

Universidad Nacional de La Matanza

Programación 1110 - Comisión 05-1965 (Lunes - Jueves - Noche)

Segundo Examen Parcial - 12/jul/2021

Sección Lenguaje C

1. Se dispone de un archivo denominado `cuentas.dat` que posee registros de estados financieros de un conjunto de clientes. Se le solicita generar un índice para dicho archivo sobre un árbol binario de búsqueda. Tenga en cuenta que el archivo se encuentra ordenado. Usted debe determinar cuál es el criterio que se ha empleado para ordenarlo. Tenga en cuenta que el árbol generado debe encontrarse lo más balanceado posible.
2. Se dispone de un archivo denominado `novedades.dat` que contiene los montos para actualizar las cuentas. Dichas novedades se imputarán como una suma sobre el saldo de la cuenta. Debido a que pueden existir más de una novedad para una cuenta, se solicita que los registros del archivo sean llevados a una lista doblemente enlazada. Se debe tener en cuenta que cada elemento de la lista contendrá un único valor resumen para cada cuenta. El archivo de novedades solo podrá ser recorrido una sola vez.
3. Cargue en una cola implementada sobre una lista circular el contenido de la lista con el resumen de novedades generada en el **punto 2**. Vacíe la lista.
4. Actualice el archivo de registros de estados financieros del **punto 1** a partir del contenido de la cola del **punto 3**. Tenga en cuenta que para acceder al registro de novedades debe utilizar el índice que generó en el punto 1. En caso de que alguno de los elementos contenidos en la cola no corresponda a cuentas existentes, debe informarlos por el `FILE` recibido por parámetro en la función.
5. Poda el árbol generado en el punto 1 a la altura que se indica en el macro reemplazo `ALTURA_PODA`. Los nodos eliminados en el proceso de poda deben insertarse en una cola implementada en una lista circular.
6. A partir de la cola generada en el punto anterior, actualice el estado de los registros del archivo de estados financieros `cuentas.dat` cambiando su estado de Activo (A) a Baja (B) cuando corresponda. Utilice acceso aleatorio a partir de la información contenida en los elementos de la cola.

Sección Lenguaje C++

Cree un nuevo proyecto organizado en archivos separados (`main.cpp`, `xxx.h`, `xxx.cpp`). Respetando la nomenclatura de los archivos y de las clases.

Para el siguiente ejercicio, las notas musicales deben representarse de la siguiente manera:

Nota	Letra
DO	C
RE	D
MI	E
FA	F
SOL	G
LA	A

Nota	Letra
SI	B

1. Desarrollar las clases correspondientes para que pueda utilizar con el siguiente código:

```
int main()
{
    cout << "Esperado: Instrumento: Contrabajo - G"
          << endl << "-----: ";
    Cuerdas *c1 = new Cuerdas("Contrabajo", 'G');
    mostrar(c1);

    cout << "Esperado: Instrumento: Contrabajo - B"
          << endl << "-----: ";
    (*c1)++;
    ++(*c1);
    mostrar(c1);
    delete c1;

    cout << "Esperado: Instrumento: Oboe - A material:madera"
          << endl << "-----: ";
    Vientos v2("Oboe", 'A', "madera");
    mostrar(&v2);

    cout << "Esperado: Instrumento: Oboe - F material:madera"
          << endl << "-----: ";
    v2--;
    --v2;
    mostrar(&v2);

    cout << "Esperado: Instrumento: Bateria - X"
          << endl << "-----: ";
    Percusion p3("Bateria");
    mostrar(&p3);

    cout << endl
          << "Esperado: Excepcion: No permite cambio de escala"
          << endl << "-----: ";
    try
    {
        p3++;
        ++p3;
        p3--;
        --p3;
    }
    catch(CambioEscalaException &cee)
    {
        cout << "Excepcion: " << cee.what() << endl;
    }

    return 0;
}
```

La funcion mostrar es la siguiente:

```
void mostrar(Instrumento* i)
{
```

```
        cout << "Instrumento: " << i->getNombre()
            << " - " << i->getEscala()
            << " " << i->getDetalles()
            << endl;
    }
```

Adicionalmente, el siguiente código **NO** debería compilar:

```
int main()
{
    Instrumento i("", 'A');
    return 0;
}
```

2. Implemente las clases del punto anterior.

Sección Lenguaje Java

Complete las siguientes preguntas en el archivo `java.txt`.

1. ¿Qué es Java?
2. Cuales son las principales características de Java
3. ¿Qué se entiende por código manejado?
4. Nombre alguno de los IDE que se pueden utilizar para desarrollar aplicaciones Java.

Condiciones de aprobación del examen

- El programa debe compilar correctamente en todos los casos.
- Debe implementar las funciones finalizadas en “_MIO”. No debe realizar funciones monolíticas.
- La compilación debe estar exenta de **warnings** arrojados por el compilador.
- Las salidas con los resultados del archivo `pantalla.txt` y deben ser **iguales** a las que se encuentran en el archivo `pantallaBIEN.txt`.
- La interpretación de los requisitos y la estructura de la aplicación es parte de la evaluación.
- Requisitos para nota **4** (cuatro): puntos anteriores, dos ejercicios a su elección de la Sección Lenguaje C y la o las definiciones de las clases solicitadas en Sección Lenguaje C++ y dos respuestas de la sección Lenguaje Java.
- Requisitos para nota **7** (siete o más): puntos anteriores. Un ejercicio adicional de la Sección Lenguaje C, resolución completa del ejercicio 2 de la Sección Lenguaje C++ y una respuesta adicional de la sección Lenguaje Java.

Observaciones generales

- Para el proyecto de la Sección Lenguaje C recuerde tener deshabilitado el flag `-std=c99`.
- Para la Sección Lenguaje C sólo debe entregar los archivos `funciones.c` y `que_hice.h`.
- Para la Sección Lenguaje C++ sólo debe entregar los archivos `.cpp` y `.h` de la clase desarrollada.
- Para la Sección Lenguaje Java solo debe entregar el archivo `java.txt`, dicho archivo debe contener sus datos personales en la cabecera del mismo.
- Generar un archivo Zip con los archivos a entregar indicados en los apartados anteriores. El archivo debe tener el nombre:

DDDDDDDD_20210712_parcial2.zip, donde DDDDDDDD es el número de DNI del alumno. Subir el archivo como entrega en la plataforma MIeL.