

Programación

UT3 - TAREA11

En Java, crear **excepciones personalizadas** es útil cuando necesitas manejar errores específicos en tu aplicación que no están cubiertos por las excepciones estándar. Para crear una excepción personalizada, simplemente necesitas extender (heredar de) la clase **Exception** (para excepciones comprobadas) o **RuntimeException** (para excepciones no comprobadas).

Para crear una excepción personalizada en este ejemplo, primero definiremos una nueva clase que herede de Exception. Esto nos permite personalizar la excepción y darle un nombre específico que describa mejor el error. Aquí tienes cómo hacerlo:

1. Definición de la Excepción Personalizada

Creamos una clase llamada **EdadNoPermitidaException** que hereda de Exception:

```
public class EdadNoPermitidaException extends Exception {  
    public EdadNoPermitidaException(String mensaje) {  
        super(mensaje);  
    }  
}
```

2. Uso de la Excepción Personalizada

Ahora usaremos esta excepción personalizada en el método verificarEdad.

```
public class Main {  
    public static void main(String[] args) {  
  
        int edad = 16;  
  
        try { verificarEdad(edad); }  
        catch (EdadNoPermitidaException e) {  
            System.out.println("Error: " + e.getMessage());  
        }  
    }  
  
    public static void verificarEdad(int edad) throws  
    EdadNoPermitidaException {  
        if (edad < 18) {  
            throw new EdadNoPermitidaException("No se permite el acceso a  
menores de 18 años.");  
        }  
    }  
}
```

```
        System.out.println("Acceso permitido."); }  
    }
```

Explicación

1. **Definición de la excepción personalizada:** Creamos `EdadNoPermitidaException`, que hereda de `Exception` y acepta un mensaje en su constructor.
2. **Lanzamiento de la excepción:** En el método `verificarEdad`, lanzamos la excepción `EdadNoPermitidaException` si la edad es menor de 18.
3. **Manejo de la excepción:** En el método `main`, usamos un bloque `try-catch` para capturar `EdadNoPermitidaException` y mostramos el mensaje de error.

Este enfoque es más claro y ayuda a especificar el tipo de error que queremos manejar. Utilizar una excepción personalizada hace que el código sea más comprensible y facilita el manejo de errores específicos.

IMPORTANTE: Adjunta a la tarea tienes los dos archivos Java correspondientes a las clases del ejemplo. Te pueden servir de ayuda si los usas como base para resolver lo que te propongo a continuación.

(10 puntos) Teniendo en mente el ejemplo que acabas de ver, vas a desarrollar una pequeña aplicación en Java que simula un sistema de compras en línea. En este sistema, no se permite realizar compras si el total de la compra es mayor de 1000 euros. Si alguien intenta realizar una compra superior a esta cantidad, el sistema debe lanzar una excepción personalizada que informe al usuario del límite de compra permitido.

Requisitos:

1. **Excepción Personalizada:**
 - o Crea una clase llamada `LimiteCompraExcedidoException` que herede de `Exception`.
 - o Esta excepción debe lanzarse cuando el total de la compra supera los 1000 euros.
2. **Método para Realizar Compras:**
 - o Crea un método llamado `realizarCompra` que reciba como parámetro el monto total de la compra (de tipo `double`).
 - o Dentro del método, verifica si el monto de la compra es mayor a 1000.
 - o Si el monto es mayor a 1000, lanza una excepción de tipo `LimiteCompraExcedidoException` con el mensaje: "Error: No se permite realizar compras superiores a 1000 euros."
 - o Si el monto está dentro del límite, muestra un mensaje indicando que la compra ha sido realizada exitosamente.

3. Bloque try-catch en el Main:

- o En el método main, pide al usuario el monto de la compra (puedes asignarlo directamente a una variable, si prefieres).
- o Llama al método realizarCompra dentro de un bloque try-catch.
- o Si se lanza una excepción, captura el mensaje y muéstralo en la consola.

Pruebas:

- **Caso 1:** Prueba con un monto de compra menor o igual a 1000. El sistema debería mostrar un mensaje indicando que la compra ha sido realizada con éxito.
- **Caso 2:** Prueba con un monto mayor a 1000. El sistema debería mostrar el mensaje de error de la excepción personalizada.

RESPONDE AQUÍ:

Pega a partir de aquí el código de la clase LimiteCompraExcedidoException:

```
public static class LimiteCompraExcedidoException extends Exception {  
    public LimiteCompraExcedidoException(String mensaje) {  
        super(mensaje);  
    }  
}
```

Pega a partir de aquí el código de tu clase principal (en mi caso la he llamado TiendaOnline):

```
public static void main(String[] args) {  
  
    double valor = 1000;  
  
    try {  
        realizarCompra(valor);  
    } catch (LimiteCompraExcedidoException e) {  
        System.out.println("Error: " + e.getMessage());  
    }  
}  
  
    public static void realizarCompra(double valor) throws  
LimiteCompraExcedidoException {  
    if (valor > 1000) {  
        throw new LimiteCompraExcedidoException("No se permite realizar compras  
superiores a 1000 euros.");  
    } else {  
        System.out.println("La compra ha sido realizada con éxito.");  
    }  
}
```

```
1 public class UT3tarea11 {
2
3     public static class LimiteCompraExcedidoException extends Exception {
4         public LimiteCompraExcedidoException(String mensaje) {
5             super(mensaje);
6         }
7     }
8
9     public static void main(String[] args) {
10
11         double valor = 1100;
12
13         try {
14             realizarCompra(valor);
15         } catch (LimiteCompraExcedidoException e) {
16             System.out.println("Error: " + e.getMessage());
17         }
18     }
19
20     public static void realizarCompra(double valor) throws LimiteCompraExcedidoException {
21         if (valor > 1000) {
22             throw new LimiteCompraExcedidoException("No se permite realizar compras superiores a 1000 euros.");
23         } else {
24             System.out.println("La compra ha sido realizada con éxito.");
25         }
26     }
27 }
28
```