

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ  
ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА № 3**  
по дисциплине  
«ПРОГРАММИРОВАНИЕ»

Вариант № 31261

**Выполнил:**  
Студент группы Р3125  
Михайлов Павел  
Максимович  
**Преподаватель:** Максимов Андрей  
Николаевич

г. Санкт-Петербург, 2022

# СОДЕРЖАНИЕ

Задание .....	3
Диаграмма классов .....	4
Ссылка на репозиторий .....	5
Вывод .....	6

# ЗАДАНИЕ

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

Как только Винтик и Шпунтик очутились в лесу со своим вездеходом, на строительную площадку непрерывным потоком начали поступать бревна, брусья, доски, планки, рейки, штакетник и другие пиломатериалы. Нечего, конечно, и говорить, что вся работа на строительстве велась в условиях невесомости, что очень облегчало труд коротышек и ускоряло работу. Увидев, что Винтик и Шпунтик завалили пиломатериалами чуть ли не всю стройплощадку. Знайка велел им прекратить пока это дело и заняться починкой испорченных лунатиками приборов. Сам Знайка вместе с Фуксией и Селедочкой были заняты исследованием свойств лунита и антилунита. Заменяя в приборе невесомости кристаллы лунита, они обнаружили, что величина зоны невесомости находится в прямой зависимости от величины кристалла: чем больше был кристалл, тем больше была и зона. Поместив кристалл лунита между полюсами подковообразного магнита, Фуксия обнаружила, что зона невесомости перестала распространяться во все стороны, а распространяется лишь в одном направлении, на манер светового луча.

## 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.

```

classDiagram
    class Passenger {
        <<interface>>
        +search(Rover)
        +comeOut()
    }
    class Rover {
        +passengers: List<Passenger>
        +location: Location
        +hp: int
        +setInitialNumber(int)
        +addPassenger(Passenger)
        +removePassenger(Passenger)
        +goToLocation(Location)
        +Reverse(int)
    }
    class Shpuntik {
        +fixDevice(Device)
        +search(Rover rover)
        +clutterSite(Lumber, Lumber)
        +comeOut()
        +Vinsnik(String, Location)
        +Vinsnik()
        +Vinsnik(String, Location)
    }
    class Vinsnik {
        +fixDevice(Device)
        +search(Rover rover)
        +clutterSite(Lumber, Lumber)
        +comeOut()
        +Vinsnik(String, Location)
        +Vinsnik()
        +Vinsnik(String, Location)
    }
    class Person {
        +location: Location
        +age: int
        +nativeGalaxy: String
        +Person(String, Location)
        +goToLocation(Location)
        +Person()
        +Person(String, Location)
    }
    class Lunatic {
        +crashDevices(Device...)
        +Lunatic()
        +Lunatic(String, Location)
    }
    class Znayka {
        +research()
        +orderToFix(Device, Person...)
        +replaceCrystal(LunarCrystal, Device)
        +changeCrystal()
        +replaceCrystal(LunarCrystal, Device)
        +detectCrystalDependence(LunarCrystal)
        +Znayka(int, String, Location)
        +orderToStop(Person...)
        +setSearchOnSite(Person...)
        +replaceCrystal(LunarCrystal, Device)
        +Znayka()
        +Znayka(String, Location)
    }
    class Seledochka {
        +research()
        +setCrystal(LunarCrystal, Device)
        +replaceCrystal(LunarCrystal, Device)
        +Seledochka(int, String, Location)
        +detectCrystalDependence(LunarCrystal)
        +Seledochka(String, Location)
    }
    class Fuchsia {
        +research()
        +setCrystal(LunarCrystal, Device)
        +replaceCrystal(LunarCrystal, Device)
        +detectCrystalDependence(LunarCrystal)
        +Fuchsia(int, String, Location)
        +Fuchsia(String, Location)
    }
    class LunarCrystal {
        +size: double
        +color: Colors
        +thermalConductivity: ThermalConductivity
        +density: Density
        +depositDepth: DepositDepth
        +getSize() double
        +LunarCrystal(Colors, ThermalConductivity, Density, DepositDepth)
        +LunarCrystal(double)
        +LunarCrystal(double, Colors, ThermalConductivity, Density, DepositDepth)
        +getColor() Colors
        +getDensity() Density
        +getThermalConductivity() ThermalConductivity
        +getDepositDepth() DepositDepth
    }
    class Location {
        +weightless: boolean
        +x: int
        +y: int
        +setWeightless(Boollean)
        +getWeightless() Boollean
    }
    class StorageLocation {
        +isOfPeople: Person
        +isOfDevices: Device
        +numberList: Lumber
        +huliy: boolean
        +addPeople() List<Person>
        +removePeople(Person)
        +getIsOfPeople() List<Person>
        +addDevice(Device)
        +removeDevice(Device)
        +getDevice(double) Device
        +getIsFullness() boolean
        +addLumber(Lumber)
    }
    class Lunit {
        +setWeightless(Zone)
        +setWeightlessRay()
        +Lunit(Colors, ThermalConductivity, Density, DepositDepth)
        +Lunit(Colors, ThermalConductivity, Density, DepositDepth)
        +Lunit(double)
    }
    class AntLunit {
        +AntLunit(double, Colors, ThermalConductivity, Density, DepositDepth)
        +AntLunit(Colors, ThermalConductivity, Density, DepositDepth)
        +AntLunit(double)
    }
    class Magnet {
        +setCrystal(LunarCrystal)
        +Magnet(int)
    }
    class WeightlessDevice {
        +setCrystal(LunarCrystal)
        +WeightlessDevice(int)
    }
    class Device {
        +LunarCrystal: LunarCrystal
        +serviceability: boolean
        +serialNumber: int
        +setCrystal(LunarCrystal)
        +getCrystal(LunarCrystal) LunarCrystal
        +fixDevice()
        +breakDevice()
    }
    class Lumber {
        <<enumeration>>
        LOGS
        BEAMS
        FLANKS
        SLATS
        LATHS
        BATTENS
    }
    class Colors {
        <<enumeration>>
        RED
        BLUE
        GREEN
        GRAY
        BLUSH
        VARIABLE
    }
    class Density {
        <<enumeration>>
        LOW
        MEDIUM
        HIGH
        VARIABLE
    }
    class DepositDepth {
        <<enumeration>>
        DEEP
        SURFACE
        MOUNTAIN
        VARIABLE
    }
    class ThermalConductivity {
        <<enumeration>>
        LOW
        HIGH
        MEDIUM
        VARIABLE
    }
    class Zone {
        <<enumeration>>
        ZONE
    }

    Passenger <|-- Vinsnik
    Passenger <|-- Shpuntik
    Rover --> Passenger
    Rover --> Person
    Rover --> Lunatic
    Rover --> Znayka
    Rover --> Seledochka
    Rover --> Fuchsia
    Person --> Lunatic
    Person --> Znayka
    Person --> Seledochka
    Person --> Fuchsia
    Lunatic --> Znayka
    Lunatic --> Seledochka
    Lunatic --> Fuchsia
    Znayka --> Seledochka
    Znayka --> Fuchsia
    Seledochka --> Fuchsia
    LunarCrystal --> Location
    LunarCrystal --> StorageLocation
    LunarCrystal --> Lunit
    LunarCrystal --> AntLunit
    Location --> StorageLocation
    Location --> Lunit
    Location --> AntLunit
    StorageLocation --> Lunit
    StorageLocation --> AntLunit
    Lunit --> Magnet
    Lunit --> WeightlessDevice
    AntLunit --> Magnet
    AntLunit --> WeightlessDevice
    Magnet --> Device
    WeightlessDevice --> Device
    
```

## ССЫЛКА НА РЕПОЗИТОРИЙ

Исходный код и результат выполнения: [https://github.com/lubitelkvokk/lab3\\_proga](https://github.com/lubitelkvokk/lab3_proga)

## **ВЫВОД**

В ходе выполнения лабораторной работы я ознакомился с SOLID принципами, интерфейсам и абстрактными классами, перечисляемым типом enum, методами hashCode, equals, toString.