



- Tarea UT04 -

Uso de Estructuras de Control

Enunciado

Para afianzar los conocimientos en el uso de estructuras de control de flujo así como de excepciones, vamos a realizar una serie de programas. Se deberá crear un proyecto en Netbeans diferente para cada uno de los ejercicios propuestos, cuyo nombre será PROG04_EjercX, donde X es el número de ejercicio. Cada proyecto incluirá una sola clase que contendrá un método `main`. El alumno decidirá si incluir todo el código en el método `main` o dividirlo en diferentes métodos.

- **Ejercicio 1:** Escribe un programa que permita al usuario generar secuencias numéricas ascendentes utilizando tres bucles diferentes: `for`, `while` y `do-while`.
 - El programa debe funcionar de la siguiente manera:
 - Solicita al usuario que introduzca un número entero N, que debe ser menor que 50, mediante la clase `Scanner`. Si el número ingresado es mayor o igual a 50, muestra un mensaje de error y termina el programa.
 - Si el número es válido, el programa generará las siguientes secuencias hasta el número N utilizando cada uno de los bucles mencionados:
 - Una secuencia generada con un bucle `for` que muestra los números del 1 al N.
 - Una secuencia generada con un bucle `while` que muestra los números del N al 1 (en orden descendente).
 - Una secuencia generada con un bucle `do-while` que muestra solo los números pares entre 1 y N.

- Deberá implementar la excepción “`InputMismatchException`” y la genérica “`Excepcion`”.
- **Ejercicio 2:** Deseamos implementar un juego en Java que permita al usuario adivinar un número oculto (que será aleatorio). El funcionamiento será el siguiente:
 - El programa mostrará un pequeño menú (utilizando la estructura `switch`) en pantalla con las siguientes opciones (1. Configurar, 2. Jugar, 3. Salir).
 - Si el usuario selecciona la primera opción, se le solicitará por teclado el número de intentos permitidos (`numInt`) y el número máximo (`numMax`) generado utilizando el método `readLine()`.
 - Si el usuario selecciona la opción 2, el programa generará un número aleatorio entre 0 y `numMax` que será el número a adivinar (`numOculto`). A partir de este momento, se le solicitarán al usuario números hasta adivinar el número oculto.
 - Si el usuario adivina el número, se mostrará un mensaje *"Has ganado!. Has necesitado X intentos"*.
 - Si se consume el número de intentos sin adivinar el número, se mostrará el mensaje *"Perdiste!. Intentos consumidos"*.
 - En cada intento, si el usuario no adivina el número se le proporcionará una pista, por ejemplo, *"El número oculto es menor"*.
 - En ambos casos, la siguiente acción será mostrar el menú.
 - Si el usuario selecciona Jugar sin configurar previamente el número de intentos y el número máximo generado, se tomarán como valores por defecto: `numInt=5` y `numMax=10`.
 - Si el usuario pulsa la opción 3, el programa finaliza.
 - Deberá implementarse la excepción `NumberFormatException` y la genérica “`Excepcion`”.
 - Para generar un número aleatorio en java puedes utilizar el siguiente código:

Unset

```
int numOculto = (int)Math.floor(Math.random()*20+1);  
//genera un número aleatorio entre 0 y 20, ambos incluidos.
```

- **Ejercicio 3:** Escribir un programa que imprima un triángulo de asteriscos (*) por pantalla. El número de filas del triángulo debe ser proporcionado por el usuario. Por ejemplo, si el usuario ingresa 5, el programa debe mostrar lo siguiente:

```
      *
     ***
    *****
   *********
  ***********
```

- El programa debe solicitar al usuario mediante la clase **Scanner** que ingrese un número entero positivo n , que representará la cantidad de filas del triángulo.
- Luego, debes usar un bucle para imprimir las filas del triángulo. En cada fila, se deben colocar una cantidad de asteriscos que aumenta conforme avanzan las filas.
- Además de los asteriscos, cada fila debe tener los espacios necesarios para alinear el triángulo al centro, es decir, la primera fila debe tener muchos espacios a la izquierda y, conforme suben las filas, menos espacios.
- El número de asteriscos en cada fila debe ser impar (1, 3, 5, 7, etc.).
- Deberá implementar la excepción **"InputMismatchException"** y la genérica **"Excepcion"**.

Criterios de puntuación. Total 10 puntos.

- Ejercicio 1: 2 puntos.
- Ejercicio 2: 3 puntos.
- Ejercicio 3: 4 puntos.
- Comentarios y estructuración: 1 punto.
- Total: 10 puntos.

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK y JRE 21 o superior.
- NetBeans IDE 19 o superior.

Consejos y recomendaciones.

- Añade al código fuente los comentarios que consideres oportuno.
- Estructura el código de una forma correcta para un mejor seguimiento y corrección.

Indicaciones de entrega

Cada ejercicio estará contenido en un fichero cuyo nombre sea similar a **PROG04_EjerXX**, donde la XX se sustituirá por el número del ejercicio.

Los archivos fuentes de cada ejercicio (no el proyecto completo, sólo las fuentes).

Asegúrate de que los archivos fuente (.java), contengan, como comentarios Java, tu nombre y apellidos y el número de ejercicio.

Los archivos fuente deben compilar y ejecutarse siguiendo las instrucciones de su enunciado. De no hacerlo, no se considerarán válidos y serán evaluados con un 0.

En el supuesto que tengas/quieras adjuntar un documento de texto con explicaciones adicionales, o con la solución a alguno de los puntos pedidos, debes de seguir las siguientes consideraciones que se explican en el siguiente [documento](#).

Crea una carpeta para la entrega y adjunta los archivos creados, y renombra la carpeta siguiendo las siguientes pautas:

apellido1_apellido2_nombre_PROG_Tarea04

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas debería nombrar esta tarea como...

sanchez_manas_begona_PROG_Tarea04

IMPORTANTE

Si la carpeta es muy pesada para el límite máximo de subida de la plataforma Moodle, súbela a tu unidad del Drive. Utiliza la cuenta de Educación, y en la entrega en Moodle adjunta la URL de tu entrega.

Para que la entrega se considere realizada en tiempo y forma, sube al Moodle un documento con la URL que contenga la solución de tu tarea. Deberá contener los archivos y documentos solicitados. Comparte la carpeta mediante su enlace correspondiente al Drive.

No olvides ajustar los privilegios de la carpeta en Drive para que pueda acceder a tu trabajo.

Todos los ejercicios y/o /documentos demandados que no cumplan exactamente con las instrucciones y su formato de entrega no se considerarán válidos y serán evaluados con un 0.