Cátedra Ingeniería de Software
Asignatura: Estructura de Datos
Código 00825
Primer cuatrimestre 2023
Proyectos

Proyecto 1

Objetivo de aprendizaje

- Adquirir los conocimientos básicos sobre las estructuras de datos.
- Analizar la forma en que se implementa la recursión y comprender cuándo se debe utilizar.
- Comprender el funcionamiento de algoritmos de ordenamiento.

Temas de estudio

- La API de colecciones y recursividad.
- Algoritmos de ordenamiento.

Descripción del trabajo

Precondiciones:

1. Como herramienta de desarrollo se deberá utilizar el Netbeans en su versión 8.2 o 16.
2. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
3. Cuando se soliciten o se muestren datos al usuario, debe ser a través de las pantallas gráficas, no se permite el uso de JOptionPane para solicitar datos o mostrar salidas de datos. Para ello se pueden utilizar cajas de texto, etiquetas o listas gráficas según sea.
4. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

Instrucciones:

Se deberá implementar un programa para llevar un registro de equipos participantes en la Copa del Mundo FIFA de Qatar 2022.

El programa deberá contener las siguientes funcionalidades:

Registro de equipos:

En esta pantalla se podrá incluir los datos generales de cada equipo participante, con los siguientes campos:

Nombre del campo	Tipo de dato
ID del equipo	Identificador consecutivo numérico generado por el programa.
Nombre del equipo	Texto con longitud máxima de 20 caracteres.
Posición actual en el ranking de FIFA	Número entero único entre 1 y 100.
Año de fundación	Número entero de 4 dígitos.

La información de los equipos con los campos anteriormente especificados, se guardarán en una clase “Equipo” y cada una de estas se agregará a una única colección utilizando la clase HashSet de Java.

Listado de equipos:
En esta pantalla se listarán todos los equipos registrados en la colección (con todos sus campos). En esta opción se debe utilizar una clase de tipo Iterator para recorrer la colección.

Búsqueda y edición de un equipo:
En esta pantalla se solicitará el ID de un equipo para realizar la búsqueda del registro correspondiente. Debe utilizar recursividad para esta búsqueda utilizando el campo ID del equipo.
Para la implementación de la búsqueda, se deberá crear un arreglo y se copiará en este cada uno de los ID de la colección HashSet. Una vez completado el arreglo, se realizará la búsqueda recursiva sobre este. Si el ID es encontrado, se mostrarán los campos de la clase “Equipo” correspondiente.
Los campos “Posición actual en el ranking de FIFA” y “Año de fundación” del Equipo encontrado, se mostrarán en cajas de texto para su edición o modificación.
Los campos “ID del equipo” y “Nombre del equipo”, solo se mostrarán y no podrán ser editados.
Habrá un botón GUARDAR para guardar los datos en la colección.

Ordenamiento de la lista de equipos:
En esta pantalla se mostrarán los equipos registrados ordenados ascendentemente por el campo “Posición actual en el ranking de FIFA” o por el campo “Año de fundación”, según lo indique el usuario de la aplicación.
El ordenamiento debe hacerlo utilizando el algoritmo Burbuja o el QuickSort, según lo indique el usuario.
Previo al ordenamiento, se deben copiar los datos a ordenar del HashSet a un arreglo para uso del algoritmo de ordenamiento correspondiente.
Finalmente se mostrará la lista ordenada de equipos con todos sus campos, según el criterio de ordenamiento indicado.

Debe existir un menú con cada una de las opciones.
Los resultados de las opciones se muestran en la pantalla correspondiente, sin usar mensajes de diálogo para ello.

Rúbrica

Detalle	Puntos
Registro de personas usando HashSet.	1
Utiliza todos los campos como se indica en el Registro de equipos.	1
Lista todos los registros de la colección mostrando todos los campos.	1
Utiliza una clase Iterator para recorrer la colección.	1
Búsqueda de equipo por ID.	1
Utiliza recursividad en la búsqueda de equipo.	2
Se edita correctamente los datos del equipo.	1
Uso de algoritmo de ordenamiento Burbuja para ambos criterios	2
Uso de algoritmo de ordenamiento Quicksort para ambos criterios	2
Utiliza un menú con las distintas opciones.	1

El programa está hecho en modo gráfico (sin consola y sin uso de JOptionPanel o mensajes de diálogo).	3
TOTAL	**16**

Proyecto 2

Objetivo de aprendizaje

- Analizar el funcionamiento de las pilas y colas y explicar su implementación.

Temas de estudio

- Pilas y colas.

Descripción del trabajo

Precondiciones:

1. Como herramienta de desarrollo se deberá utilizar el Netbeans en su versión 8.2 o 16.
2. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
3. Cuando se soliciten o se muestren datos al usuario, debe ser a través de las pantallas gráficas, no se permite el uso de JOptionPane para solicitar datos o mostrar salidas de datos. Para ello se pueden utilizar cajas de texto etiquetas o listas gráficas según sea.
4. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

Instrucciones:

Se deberá implementar un programa para gestionar instrumentos musicales por tipo.

El programa contendrá las siguientes pantallas o funcionalidades:

Registro de tipos de instrumento

Debe existir una pantalla gráfica que nos permite incluir tipos de instrumentos.
Cada tipo de instrumento incluido se almacenará en un arreglo de objetos de la clase “TipoInstrumento”, la cual funcionará como una cola (primer elemento en entrar, primero en salir).
Un atributo del elemento “Tipo instrumento” será un arreglo de objetos de la clase “Instrumento”, el cual se describe en la siguiente funcionalidad (Registro de instrumentos).

No se podrán usar colecciones o interfaces de Java como Queue, ArrayList o ninguna otra colección.

En esta pantalla se podrá incluir registros de tipos de instrumentos con los siguientes campos:

Nombre del campo	Tipo de dato
ID del tipo de instrumento	Número entero. Debe ser un consecutivo único generado por el sistema.
Descripción	Alfanumérico de máximo 20 caracteres.

Registro de instrumentos:

En esta pantalla se debe mostrar una lista gráfica con los diferentes tipos de instrumento registrados. El usuario deberá seleccionar uno de los tipos de instrumento y a continuación ingresar los datos del instrumento, el cual se relacionará al tipo de instrumento seleccionado.

Los campos por ingresar para un instrumento serán los siguientes:

Nombre del campo	Tipo de dato
DNI	Es un identificador único del instrumento. Debe ser un código GUID, el cuál es generado aleatoriamente y es único. Este código es un estándar universal. Ejemplo: 70aba303-60d8-4cb5-b3e7-4170c4be5642.
Nombre	Alfanumérico de máximo 20 caracteres.
Condición	Se selecciona un valor de una lista con las siguientes opciones: ● Nuevo ● Usado ● Reconstruido
Valor	Número con dos decimales.

El almacenamiento de los instrumentos por tipo de instrumento se realizará mediante un arreglo de objetos de la clase “Instrumento”, el cual funcionará como una pila (último en entrar, primero en salir).
No podrá utilizarse ninguna colección de Java como Stack, ArrayList ni ninguna otra colección.

Se deberá utilizar una pila independiente con los diferentes registros de instrumentos, dentro de cada elemento de la cola principal de tipos de instrumento.

Por ejemplo, la estructura de registros se definirá así:

Tipos de Instrumento (Cola)	Instrumentos (Pila)
Viento	Saxofón
	Trompeta
	Flauta dulce
	…
Cuerdas	Guitarra
	Contrabajo
	…
Percusión	(pila vacía)

Al inicio la Cola de Tipos de instrumentos estará vacía. Al incluir un primer tipo de instrumento, éste contendrá una pila de instrumentos vacía, hasta que el usuario incluya los instrumentos de ese tipo.

Eliminación de instrumentos

En esta pantalla se debe mostrar una lista gráfica con los diferentes tipos de instrumento registrados. El usuario deberá seleccionar uno de los tipos de instrumento y a continuación se mostrarán los instrumentos registrados en el tipo seleccionado (se deben mostrar todos los campos).

Al hacer clic en un botón “Eliminar Instrumento”, se eliminará el instrumento que se registró de último (recordemos que se trata de una pila). Luego, se refrescará la lista de instrumentos mostrando los que quedan relacionados al tipo de instrumento que habíamos seleccionado.

Eliminación de tipos de instrumento

La eliminación de tipos de instrumento consiste en una acción que intentará eliminar de la cola el primer elemento registrado (como en cualquier cola).

La eliminación se hará efectiva únicamente si el primer elemento de la cola no contiene ningún instrumento registrado (su pila de instrumentos está vacía). En caso de existir al menos un instrumento, se mostrará un mensaje (utilizando una etiqueta en la pantalla, no usando JOptionPane.showMessageDialog) indicando que no se puede eliminar el tipo de instrumento debido a que tiene instrumentos relacionados.

Si el tipo de instrumento es eliminado, se refrescará la lista de tipos de instrumentos con los elementos existentes.

Debe existir un menú con cada una de las opciones.

Rúbrica

Detalle	Puntos
Registro de tipos de instrumento usando un arreglo primitivo de objetos como una cola.	1
Registro de instrumentos usando un arreglo primitivo de objetos como una pila.	1
Eliminación de un instrumento.	1
Eliminación de un tipo de instrumento.	1
Verifica si un elemento de la cola está vacío.	1
Utiliza un menú con las distintas opciones.	1
El programa está hecho en modo gráfico (sin consola y sin uso de JOptionPane).	3
TOTAL	**9**

Proyecto 3

Objetivo de aprendizaje

- Aplicar el funcionamiento de la estructura de datos conocida como árbol y árbol binario de búsqueda.

Temas de estudio

- Árboles.
- Árboles binarios de búsqueda.

Descripción del trabajo

Precondiciones:

1. Como herramienta de desarrollo se deberá utilizar el Netbeans en su versión 8.2 o 16.
2. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
3. Cuando se soliciten o se muestren datos al usuario, debe ser a través de las pantallas gráficas, no se permite el uso de JOptionPane para solicitar datos o mostrar salidas de datos. Para ello se pueden utilizar cajas de texto, etiquetas o listas gráficas según sea.
4. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

Instrucciones:

Se debe crear un programa que gestione un árbol binario.

En una única pantalla, el usuario tendrá las siguientes opciones:

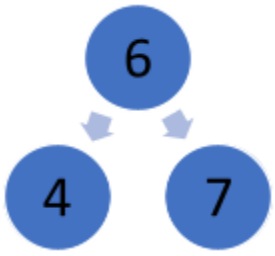
Inserción:

El usuario debe crear un nodo con un número entero y el programa la guardará en el árbol binario.

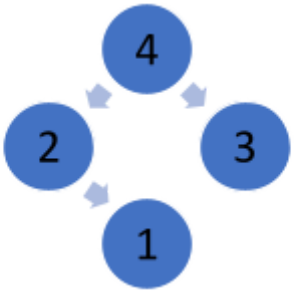
Antes de incluir el nodo, el programa de buscar el valor a ingresar en el árbol binario a ver si existe, ya que no se permitirá crear un nodo con un número que ya fue incluido anteriormente.

El primer número ingresado será la raíz. A partir del segundo número a ingresar, se debe insertar en forma ordenada, de modo que, si el número es menor a la raíz, se debe insertar a la izquierda, de lo contrario a la derecha.

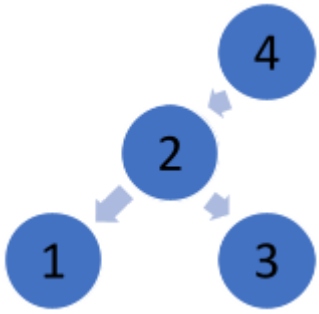
Ejemplo de árbol correctamente ordenado:



Ejemplo de árbol ordenado incorrectamente:



El ordenamiento correcto del árbol anterior es:



En el ejemplo anterior, para lograr ese ordenamiento, el primer elemento a insertar sería el 4, ya que con ello queda de raíz. A partir de ahí se empieza a ordenar cada nueva letra incluida. De esta manera, ingresó el 2, la cual al ser menor que 4 se ubica a su izquierda. Luego ingresa el 3 el cual es menor que el número de la raíz y mayor que el 2, por lo que se ubica a la derecha del 2.
Posteriormente ingresa el 1, el cual es menor que el 4 y menor que el 2, entonces va a la izquierda de este.

No se deben insertar números repetidos. El programa deberá realizar la validación correspondiente recorriendo el árbol binario (no se permite utilizar otra estructura paralela). Si el número ya existe, se muestra un mensaje indicando que no se pudo incluir el elemento (usar una etiqueta para esto, no un JOptionPane.showMessageDialog).

Eliminación:

El usuario digitará el número a eliminar. El programa procederá a eliminar el nodo correspondiente.

Altura del árbol

Esta opción mostrará la altura del árbol.

Recorrido PRE-ORDEN

Esta opción mostrará el contenido del árbol utilizando un recorrido pre-orden.

Recorrido POST-ORDEN

Esta opción mostrará el contenido del árbol utilizando un recorrido post-orden.

Recorrido IN-ORDEN

Esta opción mostrará el contenido del árbol utilizando un recorrido in-orden.

Mostrar nodos hojas

Esta opción mostrará los nodos que son hojas en el árbol.

Cada una de las funciones se pueden realizar en cualquier momento y en cualquier orden.
No se debe limpiar el árbol binario en ningún momento, a menos que sea utilizando la función de Eliminación (para cada todos los nodos) o al detener la ejecución del programa.

Rúbrica

Detalle	Puntos
Inserción de nodos	1
Validación si ya existe el número en el árbol binario recorriendo el mismo.	1
Eliminación de nodos	1
Altura del árbol	1
Recorrido pre-orden	1
Recorrido post-orden	1
Recorrido in-orden	1
Mostrar nodos hojas	1
El programa está hecho en modo gráfico (sin consola y sin uso de JOptionPane).	2
TOTAL	**10**