Проект по ООП. Тема 10 –Бази данни

1. **Увод**
   1. **Описание и идея**

Задачата е да се състави програма, поддържаща операции с прости бази от данни. Базите данни се състоят от множество таблици, всяка във отделен файл. Всяка таблица също е записана в главен файл(каталог). Таблиците трябва да поддържат колони от следните три типа – Int, double, string. Базата данни трябва да може да поддържа и изпълнява определени команди и функции, както и да може да зарежда и записва таблици и целият каталог във файл.

* 1. **Цел и задачи на разработката**

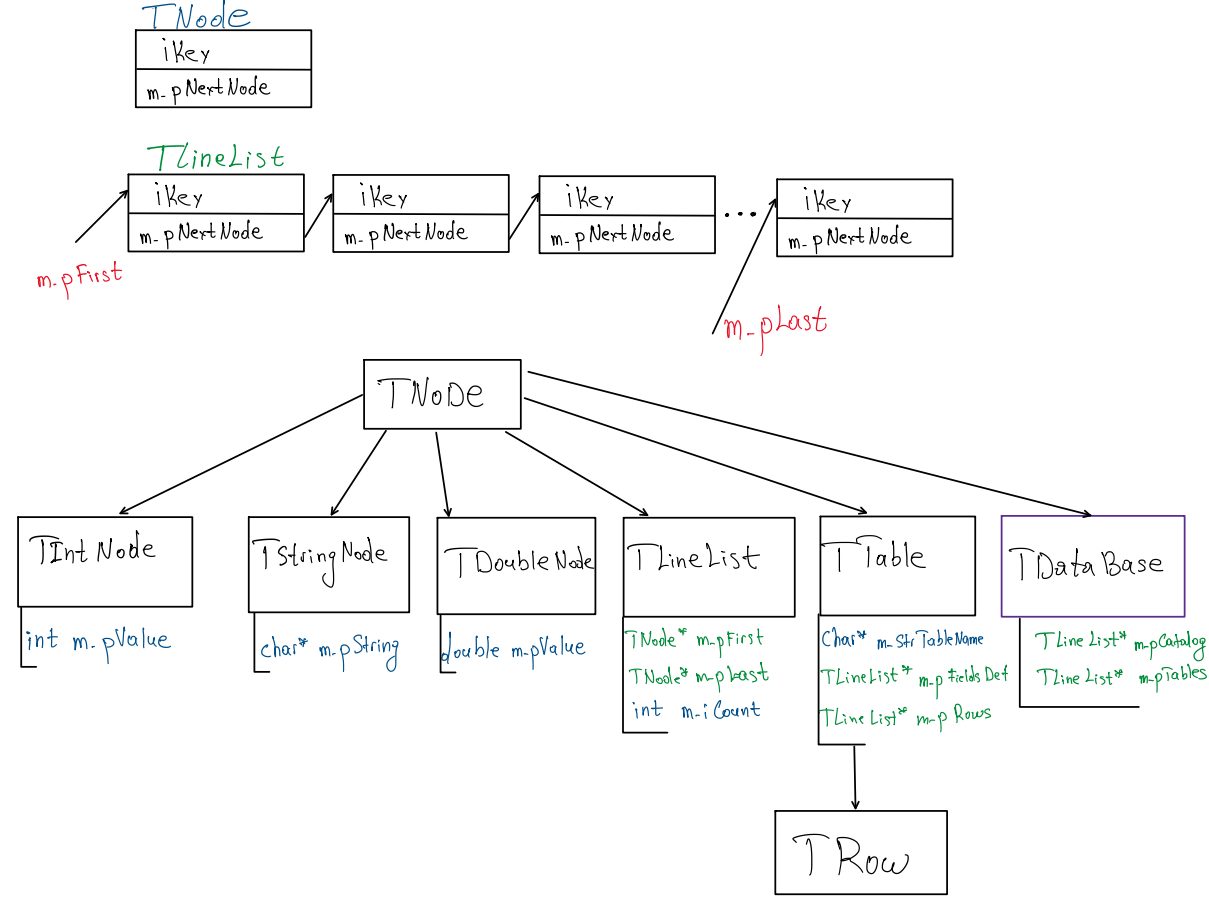
**Цел** – разработване на приложение за управление на данните в базата данни.

**Задачи за разработка** – Разработване на приложение за Visual Studio C++ с потребителски интерфейс, което да може да изпълнява всички нужни команди от заданието:

* **Import <file name>** – Добавя в базата данни нова таблица от файл. В него е записана информация за типа на всяка колона и името на таблицата. Извежда се грешка при опит за зареждане на таблица с вече съществуващо име. Новата таблица се добавя в каталога.
* **Showtables** – Показва списък с имената на всички заредени таблици.
* **Update <table name> <search column n> <search value> <target column n> <target value> -** За всички редове в таблицата, чиято колона с пореден номер съдържа стойността се променят така, че колоната им с пореден номер да получи стойност . Да се поддържа стойност NULL.
* **Describe <name>** – Извежда списък за типовете на колоните в дадената таблица.
* **Print <name>** – Извежда цялата таблица. Да се реализира диалогов режим, позволяващ съдържанието на таблицата да се преглежда по страници и командите: следваща страница, предишна страница, изход\*
* **Export <name> <file name> -** Записва таблицата във файл
* **Select <column-n> <value> <table name> -** Извежда всички редове от таблицата, които съдържат стойността <value> в клетката с дадения пореден номер.
* **Addcolumn <table name> <column name> <column type> -** Добавя нова колона (с най-голям пореден номер) в таблицата. Всички нови клетки да имат празна стойност.
* **Delete <table name> <search column n> <search value> -** Изтрива всички редове от таблицата, чиято колона с номер N съдържа value.
* **Insert <table name> <column 1> … <column n> -** вмъква нов ред в таблицата със съответните стойности. \*
* **Innerjoin <table 1> <column n1> <table 2> <column n2> -** Извършва операцията Inner Join над две таблици спрямо колоните с поредни номера в първата таблица и във втората. Създава нова таблица с автоматично генериран идентификатор и го извежда\*
* **Rename <old name> <new name> -** преименува таблица. Отпечатва грешка, ако новото име не е уникално.
* **Count <table name> <search column n> <search value> -** Намира броя на редовете в таблицата. Отпечатва грешка, ако новото име не е уникално.
* **Aggregate <table name> <search value> <target column n> <operation> -** Извършва дадена операция върху стойностите от колоната с пореден номер на всички редове, чиито колони с номер съдържат стойността. Възможните стойности на са sum, product, maximum, minimum. Системата да дава грешка, ако колоната не е от числов тип\*

1. **Преглед на предметната област**
   1. **Основни дефиниции, концепции и алгоритми**

* Основната концепция зад проекта е базовият клас TNode, който се състои от ID и указател към следващ елемент. Така този клас се наследява от всички класове по-надолу и позволява добавянето на елементи от различни типове към една и съща структура.
* Реализирането на динамичен свързан списък чрез който можем да представим всяка една таблица като списък от редове, където всеки списък пази целият ред. Използването на TNode & TLineList ни позволява почти навсякъде (освен където знаем точно какъв тип ни трябва) да се абстрахираме от типа на елемента, с който работим, а да работим само с указатели към тези елементи.

1. **Проектиране**
   1. **Обща архитектура**
2. **Реализация и тестване**
   1. **Управление на паметта и оптимизации**

* Изцяло динамично реализирана база данни.
* Използване на std::string, fstream,

1. **Заключение**
   1. **Обобщение на изпълнението на началните цели**

* Програмата успешно чете таблици от файл, обработва информацията, добавя колони и управлява цялата база данни без загуба на информация.

**Бъдещо развитие и усъвършенстване**

* Заменяне на Char\* със String за по-лесно управление на паметта и данните.
* Добавяне на допълнителни коментари
* Довършване на функционалността :
* Разбиване по страници при Print, Select (Добавяне на клас Page с редове и колони, като изходът вместо директно в конзолата се подава на клас Page, който да го форматира и през интерфейс на Page да извежда съответната страница)
* Innerjoin
* Aggregate

https://github.com/IvanBorissov/DataBase-Project2-OOP