

Tarea Final de la Unidad PROG04

Realizar un ejercicio en JAVA sobre la utilización de objetos.

Entrega: Deberás crear una carpeta comprimida en formato ‘zip’ con el nombre “Apellido_Nombre_Tarea_Final_PROG04”. Dentro de esta carpeta, incluye el proyecto o el paquete donde están las clases del ejercicio.

Finalmente, sube el archivo comprimido en la tarea correspondiente del aula virtual.

Enunciado del Problema

Título: Juego de Adivinanza con Sistema de Puntuación y Control de Intentos

Descripción: Crea un programa en Java que permita al usuario jugar a un juego de adivinanza de números. El programa generará un número aleatorio entre 1 y 100, y el usuario tendrá que adivinarlo. Además, el programa debe incluir las siguientes funcionalidades:

1. Límite de Intentos:

- El usuario dispondrá de un máximo de 10 intentos para adivinar el número.
- Si no lo adivina en esos intentos, el juego terminará indicando que ha perdido.

2. Sistema de Puntuación:

- El sistema calculará la puntuación según el número de intentos realizados.

Por ejemplo:

- Si adivina en el primer intento: 100 puntos.
- Si adivina en el segundo intento: 90 puntos.
- Y así sucesivamente, restando 10 puntos por cada intento adicional.

3. Control de Datos de Entrada:

- Utiliza estructuras try-catch para asegurarte de que el usuario introduce un número válido.

- Si el usuario introduce un dato no numérico o un número fuera del rango permitido (1-100), el programa deberá mostrar un mensaje de error y permitirle volver a intentarlo sin penalización.

4. Mensajes de Ayuda:

- Después de cada intento fallido, el programa indicará si el número introducido es mayor o menor que el número secreto.

5. Reinicio del Juego:

- Al terminar una partida (ganar o perder), el programa preguntará al usuario si desea jugar de nuevo. Si responde afirmativamente, se reiniciará el juego con un nuevo número aleatorio.

Clase JuegoAdivinanza

La clase principal que contiene la lógica del juego. Gestiona el flujo del juego y utiliza métodos auxiliares para manejar la puntuación y validar las entradas del usuario.

Atributos

1. numeroSecreto: Número aleatorio generado entre 1 y 100 que el usuario debe adivinar.
2. intentosMaximos: Número máximo de intentos permitidos (por defecto, 10).
3. intentosRealizados: Contador de los intentos realizados.
4. puntuacion: Puntuación obtenida por el usuario al final del juego.

Métodos

1. iniciarJuego()

Controla el flujo del juego:

- Genera el número secreto.
- Solicita entradas al usuario hasta que adivine el número o alcance el máximo de intentos.
- Proporciona mensajes sobre si el número introducido es mayor o menor que el número secreto.
- Al finalizar, calcula y muestra la puntuación o informa de la derrota.

2. calcularPuntuacion(int intentos)

Calcula la puntuación basada en el número de intentos:

- Si el usuario adivina en el primer intento, obtiene 100 puntos. Si agota los 10 intentos, la puntuación es 0. Según los intentos, calcular la puntuación.
3. validarEntrada(String entrada)
- Comprueba si la entrada es válida:
- Verifica que el número esté entre 1 y 100.
 - Lanza una excepción (NumberFormatException o personalizada) si la entrada no es válida.
4. mostrarMensajeFinal(boolean ganador)
- Muestra un mensaje al final del juego indicando si el usuario ganó o perdió. Si ganó, muestra la puntuación obtenida.

Clase Main

Clase que contiene el método main y sirve como punto de entrada para el programa.

1. Crea una instancia de la clase JuegoAdivinanza.
2. Llama al método iniciarJuego() para comenzar el juego.
3. Permite reiniciar el juego si el usuario desea jugar otra partida.

Ayuda, generación de un número secreto entre 1 y 100:

```
numeroSecreto = (int) (Math.random() * 100) + 1;
```

Rúbrica de Evaluación

Rúbrica de Evaluación

Criterio	Peso	Descripción
Correcta implementación del límite de intentos	2 puntos	El juego finaliza correctamente al alcanzar los 10 intentos máximos, mostrando un mensaje claro.
Sistema de puntuación	2 puntos	La puntuación se calcula correctamente según los intentos realizados.
Control de datos de entrada	2 puntos	El uso de try-catch maneja adecuadamente las entradas inválidas, permitiendo reintentos.
Mensajes de ayuda y lógica del juego	2 puntos	Se muestran mensajes claros sobre si el número es mayor o menor, y el flujo del juego es correcto.
Interactividad y reinicio del juego	2 puntos	El programa permite reiniciar el juego correctamente tras finalizar una partida.