**Excepciones:**

**ESPECIFICAS**

Ej) ValidarNumero

Syso ("Numero")

5 int n=Entrada.entero()

Syso=("leido")

->Esto da ERROR, ¿porque? En el starTrade dice que el error está en la línea 5 en el Pareint que se llama: java.lenght.Number.format

\*Si no se hace nada con las excepciones se propagan del interior al exterior.

1) Debemos meter la excepción en un TRY que será como un contenedor.

Es como decir: Si se produce un error dentro de estas líneas del try hacer esto (CATCH)

Catch se crea con el nombre de la excepción.

ej) try{

n=Entrada.entero()

}

catch (Number.format **e**){ <- e= identificador

Syso ("numero incorrecto") <- mensaje de error

}

Syso ("numero leído"+n

)

2) Sigue sin funcionar porque sale:

numero incorrecto

numero = 0 <- eso no debería salir

3) Debemos ordenar la clase

int n=0

syso ("numero")

try{

n=Entrada.entero()

syso ("numero leido"+n)

}

catch (Number.format **e**){

Syso ("numero incorrecto")

**Syso(e.getMessage())**

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TIPOS DE EXCEPCIONES:**

-**Comprobada** (CHEKED) son de la clase Exception.

Obligan al programador a hacer algo. Es decir, cuando te llega un excepción puede manejarla (try/catch), enviársela a otro (trow)...

-**No Comprobada** (UNCHECKED) de la clase RunTimeException

No obligan al programador a hacer algo. Están ahí pero no hace falta hacer algo contra ella.

Las excepciones de java suelen ser no comprobadas

\*trows Exception (obliga a excepcion)

Ejemplos)

**\*\*Surround with**, seleccionando una frase, botón derecho y podemos poner condiciones, try/cath, bucle... solos