“Университет ИТМО”

Факультет программной инженерии

и компьютерной техники

Лабораторная работа №3

Вариант #21122

Выполнил:

Студент группы Р3112

Шаматульский Роман Константинович

Преподаватель

Харитонова Анастасия Евгеньевна

Санкт-Петербург, 2023

Оглавление

[Задание 3](#_Toc149043784)

[Основные этапы вычислений 4](#_Toc149043785)

[Результат работы программы 6](#_Toc149043786)

[UML-диаграмма классов 8](#_Toc149043787)

[Выводы 9](#_Toc149043788)

# Задание

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

*Избавившись от преследования, Незнайка во весь дух помчался по улице, огороженной с обеих сторон высокими заборами. Из-за заборов раздавался непрерывный собачий лай, и Незнайке казалось, что свирепые псы все еще гонятся за ним. От страха он даже не замечал, где бежал, и начал приходить понемножку в себя, когда очутился на улице с оживленным движением. Тут только он оглянулся и увидел, что позади уже нет напугавших его собак. Вокруг по тротуарам шагали лунные коротышки: никто никуда не бежал, никто никого не преследовал, никто никаких враждебных действий по отношению к Незнайке не предпринимал. Здесь уже не было глухих дощатых заборов. По обеим сторонам улицы стояли высокие дома, в нижних этажах которых помещались различные магазины.*

# Основные этапы вычислений

import objects.\*;

import enums.\*;

public class Main{

public static void main(String[] args) {

// Одушевленные объекты

Entity dunno = new Entity("Незнайка");

Entity barking = new Entity("Лай");

Entity dogs = new Entity("Собаки");

Entity moonShorties = new Entity("Лунные коротышки");

Entity nobody = new Entity("Никто");

// Неодушевленные объекты

Inanimate fence = new Inanimate("Забор");

Inanimate houses = new Inanimate("Дома");

Inanimate shops = new Inanimate("Магазины");

// Места (окружение)

Environment street = new Environment("Улица");

Environment sidewalk = new Environment("Тротуар");

Environment lowerFloors = new Environment("Нижние этажи");

//------1-предложение-------------------------------------

dunno.breakOut("преследования");

dunno.run(street);

street.surrounded(fence, Prepositions.AROUND);

//------2-предложение-------------------------------------

barking.distributed(fence, Prepositions.FROM);

dunno.think(dogs, "гонятся за ним");

//------3-предложение-------------------------------------

dunno.wasScared();

dunno.notMention("где бежал");

dunno.comeRound(street);

//------4-предложение-------------------------------------

dunno.saw(dogs, "пропали");

moonShorties.go(sidewalk);

//------5-предложение-------------------------------------

nobody.doNthnAbout(dunno);

//------6-предложение-------------------------------------

fence.lack();

//------7-предложение-------------------------------------

street.surrounded(houses, Prepositions.BYTWOSIDES);

shops.located(lowerFloors);

}

}

Main-класс

package objects;

import enums.\*;

import interfaces.\*;

public class Entity extends GeneralObj implements Move, Thoughts{

public Entity(String name){

super(name);

}

@Override

public String toString(){

return "Entity{name=" + this.getName() + "}";

}

public void breakOut(String addition){

System.out.println(this.getName() + " избавился от " + addition);

}

@Override

public void run(GeneralObj obj){

System.out.println(this.getName() + " помчался по " + obj.getName());

}

public void distributed(GeneralObj obj, Prepositions prep){

System.out.println(this.getName() + " раздавался " + prep.getValue() + " " + obj.getName());

}

@Override

public void think(GeneralObj obj, String action){

System.out.println(this.getName() + " думал, что " + obj.getName() + " " + action);

}

public void notMention(String action){

System.out.println(this.getName() + " не замечал, " + action);

}

public void wasScared(){

System.out.println(this.getName() + " боялся");

}

public void comeRound(Environment place){

System.out.println(this.getName() + " начал приходить в себя на " + place.getName());

}

public void saw(GeneralObj obj, String action){

System.out.println(this.getName() + " увидел, что " + obj.getName() + " " + action);

}

public void go(Environment place){

System.out.println(this.getName() + " шёл по " + place.getName());

}

public void doNthnAbout(GeneralObj obj){

System.out.println(this.getName() + " ничего не делал, не наносил вред " + obj.getName());

}

}

Реализация класса Entity

package interfaces;

import objects.GeneralObj;

public interface Move {

public void run(GeneralObj obj);

}

public interface Move {

public void run(GeneralObj obj);

}

Реализация интерфейсов Move и Thoughts

package objects;

public abstract class GeneralObj {

private String name;

public GeneralObj(String name){

this.name = name;

}

@Override

public boolean equals(Object obj){

return this.hashCode() == obj.hashCode();

}

@Override

public int hashCode(){

return super.hashCode() + this.getName().hashCode();

}

public String getName(){

return name;

}

}

Реализация абстрактного класса – родителя всех классов

package enums;

public enum Prepositions{

BYTWOSIDES("с обеих сторон"), BEHIND("позади"), AROUND("вокруг"), FROM("из-за");

private String value;

Prepositions(String value){

this.value = value;

}

public String getValue(){

return value;

}

}

Реализация enum’a

# Результат работы программы

*Незнайка избавился от преследования*

*Незнайка помчался по Улица*

*Забор cтоит вокруг Улица*

*Лай раздавался из-за Забор*

*Незнайка думал, что Собаки гонятся за ним*

*Незнайка боялся*

*Незнайка не замечал, где бежал*

*Незнайка начал приходить в себя на Улица*

*Незнайка увидел, что Собаки пропали*

*Лунные коротышки шёл по Тротуар*

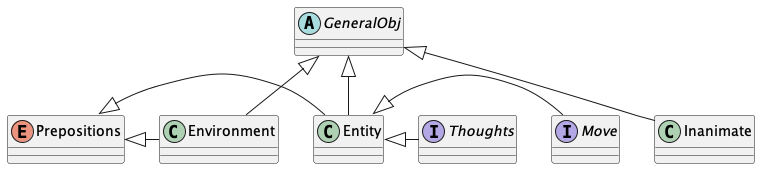
*Никто ничего не делал, не наносил вред Незнайка*

*Забор здесь отсутствовал*

*Дома cтоит с обеих сторон Улица*

*Магазины находится на Нижние этажи*

# UML-диаграмма классов

**

# Выводы

Во время написания данной лабораторной работы я изучил принципы SOLID, а также использовал абстрактные классы, интерфейсы и enum для описания ситуации в задании, а также научился переопределять некоторые методы.