**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет Программной Инженерии и Компьютерной Техники**

**Отчет по лабораторной работе №3:**

**Вариант 123456789**

Студент:

Иванов Илья Андреевич

Р3117

Преподаватель:

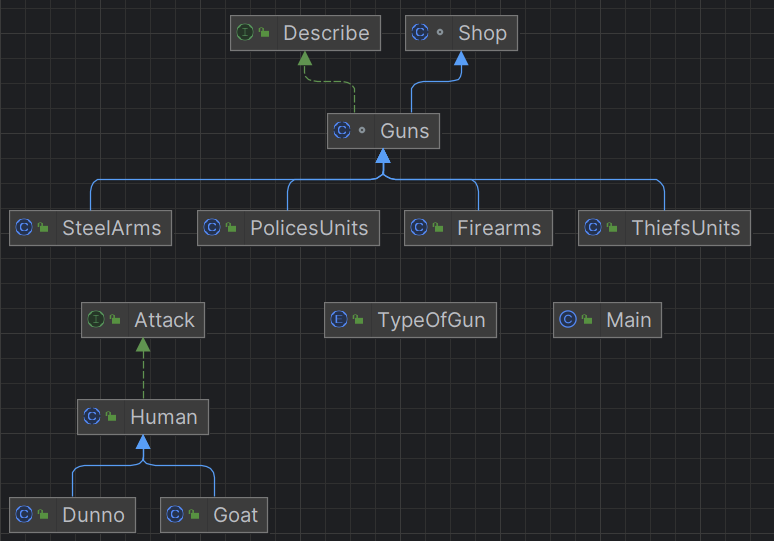
Карасёва Мария Александровна

Санкт-Петербург, 2023

# Текст задания

# Увидев над дверью одного из магазинов вывеску с надписью "Продажа разнокалиберных товаров", Незнайка и Козлик вошли в магазин и только тогда поняли, какого рода здесь продавались товары. Первое, что сразу бросалось в глаза, были ружья различных систем и калибров, стоявшие стройными рядами на специальных деревянных подставках. На прилавке в образцовом порядке лежали различные пистолеты, ножи, финки, кинжалы, кистени и кастеты. Вдоль стен были устроены освещенные изнутри витрины, в которых, словно на выставке, красовались наборы воровских отмычек, стальные пилочки, сверла, клещи, кусачки, ломики, фомки для взламывания замков, автогенные аппараты для разрезания несгораемых шкафов и сундуков. В витрине, над которой имелась надпись "Полицейская утварь", были выставлены резиновые электрические дубинки разных фасонов, стальные наручники, кандалы, зажигательные и слезоточивые бомбы и другие предметы полицейского обихода.

# Диаграмма классов реализованной объектной модели



# Коды программ

public class Main {  
 public static void main(String[] args) {  
 Dunno person1 = new Dunno("тупенький"**,** (int) (Math.*random*() \* **10000**))**;** Dunno person2 = new Dunno("тупенький2"**,** (int) (Math.*random*() \* **10000**))**;** Guns pistol = new Firearms("tec9")**;** pistol.setType(TypeOfGun.*PISTOL*)**;** pistol.setPrice(**132**)**;** //System.out.println(person1.getValue());  
 person1.buy(pistol)**;** person1.use(pistol)**;** person2.buy(pistol)**;** //System.out.println(pistol.getClass());  
 System.*out*.println(person1.equals(person2))**;** }  
}

Абстрактные классы

import java.util.Arrays**;**import java.util.Objects**;**public abstract class Human implements Attack {  
 private final String name**;** private int cash**;** private Guns item**;** Guns[] items = new Guns[**3**]**;** public Human(String name**,** int cash) {  
 this.name = name**;** this.cash = cash**;** }  
  
 public Human(int cash) {  
 this.name = "неизвестный"**;** this.cash = cash**;** }  
  
 public void addValue(int addValue) {  
 this.cash += addValue**;** }  
  
 public int getValue() {  
 return this.cash**;** }  
  
 public String getItems() {  
 for (Guns elem : items) {  
 if (elem != null) {  
 return elem.getName()**;** }  
 }  
 return "No items"**;** }  
  
 public String getName() {  
 return this.name**;** }  
  
 @Override  
 public String toString() {  
 return "Человек {" +  
 "name='" + name + '\'' +  
 ", cash=" + cash + '}'**;** }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name**,** cash**,** Arrays.*hashCode*(items))**;** }  
  
 @Override  
 public boolean equals(Object obj) {  
 if (this == obj)  
 return true**;** if (obj == null || getClass() != obj.getClass())  
 return false**;** Human person = (Human) obj**;** return cash == person.cash && name.equals(person.name) && items == person.items**;** }  
  
 public void buy(Guns item) {  
 if (Arrays.*asList*(items).contains(null)) {  
 if (item.getPrice() <= getValue()) {  
 for (int i = **0;** i < **3;** i++) {  
 if (items[i] == null) {  
 items[i] = item**;** break**;** }  
 }  
 addValue(-item.getPrice())**;** } else {  
 System.*out*.println("Не хватает денег на это оружие")**;** }  
 } else {  
 System.*out*.println("Убери какое-нибудь оружие")**;** }  
 }  
  
 public void use(Guns item) {  
 if (Arrays.*asList*(items).contains(item)) {  
 System.*out*.println(getClass().toString().substring(**6**) + " " + getName() + " " + item.describe() + item.getName())**;** } else {  
 System.*out*.println("У него нет такого оружия")**;** }  
 }  
}

abstract class Guns extends Shop implements Describe {  
 private TypeOfGun type**;** private final String name**;** public Guns(String name) {  
 this.name = name**;** }  
  
 public String getName() {  
 return this.name**;** }  
  
 public void setType(TypeOfGun type) {  
 this.type = type**;** }  
  
 public TypeOfGun getType() {  
 return this.type**;** }  
  
}

abstract class Shop {  
 private int price**;** public void setPrice(int price) {  
 this.price = price**;** }  
  
 public int getPrice() {  
 return this.price**;** }  
  
}

Интерфейсы

public interface Describe {  
 String describe()**;**}

public interface Attack {  
 void use(Guns item)**;**}

Другие классы

public class Dunno extends Human {  
 public Dunno(String name**,** int cash) {  
 super(name**,** cash)**;** }  
  
 public Dunno(int cash) {  
 super(cash)**;** }  
  
 @Override  
 public String toString() {  
 return "Незнайка {" +  
 "name='" + super.getName() + '\'' +  
 ", cash=" + super.getValue() + '}'**;** }  
}

public class Goat extends Human {  
 public Goat(String name**,** int cash) {  
 super(name**,** cash)**;** }  
  
 public Goat(int cash) {  
 super(cash)**;** }  
  
 @Override  
 public String toString() {  
 return "Козлик {" +  
 "name='" + super.getName() + '\'' +  
 ", cash=" + super.getValue() + '}'**;** }  
}

public class PolicesUnits extends Guns {  
 public PolicesUnits(String name) {  
 super(name)**;** }  
  
 public String describe() {  
 return " использует полицейское снаряжение в виде: "**;** }  
}

public class Firearms extends Guns {  
 public Firearms(String name) {  
 super(name)**;** }  
  
 public String describe() {  
 return "стреляет из "**;** }  
}

public class SteelArms extends Guns {  
 public SteelArms(String name) {  
 super(name)**;** }  
  
 public String describe() {  
 return " режет, используя "**;** }  
}

public class ThiefsUnits extends Guns {  
 public ThiefsUnits(String name) {  
 super(name)**;** }  
  
 public String describe() {  
 return " крадет что-то с помощью "**;** }  
}

Перечисляемый тип

public enum TypeOfGun {  
 *BATON***,** *HANDCUFFS***,** *SHACKLES***,** *BOMB***,** *MASTERKEYSSET***,** *STEELFILE***,** *DRILL***,** *PLIER***,** *WIRECUTTER***,** *CROWBAR***,** *AUTOGEN***,** *GUN***,** *PISTOL***,** *KNIFE***,** *FINK***,** *DAGGER***,** *FLAIL***,***KNUCKLE***;**}

# Результат работы программы

Dunno тупенький стреляет из tec9

false

# Вывод

Я научился создавать свою объектную модель, использовать перечисляемый тип и интерфейсы.