Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

ИТМО»

Лабораторная работа
№3-4

по дисциплине “Программирование”

Студент:

Новиков Даниил Дмитриевич, P3131

Преподаватель:

Бобрусь Александр Владимирович

Санкт-Петербург

2024

Задание

Введите вариант: 9012

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

> Как я ожидал, так и вышло: лишь только начался прилив, дикари сели в лодки и отчалили. Я забыл сказать, что за час или за полтора до отъезда они плясали на берегу: я ясно различал в трубку их странные телодвижения и прыжки. Я видел также, что все они были нагишом, но были ли то мужчины или женщины - не мог разобрать. Как только они отчалили, я спустился с горы, вскинул на плечи оба свои ружья, заткнул за пояс два пистолета, тесак без ножен и, не теряя времени, отправился к тому холму, откуда открыл первые признаки этих людей. Добравшись туда (что заняло не менее двух часов времени, так как я был навьючен тяжелым оружием и не мог итти скоро), я взглянул в сторону моря и увидел еще три лодки с дикарями, направлявшиеся от острова к материку.

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.

UML диаграмма

https://github.com/Buratishkin/ITMO/tree/main/prog/lab3-4/lab3-4.pdf

Исходный код программы

https://github.com/Buratishkin/ITMO/tree/main/prog/lab3-4/src

Результат работы программы

https://github.com/Buratishkin/ITMO/tree/main/prog/lab3-4/test.txt

Выводы к работе

В этой работе я:

- Изучил и применил record, exception, interface, abstract class, enums в работе
- Изучил принципы ООП: SOLID и STUPID
- Узнал, что такое ArrayList
- Понял, как работает множественное наследование