

Федеральное государственное автономное образовательное
учреждение высшего образования „Национальный исследовательский
университет ИТМО“

Системное и прикладное программное обеспечение

Лабораторная работа №3-4
Вариант 8708

Выполнил:
Зуев Даниил Андреевич, Р3111

Санкт-Петербург 2025

Содержание

Задание.....	3
UML-диаграмма.....	4
Исходный код программы.....	5
Заключение.....	6

Задание

Лабораторная работа #3-4

В соответствии с выданным вариантом на основе предложенного текстового отрывка из литературного произведения создать объектную модель реального или воображаемого мира, описываемого данным текстом. Должны быть выделены основные персонажи и предметы со свойственным им состоянием и поведением. На основе модели написать программу на языке Java.

Введите вариант:

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

Однажды ранним утром, когда мы бедствовали таким образом - ветер все еще не сдавал - один из матросов крикнул: "Земля!", но не успели мы выскочить из каюты в надежде узнать, где мы находимся, как судно село на мель. В тот же миг от внезапной остановки вода хлынула на палубу с такой силой, что мы уже считали себя погибшими: стремглав бросились мы вниз в закрытые помещения, где и укрылись от брызг и пены. Тому, кто не бывал в подобном положении, трудно дать представление, до какого отчаяния мы дошли. Мы не знали, где мы находимся, к какой земле нас прибило, остров это или материк, обитаемая земля или нет. А так как буря продолжала бушевать, хоть и с меньшей силой, мы не надеялись даже, что наше судно продержится несколько минут, не разбившись в щепки; разве только каким нибудь чудом ветер вдруг переменится. Словом, мы сидели, глядя друг на друга и ежеминутно ожидая смерти, и каждый готовился к переходу в иной мир, ибо в здешнем мире нам уже нечего было делать. Единственным нашим утешением было то, что, вопреки всем ожиданиям, судно было все еще цело, и капитан оказал, что ветер начинает стихать.

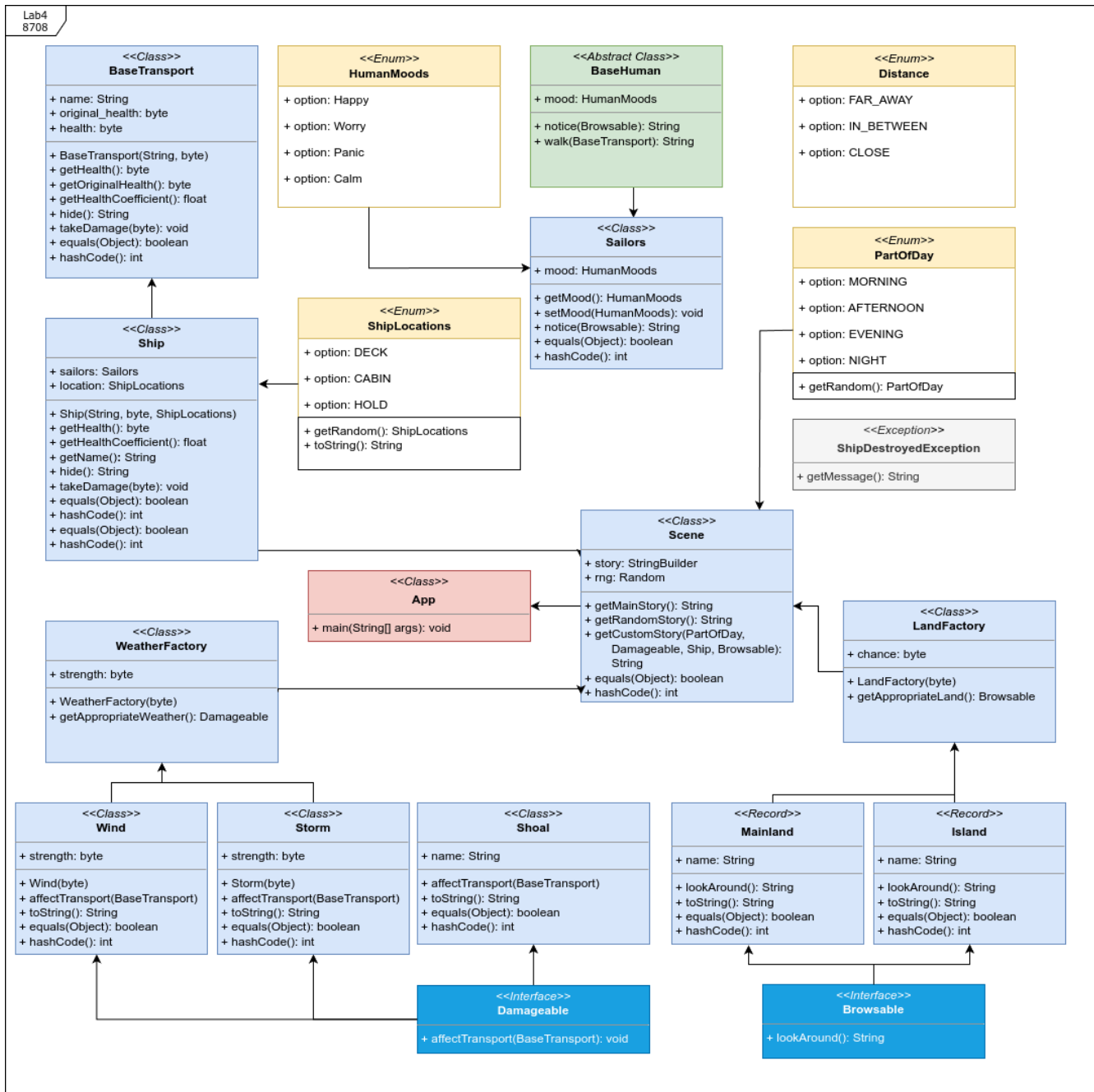
Требования к объектной модели, сценарию и программе:

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 принципы 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.

Содержание отчёта по работе:

1. Текст задания.
2. Диаграмма классов объектной модели.
3. Исходный код программы (можно в виде ссылки на репозиторий).
4. Результат работы программы.
5. Выводы по работе.

UML-диаграмма



Исходный код программы



[Ссылка на репозиторий.](#)

Заключение

Во время выполнения лабораторной работы я изучил принципы ООП SOLID; построил объектную модель по описанию предметной области, учитывая принципы SOLID; создал диаграмму UML для структуры проекта; до конца разобрался с интерфейсами и абстрактными классами, классом Object и его методами.