Выполнил

студент группы КТбо1-2 Цыбко В. Д.

Принял

доцент кафедры САиТ Лапшин В. С.

Таганрог 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра системного анализа и телекоммуникаций

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Основы алгоритмизации и программирования»

**«РАЗРАБОТКА ПРИЛОЖЕНИЙ НА С++   
В ОБЪЕКТНОМ СТИЛЕ»**

Вариант 3

# Содержание

Содержание 2

Техническое задание 3

Цель задания 3

Задача 3

Ход работы 4

Написание программы 4

Пример работы программы 4

Вывод 5

Листинг 6

app.cpp 6

datamanager.h 7

datamanager.cpp 9

# Техническое задание

## Цель задания

Цель задания: ознакомиться с возможностью создания классов в С++, которые можно в последствии использовать для разработки приложения под Windows (CLI-приложение).

## Задача

Задачи лабораторной работы:

1. Разработать объектную модель приложения.
2. Создать CLI-консольное приложение.
3. Создать в среде VS классы разработанной объектной модели.
4. Отладить методы классов.

Заданием моего варианта является работа с тремя файлами: список книг Books.txt, список дисциплин Subjects.txt, список связей между книгами и дисциплинами Links.txt.

# Ход работы

## Написание программы

Для работы с тремя файлами я сделал свой класс Resource. Рассмотрю его поля и методы.

Класс имеет private поля String^ fileLinksPath, fileBooksPath, fileSubjectsPath. Они содержат адреса соответствующих текстовых файлов. Также у класса три private поля listLinks, listBooks, listSubjects. Они и будут хранить данные из трех файлов.

Рассмотрим public методы класса:

В конструкторе класса определяются пути к файлам Links.txt, Books.txt, Subjects.txt, а также списки через gcnew.

Функция ReadDataFiles читает данные из файлов при помощи System::IO::StreamReader в списки, и закрывает файлы.

Функция WriteDataFiles записывает в файлы данные из списков при помощи System::IO::StreamWriter. Строка для записи получается при помощи форматирование строк структуры данных в одну через String::Format.

Функция MakeNode принимает строку и формат данных, на ее основе создает соответствующую структуру данных и возвращает ее как Object^.

Функция AddNode принимает Object^ и формат данных и в соответствии с форматом записывает в нужный список объект.

Функция RemoveNode принимает Object^ и формат данных, ищет в списке данные по объекту, и удаляет вхождения. Если удаляется книга или дисциплина, также удаляются все связанные с ними связи.

Функция FindNode принимает Object^ и формат данных, ищет в списке похожие данные (при помощи (String^) line->Contains()), собирает найденные объекты в List<Objects^>^, и возвращает его.

## Пример работы программы

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

# Вывод

Итогом работы стали три класса, которые работают с тремя файлами с данными, могут оперировать ими (добавлять/удалять/искать).