

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Лабораторная работа №3-4

по дисциплине «Программирование»

Студент:

Лабин Макар Андреевич

Преподаватель:

Наумова Надежда Александровна

г. Санкт-Петербург, 2024

Текст задания

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:

Доктор Пилюлькин сказал, что это ничего, что Пончик скоро оправится от потрясения и заговорит по-прежнему, как все нормальные коротышки. Надо только дать ему немного покушать, тогда он войдет в свою обычную колею. Так и на самом деле случилось. Пончика усадили за стол, поставили перед ним тарелку борща и тарелку каши. Пончик быстро уписал все это и тут же начал рассказывать о том, что произошло с ним: и о том, как они вместе с Незнайкой залезли тайком в ракету и отправились на Луну; как путешествовали по Луне и попали в пещеру; как Незнайка провалился в подлунный мир, после чего Пончик остался совсем один; как он сидел в ракете, пока не прикончил все пищевые запасы, после чего тоже провалился в подлунный мир и попал в город Лос-Паганос, где принялся торговать солью; как сначала разбогател, потом разорился, потом стал работать на чертовом колесе и сделался членом Общества свободных крутильщиков.

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.

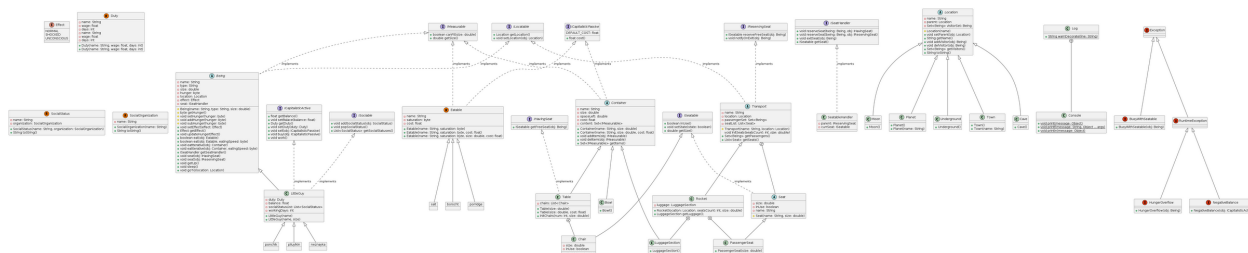
Текст, выводящийся в результате выполнения программы не обязан дословно повторять текст, полученный в исходном задании. Также не обязательно реализовывать грамматическое согласование форм и падежей слов выводимого текста.

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 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.

Диаграмма классов объектной модели



Исходный код программы

С исходным кодом программы можно ознакомиться в репозитории GitHub по ссылке: https://github.com/MrDvD/itmo_labs/tree/master/programming/lab3/lab.

Результат работы программы

С результатом работы программы можно ознакомиться в репозитории GitHub по ссылке: https://github.com/MrDvD/itmo_labs/blob/master/programming/lab3/file.out.

Выводы по работе

В ходе выполнения лабораторной работы я познакомился с принципами SOLID и STUPID кода. В процессе проектирования объектной модели я постарался учесть все советы и спроектировал примитивную модель, которая позволит эмулировать поведение объектов, напоминающее ситуацию, представленную в приведённом отрывке художественного произведения. При реализации данной модели нужно было учитывать особенности использования интерфейсов, модификаторов доступа, абстракций. На завершающем этапе работы были внедрены инструменты логирования и обработки исключений, которые позволят представить в текстовом виде взаимодействия между объектами.