Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

Вариант 43

Выполнил:

Козаченко Данил Александрович

Группа P3112

Проверил:

Карасева Мария Александровна

Задание

**Описание предметной области, по которой должна быть построена объектная модель:**

Не забыл я также вознестись к небу благодарной душой. Да и мог ли я не проникнуться благодарностью к тому, кто столь чудесным образом охранял меня в пустыне и не дал мне погибнуть в безотрадном одиночестве? И кого мог я благодарить за свое избавление, как не того, кто источник всех благ, всякого утешения и отрады? Когда мы немного успокоились, капитан сказал мне, что привез мне кое чего подкрепиться из корабельных запасов, которых еще не успели расхитить негодяи, так долго хозяйничавшие на корабле. Вслед затем он крякнул матросам, сидевшим в лодке, выгрузить на берег тюки, предназначенные для губернатора. Их было столько, что могло показаться, будто я вовсе не собираюсь уезжать с ним, а остаюсь на острове до конца моих дней.

**Этапы выполнения работы:**

1. Получить вариант
2. Нарисовать UML-диаграмму, представляющую классы и интерфейсы объектной модели и их взаимосвязи;
3. Придумать сценарий, содержащий действия персонажей, аналогичные приведенным в исходном тексте;
4. Согласовать диаграмму классов и сценарий с преподавателем;
5. Написать программу на языке Java, реализующую разработанные объектную модель и сценарий взаимодействия и изменения состояния объектов. При запуске программа должна проигрывать сценарий и выводить в стандартный вывод текст, отражающий изменение состояния объектов, приблизительно напоминающий исходный текст полученного отрывка.
6. Продемонстрировать выполнение программы на сервере helios.
7. Ответить на контрольные вопросы и выполнить дополнительное задание.

Текст, выводящийся в результате выполнения программы не обязан дословно повторять текст, полученный в исходном задании. Также не обязательно реализовывать грамматическое согласование форм и падежей слов выводимого текста.

Стоит отметить, что цель разработки объектной модели состоит не в выводе текста, а в эмуляции объектов предметной области, а именно их состояния (поля) и поведения (методы). Методы в разработанных классах должны изменять состояние объектов, а выводимый текст должен являться побочным эффектом, отражающим эти изменения.

**Требования к объектной модели, сценарию и программе:**

1. В модели должны быть представлены основные персонажи и предметы, описанные в исходном тексте. Они должны иметь необходимые атрибуты и характеристики (состояние) и уметь выполнять свойственные им действия (поведение), а также должны образовывать корректную иерархию наследования классов.
2. Объектная модель должна реализовывать основные принципе ООП - инкапсуляцию, наследование и полиморфизм. Модель должна соответствовать принципам SOLID, быть расширяемой без глобального изменения структуры модели.
3. Сценарий должен быть вариативным, то есть при изменении начальных характеристик персонажей, предметов или окружающей среды, их действия могут изменяться и отклоняться от базового сценария, приведенного в исходном тексте. Кроме того, сценарий должен поддерживать элементы случайности (при генерации персонажей, при задании исходного состояния, при выполнении методов).
4. Объектная модель должна содержать как минимум один корректно использованный элемент каждого типа из списка:
   * абстрактный класс как минимум с одним абстрактным методом;
   * интерфейс;
   * перечисление (enum);
   * запись (record);
   * массив или ArrayList для хранения однотипных объектов;
   * проверяемое исключение.
5. В созданных классах основных персонажей и предметов должны быть корректно переопределены методы equals(), hashCode() и toString(). Для классов-исключений необходимо переопределить метод getMessage().
6. Созданные в программе классы-исключения должны быть использованы и обработаны. Кроме того, должно быть использовано и обработано хотя бы одно unchecked исключение (можно свое, можно из стандартной библиотеки).
7. При необходимости можно добавить внутренние, локальные и анонимные классы.

Диаграмма классов реализованной объектной модели

Представлена на Рисунке 1

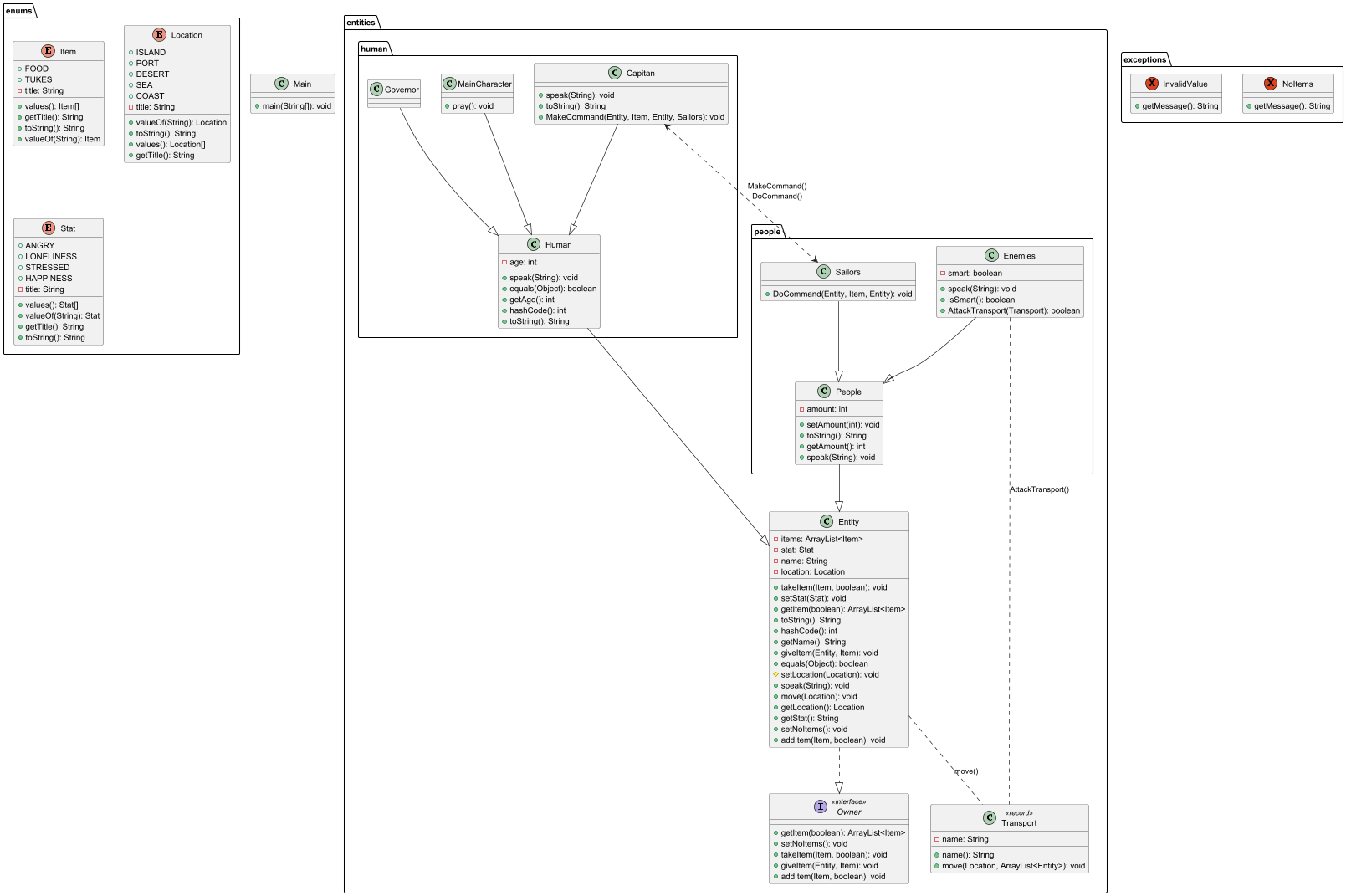


Рисунок 1

Исходный код

<https://github.com/Dkanil/ITMO/tree/main/semester1/programming/lab3>

Результат работы программы

Гена произнёс: Я так устал испытывать Одиночество, прошу, избавь меня от этого чувства

Гена произнёс: Всевышний услышал мои молитвы и избавил меня от Одиночество

\*\*Гена теперь испытывает Счастье\*\*

\*\*Гена теперь в Побережье\*\*

Гена произнёс: Джек Воробей, я так голоден и хочу добраться до дома, прошу Вас помочь мне

\*\*Джек Воробей передал Еда к Гена\*\*

Благодаря Корабль:

\*\*Джек Воробей теперь в Остров\*\*

\*\*Гена теперь в Остров\*\*

\*\*Матросы теперь в Остров\*\*

О нет! Аборигены захватили Корабль!

Аборигены промычали: У-А-Ы-УНГА-БУНГА

Аборигены оказались слишком глупы не поняли, как управлять Корабль.

Корабль освобождён!

Благодаря Корабль:

\*\*Джек Воробей теперь в Порт\*\*

\*\*Гена теперь в Порт\*\*

\*\*Матросы теперь в Порт\*\*

Капитан Джек Воробей произнёс: Йо-хо-хо! Матросы, выполнить команду передать мои Тюки к Губернатор!

Матросы произнесли: Так точно, капитан!

\*\*Джек Воробей передал Тюки к Матросы\*\*

\*\*Матросы передал Тюки к Губернатор\*\*

Матросы произнесли: Приказ успешно выполнен, капитан!

Гена произнёс: Груза было настолько много, что казалось, будто я вовсе не собираюсь уезжать с капитаном, а остаюсь на острове до конца моих дней

Вывод

Во время выполнения лабораторной работы я научился работать с абстрактными классами, интерфейсами и перечислениями. Лучше освоил работу с полями, методами и классами. Научился строить сложные UML диаграммы.