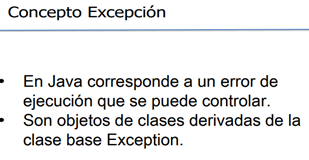
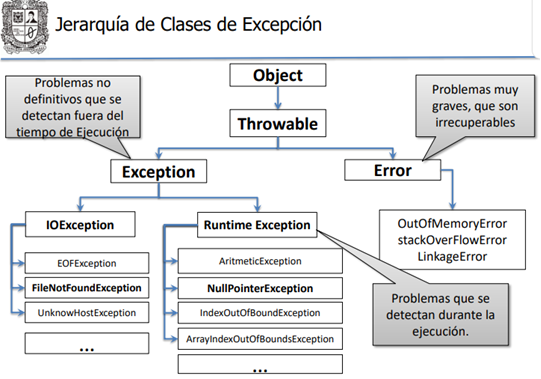
**Multi302 Abril 24 Manejo de excepciones con Try-Catch**

1. Concepto Excepción
2. Usar un controlador desde otro Controlador
3. Uso de Try- Catch
4. Ejercicio de aplicación

**Desarrollo**

1. Concepto Excepción

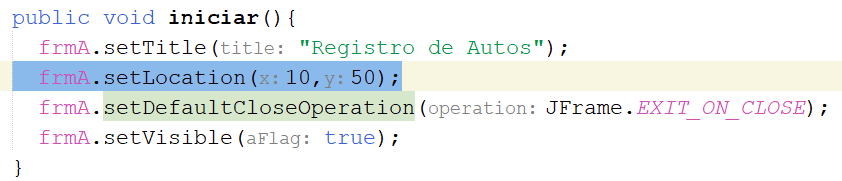
1. Usar un controlador desde otro Controlador

Para el ejercicio proyVehiculos, se están trabajando dos tipos de **JFrames**, un MDI y un JInternal, aunque se puede controlar todo desde el MDI, el método **actionPerformed**() quedaría muy cargado, por lo tanto, se sugiere hacer uso de los controladores por separado, para ello se realizarán los siguientes procesos:

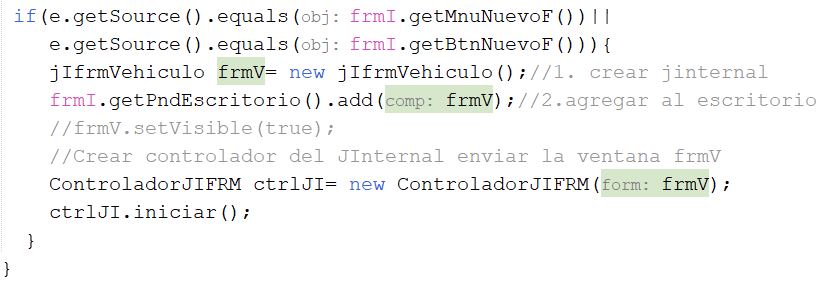
1. Duplicar el **ControladorFRM** para crear el controlador del JinternalFrame de Registro de Vehículo y crear un nuevo constructor para que reciba un JinternalFrame



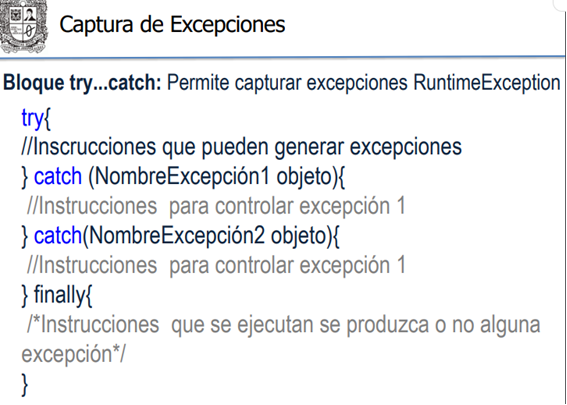
1. Verificar que el JInternalFrame ya tenga los getters and setters
2. Agregar buttonGroup btngTipoV y generar getters and setter (para controlar los radioButton)
3. Modificar el método para ubicar el formulario en la ventana en el método iniciar()



1. En ControladorMDI, dentro del método actionPerformed() modificar el condicional que abre el registro de formulario desde la opción de menú mnuBuevoF o desde el botón de la barra de herramientas btnNuevoF. Se crea la nueva ventana frmA pero en lugar de hacerla visible se envía al controladorJIFrm y se ejecuta el controlador mediante el método iniciar(), tal como sigue:



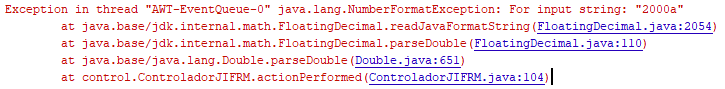
1. **Uso de Try-Catch**



1. **Ejercicio de aplicación**

El objetivo es controlar una excepción de tipo **NumberFormatException**

Al ingresar un valor alfabético en alguno de los campos numéricos (Campo Modelo y valor en ejercicio **proyVehiculos**) que hacen uso de los métodos de conversión .**parseDouble**() o **parseInteger**(), activan la excepción **NumberFormatException**, generando la excepción como aparece a continuación e impidiendo la ejecución normal del programa.

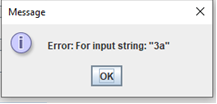


Para controlar estas excepciones se utiliza el bloque **try-catch**, que deberá contener las instrucciones que pueden activar la excepción, en el botón **Agregar Vehiculo** donde se envían los datos de las cajas de texto al objeto Vehiculo, este uso se hace dentro del método **actionPerformed**() del formulario JInternal de registro de vehiculo **JIFrmVehiculo**, en el condicional del botón Agregar Vehículo **BtnAgregarV**, como aparece a continuación.

En la cláusula **Catch** se pueden utilizar los diferentes métodos para visualizar la descripción del error generado y de esta forma se puede controlar el error en el momento de la ejecución con una caja de mensaje que le indique al usuario cual es el dato que genera el error.

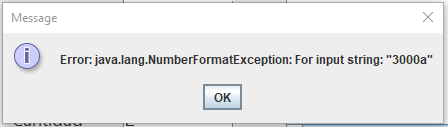
* **Mensaje con método getMessage() del error:** genera la información de la excepción indicando que la genera y el valor asociado





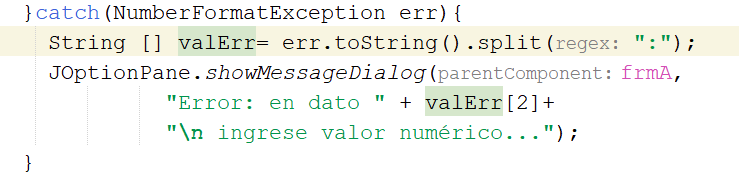
* **Mensaje con método toString() del error:** genera toda la información, nombre de la excepción, que la genera y el valor asociado





* **Mensaje Personalizado**

**Usando split del getMessage().** Para personalizar el mensaje de salida se puede usar el siguiente código:



**Próxima Clase**

- Traer avance ejercicio Hospedaje/Alimentación, aplicando los conceptos MDI y excepciones trabajadas hasta la fecha.

- Traer presentación ejercicio de control asignado en clase