# EXAMEN FEBRERO
# 10 DE FEBRERO DE 2023

## Instrucciones de entrega

Una vez realizado el examen, comprimirás todas las carpetas de los proyectos Netbeans en un único archivo en formato ZIP que enviarás para su corrección.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**Apellido1_Apellido2_Nombre_PROG_ExamenFebrero.zip**

## Programas

1. **(1,5 puntos)** Crea un proyecto en Netbeans llamado **CuadradoNumero** en donde se implementará en Java un programa con las funciones necesarias para calcular el valor del cuadrado de un número siguiendo el siguiente algoritmo:

> *"El cuadrado de un número entero puede calcularse como la suma de tantos números impares consecutivos (incluido el 1), como indica el valor absoluto de la base."*
> *Ejemplo 1: $3^2 = 1 + 3 + 5$ (se suman 3 números impares consecutivos desde el 1).*
> *Ejemplo 2: $(-5)^2 = 1 + 3 + 5 + 7 + 9$ (se suman 5 números impares consecutivos desde el 1).*

El programa se compondrá de los siguientes elementos:

- **(1 punto)** ***public static int cuadradoSumando(int n)***. Recibirá un valor entero por parámetro y devolverá el cálculo de la potencia de dicho número empleando el algoritmo descrito anteriormente.
- **(0,5 puntos)** Programa principal ***main***. Pedirá al usuario que introduzca un valor entero y mostrará en pantalla el cálculo de su potencia. El programará finalizará cuando se introduzca el valor 0. Se debe validar adecuadamente que el usuario introduce un valor numérico.

2. **(2 puntos)** Crea un proyecto en Netbeans llamado **Submatriz** en donde se implementará en Java una función que devolverá una submatriz a partir de una matriz recibida por parámetro.

- **(1,5 puntos)** ***public static int[][] submatriz(int [][] matriz, int filaInicial, int colInicial, int filaFinal, int colFinal)***. Recibirá por parámetro una matriz cuadrada (NxN) y la coordenada inicial y final de dicha matriz a partir de la cual se generará la submatriz. La submatriz se debe devolver como una estructura nueva con los valores copiados de la matriz original. Si las coordenadas no son correctas o no permiten generar una **submatriz de al menos 2x2** se lanzará la excepción *IllegalArgument Exception*.
- **(0,5 puntos)** Programa principal ***main***. Pedirá al usuario la dimensión de la matriz y generará una matriz aleatoria. Posteriormente, pedirá al usuario los índices de la submatriz a generar, que se mostrará por pantalla tras ello.

3. **(6,5 puntos)** Crea un proyecto Netbeans que se llame **Conjunto** que contendrá la clase ***ConjuntoOrdenadoEnteros***. Esta clase se implementará del siguiente modo.

Un conjunto es una estructura de datos dinámica con un funcionamiento especial, ya que se pueden añadir elementos sin límites de forma transparente al usuario, aunque internamente se puede usar una estructura estática como un array. Además, un conjunto es una estructura que **no permite elementos duplicados**, por lo que se debe comprobar en todo momento que no se introducen valores ya existentes. Por último, **queremos que el conjunto esté ordenado**, por lo que la inserción de nuevos elementos deberá mantener el orden interno de los mismos **de menor a mayor**.

Implementaremos esta clase para elementos de tipo entero. Contendrá los siguientes atributos, constructores y métodos:

- **Atributos privados**:
  - ***int [] conjunto***. Será el array donde almacenaremos los elementos. Tendrá una capacidad inicial por defecto de 100 elementos.
  - **int elementos**. Llevará la cuenta del número de elementos almacenados en el conjunto. Inicialmente valdrá 0 y se actualizará conforme se añadan o saquen elementos.
- **Constructores (reutilizarán código con this() si es posible):**
  - **(0,5 puntos)** ***public ConjuntoOrdenadoEnteros()***. Constructor por defecto que construye el array con la capacidad por defecto e inicializa el valor de *elementos*.
  - **(0,5 puntos)** ***public ConjuntoOrdenadoEnteros(int capacidad)***. Construye el array con la capacidad indicada por parámetro e inicializa el valor de *elementos*.
  - **(0,5 puntos)** ***public ConjuntoOrdenadoEnteros(ConjuntoOrdenadoEnteros c)***. El constructor copia creará un nuevo conjunto cuya capacidad será igual al número de elementos del conjunto recibido por parámetro y tendrá los mismos valores que este.
- **Métodos públicos:**
  - **(1 puntos)** ***public void add(int valor)***. Se añadirá el nuevo valor al conjunto de forma que se mantenga el orden en el mismo. Si el valor ya existe no se añadirá. Si no hay espacio se deberá crear un nuevo array con 10 elementos más donde se copien los valores del conjunto añadidos hasta el momento.
  - **(1 puntos)** ***public void add(ConjuntoOrdenadoEnteros c)***. Se añadirán todos los valores del conjunto recibido por parámetro que no existen en este conjunto manteniendo el orden del mismo. Si no hay espacio se deberá crear exactamente el hueco necesario.
  - **(1 puntos)** ***public int search(int valor)***. Devuelve la posición del valor recibido por parámetro en el conjunto. Si el valor no existe, se devolverá un valor negativo.
  - **(1 puntos)** ***public int del(int posicion)***. Devuelve el valor del elemento indicado por el parámetro *posicion*, eliminándolo del conjunto y compactando el array para eliminar los posibles huecos. Si la posición es mayor al número de elementos se lanzará la excepción *IllegalArgumentException*.
  - **(1 puntos)** ***public ConjuntoOrdenadoEnteros subconjunto(int indiceInicial, int indiceFinal)***. Devolverá un objeto ConjuntoOrdenadoEnteros que contendrá los elementos entre el índice inicial (incluido) y el índice final (excluido). Si no es posible crear el subconjunto se lanzará la excepción *IllegalArgumentException*.

Implementa una función ***main*** para probar la clase y todos sus constructores y métodos.

## Resultados de Aprendizaje, Criterios de Evaluación y Rúbrica

En cada ejercicio / apartado se tocan los siguientes resultados de aprendizaje y criterios de evaluación:

| Resultado de Aprendizaje | Criterios de evaluación |
|---|---|
| **RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.** | **d, e, f, g, h, i** |
| **RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.** | **a, b, c, d, e, f, g, h, i** |
| **RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.** | **a, b, e, f, g** |
| **RA 4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.** | **a, b, c, d, e, f, h** |
| **RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.** | **a** |
| **RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.** | **a** |

Cada ejercicio / apartado se valorará con la siguiente rúbrica.

| # | Criterio | Porcentaje |
|---|---|---|
| **1** | El programa/función implementado no cumple los requisitos, no soluciona de forma algorítmica el ejercicio o las soluciones obtenidas por el programa no son las esperadas, el código no compila o contiene errores. | 0% |
| **2** | El programa/función implementado soluciona de forma algorítmica el ejercicio pero falla con datos de entrada no permitidos. | 25% |
| **3** | El programa/función implementado tiene fallos inesperados en situaciones específicas o concretas, es decir, falla para un determinado caso o valor de entrada, pero en general el resultado obtenido es válido. | 50% |
| **4** | El programa/función implementado cumple los requerimientos pero:<br>• El código no es legible o no está bien estructurado. | 75% |
| **5** | El programa/función implementado se ajusta perfectamente a la especificación:<br>• Se validan los datos de entrada.<br>• El resultado obtenido es válido para cualquier dato de entrada.<br>• El código es modular y se emplean funciones/métodos adecuadamente.<br>• El código es legible y usa comentarios relevantes y/o Javadoc. | 100% |