



**Вариант №84222**  
**Лабораторная работа №3**  
**по дисциплине**  
***’Программирование’***

**Выполнил:**  
**Студент группы Р3113**  
***Крутько Никита : 242570***  
**Преподаватель:**  
***Письмак Алексей Евгеньевич***

**Санкт-Петербург 2019 г.**

## Содержание

<b>1</b>	<b>Задание</b>	<b>2</b>
1.1	Программа должна удовлетворять следующим требованиям: . . . . .	2
1.2	Порядок выполнения работы: . . . . .	2
1.3	Текст задания. . . . .	2
1.4	Принципы объектно-ориентированного программирования SOLID и STUPID.	2
1.5	Задание: . . . . .	3
<b>2</b>	<b>Исходный код</b>	<b>3</b>
<b>3</b>	<b>Вывод</b>	<b>3</b>

## 1. Задание

### 1.1. 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).

### 1.2. 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.
5. Отчёт по работе должен содержать:

### 1.3. Текст задания.

1. Диаграмма классов объектной модели.
2. Исходный код программы.
3. Результат работы программы.
4. Выводы по работе.
5. Вопросы к защите лабораторной работы:

#### 1.4. Принципы объектно-ориентированного программирования SOLID и STUPID.

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

#### 1.5. Задание:

*Но лунатики знали, что вечно так продолжаться не может, что со временем воздух вокруг Луны совсем рассеется, отчего поверхность Луны, не защищенная значительным слоем воздуха, будет сильно прогреваться солнечными лучами и на Луне даже под стеклянным колпаком невозможно будет существовать. Вот поэтому-то лунатики стали переселяться внутрь Луны и теперь живут не с наружной, а с внутренней ее стороны, так как на самом деле Луна внутри пустая, вроде резинового мяча, и на внутренней ее поверхности можно так же прекрасно жить, как и на внешней. Эта Знайкина книжка наделала много шума. Все коротышки с увлечением читали ее. Многие ученые хвалили эту книжку за то, что она интересно написана, но все же высказывали недовольство тем, что она научно не обоснована. А действительный член академии астрономических наук профессор Звездочкин, которому тоже случилось прочитать Знайкину книжку, просто кипел от негодования и говорил, что книга эта – вовсе не книга, а какая-то, как он выразился, чертова чепуха. Этот профессор Звездочкин был не то чтобы какой-нибудь очень сердитый субъект. Нет, он был довольно добрый коротышка, но очень, как бы это сказать, требовательный, непримиримый. Во всяком деле он ценил больше всего точность, порядок и терпеть не мог никаких фантазий, то есть выдумок.*

## 2. Исходный код

[github.com/KrutNA/itmo-programming/tree/lab-3/](https://github.com/KrutNA/itmo-programming/tree/lab-3/)

## 3. Вывод

В ходе лабораторной работы я познакомился с интерфейсами, абстрактными классами и енами. Попытался в dependency injection, не разобрался и сделал без него :с.