



Вариант №1545346
Лабораторная работа №3
По дисциплине
Программирование

Выполнил студент группы Р3115:
Барсуков Максим

Преподаватель:
Сорокин Роман Борисович
Письмак Алексей Евгеньевич

1. Текст задания

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

Одним прыжком Знайка достиг беседки и заглянул внутрь. Винтика и там не было. Коротышки принялись тянуть веревку и притянули Знайку обратно к дому. Знайка мгновенно вскарабкался по водосточной трубе на крышу и уже хотел оглядеться по сторонам, но налетевший неожиданно порыв ветра сдул его с крыши и понес в сторону. Это не испугало Знайку, так как он знал, что коротышки в любой момент могут притянуть его на веревке обратно.

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.

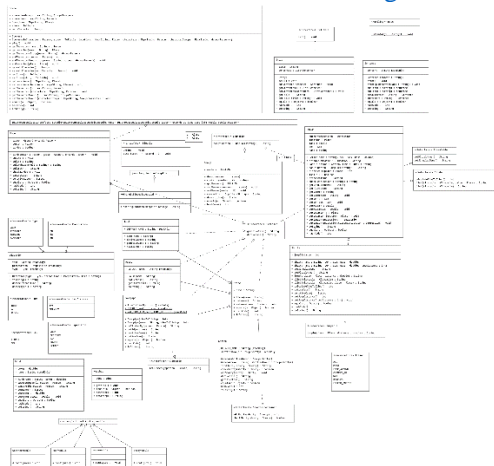
2. Исходный код программы.

Репозиторий:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab3>

3. Диаграмма классов реализованной объектной модели.

(<https://github.com/maxbarsukov/itmo/blob/master/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab3/docs/diagrams/uml.png>)



4. Результат работы программы:

```
Starting a Gradle Daemon (subsequent builds will be faster)
```

```
> Task :app:run
```

Одним прыжком Знайка достиг беседки и заглянул внутрь.

Винтика и там не было.

Коротышки принялись тянуть веревку и притянули Знайку обратно к дому.

Знайка мгновенно вскарабкался по водосточной трубе на крышу и уже хотел оглядеться по сторонам, но налетевший неожиданно порыв ветра сдул его с крыши и понес в сторону.

Это не испугало Знайку, так как он знал, что коротышки в любой момент могут притянуть его на веревке обратно.

```
BUILD SUCCESSFUL in 10s
```

```
2 actionable tasks: 2 executed
```

5. Вывод

Во время выполнения данной лабораторной работы я научился декомпозировать программу на классы и интерфейсы, узнал, что такое Dependency Injection, поработал с библиотекой Google Guice, улучшил свои навыки составления UML диаграмм.

Мем в конце отчёта (текстом):

> If you have a problem and decide to use Java... Now you have a Problem, ProblemImpl, ProblemException, and a ProblemFactory.