Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение*

*Дисциплина «Программирование»*

**Отчет**

**По лабораторной работе №3**

**Вариант №27110**

Выполнил:

Кирячек Тимофей Алексеевич

Группа: Р3109

Преподаватель:

Письмак Алексей Евгеньевич

Г. Санкт-Петербург, 2022 г.

**Оглавление**

[Задание 2](#_Toc150092284)

[UML-диаграмма классов: 3](#_Toc150092285)

[Код программы: 4](#_Toc150092286)

[Главный класс (Main.java): 5](#_Toc150092287)

[Результат работы программы: 7](#_Toc150092288)

[Заключение 8](#_Toc150092289)

[Список литературы. 9](#_Toc150092290)

# Задание

Построить классово-объектную модель на основе текста из данного варианта.

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

Незнайка и Козлик смотрели фильм с середины и никак не могли понять, где и какие ценности преступники похитили. Им все же почему-то хотелось выяснить этот вопрос. В то же время им чрезвычайно интересно было узнать, поймают в конце концов преступников или нет. Картина между тем становилась все напряженнее и стремительнее. Одна за другой возникали головокружительные погони, массовые драки и оглушительные перестрелки. На самом интересном месте, когда главаря шайки вот-вот должны были схватить, телевизор вдруг выключился, вверху снова высунулся язычок и замигала надпись: "5 сантиков".

# UML-диаграмма классов: Изображение выглядит как текст, снимок экрана, схема Автоматически созданное описание

# Код программы:

Исходный код может быть найден в [этом репозитории](https://github.com/TecHeReTiC3141/Programming_Lab3)

# Главный класс (Main.java):

package lab3;  
  
import lab3.heros.\*;  
import lab3.actions.\*;  
import lab3.entities.\*;  
import java.util.ArrayList;  
public class Main {  
 public static void main(String[] args) {  
  
 Sentence sentence1 = new Sentence();  
 Neznaika neznaika = new Neznaika();  
 Goat goat = new Goat();  
 EntityArray heros1 = new EntityArray(){{  
 addEntity(neznaika);  
 addEntity(goat);  
 }};  
 Action watch = new Watch(new ArrayList<String>() {{ add("с середины"); }});  
 watch.attachEntity(new Film());  
 Action dontUnderstand = new DontUnderstand();  
 dontUnderstand.addModifier(ActionModifier.NO\_WAY);  
 sentence1.addClause(heros1.executeActions(new ArrayList<Action>()  
 {{ add(watch); add(dontUnderstand); }}));  
 Crimes crimes = new Crimes();  
 Action steal = new Steal();  
 steal.attachEntity(new Valuables());  
 steal.addModifier(ActionModifier.WHERE);  
 steal.addModifier(ActionModifier.WHICH);  
 sentence1.addClause(crimes.executeAction(steal));  
 System.out.println(sentence1);  
  
 Sentence sentence2 = new Sentence();  
  
 FindOut findOut = new FindOut(new ArrayList<String>() {{ add("этот"); }});  
 findOut.attachEntity(new Question());  
 Want want = new Want(findOut, new ArrayList<String>() {{ add("им"); }});  
 want.addModifier(ActionModifier.STILL);  
 want.addModifier(ActionModifier.FOR\_SOME\_REASON);  
 sentence2.addClause(want.executeSelf());  
 System.out.println(sentence2);  
  
 Sentence sentence3 = new Sentence();  
 Be be = new Be(findOut, new ArrayList<>() {{ add("им"); }});  
 Adjective interesting = new Adjective("интересно");  
 Adjective really = new Adjective("чрезвычайно");  
 be.attachObject("им", true);  
 be.addModifier(ActionModifier.AT\_THE\_SAME\_TIME);  
 be.addAdjective(really);  
 be.addAdjective(interesting);  
 findOut.clearObjects();  
 sentence3.addClause(be.executeSelf());  
  
 Catch catchAction = new Catch();  
 catchAction.addModifier(ActionModifier.FINALLY);  
 catchAction.addModifier(ActionModifier.OR\_NOT);  
 catchAction.attachEntity(crimes);  
 sentence3.addClause(catchAction.executeSelf());  
 System.out.println(sentence3);  
  
 Sentence sentence4 = new Sentence();  
 Movie movie = new Movie();  
 Become become = new Become();  
 become.addModifier(ActionModifier.BY\_THE\_WAY);  
 become.addModifier(ActionModifier.PROGRESSING);  
 become.addAdjective(new Comparative("напряженно"));  
 become.addAdjective(new Comparative("стремительно"));  
 sentence4.addClause(movie.executeAction(become));  
 System.out.println(sentence4);  
  
 Sentence sentence5 = new Sentence();  
 Pursuits pursuits = new Pursuits();  
 Shootouts shootouts = new Shootouts();  
 Fights fights = new Fights();  
 EntityArray heros2 = new EntityArray(){{  
 addEntity(pursuits);  
 addEntity(shootouts);  
 addEntity(fights);  
 }};  
 Appear appear = new Appear();  
 appear.addModifier(ActionModifier.ONE\_BY\_ONE);  
 sentence5.addClause(heros2.executeAction(appear));  
 System.out.println(sentence5);  
  
 Sentence sentence6 = new Sentence(true);  
 CrimesLeader leader = new CrimesLeader();  
 catchAction.attachEntity(leader, true);  
 catchAction.clearModifiers();  
 be.setSubAction(catchAction);  
 be.addModifier(ActionModifier.NEARLY, true);  
 be.clearObjects();  
 be.clearAdjectives();  
 Must must = new Must(be);  
 sentence6.addClause(ActionModifier.WHEN.toString() + " " + must.executeSelf());  
 TV tv = new TV();  
 TurnOff turnOff = new TurnOff();  
 turnOff.addModifier(ActionModifier.SUDDENLY);  
 turnOff.addModifier(ActionModifier.AT\_INTERESTING\_PLACE);  
 sentence6.addClause(tv.executeAction(turnOff));  
 Tongue tongue = new Tongue();  
 LeanOut leanOut = new LeanOut();  
 leanOut.addModifier(ActionModifier.AGAIN);  
 leanOut.addModifier(ActionModifier.FROM\_ABOVE);  
 sentence6.addClause(tongue.executeAction(leanOut));  
  
 Sign sign = new Sign();  
 Blink blink = new Blink();  
 sentence6.addClause(sign.executeAction(blink));  
 System.out.println(sentence6);  
 }  
}

# Результат работы программы:

Незнайка и Козлик смотреть фильм и с середины и не может понять никак, преступники похитить где и какие ценности.  
Хочется все же и почему-то им выяснить этот вопрос.

Было в то же время чрезвычайно и интересно им выяснить, поймать в конце концов и или нет преступники.

Картина становиться между тем и все более напряженно и более стремительно.

Головокружительные погони, оглушительные перестрелки и массовые драки возникать один за другим.

Когда должны было вот-вот поймать главарь шайки, телевизор выключиться вдруг и на самом интересном месте, язычок высунуться снова и вверху и надпись: "5 сантиков" замигать

# Заключение

В процессе выполнения лабораторной работы я познакомился с принципами SOLID и антипринципами STUPID, получил опыт проектирования объектной модели по заданной предметной области, узнал больше о классу Object и его методах по умолчанию, а также когда и согласно каким протоколам переопределять их, познакомился с базовыми понятиями функционального программирования

# Список литературы.

1. https://se.ifmo.ru/courses/programming
2. <https://pokemondb.net/pokedex/>
3. <https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhnih-interfeysih>
4. <https://javarush.ru/groups/posts/591-vot-tak-final>
5. <https://www.youtube.com/watch?v=3Xo6zSBgdgk>
6. <https://javarush.ru/groups/posts/1985-raznica-mezhdu-abstraktnihmi-klassami-i-interfeysami>
7. <https://skillbox.ru/media/base/klassy-i-obekty-v-java/>
8. <https://www.jetbrains.com/help/idea/class-diagram.html#analyze_graph>
9. <https://metanit.com/java/tutorial/3.3.php>
10. JAVA from EPAM, 2-е издание, исправленное. И. Н. Блинов, В. С. Романчук
11. <https://javarush.ru/groups/posts/1967-otnoshenija-mezhdu-klassami-nasledovanie-kompozicija-i-agregirovanie->
12. <https://javarush.ru/groups/posts/599-inicializacija-poley>
13. https://javarush.ru/groups/posts/2347-klass-string-v-java