



Вариант №76160  
Лабораторная работа №3  
По дисциплине  
Программирование

Выполнил студент группы Р3110:  
Ахроров Кароматуллохон Фирдавсович

Преподаватель:  
Гаврилов Антон Владимирович  
Мустафаева Айнур Вугар Кызы

## 1. 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:**

- Потому что я не могла приносить тебе другого корма, пока ты жил на своей скале, - отвечала ему Акка. - Но не горюй, из тебя все-таки выйдет хорошая птица. Осенью, когда гуси двинулись в далекий путь на юг, Горго полетел с ними. Но он никак не мог научиться держать строй во время полета. То и дело он улетап далеко вперед, потом снова возвращался и кружился над стаей.

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

- Потому что я не могла приносить тебе другого корма, пока ты жил на своей скале, - отвечала ему Акка. - Но не горюй, из тебя все-таки выйдет хорошая птица. Осенью, когда гуси двинулись в далекий путь на юг, Горго полетел с ними. Но он никак не мог научиться держать строй во время полета. То и дело он улетап далеко вперед, потом снова возвращался и кружился над стаей.

**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.

## 2. Исходный код программы.

Репозиторий:

[https://github.com/Ahrorovk/programming\\_lab\\_34](https://github.com/Ahrorovk/programming_lab_34)

## 3. Диаграмма классов реализованной объектной модели.

Вывод в UML-формате см. в репозитории.

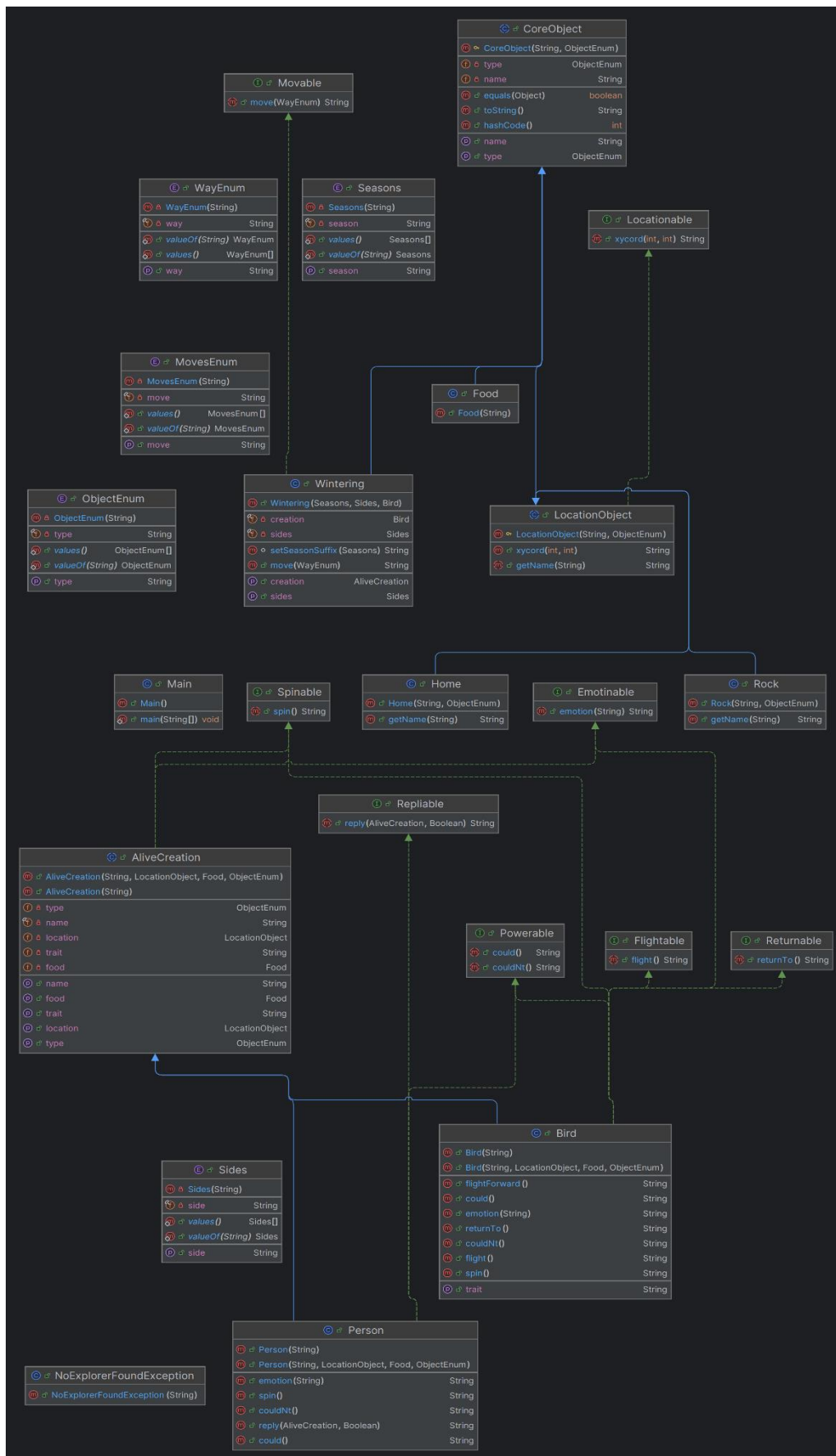


Рисунок 2. Диаграмма

#### 4. Результат работы программы:

См. в репозитории: [https://github.com/Ahrorovk/programming\\_lab\\_34/logs/results.log](https://github.com/Ahrorovk/programming_lab_34/logs/results.log)

## **5. Вывод**

Во время выполнения данной лабораторной работы я улучшил и укрепил свои знания языка программирования Java, использовал систему сборки Gradle Kts, научился пользоваться инструментом для генерации UML-диаграмм PlantUML, закрепил свои знания об SOLID и повторил их на практике.