**Технически университет – Варна**

Факултет по изчислителна техника и автоматизация

Катедра: ,,Софтуерни и интернет технологии”

Специалност: ,,Софтуерни и интернет технологии”

Тема: ,, Приложение за работа с електронни таблици”

Студент: Сезгин Бейханов Шабанов

Факултетен номер: 20621521

1

**Увод**

**Основна идея:** Основната идея на приложението е да позволи на потребителя работа с електронни таблици. Като основни операции, които може да бъдат извършени са:

* Зареждане на вече съществуваща таблица или създаване на нова такава.
* Принтиране на таблицата на конзолата.
* Редактиране на таблица.
* Работа с формули.
* Записване на направените промени обратно в същия файл.
* Записване на направените промени в нов файл.

**Цел и задачи на разработката:** Основната цел беше да се направи напълно функционално приложение, което да позволява на потребителите лесна и ефективна работа с електронни таблици.

Задачите, които бяха разработени са:

* Прочитане на данните от текстов файл и вкарването им в електрона таблица.
* Валидация на типа данни, с които беше указано в заданието на проекта, че таблицата трябва да работи.
* Меню чрез, което се дава информация на потребителя как и какви операции може да извършва.
* Имплементация на функционалността **print:** за принтиране на таблицата на конзолата.
* Имплементация на функционалността **save:** за запазване на направените промени в текущо отворения файл.
* Имплементация на функционалността **save as:** за запазване на направените промени в нов файл.
* Имплементация на функционалността **edit:** за редактиране на таблицата.
* Имплементация на функционалност, която позволява на потребителя да работи с формули.

**Структура на документацията:** Структурата на документацията е следната. Заглавната страница включва информация за студента, факултета и темата на проекта.

**Увод** – тук се описват основната идея, цели и задачи на проекта.

**Преглед на предметната област** – описват се основни дефиниции, концепции и алгоритми, както и подходи и методи на решаване на поставените проблеми.

**Проектиране** – обща структура на проекта, диаграми/блок-схеми.

**Реализация и тестване** – реализация на класове, алгоритми и оптимизации. Създаване на тестови сценарий.

**Заключение** – Обобщение, насоки за бъдещо развитие и усъвършенстване.

2

**Преглед на предметната област**

По време на разработка на проекта е обърнато внимание на това да се използват принципите на Обектно-Ориентираното Програмиране. В проекта за използвани принципи като: абстракция, полиморфизъм както и *SOLID* принципите. Обърнато е внимание на това проектът да е затворен за модификация и отворен за разширения, класовете и методите, са написани така, че да имат една единствена отговорност. Също така и на това всеки обект да може да се замени с бащиния тип без това да променя верността на програмата, както и на това интерфейсите да са по-малки по обем но повече на брой. И на последно място е приложен и принципът за това, че абстракциите не зависят на нищо, а конкретните класове зависят на абстракциите. Един от проблемите, които възникнаха по време на разработка беше валидацията на входните данни. За решаването на този проблем беше използван *regex*, за да може входните данни да са валидни така както е изискано да бъдат по условие. Друг проблем беше това, че входните данни не винаги идват по най-лесният начин за обработка. Този проблем беше решен като се правят няколко различни проверки, какъв е бил входа и според указаното в заданието как трябва да бъдат обработени данните, те биват обработени и представени в таблицата.

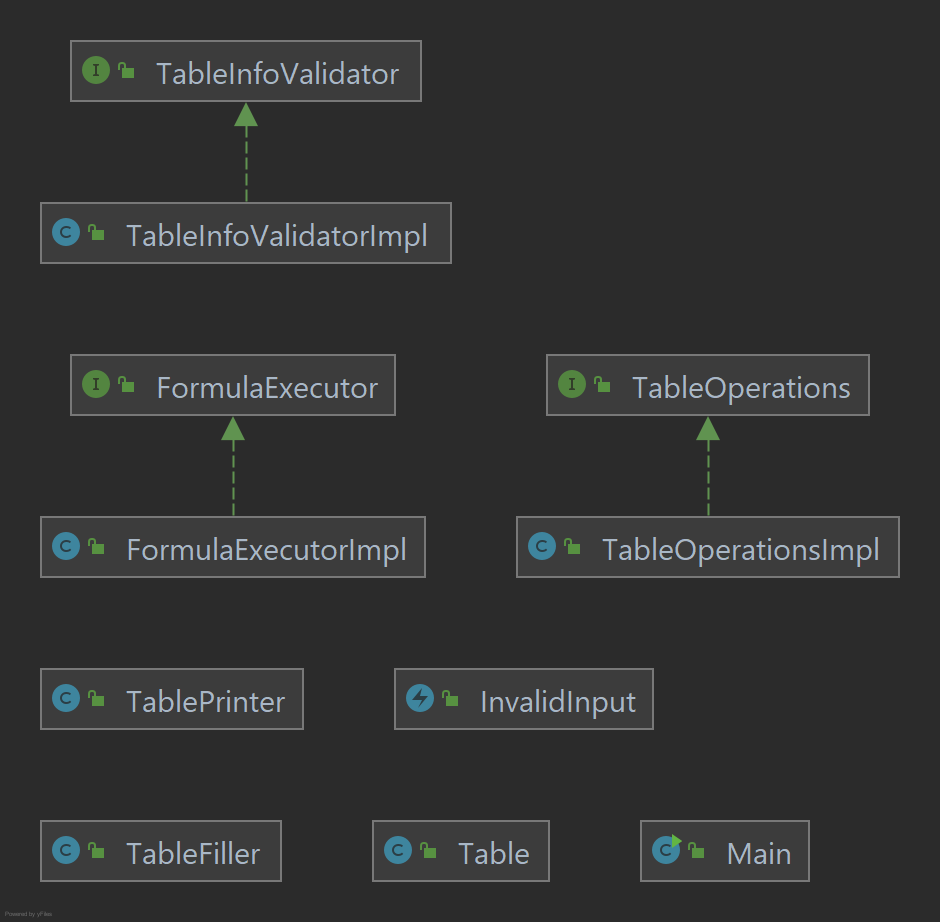
3

**Проектиране**

Проектът се състои от три пакета

1. **src –** тук се намират класовете и интерфейсите както и другите два пакета от които зависи проекта.
2. **exceptions** – пакет за грешки.
3. **files –** пакет в който се намират текстовите файлове с примерни входни данни.

**Диаграма на класовете и интерфейсите в проекта**

****

4

**Реализация и тестване**

**Main –** Стартова точка на програмата. В този клас се обработват командите, които потребителя въвежда, за да избере каква операция иска да извърши. След като бъде въведена някоя команда Main делегира самата бизнес логика на класа, който е отговорен за свършването **ѝ**. Тук е и информацията за това какви операции поддържа приложението (менюто).

**Table -** Това е класът, който използваме, за да репрезентираме електронна таблица в приложението. Той има две частни полета, като първото поле носи информация за това, кой е файлът от който е заредена информацията за таблицата, а второто е матрица от низове, която репрезентира редовете, клетките и данните в тези клетки в таблицата.

**TablePrinter** – Това е класът, който има за отговорност да принтира таблицата на конзолата по възможно най *user friendly* начин.

**TableInfoValidatorImpl** – имплементира интерфейса **TableInfoValidator**, който има един единствен метод, който трябва да валидира данните, които ще бъдат попълнени в таблицата. Това валидиране на данните го постигаме, като използваме *regex* и няколко проверки, за това дали данните са: низ, цяло число, дробно число или пък формула. В случай, че данните са валидни те биват вкарани в таблицата, а в противен случай бива хвърлена грешката **InvalidInput**, която описва на кой ред в кой файл се е получила тази грешка и защо данната е невалидна.

**TableFiller** – Този клас го използваме, за да записваме данни в таблицата.