



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика



Курсов проект по ООП

Летен семестър 2022/2023

Проект: Работа със CSV файлове

Изработил: Мария Стефанова Маргаритова

Специалност: Информационни системи

Курс: 2

Факултетен номер: 1MI0700004



Глава 1. Увод

1.1 Описание и идея на проекта

Идеята на проекта е да се реализира програма, която зарежда таблици от CSV файлове и може да извършва различни действия с тях. Идеята е да се усвоят знания за това как се работи с такъв тип таблици, как може те да се обхождат, да се изтриват редове, колони или пък да се извършват сортировки. Проектът е интересен, защото нагледно се запознаваме с процеса, който се случва „задкулисно“ докато работим с такъв тип файлове.

1.2 Цел и задачи на разработката

Основните цели на нашата програма е тя да може:

- Да се прочете файла, като програмата трябва да го съхрани в подходяща структура от данни в паметта.
- Обекта да се изведе в четлив вид.
- Запазване на промените, извършени върху данните в текущия файл.
- Запазване на промените, извършени върху данните в нов файл по подадено име.
- Премахване на дадена колона (по име или номер).
- Промяна на реда на колоните - с подаване на нова пермутация на номерата на колоните. Например за таблица с три колони, пермутацията 2 3 1 означава, че първата колона ще стане последна.

В края на таблицата може да бъде добавен нов ред, който да представлява:

- Копие на ред с подаден номер;
- минималните стойности за всяка колона;
- Максималните стойности за всяка колона;
- най-често срещаните стойности за всяка колона. Ако има повече от една с еднаква честота, да се използва най-малката от тях.

Глава 2. Преглед на предметната област

2.1 Основни дефиниции, концепции и алгоритми

Класът `Commands` съдържа командния интерфейс, с който потребителя работи, когато иска да извършва действията върху файловете и таблиците. Създава се обект от тип `CsvCtroller`.

В `CsvController` клас е създадена цялата бизнес логика на проекта. Така чрез този обект ние извикваме всичките методи, които са достъпни и се извършва реализацията на командния интерфейс.



В проектът се създава абстрактен клас `Cell`, който бива наследяван от четири класа, които са типовете клетки, които можем да имаме в един `Csv` файл. Създаваме клас `Column`, който се състои от име, номер и вектор от `Cell*`, по този начин ние показваме какво ще съдържа една колона. След това създаваме клас `Table`, който всъщност е вектор от `Column`. По този начин се оформя нашата таблица и когато при отваряне на файл се опитваме да заредим данните. Всичко ще се зарежда точно в една такава таблица. В класа `CsvController` създаваме таблица от тип `Table` и когато извършваме действия, то ние работим именно с нея.

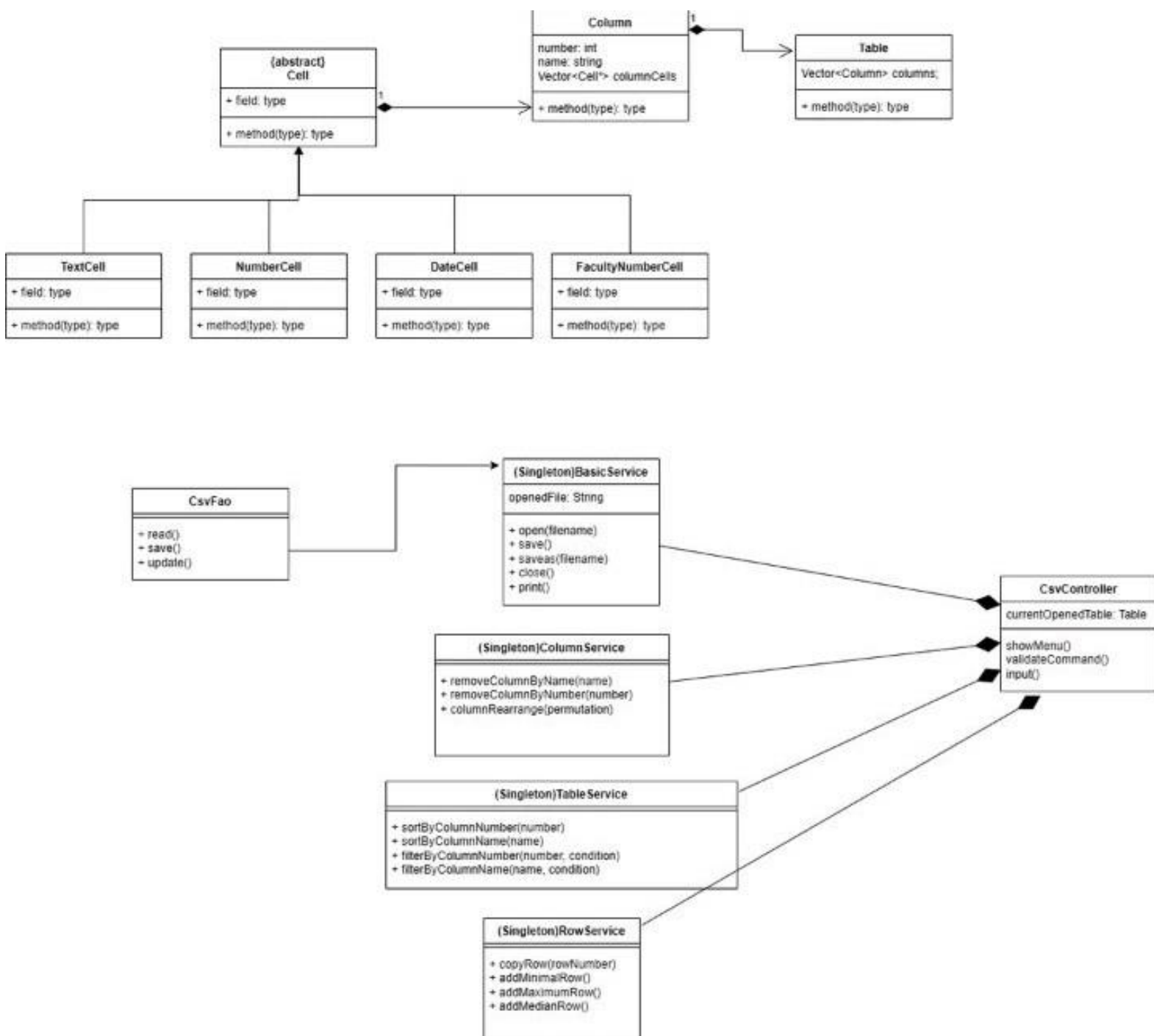
2.2 Основни проблеми, решения и определяне на сложност на задачата

Основните проблеми в проекта са: да се прочетат данните от файловете и да се запишат на обекта, да се извършват сложни операции от типа: намиране на най-често срещан елемент за всяка колона и създаване на нов ред като се взимат всички най-често срещани от съответната колона. Сходни операции изискват поне 2 вложени `for`-жикъла, които усложняват решението на задачата. Триенето на колони по дадено условие, създаване на опция за `undo`, също са част от основните проблеми.

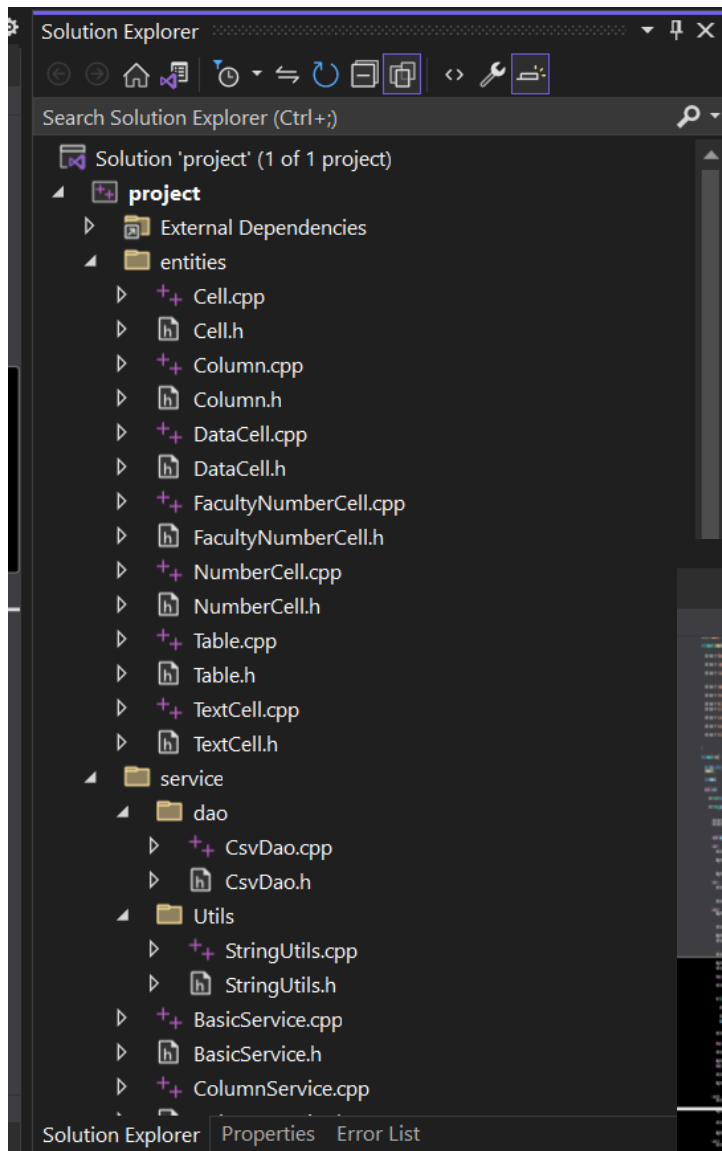
Глава 3. Обща архитектура

На фигура 1 се показва доста ясно каква е общата архитектура на проекта. За подобряването на четимостта на кода са създадени класовете `BasicService`, `ColumnService`, `TableService` и `RowService`. Реално в тях е описана бизнес логиката, а в `CsvController` ние правим обект от всеки един от тези класове и ги извикваме заедно с таблицата, която я подаваме по референция, за да може стойността и да се променя извиквайки методите на някои от тези класове. Програмата може да се напише и като всичката логика е само в главния клас, но това ще затрудни четимостта му и може да предизвика доста обърквания. Именно заради това е избрана тази архитектура в която те са разделени. Друга основна част е начина по, който се създава класа `Table`, но той е описан малко по-горе в документацията. Най-важните компоненти са абстрактния клас, вектора от поинтър от абстрактния клас (тъй като знаем, че обект от абстрактен клас не може да се създаде). И най-накрая създаване на вектор от тип `Column`.

Създаваме и един клас `csvDao`, в който единствено и само имаме работа с файл – четене и писане. Защото доста пъти се налага да ги използваме, при създаване на бизнес логиката и именно заради това ги отделяме, за да можем да ги извикваме и това да подобри ефективността на кода ни.

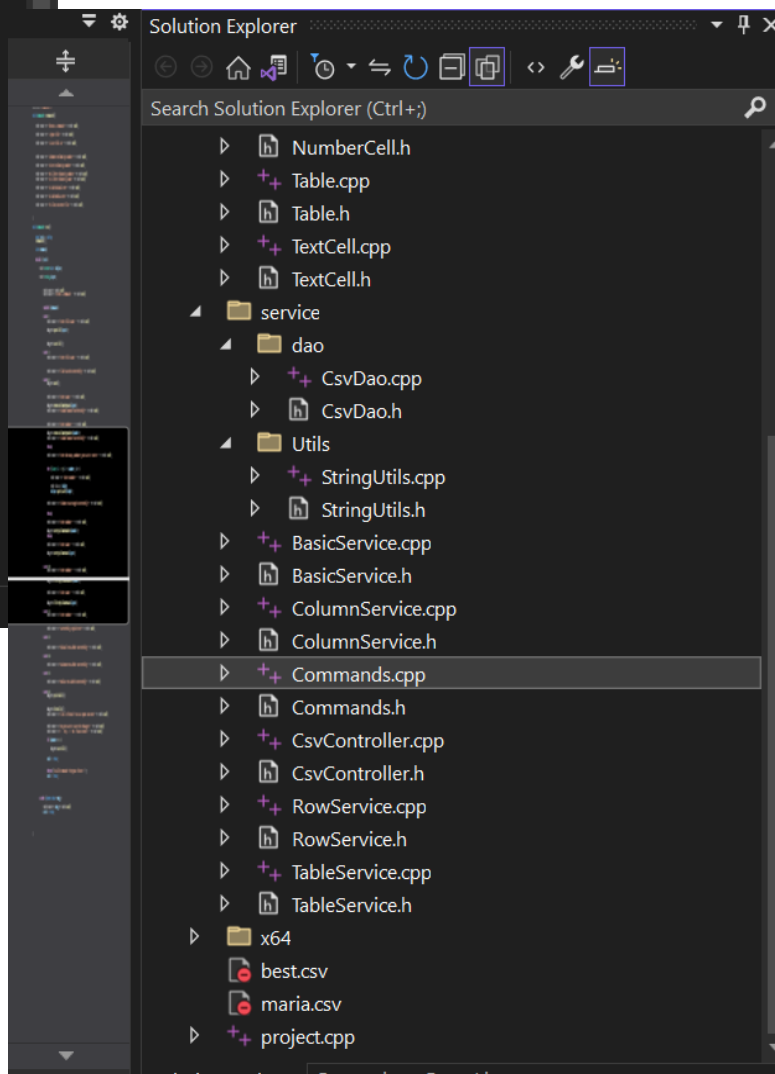


Фигура 1



Фигура 2

На фигури 1 и 2 се вижда се цялата структура на проекта. Това са .cpp и .h файловете, които описват по-горната схема. Има и папка Utils, в която имаме помощни материали, които ни позволяват да извличаме данните от файловете. Така нареченото parse по даден елемент и в случая това е запетайката.



Фигура 3



```
10
11 class CsvController {
12     private:
13         BasicService basicService;
14         ColumnService columnService;
15         RowService rowService;
16         TableService tableService;
17
18         Table previousState;
19         Table currentOpenTable;
20     public:
21
22         void printTable();
23         void setCurrentOpenTable(Table currentOpenTable);
24         Table& getCurrentOpenTable();
25
26         // Basic service
27         void openFile(std::string fileName);
28         void saveFile();
29         void saveFileAs(std::string fileName);
30         void undo();
31         void closeFile();
32
33         // Column service
34         void removeColumnByName(std::string name);
35         void removeColumnByNumber(size_t number);
36         void columnRearrange(std::vector<int> permutation);
37
38         // Table service
39         void sortByColumnNumber(size_t number);
40         void sortByColumnName(std::string& name);
41         void filterByColumnNumber(size_t number);
42         void filterByColumnName(std::string& name);
43
44         // Row service
45         void copyRow(int rowNumber);
46         void addMinimalRow();
47         void addMaximumRow();
48         void addMedianRow();
49     };
50
51 #endif
52
53
```

Представени са двата най-основни класа, CvsController (фигура 4), в който е цялата имплементация на бизнес логиката и класа Table (фигура 5), където реално се създава връзката вектора от Column и чрез този клас ние ще имаме достъп до колоните, възможност за добавяне на редова, за извличане на редовете. Да добавяме колонии и да ги извличаме.

Фигура 4

```
4 #include <vector>
5 #include "Column.h"
6
7 class Table {
8     private:
9         std::vector<Column> columns;
10
11     public:
12         Table();
13
14         Table(std::vector<Column> columns);
15
16         void addColumn(const Column& newColumn);
17
18         void addRow(std::vector<Cell*> data);
19
20         std::vector<Cell*> getRow(int number);
21
22         std::vector<Column>& getColumns();
23
24         void setColumns(const std::vector<Column>& columns);
25
26         Table& operator=(const Table& other);
27
28     };
29
30 #endif
31
32
33
```

Фигура 5



Глава 4. Заключение

Проекта позволява, четенето и писане в Csv файл. Позволява ни да работим с таблици и да извършваме множество операции върху тях. Като добавяне на ред по дадени критерии, триене на колона по дадени критерии, отпечатване на екрана. Създаване на нови файлове и промяна по старите. Този проект има много поле за бъдещето развитие като за начало ще се добавят множество от валидации за проверка на коректни данни, ще се осъществят сортировки и филтрации на данни и ще се създадат по-добри алгоритми и по-бързи такива, с които да подобрим ефективността на програмата

Използвана литература:

1. <https://www.w3schools.com/>
2. <https://www.geeksforgeeks.org/>
3. <https://9to5answer.com/>