**ТУ – Варна**

**Проект по дисциплина ООП - част 1**

**Изготвил: Михаил Терзиев, ф.к. 20621672**

**Глава 1. Увод**

* 1. **Описание и идея на проекта**

Проекта играе роля на функционална система, която пресъздава вселена от поредицата Star Wars. За целта системата позволява създаването на планети и джедаи, които ги населяват. Могат дa бъдат извършвани определени операции, като обработване, принтиране и записване на информация за тях. Системата реализира и функционалност за работа с файлове.

* 1. **Цел и задачи на разработка**

Целта на проекта е да покаже работа с колекции, въвеждане и извличане на подаваните данни, както и промяна на записи. Разработката на системата включва работа с вход и изход, създаване на обекти, които могат да извършват действия, според изискваната функционалност.

**1.3. Структура на документация**

Документацията съдържа основни дефиниции на класовете, дефиниране на проблеми и тяхното решаване. Вкючени са схема на структурата на проекта и примери как се работи с потребителската система. Добавено е описание на методите за имплементираната функционалност. Завършва с извод и линк към кода на проекта.

**Глава 2. Преглед на предметната област**

**2.1. Основни дефиниции, концепции и алгоритми, които ще бъдат използвани**

Клас Jedi, който описва характеристиките на един джедай, а именно – име, ранг, възраст, цвят на меч, локация и сила.

Клас Planet, който представлява контейнер за обекти от класа Jedi, реализира населението на планета, която също има и атрибут име, както и незадължителен брой спътници луни, които се съхраняват също в контейнер. Наследява абстрактен клас Controller.

Клас Moon, описващ спътник (луна) на планета. Класа също има атрибут име и може да има и население от джедаи. Наследява абстрактен клас Controller.

Клас Controller, който имплементира основната функционалност за работа с гореописаните класове. Той съдържа концепции за работа с контейнери, алгоритми за търсене и сортиране, извличане и следене на записаната информация.

Клас FileManinpulator за работа с файлове. Записва и зарежда инстанция на Controller класа, съдържаща данните за планети, луни и джедаи.

**2.2. Дефиниране на проблеми и сложност на поставената задача**

- да се създават планети, луни и джедаи, които трябва да се добавят.

- да се премахва джедай от дадена планета.

- да се вдига и намаля ранг и сила на джедай.

- извличане на информация за планета – кои джедаи я населяват, сортирани по име и ранг, и всичките луни, заедно с тяхното население.

- извличане на данните за даден джедай.

- да се намира най-силен или най-млад джедай, според локация или ранг.

- да се изведе най-често срещан цвят на светлинен меч.

- да се изведе сумирана инфорамция за две планети.

**2.3. Подходи, методи (евентуално модели и стандарти) за решаване на поставените проблеми**

Реализирани методи:

* Public addPlanet <name> – добавя планета, ако не е добавена преди.
* Public addMoon <name> <planet\_name> - добавя луна към избрана планета, ако съществува.
* Private addJediToLocation <planet/moon> <jedi> - добавя джедай към колекция джедаи на планета или луна.
* Public createJedi <planet/moon> <name> <rank> <age> <saber\_color> <force> - създава джедай и извиква горния метод за добавяне към планетата.
* Public removeJedi <jedi\_name> <planet/moon> - премахва джедай от колекцията на избрана планета.
* Private getUpperRank <rank> - връща по-горен ранк от подадения.
* Public promoteJedi <jedi\_name> <multiplier> - промотира джедай като извиква горния метод и увеличава силата му с multiplier.
* Private getLowerRank <rank> - връща по-долен ранк на подадения.
* Public demoteJedi <jedi\_name> <multiplier> - демотира джадай като извиква горния метод и намалява силата му с multiplier.
* Public getStrongestJedi <planet\_name> - връща информация за джедая с най-голяма сила, намиращ се на подадената планета.
* Public getYoungestJedi <planet\_name> <jedi\_rank> - връща информация за най-младият джедай от съответния ранк.
* Public getMostUsedSaberColor <planet\_name> <jedi\_rank> - връща най-често срещания цвят на сабя по зададена планета, ако тя има джедаи от подадения ранк.
* Public getMostUsedSaberColor <planet\_name> - връща най-често срещания цвят на сабя по зададена планета, ако тя има джедаи от ранк GRAND\_MASTER.
* Public print <name> - връща информация за планета, луна или джедай.
* Private sortAlpha <jedies\_list> - сортира лист от джедаи по име възходящо.
* Public add <planet\_name> <planet\_name> - връща сумирана информация за джедаите, сортирани по име възходящо, намиращи се на двете планети.

**2.4 Потребителски и качествени изисквания**

- командита да се въвеждат от потребителят по посочения начин показан в точка 2.3.

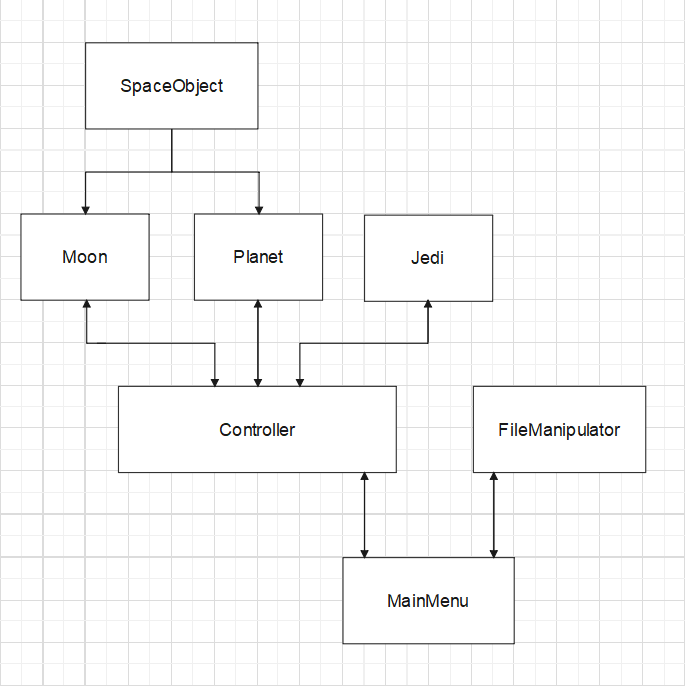
- имената на обектите и командите да се въвеждат правилно.

**Глава 3. Проектиране**

**3.1. Обща структура на проекта пакети който ще се реализират**

Проекта се съдържа в главен пакет StarWars, в който са реализирани класовете SpaceObject, Planet, Moon, Jedi, JediRanks, Controller, FileManipulator и MainMenu. Към пакета е добавена и основна директория за съдържане на файловете.

**3.2. Диаграми и Блок схеми**



**Фиг.1 Структура на проекта**

**Глава 4. Реализация и тестване**

**4.1. Реализация на класове.**

- Абстрактен клас SpaceObject – съсържа име и колекция за джедаите.

- Клас Planet – наследява SpaceObject, има колекция за луните.

- Клас Moon – наследява SpaceObject, има атрибут за планетата, която обикаля.

- Енум клас JediRanks – съдържа всички възможни рангове за джедай.

- Клас Jedi – съдържа име, възраст, ранк, локация, сила на джедай и цвят на меча.

- Клас FileManupulator – работа с файлове – open, close, save, saveas.

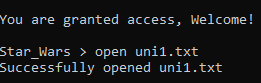
- Клас Controller – работа с класовете Moon, Planet, Jedi.

- Клас MainClass – меню за контол на операциите.

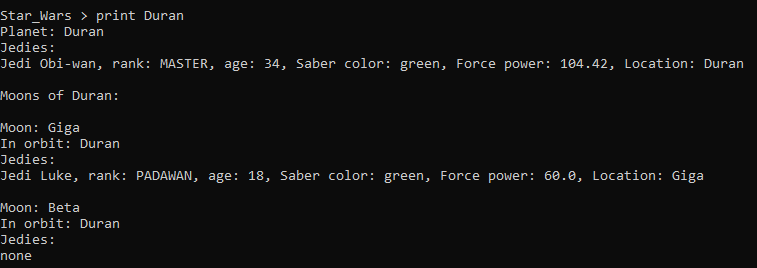
**4.2. Алгоритми и Оптимизации.**

Използвани са алгоритъм за сортиране на колекции и търсене на обекти в тях. Алгоритъмът за търсене е оптимизиран да извършва възможно най-малък брой итерации за намирането на желания обект в 2 нива на итериране. Съдържа операции за промяна на извлечената информация, добавяне и премахване.

**4.3. Примери**

****

**Фиг.2 Зареждане на файл**

****

**Фиг.3 Принтиране на информация за планета заредена от отворен файл**

****

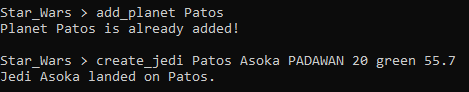
**Фиг.4 Вдигане ранка на джедай**

****

**Фиг.5 Принтиране на информация за джедай**

****

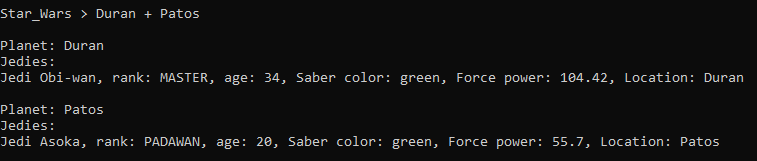
**Фиг.6 Понижаване ранка на джедай**

****

**Фиг.7 Добавяне на джедай към планета**

****

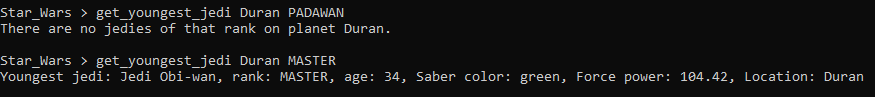
**Фиг.8 Добавяне на луна към планета**

****

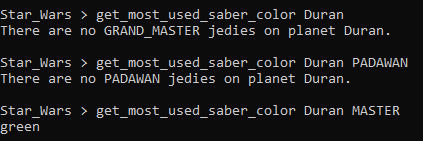
**Фиг.9 Принтиране на сумарна информация за две планети**

****

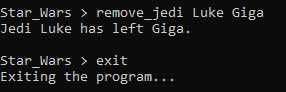
**Фиг.10 Извеждане на най-силен джедай за избрана планета**

****

**Фиг.11 Извеждане на най-млад джедай за избрана планета**

****

**Фиг.11 Връщане на най-често срещан цвят на светлинен меч**

****

**Фиг.12 Премахване на джедай и излизане от програмата**

**Глава 5. Заключение**

В заключение проекта реализира описаните в условието задачи за работа с дадените обекти и техните функционалности. Възможни са подобрения по работата с файлове и добавяне на повече методи за взаимодействие между системата и потребителя, както и смяна на подход или алгоритми при имплементиране на логиката.

Линк към Github: <https://github.com/MihailTerziev/Java_OOP_Project>