

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «Санкт-Петербургский национальный исследовательский
университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 8765433

Выполнил:

Студент группы Р3116

Билошицкий Михаил Владимирович

Преподаватель:

Письмак Алексей Евгеньевич

Санкт-Петербург, 2022

Содержание

Задание.....	3
Исходный код программы	4
Результат работы программы	5
UML диаграмма классов	6
Вывод	7

Задание

Лабораторная работа #3

Введите вариант: 8765433

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

Все засмеялись, а Ворчун сказал: Воздушный шар поднялся еще выше, и весь Цветочный город был виден как на ладони. Дома казались совсем крошечными, а коротышек уж и совсем нельзя было разглядеть. Воздушный шар относил ветром, и скоро весь город виднелся далеко позади.

Программа должна удовлетворять следующим требованиям:

1. Доработанная модель должна соответствовать принципам SOLID.
2. Программа должна содержать как минимум два интерфейса и один абстрактный класс (номенклатура должна быть согласована с преподавателем).
3. В разработанных классах должны быть переопределены методы `equals()`, `toString()` и `hashCode()`.
4. Программа должна содержать как минимум один перечисляемый тип (`enum`).

Порядок выполнения работы:

1. Доработать объектную модель приложения.
2. Перерисовать диаграмму классов в соответствии с внесёнными в модель изменениями.
3. Согласовать с преподавателем изменения, внесённые в модель.
4. Модифицировать программу в соответствии с внесёнными в модель изменениями.

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Принципы объектно-ориентированного программирования SOLID и STUPID.
2. Класс `Object`. Реализация его методов по умолчанию.
3. Особенности реализации наследования в Java. Простое и множественное наследование.
4. Понятие абстрактного класса. Модификатор `abstract`.
5. Понятие интерфейса. Реализация интерфейсов в Java, методы по умолчанию. Отличия от абстрактных классов.
6. Перечисляемый тип данных (`enum`) в Java. Особенности реализации и использования.
7. Методы и поля с модификаторами `static` и `final`.
8. Перегрузка и переопределение методов. Коварианты возвращаемых типов данных.
9. Элементы функционального программирования в синтаксисе Java. Функциональные интерфейсы, лямбда-выражения. Ссылки на методы.

Исходный код программы

Ссылка на GitHub с исходным кодом:

https://github.com/michael-bill/labs_programming_itmo/tree/main/Laba3

Результат работы программы

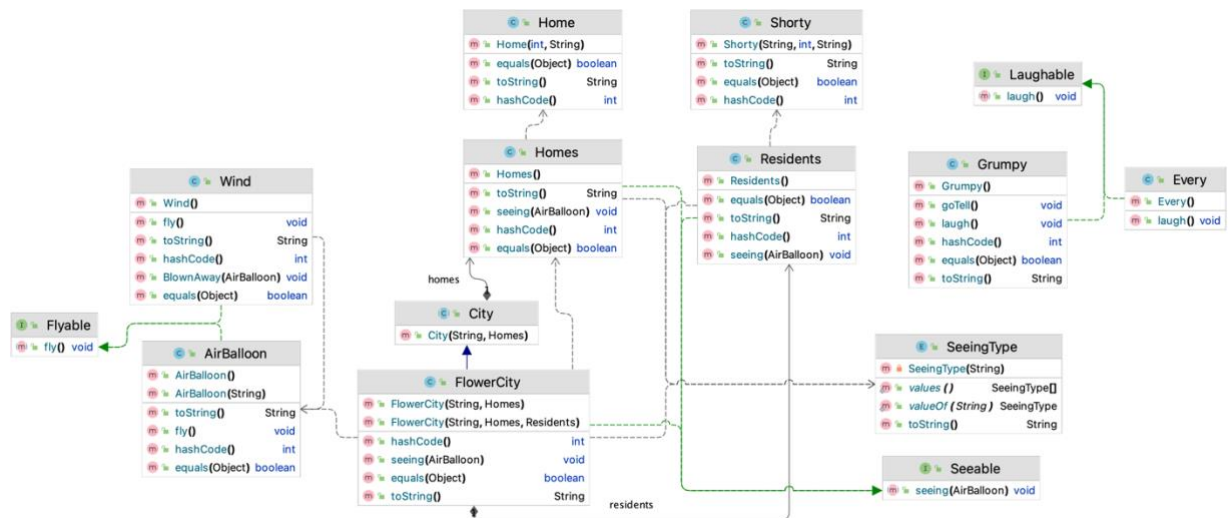
Через Helios:

```
[s367101@helios ~/programming]$ java -jar Laba3.jar  
Все засмеялись, а Ворчун сказал: Воздушный шар поднялся еще выше, и весь Цветочный  
город был виден как на ладони. Дома казались совсем крошечными, а коротышек уж и  
совсем нельзя было разглядеть. Воздушный шар относил ветром, и скоро весь город в  
иднелся далеко позади.
```

Текст:

Все засмеялись, а Ворчун сказал: Воздушный шар поднялся еще выше, и весь Цветочный город был виден как на ладони. Дома казались совсем крошечными, а коротышек уж и совсем нельзя было разглядеть. Воздушный шар относил ветром, и скоро весь город виднелся далеко позади.

UML диаграмма классов



Чтобы ознакомиться с ней поближе, ссылка на .png файл также прикреплена в GitHub репозитории.

https://github.com/michael-bill/labs_programming_itmo/blob/main/Laba3/YML%20Diagram.png

Вывод

Во время выполнения работы я ознакомился с абстрактными классами, интерфейсами и перечисляемым типом данных Enum в Java. Научился работать с ними и на их основе строить объектные модели с соблюдением принципов SOLID. Полученные знания понадобятся нам в процессе дальнейшего обучения.