Informe TP7

* En el primer enunciado, se pide realizar un programa que valide una contraseña con algunas reglas y excepciones, por ejemplo, si la contraseña no contiene nada (vacía o null) o si la contraseña tiene menos de 8 caracteres, para ello, se creó una excepción personalizada llamada PasswordException, que me devuelve una excepción si la contraseña tiene menos de 8 caracteres, luego se captura la excepción que me larga si lo ingresado es vacío o nulo.

Se podría haber puesto otra excepción si no contiene un número, pero ya cumple con la consigna

* En el segundo enunciado, se pide realizar un programa que valide las calificaciones de un alumno y utilizar excepciones para indicar errores, como si se excede del rango de 0 a 10 mostrar una excepción, para ello, se crea la excepción personalizada CalificationException, que compila mientras esté en ese rango, si no, se muestra un mensaje de error, también se captura una excepción de Java, InputmismatchException, que si se ingresa una letra o carácter especial, también larga un error, que solo debe ingresar números.

Podría haber puesto un while para poner varias veces las calificaciones hasta que no cumpla esa condición.

* En el tercer enunciado, se pide simular un sistema de reservas de cine, que tenga una Clase Cine y otra Asiento, y de largar una excepción si el asiento ingresado ya está ocupado, para ello se crea una excepción personalizada llamada AsientoOcupadoException, qué al ingresar otra vez el mismo número, ya larga la excepción pedida.

La clase Asiento lo único que hace es saber cuantos asientos hay y si están ocupados o no.

La clase Cine crea la cantidad ingresada por el usuario, fijándose si no hay una excepción primero, como la que si ingresa un valor menor a 0, le pide ingresar un numero positivo, y se termina con el System.exit para que no vaya a la parte del menú, si lo dejo correr, podría poner una excepción NullPointerException diciendo que no se pudo crear la sala debido al carácter o numero negativo ingresado, que salga del programa e ingrese un valor positivo para correrlo correctamente, pero queda mejor con el exit. Al haber ingresado ese valor positivo, se crea el cine con los asientos, y el usuario puede elegir tres opciones: Mostrar Asiento , Reservar Asiento, Salir (Las dos primeras corresponden con métodos de la Clase Cine).

Si ingresa 1, se muestra los asientos que fueron creados anteriormente, y muestran su estado, si están libres o ocupados, se lo hace con un ciclo for y se pregunta si está ocupado o libre dependiendo del estado del asiento, inicialmente no ocupado, entonces libre.

Si ingresa 2, el usuario puede elegir cual asiento se quiere sentar, al principio todos están libres, si ingresa cualquiera de esos creados, automáticamente se marca como ocupado, y lo puedo corroborar otra vez con la opción Mostrar Asiento, si ingresa de nuevo el asiento que ya esta ocupado, automáticamente larga la primera excepción ya dicha, si ingresa un numero fuera del rango creado, se muestra otra excepción personalizada llamada AsientoException con su mensaje de que el valor esta fuera del rango.

Si ingresa 3, termina el programa.

En el main tengo dos try para capturar las excepciones de Java y las personalizadas ya creadas y nombradas anteriormente, una va con la cantidad de asientos y el otra con el menú que se muestra, ya que primero se crea la sala con los asientos (fundamental, si no termina el programa), y en base a eso se crea las opciones para que el usuario elija y ocurran o no las excepciones.