

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ  
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ  
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ  
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3-4  
курса «Программирование»**

**по теме: «Принципы объектно-ориентированного  
программирования SOLID и STUPID»**

**Вариант № 3109007**

Выполнил студент:  
Бых Даниил Максимович  
группа: Р3109

Преподаватель:  
Гаврилов А. В.,  
Наумова Н. А.

**ИТМО**

Санкт-Петербург, 2025 г.

# Содержание

<b>Лабораторная работа № 3-4.</b>	<b>2</b>
1. Задание варианта № 3109007 . . . . .	2
2. Выполнение задания. . . . .	5
1. Листинги кода . . . . .	5
3. Результат работы программы. . . . .	7
4. Вывод . . . . .	8

# Лабораторная работа № 3-4

## 1. Задание варианта № 3109007

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:**

Хотя ракета мчалась со страшной скоростью, покрывая пространство в двенадцать километров в одну секунду, Незнайке казалось, что она застыла на месте и ни на полпальца не приближается к Луне. Это объяснялось тем, что расстояние от Земли до Луны очень большое – около четырехсот тысяч километров. При таком огромном расстоянии скорость двенадцать километров в секунду не так велика, чтобы ее можно было заметить на глаз, да еще находясь в ракете. Прошло два или три часа, а Незнайка все смотрел на Луну и никак не мог от нее оторваться. Луна словно притягивала к себе его взоры. Наконец он почувствовал какое-то мучительное посасывание в животе и только тогда сообразил, что наступила пора обедать. Он поскорей спустился в пищевой отсек и увидел, что Пончик проснулся и уже что-то жует с аппетитом.

**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 разработаные объектную модель и сценарий взаимодействия и изменения состоя-

ния объектов. При запуске программы 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 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. Выполнение задания.

Задание было выполнено в редакторе кода Visual Studio Code, собрано в jar файл lab3.jar и загружено в Git репозиторий на GitHub.

### 2. 1. Листинги кода

Точка входу в программу из файла [1.1](#)

```
1 import model.TaskStory;
2 import model.exceptions.NotUniqueCompartmentsException;
3
4
5 public class Main {
6     public static void main(String[] args) {
7         try {
8             TaskStory story = new TaskStory();
9             story.tell();
10        } catch(NotUniqueCompartmentsException e) {
11            System.out.println(e.getMessage());
12            System.exit(1);
13        }
14    }
15}
```

Листинг 1.1: Исходный код главного класса программы

UML-диаграмма классов [1.1](#)

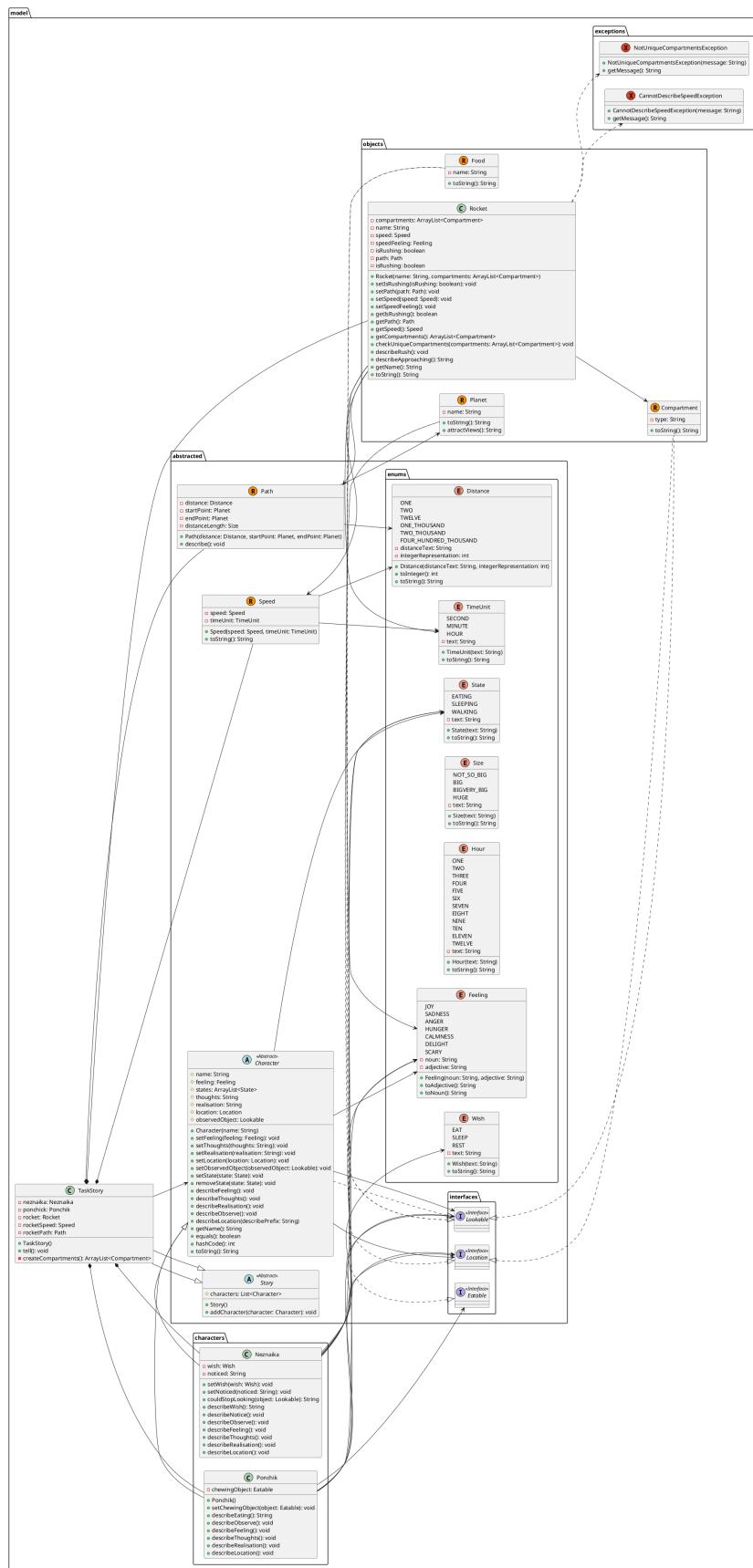


Рис. 1.1: UML диаграмма

### 3. Результат работы программы.

```
1 ракета мчалась со страшный скоростью двенадцать километров в сек
2 Незнайка думал что ракета ни на пол пальца не приближается к Луна
3 расстояние от Земля до Луна огромное -- около четыреста тысяч километров
4 прошло два часа или три часа
5 Незнайка смотрел на Луна и не мог оторваться
6 Луна словно притягивала к себе взоры
7 Незнайка почувствовал голод
8 Незнайка сообразил что настало время поесть
9 Незнайка спустился в пищевой отсек
10 Незнайка увидел что Пончик уже проснулся и Пончик ест чтото-
11
```

Листинг 1.2: Резульат выполнения программы

## **4. Вывод**

Во время выполнения лабораторной работы я изучил основные принципы объектно-ориентированного программирования в Java. Освоил работу с наследованием, абстрактными классами, интерфейсами, изучил особенности использования различных модификаторов доступа и методов. Рассмотрел принципы SOLID, STUPID, особенности работы с коллекциями, системой обработки исключений и современными возможностями языка, такими как enum и record. Полученные знания создают надежную базу для дальнейшего изучения языка программирования и разработки программных решений.

# Литература

- [1] Ссылка на личный репозиторий GitHub: <https://github.com/DaniilRen/ITMO-labs/tree/main/Java/Lab3>
- [2] Ссылка на сайт с информацией о покемонах: <https://pokemondb.net>
- [3] Ссылка на документацию по jar библиотеке с покемонами: <https://se.ifmo.ru/~tony/doc/>