

Санкт-Петербургский Национальный Исследовательский Университет
ИТМО
Факультет Программной Инженерии и Компьютерной Техники



Вариант №408300
Лабораторная работа №3-4
По дисциплине
Программирование

Выполнил студент группы Р3212:
Бондарев Алексей Михайлович

Преподаватель:
Гаврилов Антон Валерьевич

Санкт-Петербург 2024 г.

1. Текст задания

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, выводимый 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 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.

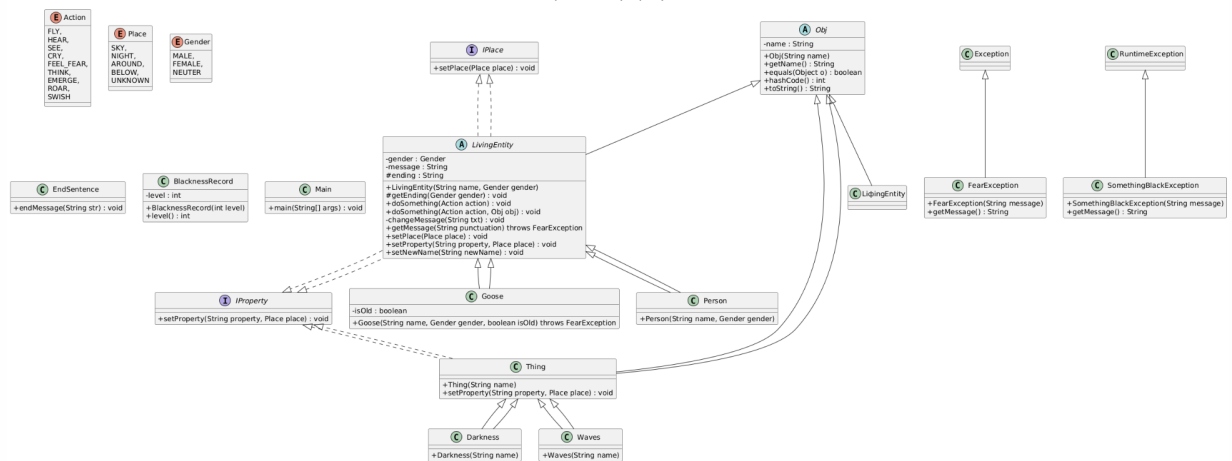
Содержание отчёта по работе:

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.

Вопросы к защите лабораторной работы:

1. Принципы объектно-ориентированного программирования SOLID и STUPID.
2. Класс Object. Реализация его методов по умолчанию.
3. Простое и множественное наследование. Особенности реализации наследования в Java.
4. Понятие абстрактного класса. Модификатор abstract.
5. Понятие интерфейса. Реализация интерфейсов в Java. Отличие интерфейсов от абстрактных классов.
6. Модификаторы default, static и private для методов интерфейса.
7. Перечисляемый тип данных (enum) в Java. Особенности реализации и использования.
8. Тип запись (record) в Java. Особенности использования.
9. Методы и поля с модификаторами static и final.
10. Перегрузка и переопределение методов.
11. Обработка исключительных ситуаций, три типа исключений.
12. Стандартный массив и динамический массив (ArrayList). Основные различия.
13. Вложенные, локальные и анонимные классы.

2. Диаграмма классов объектной модели.



3. Исходный код программы

<https://github.com/666Daredevil666/javaproga.git>

(нужна ветка master)

4. Результат работы программы.

===== Начинается сценарий =====

Акка испугался в ночи летит . Гусь №1 испугался в ночи летит . Гусь №2 испугался в ночи летит .

.

Акка видита слышита , Гусь №1 видит слышит , Гусь №2 видит слышит , Акка кричит .

Нильс подумал .

не могут (UNKNOWN) грохочето грохочето свистито вниз, быстро стущающаяся (UNKNOWN) не может быть чернее (AROUND) появился ещё чернее, чем небо (UNKNOWN) .

===== Сценарий завершён =====

5. Вывод

Во время выполнения данной лабораторной работы мною были изучены приемы работы с различными структурами данных в Java и файлами, а также освоены знания о ООП в Java, параметризованных типах, wildcard-параметрах и утилитах javadoc.