Тема №6 Бази от данни

Изготвена от Станислав Стаматов ИС 1МИ0700208

# Анализ на задачата

Основният ход на програмата ще преминава през клас, който чете потребителски вход и управлява друг клас — База от данни. Класът База от данни ще съдържа вектор от таблици. Всяка таблица ще бъде представена като матрица и ще поддържа колони с различни типове. За да се реализира хетерогенен контейнер, ще бъде използван полиморфизъм и йерархия на наследяване между различните типове данни. Като транзакционност би се постигна чрез създаване на копие на определените редове/колони от таблицата, върху които ще се извършват операции.Ако някоя операция се провали, данните се възстановяват от това копие.

# Подход

* Типовете данни са реализирани като отделни класове (Integer, Decimal, MyString, Date), за да се позволи добавяне на специфични методи. Вместо да бъде един шаблонен клас, ограничаващ типовете до еднакви функционалностти за всички
* Имплементирана е йерархия с абстрактен базов клас Data, от който всички останали типове наследяват.
* За постигане на хетерогенна матрица се използва vector<vector<Data \*>> където указателите се освобождават ръчно в деструктора.
* Матрицата се пази ред по ред, вместо колона по колона. Като се запазва наредбата на самите редове при различните функционалностти
* Null стойност се отбелязва чрез булева член данна isNull в класа Data.
* За класът Date са използвани три целочислени променли за ден, месец и година
* MyString пази своите данни форматирани, пази данните във форматиран вид, а чрез deformatString (външна функция)
* Null стойностите се записват като текст
* Четенето на редовете става чрез getline последвано разделяне на подстрингове
* Като има и файлове Factory и Utils за споделени функционалности
* За сравняването на различните типове данни е използван оператор ==, в който се използва динамично кастване, защото самият клас Data няма функция за достъп до променливата поради причината, че не знае кой тип да върне
* В класът Table има най-различни функции за принтиране заради различните нужди на самата реализация като: ограничено принтиране на определени редове или на конкретен ред. Направено с цел преизползване.

# Стъпки за решаване

* В началото са имплементирани основните типове: Integer,Decimal, MyString, Date
* След това е създаден класът Table с нужните му функционалности.
* Таблиците се капсулират в Database
* И за накрая CommandHandler
* По време на разработка са създавани тестови функции по самото решение.

# Описание на класовете

### Абстрактен клас Data

### Член данна

* bool isNull

### Функции

* bool isDataNull() const; - за извеждане на състояние

Pure virtual функции,които се преписват от останалите типове

* read - за четене от поток
* print - за писане на поток
* getName - за извеждане тип на класа
* Clone - за създаване на копие
* emptyClone - за празно копие
* operator== и != - за сравняване
* converTo - за преобразуване до други типове

### клас Integer, Decimal

Реализират базовата функционалност от Data

### клас MyString

Добавя методи за форматиране и преформатиране

### клас Date

Съдържа функции за валидации при създаване на класа

### клас Table

Съдържа матрица от указатели Data и име на таблицата

Като има допълнителни помощни методи като:

* copyConvertedTargetColumn - която в буфер записва избрана от нас колона преобразувана в избран тип
* copyConvertedColumnWithSelect - копира в буфер колони където на избрани редове слага конкретна стойност, а на останалите стойност по подразбиране
* swapRows - функция, която вече трие данната в конкретна колна за всички редове и я променя с избрана от променлива
* copySpecificRows - копира в буфер конкретни редове
* getRowIndWithValue - извлича индексите на редове, които в избрана колона имат конкретна стойност

### клас Database

Поддържа вектор от таблици и има допълнителна член функция за търсене на таблица по име

### клас CommandHandler

Обединява функциите за взаимодействие с Databse и потребителския интерфейс в функция start

# Бъдещи подобрения

### Проблеми за оправа

* Заради четене от файл по стрингове и преобразуването им към double се губи информация след запетаята. Както и конвертиране от MyString към Double.
* Да се добави шаблонен метод във всеки тип за връщане на стойността му, за да се използва в оператор ==, вместо dynamic\_cast
* Липсва именуване на колони и частично съвпадение на низове
* addColumn не задава стойности по подразбиране при добавяне

### Подобрения

* Възможно е Database да използва по-добра структура от vector, например дърво
* Да се направи клас за съобщения с throw за по-ясно управление на грешки
* Клас Date може да използва uint8 и uint16 за оптимизация откъм размер
* Да се премахне повтарящ код в Table