparece al iniciar la aplicación**

La acción sobre el botón **Convertir** deberá mostrar el resultado de la operación de conversión del **valor** ingresado por el usuario de la **Moneda Origen** a la **Moneda destino** junto con la imagen de la moneda destino tal como se muestra en la Imagen 2.

Para calcular la conversión utilice la fórmula:

resultado = valorIngresado \* tasaConsersión (de la Moneda Origen a la Moneda Destino)

****

**Imagen2-** El valor resultante de convertir **100 dólares a euros** es 90 euros.

Asegúrese que el usuario ha ingresado un valor en la caja de texto y seleccionado un elemento de cada combobox, caso contrario presente una alerta al usuario solicitando completar la información.

Rúbrica

1. Crear el fxml de la vista (12 puntos)
   1. controles correctos - 4
   2. contenedores correctos - 3
   3. colocar los id a los controles (el textfiled y los combobox) - 3 puntos
   4. colocar el manejar el boton a la accion de dar click - 2 puntos
2. Controlador (29 puntos)
   1. en el initiliaze (6 puntos)
      1. obtener la lista de las monedas - 2 puntos
      2. cargar los combobox - 4 puntos
   2. manejador del evento del botón convertir (23 puntos)
      1. obtener los valores de los combos y el textfield - 3 puntos
      2. si el textfield está vacío lanzar una alerta - 5 puntos
      3. obtener la lista de TasasConversion llamando al método correcto - 2 puntos
      4. Recorrer la lista de tasas y obtener las tasa que corresponda usando las moneda de origen y destino - 4 puntos
         1. realizar la conversion - 3 puntos
         2. colocar el textfield con la conversion - 3 puntos
         3. colocarImagen - 3 puntos