

Laboratorio 6 – Herencia, Polimorfismo y Sobrecarga

Valor 5.0 pts. (19.7%)

Enunciado:

Crear una aplicación en C++ que use **Herencia** por medio de una jerarquía de al menos tres niveles, implemente **Composición**, **Polimorfismo**, **sobrecarga de métodos**, **sobrecarga de constructores** o **un constructor con argumentos predeterminados** en dos de las entidades (Clases) definidas para el proyecto final de curso.

La cantidad de atributos de las clases es establecida por el grupo de programadores y depende del sistema propuesto.

La **Composición** se puede usar en cualquiera de las clases a implementar.

Una clase (entidad 1) debe implementar **sobrecarga de constructores** (al menos dos)

La otra clase (entidad 2) debe implementar **un constructor con argumentos predeterminados**

Cualquiera de las clases debe **sobrecargar al menos 2 veces uno de sus métodos**, bien sea para dar valores a los atributos o realizar cualquier operación sobre los mismos.

En las pruebas a realizar se debe evidenciar el uso del Polimorfismo y la sobrecarga de métodos. Informar por pantalla en cual prueba se usará esta característica.

Los objetos de las clases se almacenan en vectores diferentes.

La aplicación muestra un menú en pantalla con las opciones:

1. Adicionar entidad 1: el objeto se crea mediante el constructor sobrecargado 1
2. Adicionar entidad 1: el objeto se crea mediante el constructor sobrecargado 2
3. Adicionar entidad 2: se llama al constructor con argumentos predeterminados
4. Listar: se muestra un submenú para seleccionar la entidad a listar y se muestran todos los objetos y atributos de esta
5. Acerca de... Por pantalla aparece los datos de los programadores en el formato:

Programadores

Nombre - Código Estudiantil

Nombre - Código Estudiantil

Nombre - Código Estudiantil

6. Salir

Sistema de evaluación:

- **RA1 y RA2:** Crea nuevos tipos de dato (clases). Diseño básico de una clase con atributos y métodos set y get para su funcionamiento. Separa la interfaz de la implementación. Usa adecuadamente las referencias a objetos. Usa un vector para guardar objetos de clase
- **RA3:** Diseña el programa de forma flexible para permitir extenderlo en el futuro. Análisis de problemas y diseño de clases. Implementa la sobrecarga como mecanismo para aumentar la funcionalidad del código

- **RA4 y RA5:** Documentación del código fuente y selección adecuada de nombres (identificadores) de variables, atributos, clases, métodos y objetos. Creación de la documentación en formato de página web mediante cualquier generador de uso gratuito, por ejemplo, [Doxigen](#). Trabaja en equipo
- **Cumplimiento de requerimientos y funcionamiento de la aplicación**

Observaciones:

- **Metodología:** equipos de trabajo del proyecto final
- **Entregable:**
 - diagrama de clases
 - carpeta .zip con el proyecto en Code::Blocks
 - carpeta .zip con la documentación del código fuente generada en formato HTML
- **Método de entrega:** enlace campus virtual
- **Fecha máxima de entrega:** 05-12-2023 18:30