Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение

Дисциплина «Программирование»

Отчёт по лабораторной работе №3

Вариант №110555

Выполнил

Галак Екатерина Анатольевна

P3115

Проверил

Кулинич Ярослав Вадимович

Санкт – Петербург, 2024

Содержание

Задание

Вариант №110555

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:

```
К тому времени и все остальные коротышки разделились, если можно так
выразиться, по интересам. Помимо шарашников, здесь были карусельщики,
колесисты, чехардисты, киношники, картежники и козлисты. Нетрудно
догадаться, что карусельщиками называли тех коротышек, которые по целым
дням вертелись на каруселях; колесистами -- тех, что предпочитали
вертеться на чертовом колесе. Чехардисты, естественно, были те, которые
не признавали ничего, кроме игры в чехарду. Козлисты день-деньской сидели
за столиками и изо всех сил стучали костяшками домино, играя в "козла".
Картежники, расположившись партиями по четыре, сидели на травке и играли
в карты, преимущественно в подкидного дурака. Наконец, киношники с утра и
до ночи сидели в кинотеатре и сеанс за сеансом смотрели различные
кинофильмы. Нечего, конечно, и говорить, что такое однообразие в занятиях
притупляло умственные способности коротышек, исподволь подготовляя
переход их в животное состояние. Считалось, между прочим, что смотрение
кинофильмов является более интеллектуальным, то есть более полезным для
ума занятием, нежели игра в шарашки или в "козла". Это, однако, ошибка,
так как содержание фильмов было слишком бессмысленным, чтобы давать
какую-нибудь пищу для ума. Глядя изо дня в день, как герои всех этих
кинокартин бегали, прыгали, падали, кувыркались и палили из пистолетов,
можно было лишь поглупеть, но ни в коем случае не поумнеть.
```

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.

Диаграмма классов, реализованной объектной модели

Диаграмма классов, реализованной объектной модели представлена в репозитории на Github:

https://github.com/KatyaGalak/ITMO-University/blob/main/First%20Year/Fall%20Semester/Programming/Lab3/scheme/Scheme.png

Исходный код программы

Исходный код программы представлен в репозитории на Github:

https://github.com/KatyaGalak/ITMO-University/tree/main/First%20Year/Fall%20Semester/Programming/Lab3/prog_lab34

Результат работы программы

```
К тому времени все коротышки разделились, если можно так выразиться, по интересам. Здесь были Шарашники, Карусельщики, Колесисты, Чехардисты, Доминошники, Картежники, Киношники. Шарашники занимались различными изобретиниями и научными экспериментами. Карусельщики целыми днями вертелись на каруселях. Колесисты вертелись на чертовом колесе. Чехардисты не признавали ничего, кроме игры в чехарду. Доминошники день-деньской сидели за столиками и стучали костяшками домино, в особенности в козёл. Картежники день-деньской располагались на траве и играли в карты, преимущественно в подкидной дурак в группах по 4 коротышки. Киношники с утра до вечера сидели в кинотеатре и смотрели различные фильмы. Нечего, конечно, и говорить, что такое однообразие в занятиях притупляло умственные способности коротышек, исподволь подготавливая переход их в животное состояние. Считалось, между прочим, что Киношники занимаются более интеллектуальным занятием, чем Шарашники, Доминошники. Это, однако, ошибка, так как содержание фильмов было слишком бессмысленным, чтобы давать какую-нибудь пищу для ума. Киношники, глядя как герои кинокартин бегали, прыгали, палили из пистолета, падали, кувыркались, могли лишь поглупеть, но ни в коем случае не поумнеть.
```

Вывод

В процессе выполнения лабораторной работы я познакомилась с принципами SOLID, научилась использовать абстрактные классы, интерфейсы, record, enum, исключения, выполнять парсинг конфигурационного файла в формате json с использованием библиотеки Gson, парсинг аргументов командной строки для кода на Java, писать тесты для проверки корректности исключений, а также использовать Gradle для сборки проекта.