**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Программирование**

**Лабораторная работа №3-4**

**Вариант 5514**

Выполнил: Михайлов Петр Сергеевич

Группа: Р3111

Преподаватель: Письмак Алексей Евгеньевич

Санкт-Петербург 2024г.

Содержание

[Текст задания 3](#_Toc184637618)

[Диаграмма классов реализованной объектной модели 5](#_Toc184637619)

[Исходный код программы 6](#_Toc184637620)

[Результат работы программы 7](#_Toc184637621)

[Выводы по работе 8](#_Toc184637622)

# Текст задания

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.

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

Я исходил весь остров, отыскивая самые глухие места, и, наконец, выбрал один уголок, так хорошо укрытый от нескромных взоров, что лучше нельзя было и желать. Это была небольшая полянка в низине, в чаще леса - того самого леса, где я заблудился, когда возвращался домой с восточной части острова. Вся полянка занимала около трех акров; лес обступал ее со всех сторон почти сплошной стеной, образуя как бы естественную ограду; во всяком случае, устройство ограды потребовало от меня гораздо меньше труда, чем в других местах. Я немедленно принялся за работу, и недели через четыре мой новый загон был огорожен настолько плотно, что можно было перевести в него коз. Теперь это не представляло большого труда, так как новые поколения коз, вырешенные в огороженных загонах, привыкли ко мне и утратили свою природную дикость. Я, не откладывая, отделил от стада десять коз и двух козлов и перевел их в новый загон. После того я употребил еще некоторое время на окончательное укрепление изгороди, но делал это не торопясь, очень медленно.

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.

1. В созданных классах основных персонажей и предметов должны быть корректно переопределены методы equals(), hashCode() и toString(). Для классов-исключений необходимо переопределить метод getMessage().
2. 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).
3. 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.

# Диаграмма классов реализованной объектной модели

Вывод UML-диаграммы находится по ссылке в репозитории: <https://clck.ru/3F7rSR>

# Исходный код программы

Исходный код программы находится по ссылке в репозитории: <https://clck.ru/3F7reg>

# Результат работы программы

Вывод программы находится log-файле по ссылке в репозитории: <https://clck.ru/3F7rhB>

# Выводы по работе

Во время выполнения данной лабораторной работы, я изучил синтаксис PlantUML для создания UML-диаграмм, узнал про абстрактные классы и Enum, научился переопределять методы equals(), hashCode(), toString() и разобрался с принципами SOLID.