**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**Лабораторная работа №3.**

ФИО студента, вариант: Захаров Егор Вячеславович, 322043

Направление подготовки(специальность): 09.03.04

Группа: P3119

ФИО преподавателя: Письмак Алексей Евгеньевич

**Санкт-Петербург, 2021**

Оглавление

[UML таблица 2](#_Toc89254546)

[Исходный код программы 3](#_Toc89254547)

[Вывод 6](#_Toc89254548)

Текс задания.

ВАРИАНТ 322043

Эта сцена произвела неизгладимое впечатление на телезрителей, а в

особенности на полицейских, которые смотрели в тот день передачу. Многие

из них впервые поняли, что теперь наконец настала пора, когда нельзя уже

безнаказанно хвататься за оружие и палить из него в кого попало. Всем

стало ясно, что по-прежнему жить скоро будет нельзя.

Нечего и говорить, что полицейские боялись теперь и близко подходить

к ракете, а не то что стрелять возле нее. Деревенские жители могли

беспрепятственно приходить к космонавтам и получать у них семена

гигантских растений. Теперь гигантские семена сажали не только в деревне

Нееловке, но и в селе Голопяткине, Бесхлебове, Голодаевке, Непролазном и

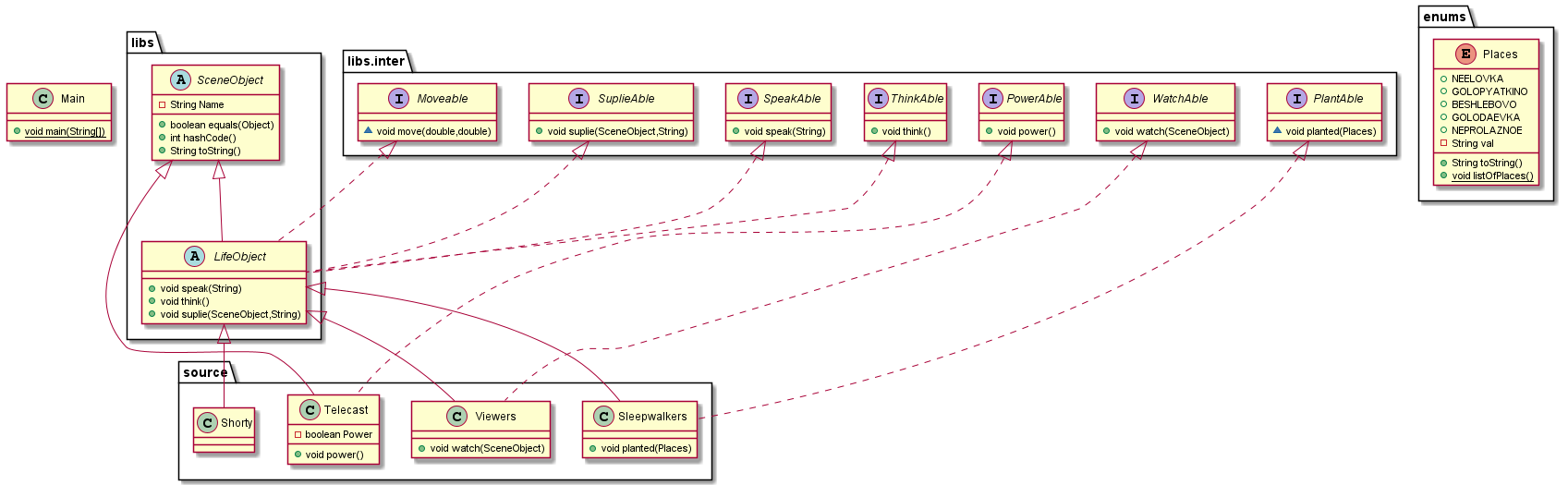
во многих других. Знайка распорядился, чтоб лунатикам давали не только

нужные им семена, но снабжали их приборами невесомости, а также

антилунитом и объясняли им, как всем этим пользоваться, чтоб защититься от

полицейских.

# UML таблица



# Исходный код программы

Places.java

package enums;

import java.util.Arrays;

public enum Places {

    NEELOVKA("деревне Нееловке"),

    GOLOPYATKINO("селе Голопяткине"),

    BESHLEBOVO("селе Бесхлебове"),

    GOLODAEVKA("селе Голодаевке"),

    NEPROLAZNOE("селе Непролазном");

    private String val;

    private Places(String s){

        val = s;

    }

    public String toString(){

        return val;

    }

    public static void listOfPlaces(){

        System.out.println(Arrays.toString(Places.values()));

    }

}

MoveAble.java

package libs.inter;

public interface MoveAble {

    default void move(double x, double y){

        if (x>0){

            System.out.println(x);

        } else {

            if (x==0){

                System.out.println(0);

            } else {

                System.out.println(-x);

            }

        }

        if (y>0){

            System.out.println(y);

        } else {

            if (y==0){

                System.out.println(0);

            } else {

                System.out.println(-y);}

        }

    };

}

PlantAble.java

package libs.inter;

import enums.Places;

public interface PlantAble{

    void planted(Places place);

}

PowerAble.java

package libs.inter;

public interface PowerAble {

    public void power();

}

SpeakAble.java

package libs.inter;

public interface SpeakAble {

    public void speak(String phrase);

}

SuplieAble.java

package libs.inter;

import libs.SceneObject;

public interface SuplieAble {

    public void suplie(SceneObject obj, String name);

}

ThinkAble.java

package libs.inter;

public interface ThinkAble {

    public void think();

}

WatchAble.java

package libs.inter;

import libs.SceneObject;

public interface WatchAble {

    public void watch(SceneObject obj);

}

LifeObject.java

package libs;

import libs.inter.MoveAble;

import libs.inter.SpeakAble;

import libs.inter.SuplieAble;

import libs.inter.ThinkAble;

public abstract class LifeObject extends SceneObject implements SpeakAble, ThinkAble, MoveAble, SuplieAble{

    public LifeObject(String s) {

        super(s);

    }

    @Override

    public void speak(String phrase){

        System.out.printf("объект %s говорит - %s \n",toString(), phrase);

    }

    @Override

    public void think(){

        System.out.printf("%s думает \n",toString());

    }

    @Override

    public void suplie(SceneObject obj, String name){

        System.out.printf("%s снабажет %s %s",toString(), obj, name);

    }

}

SceneObject.java

package libs;

public abstract class SceneObject {

    private String Name;

    public SceneObject(String s) {

        this.Name = s;

        System.out.printf("объект %s создан \n", toString());

    }

    @Override

    public boolean equals(Object obj) {

        if (this == null || this.getClass()!= obj.getClass()) return false;

        SceneObject another = (SceneObject)obj;

        return this.toString() == another.toString();

    }

    public int hashCode() {

        return this.Name.hashCode();

    }

    public String toString() {

        return this.Name;

    }

}

Shorty.java

package source;

import libs.LifeObject;

public class Shorty extends LifeObject{

    public Shorty(String s){

        super(s);

    }

}

Sleepwalkers.java

package source;

import enums.Places;

import libs.LifeObject;

import libs.inter.PlantAble;

public class Sleepwalkers extends LifeObject implements PlantAble{

    public Sleepwalkers(String s){

        super(s);

    }

    @Override

    public void planted(Places place) {

        System.out.printf("объект %s сажали семена в %s \n", this, place);

    }

}

Telecast.java

package source;

import libs.SceneObject;

import libs.inter.PowerAble;

public class Telecast extends SceneObject implements PowerAble{

    private boolean Power=false;

    public Telecast(String s) {

        super(s);

    }

    public void power(){

        if (Power == false){

            System.out.println("Передача включена");

            this.Power = true;

        }

        else{

            System.out.println("Передача выключена");

            this.Power = false;

        }

    }

}

Viewers.java

package source;

import libs.LifeObject;

import libs.SceneObject;

import libs.inter.WatchAble;

public class Viewers extends LifeObject implements WatchAble{

    public Viewers(String s) {

        super(s);

    }

    @Override

    public void watch(SceneObject obj) {

        System.out.printf("объект %s смотрят объект %s \n",toString(),obj);

    }

}

Main.java

import enums.Places;

import source.Shorty;

import source.Sleepwalkers;

import source.Telecast;

import source.Viewers;

public class Main {

    public static void main(String[] args) {

        Telecast telecast = new Telecast("передача");

        Viewers policemen = new Viewers("полицейские");

        policemen.watch(telecast);

        Sleepwalkers lunatiki = new Sleepwalkers("лунатики");

        Places.listOfPlaces();

        lunatiki.planted(Places.NEELOVKA);

        lunatiki.planted(Places.GOLOPYATKINO);

        lunatiki.planted(Places.BESHLEBOVO);

        lunatiki.planted(Places.GOLODAEVKA);

        lunatiki.planted(Places.NEPROLAZNOE);

        Shorty znaika = new Shorty("Знайка");

        znaika.suplie(lunatiki, "семена растений");

    }

}

Вывод программы:

объект передача создан

объект полицейские создан

объект полицейские смотрят объект передача

объект лунатики создан

[деревне Нееловке, селе Голопяткине, селе Бесхлебове, селе Голодаевке, селе Непролазном]

объект лунатики сажали семена в деревне Нееловке

объект лунатики сажали семена в селе Голопяткине

объект лунатики сажали семена в селе Бесхлебове

объект лунатики сажали семена в селе Голодаевке

объект лунатики сажали семена в селе Непролазном

объект Знайка создан

Знайка снабажет лунатики семена растений

# Вывод

Во время выполнения лабораторной работы я научился работать с абстрактными классами, интерфейсами и перечислениями. Лучше освоил работу с полями, методами и классами и построение UML диаграмм. Узнал о функциональной части Java.