Diseño y Programación Orientados a Objetos 1er. Semestre 2024

Certamen II ELO-329 Diseño y Programación Orientada a Objetos (1s2024)

En este certamen usted no podrá hacer preguntas. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.

Segunda parte, **con apuntes**.

Segunda pregunta (35 pts.) La UTFSM le ha solicitado a ud. como programador la creación de una aplicación en C++ que gestione una lista de estudiantes de sus cursos. Cada estudiante tiene un nombre, un número de identificación y la nota final obtenida en un curso. La universidad necesita una solución sencilla que permita a los administradores del curso realizar varias operaciones esenciales de gestión de estudiantes. Su tarea es desarrollar esta aplicación siguiendo los requisitos especificados.

Requerimientos

La aplicación debe permitir al usuario realizar las siguientes operaciones:

- Añadir un nuevo estudiante a un curso.
- Mostrar la información de todos los estudiantes de un curso.
- Buscar un estudiante por su número de identificación y mostrar su información.

Instrucciones

1. Defina e implemente una clase **Estudiante** con los atributos y métodos necesarios.
2. Defina e implemente la clase **Curso** con los atributos y métodos necesarios.
3. Use un vector para almacenar la lista de estudiantes en la clase **Curso**.
4. Considere el código **main** disponible en el siguiente enlace como base para su desarrollo

```
// Agregue los archivos de cabecera que sean necesarios para su desarrollo


int main() {
    Curso curso;
    int opcion;
    do{
        std::cout << "\nMenú:" << std::endl;
        std::cout << "1. Añadir estudiante" << std::endl;
        std::cout << "2. Mostrar todos los estudiantes" << std::endl;
        std::cout << "3. Buscar estudiante por ID" << std::endl;
        std::cout << "4. Salir" << std::endl;
        std::cout << "Seleccione una opción: ";
        std::cin >> opcion;


        switch (opcion){
```

```
            case 1:
                curso.add_student();
                break;
            case 2:
                std::cout << curso << std::endl;
                break;
            case 3:
                curso.search_student();
                break;
        }
    }while(opcion != 4);


    return 0;
}
```

Una solución posible para esta pregunta está aquí.

Asignación de puntajes

Clase Estudiante (10 pts)
- Definición de la clase Estudiante (1 pt)
- Atributo nombre (1 pt)
- Atributo id (1 pt)
- Atributo nota_final (1 pt)
- Constructor para inicializar los atributos (2 pts)
- Métodos getNombre(), getId(), getNotaFinal() (3 pts)
- Métodos setNombre(), setId(), setNotaFinal() (1 pt)

Clase Curso (10 pts)
- Definición de la clase Curso (1 pt)
- Atributo vector de estudiantes (2 pts)
- Método para añadir estudiante (3 pts)
- Método para mostrar todos los estudiantes (2 pts)
- Método para buscar estudiante por ID y mostrar información (2 pts)

Código main y Funcionalidades (15 pts)
- Implementación del main (2 pts)
- Opción para añadir un nuevo estudiante (4 pts)
- Opción para mostrar la información de todos los estudiantes (4 pts)
- Opción para buscar un estudiante por su número de identificación y mostrar su información (5 pts)