**Документация по курсов проект – Електронни таблици**

[**https://github.com/Kiiara21**](https://github.com/Kiiara21)

Глава 1: Увод

1.1 Описание и идея на проекта:

Работа с електронни таблици. Съхранение на данни от различни типове в текстов файл и обработката им.

1.2 Цел и задачи на разработката:

Задачи:

• Да се създадат основни класове

• Да се реализират необходимите структури от данни и алгоритми

• Да се направят валидации на данните

• Да се записва информацията от програмата във файлове

• Да се зарежда запазената информация от файловете

• Да се напишат функционалности за таблицата и отделните редове

• Да се направи потребителски интерфейс

1.3 Структура на документацията:

Документацията е структурирана по следният начин:

• Увод

• Преглед на предметната област

• Проектиране

• Реализация, тестване

• Заключение

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

Проблемите, за които програмата е създадена да решава са:

Обработка на различни типове данни в таблица

2.3. Подходи, методи за решаване на поставените проблемите

Външни библиотеки, които съм използвала: vector, string, sstring, fstream, iostream,

За решаване на гореспоменатите проблеми, програмата използва множество функции и класове.

* Класът Cell – интерфейс на клетка

o Функция за клониране на клетка

o Функция за сериализация на клетка

o Функция за взимане на стойността като стринг

* Класовете DoubleCell, IntCell, StringCell и EmptyCell са наследници на Cell и реализират конкретният тип клетка
* Класът Row съдържа статична член-данна, която да брои редовете и вектор от указатели към клетки

o Функции за добавяне на различен тип клетки

o Функция за записване във файл

o Функция за сетване на елементи във вектор

o Оператори за достъп до клетки

o Функция за взимане на размера на реда (броя клетки)

* Класът Table съдържа вектор от указатели към редове

o Функция за добавяне на ред

o Функция за добавяне на празен ред с <брой> клетки

o Функция за принтиране на таблица

o Функция за взимане на размера на таблицата(брой редове)

o Оператори за достъп до елемент по индекс

o Функция за запис във файл

o Функция за четене от файл

o Функция за изчистване на таблица (изтриване на редовете и клетките)

* Класът IO – потребителски интерфейс
* Класът ISerialize – интерфейс за функциите за запис във файл и четене от файл
* Класът IPrintable – интерфейс за функцията за принтиране
* Класът Utils – съдържа помощни функции
* Класът Validate – съдържа функции за валидиране на данните

Глава 3: Проектиране

3.1. Обща архитектура

Основни класове които се използват в последствие за структурата на други класове

• Vector, Cell (DoubleCell, StringCell, IntCell, EmptyCell)

Следващите класове са подредени по начина в който се използват един в друг:

• Row

Table

Файлът main.cpp контролира входа и изхода(интерфейса на програмата

Глава 4: Реализация, тестване

Управление на паметта и алгоритми. Оптимизации.

Всеки клас е дефиниран със своите полета и функционалности. Всеки клас отговаря собственоръчно за управлението на паметта си. Всеки клас, който използва динамична памет в последствие я освобождава при приключването на програмата. Реализирани са тестове с 80% code coverage (във файла Tests.cpp !трябва да се пусне отделно от main.cpp!)

Глава 5. Заключение

5.1. Обобщение на изпълнението на началните цели

Програмата се справя с работа с файлове, поддържа използването на електронни таблици с всички нужни функционалности без подравняване на таблицата.

5.2. Насоки за бъдещо развитие и усъвършенстване

В бъдещи версии на програмата предстои да се оптимизират структурите използвани в програмата и алгоритмите, които те използват. Така че програмата да стане като прототип на известни други програми, създадени от известни компании.