**Проект по Обектно-ориентирано програмиране**

**Тема №7 Бази от данни**

Изготвил: Ростислав Миленов Митев, ФН: 2MI0400008

**Съдържание**

**Глава 1. Увод**

1.1. Описание и идея на проекта

1.2. Цел и задачи на разработката

**Глава 2. Преглед на предметната област**

2.1. Основни дефиниции, концепции и алгоритми, които ще бъдат използвани.

2.2. Дефиниране на проблеми и сложност на поставената задача.

2.3. Подходи, методи (евентуално модели и стандарти) за решаване на поставените проблемите.

**Глава 3. Проектиране**

3.1. Обща архитектура – ООП дизайн.

3.2. Диаграми (на структура и поведение – по обекти, слоеве с най-важните извадки от кода).

**Глава 4. Реализация, тестване**

4.1. Реализация на класове (включва важни моменти от реализацията на класовете и малки фрагменти от кода).

4.2. Управление на паметта и алгоритми. Оптимизации.

4.3. Планиране, описание и създаване на тестови сценарии (създаване на примери).

**Глава 5. Заключение**

5.1. Обобщение на изпълнението на началните цели.

5.2. Насоки за бъдещо развитие и усъвършенстване.

**1. Увод**

1.1. Описание и идея на проекта

В рамките на този проект трябваше да се реализира приложение, което да поддържа операции с прости бази от данни. Базите от данни се състоят от множество таблици, от която всяка е записана в собствен файл. Имаме главен файл (каталог), който съдържа имената на всички таблици в базата.

Една таблица се състои от колони, като всяка колона има име, тип и множество от съответния тип данни. Поддържаните типове са: цяло число, дробно число и символен низ.

1.2. Цел и задачи на разработката

Подходът ми за решение на задачата бе да направя общ клас Column, който служи като база на други 3 класа, а именно ColumnInt, ColumnDouble и ColumnString. В базовия клас имаме общи променливи и функции, а в другите 3 класа пазим информацията под формата на вектор от съответния тип.

Допълнително създадох клас Table, който представлява информация за една таблица. В нея имаме член-данни вектор от Column\*, за да можем да имаме множество от колони от различни типове, също имаме член-данни за името на таблицата, името на файла свързан с таблицата и размера(дължината) на колоните.

Главният клас, в който са всички функции, които са искани по условие, AllTables съдържа вектор от таблици. AllTables е главния начин, чрез който използваме програмата.

Подходът, който използвах бе чрез абстрактни класове.

**2. Преглед на предметната област**

2.1. Основни дефиниции, концепции и алгоритми, които ще бъдат използвани.

Проектът използва като основа библиотеките <string>, <vector>, <fstream>, <iostream>. Също така проекта съдържа абстрактни класове, полиморфизъм и колекции от обекти. Чрез тях е възможно осъществяването на много от функциите в проекта. Главните алгоритми в проекта са такива за търсене във файл.

2.2. Дефиниране на проблеми и сложност на поставената задача.

Основните проблеми при изработката на поставанета задача бяха при нагласата на различните класове. В последсвие и с тестването и дебъгването на функциите.

2.3. Подходи, методи (евентуално модели и стандарти) за решаване на поставените проблемите.

За реализация на приложението се използва обектно-ориентирано програмиране (ООП) за моделиране на бази от данни, таблици и операции върху тях. Подходящи методи включват четене и записване на файлове за управление на данните и използване на структури от данни като списъци и речници за съхранение и манипулиране на информацията в паметта. В допълнение, диалоговият режим за преглеждане на таблици е реализиран чрез итеративни цикли и команди за потребителския интерфейс.

**Глава 3. Проектиране**

3.1. Обща архитектура – ООП дизайн.

Проектът съдържа две основни архиттектури, тази на колона и тази на база от данни. Архиктектурата за колона съдържа абстрактен клас „Column“ и негови три подкласа „ColumnDouble“, „ColumnInt“ и „ColumnString“, които са съответно класове за колона с данни от тип число с плаваща запетая, колона с данни от тип цяло число и колона с данни от тип множество от символи.

3.2. Диаграми (на структура и поведение – по обекти, слоеве с най-важните извадки от кода).

A screenshot of a computer

Description automatically generatedДиаграма на клас Column:

A screenshot of a computer

Description automatically generated

Диаграма на клас ColumnInt:

A screenshot of a computer

Description automatically generatedДиаграма на клас ColumnString:

A screenshot of a computer

Description automatically generatedДиаграма на клас ColumnDouble:

A screenshot of a computer program

Description automatically generatedДиаграма на клас Table:

A screenshot of a computer

Description automatically generatedДиаграма на клас DataBase:

**Глава 4. Реализация, тестване**

A computer screen with white text and green text

Description automatically generated4.1. Реализация на класове (включва важни моменти от реализацията на класовете и малки фрагменти от кода).

Класът Column съдържа две член данни споделени от класовете които го наследяват. Първата е име на колоната, защото трябва да знаем какво представляват данните, които искаме да покажем, а втората член данна е от тип DataType, който е изброен тип служещ за разпознаването на типа данни, които се съдържат в дадена колона.

A screen shot of a computer

Description automatically generated Класовете ColumnInt, ColumnDouble и ColumnString имат много подобна струтура. Единсвената разлика при тях е вида данни, които се съдържат във вектора data\_. В ColumnDouble вида е double, а в ColumnString вида е std::string.

A computer screen shot of a code

Description automatically generated

Класът Table съдържа колекция от колони, реализирана чрез вектор, който съдържа данни от тип Column\*, така една таблица множе да съдържа колони от няколко типа, а не само един. Една таблица има също така име и име на файл, койото съдържа данните от обекта. Файлът е форматиран по специален начин чрез който може да се четат данните в него.

A screen shot of a computer

Description automatically generatedКласът DataBase съдържа само вектор с данни от тип Table като член данни. В този клас са всички методи, които са подадени от задачата.

4.2. Управление на паметта и алгоритми. Оптимизации.

Голяма част от управлението на паметта се осъществява чрез string и vector обектите. Те имат вградени методи и начини за работа с динамична памет, от които този проект се възползва.

4.3. Планиране, описание и създаване на тестови сценарии (създаване на примери).

В DatabaseProjectOOP.cpp файла можете да намерите do while цикъл, чрез който можете да тествате всички методи. Като примерни таблици са създадени три текстови файла (Shop.txt, Apartment.txt, Work.txt), те съдържат примерни данни.

**Глава 5. Заключение**

5.1. Обобщение на изпълнението на началните цели.

Създаденият проект е приложение за работа с прости бази от данни. То поддържа различни типове данни като цели числа, дробни числа и символни низове, както и специална стойност "NULL". То позволява основни операции като добавяне, показване, описване, експортиране и манипулиране на таблици чрез различни команди, включително импортиране на таблици от файлове, извеждане на съдържание, добавяне на колони, актуализиране и изтриване на редове, и вмъкване на нови редове. Това е постигнато чрез четене и записване на файлове, използване на структури от данни и реализиране на диалогов режим за взаимодействие с потребителя.

5.2. Насоки за бъдещо развитие и усъвършенстване.

Ето няколко насоки за бъдещо усъвършенстване на проекта:

* Проверки за невалидни формати на таблици.
* Функция, която добава или маха таблици в каталога на база на отворените таблици в базата от данни.
* Прилежно форматиране на файловете.