**Star Wars Universe 0.1**

Проект на Георги Кънчев

Github: <https://github.com/Invadoge/OOP-2022/tree/main/Star%20Wars>

1. **Въведение в проекта**

Star Wars Universe цели да даде на потребителя възможност да създаде своя вселена по подобие на франчайза Star Wars. Потребителят може да прави планети и да добавя джедаи към тях, като определя тяхната сила, име, възраст и дори цвета на светлинния им меч.

За реализацията на тази концепцията се изисква намиране на начин за записване на създадените от потребителя планети и джедаи, както и за тяхното зареждане.

**Съдържание на документацията:**

Класове в реализацията(стр.1 – 2)  
Схема на включванията в header файловете(стр. 3)   
Реализация записване на данни(стр.3)  
Алтернативни имплементации(стр.3)  
Тестове(стр.4)  
Заключение и възможни подобрения(стр.4)

1. **Класове в реализацията**

**NamedClass** е базовият клас, който отговаря за имената на планетите и Джедаите. Също така помага да се сортират джедаите и планетите поименно, защото в него има оператор за сравнение.

**JediRank** служи за улесняване за добавянето на рангове на джедаи. Масивът в този клас складира всички рангове, които са имплементирани, за да може с лекота да се видят и при нужда, да се премахват или добавят.

**Jedi –** в този клас се складира цялата информация, необходима за джедай. Наследява NamedClass.

**SortedVector –** отговаря за подреждането на планетите и техните джедаи в азбучен ред. Има вградена функция за сортиране, която приема различни критерии за подреждане на елементите си.

**Planet –** Наследява NamedClass и има SortedVector от джедаи. Дава достъп до метода, който комбинира две планети и подрежда членовете им в азбучен ред.

**StarWars –** класът, отговорен за зареждането на цялата вселена и за взаимодействията на нейните елементи.

**Command –** клас, който чрез динамичен полиморфизъм позволява добавянето и изпълнението на различни команди. Методът doExecute(size\_t) приема число като стойност, за да може да се разглеждат различни стадии при изпълнение на програмата.

Invalid – изписва съобщение за грешка, когато не е въведена правилна команда.  
Help – дава списък с валидни команди за този момент. Приема като аргумент стадии на програмата, за да знае какви команди могат да се напишат.

Следните команди имат показател към класа StarWars, за да могат да го променят:

Open – отваря файл и зарежда данните му в StarWars

Close – премахва записаното име на файла за прочит и изчиства данните, записани в StarWars

Save и SaveAs – записват данните от StarWars в текущият файл или в нов файл

AddPlanet – добавя планета към StarWars

CreateJedi – добавя Jedi към Planet в StarWars

RemoveJedi – премахва Jedi от StarWars

PromoteOrDemoteJedi – повишава или понижава рангът и силата на конкретен Jedi от StarWars

GetStrongestJedi – изписва най-силният Jedi на дадена планета

GetYoungestJedi – ипзисва най-младият Jedi с конкретен ранг на дадена планета

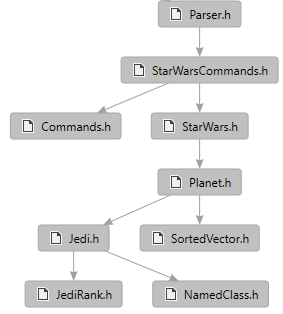
GetMostUsedSaber – изписва най-често използваният лазерен меч на дадена планета от джедаи с конкретен ранг

PrintJediOrPlanet – изписва информацията за планета или джедай

CombinePlanets – комбинира данните на две планети и я изписва

**CMDParser –** отговаря за прочита на командите

**UserInterface** – има като член данна CMDParser, чрез който чете и изпълнява команди, написани от потребителя.



Фиг.1 Схема на включванията в header файловете(.cpp файловете не са включени в схемата, но всеки клас има един без SortedVector, защото е шаблонен клас)

1. **Реализация записване на данни**

Понеже потребителят няма директен достъп до файловете, които съхраняват Star Wars вселената му, няма конкретни изисквания към това как ще се запазва информацията. В конкретната имплементация е избран да се съхранява във файлове с .txt разширение, като всяка отделна информация се записва на нов ред:



**Фиг 2**: Пример за съхраняването на данни

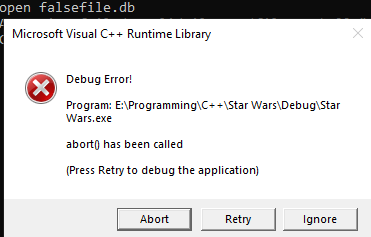
1. **Алтернативни имплементации**

В началото беше направен опит SortedVector да бъде реализиран чрез динамичен полиморфизъм, но поради нечетимостта на кода и трудността за неговата разработка, тази идея беше отхвърлена.

Алтернативен начин за принтиране на planet, е да се разшири функцията sort() на Planet, така че тя да приема булева функция, сравняваща Jedi. Така може sort да се извика от Starwarscommands, когато има нужда. Сегашното решение за принтирането на Planet е специално дефинирана функция, която принтира класа по различни начини. Причината за тази имплементация е запазването подредбата на елементите в класа Planet. До този момент не се позволява на външни функции да променят подредбата на класа.

1. **Тестове**

Показаните тук тестове не са напълно изчерпателни. Те служат за демонстрация на някои възможни ситуации при използването на програмата.



**Фиг. 3:** Тест за неправилно въведен файл.



**Фиг. 4:** Тест за неправилно въведен ранк.



**Фиг. 5:** Тест, когато няма намерена планета или джедай.



**Фиг. 6:** Несъществуваща планета при опит за създаване на джедай.

1. **Заключение и възможни подобрения**

Като цяло, Star Wars Universe 0.1 e прототип, който дава възможност да бъде разширен според бъдещите нужди на програмата. Предоставя онсовни средства за създаване на Star Wars вселена, но със сигурност има нужда от повече. Едно примерно допълнение ще е възможността да се изтрие планета, било то празна или населена с джедаи.