Университет ИТМО

Кафедра ВТ

Программирование

Лабораторная работа №3

Вариант 959

Группа P3110

Нгу Фыонг Ань

ПРОВЕРИЛ:

ГАВРИЛОВ АНТОН ВАЛЕРЬЕВИЧ

2017 год

**Задание:**

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

А Гвоздик, довольный тем, что его похвалили, еще старательнее стал работать. Каждому ведь нравится, когда его хвалят.

**Глава двадцать шестая. ВОЗВРАЩЕНИЕ ГВОЗДИКА**

После того как Гвоздик не вернулся домой, никто из жителей Змеевки не осмеливался отправиться в Зеленый город. Разнесся слух, что стоголовый дракон скоро прикончит всех малышек, а потом явится в Змеевку и начнет глотать малышей. Время шло, но дракон не появлялся, а вместо него в одно прекрасное утро в Змеевке появился совсем незнакомый малыш. Он рассказывал, что летел на воздушном шаре вместе со своими товарищами и спрыгнул с парашютом, когда шар стал падать. Он приземлился в дремучем лесу и с тех пор скитался по полям, по лесам, разыскивая своих товарищей, которые улетели дальше на воздушном шаре.

**Программа должна удовлетворять следующим требованиям:**

1. Доработанная модель должна соответствовать [принципам SOLID](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)).
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).

**Диаграмма классов:**

**Исходный код:**

***Файл lab3\_959.java:***

**package lab3\_959;**

**import Pack.\*;**

**public class Lab3\_959 {**

**public static void main(String[] args) {**

**//Nick**

**Position pstNick = new Position(0 , 5);**

**UnfamiliarMan Nick = new UnfamiliarMan("Nick", 23, pstNick,5,25);**

**House houseN = new House(HouseType.BEAUTIFULL);**

**City city3 = new City(Nick, houseN);**

**System.out.println(city3);**

**Nick.skill();**

**Nick.run();**

**Nick.punch();**

**Nick.jump();**

**Nick.Speak();**

**//Phil**

**Position pstPhil = new Position(0 , 5);**

**WorkMan Phil = new WorkMan("Phil", 35, pstPhil, 5, 15, "BUSINESSMAN", true);**

**House houseP = new House(HouseType.BIG);**

**City city1 = new City(Nick, houseP);**

**System.out.println(city1);**

**Phil.skill();**

**Phil.run();**

**Phil.punch();**

**Phil.jump();**

**Phil.WorkAs();**

**Phil.Speak();**

**//Carnation**

**Position pstCarnation = new Position(0 , 14);**

**WorkMan Carnation = new WorkMan("Carnation", 27, pstCarnation, 5, 15, "BAKER", false);**

**House houseCar = new House(HouseType.SMALL);**

**City city2 = new City(Carnation, houseCar);**

**System.out.println(city2);**

**Carnation.skill();**

**Carnation.run();**

**Carnation.punch();**

**Carnation.jump();**

**Carnation.WorkAs();**

**Carnation.Speak();**

**Check check = new Check();**

**check.checkLocation(Nick, Carnation);**

**check.checkLocation(Nick, Phil);**

**Position pstDraco = new Position(0 , 14);**

**Dragon Draco = new Dragon("Draco", 27, pstCarnation, 5, 15, 300, "Green City");**

**Draco.location();**

**Draco.skill();**

**Draco.fly();**

**Draco.fireBreath();**

**}**

**}**

***Файл* Animal*.java:***

**package Pack;**

**public class Animal {**

**final String name;**

**public int age;**

**public int volum;**

**public int speed;**

**public Position position;**

**public Animal(String name , int age, Position position, int volum, int speed){**

**this.name = name;**

**this.age = age;**

**this.position = position;**

**this.volum = volum;**

**this.speed = speed;**

**}**

**public int getAge() {**

**return age;**

**}**

**public int getSpeed() {**

**return speed;**

**}**

**public int getVolum() {**

**return volum;**

**}**

**public Position getPosition() {**

**return position;**

**}**

**public void setAge(int age) {**

**this.age = age;**

**}**

**public void setSpeed(int speed) {**

**this.speed = speed;**

**}**

**public void setVolum(int volum) {**

**this.volum = volum;**

**}**

**public void setPosition(Position position) {**

**this.position = position;**

**}**

**}**

***Файл* Check*.java:***

**package Pack;**

**public class Check {**

**public void checkLocation(Human Man1, Human Man2){**

**//equal() and hashCode()**

**if (Man1.position.equals(Man2.position))**

**System.out.println(Man1.name + " AND " + Man2.name + " ARE AT THE SAME PLACE"); else**

**System.out.println(Man1.name + " AND " + Man2.name + " ARE NOT AT THE SAME PLACE");**

**}**

**}**

***Файл* City*.java:***

**package Pack;**

**public class City {**

**//Attributes**

**private Human human;**

**private House house;**

**private int population = 12345 ;**

**//Constructor**

**public City(Human human, House house){**

**this.human = human;**

**this.house = house;**

**}**

**//Getters and Setters**

**public Human getHuman() {**

**return human;**

**}**

**public House getHouse() {**

**return house;**

**}**

**public int getPopulation()**

**{**

**return population;**

**}**

**//Equals**

**@Override**

**public boolean equals(Object obj) {**

**if (obj == this ) return true;**

**if (obj instanceof City){**

**City city = (City)obj;**

**if(city.getHuman().equals(this.getHuman()) && city.getHouse().equals(this.getHouse())) return true;**

**}**

**return false;**

**}**

**//HashCode**

**@Override**

**public int hashCode(){**

**return human.hashCode() + house.hashCode();**

**}**

**//The toString() describes itself**

**@Override**

**public String toString(){**

**return human + " " + house;**

**}**

**}**

***Файл* Dragon*.java:***

**package Pack;**

**public class Dragon extends Animal{**

**int degree;**

**String city;**

**public Dragon(String name, int age, Position position, int volum, int speed, int degree, String city) {**

**super(name, age, position, volum, speed);**

**this.degree = degree;**

**this.city = city;**

**}**

**public void location(){**

**System.out.println("\*" + name + " IS AT " + city);**

**}**

**public void skill() {**

**System.out.println("SKILL: Fly, Firebreath");**

**}**

**public void fly(){**

**System.out.println("FLY WITH " + speed + "KM/H");**

**}**

**public void fireBreath() {**

**System.out.println("FIRE BREATH OF " + degree + "oC");**

**}**

**@Override**

**public String toString() {**

**return "\*" + name + ": " + age + " YEARS OLD ";**

**}**

**}**

***Файл* House*.java:***

**package Pack;**

**public class House {**

**private HouseType type;**

**public House(HouseType type) {**

**this.type = type;**

**}**

**public HouseType getType() {**

**return type;**

**}**

**public void setType(HouseType type) {**

**this.type = type;**

**}**

**//Equals**

**@Override**

**public boolean equals(Object obj) {**

**if (obj == this) return true;**

**if (obj instanceof House){**

**House house = (House)obj;**

**if (house.type == this.type) return true;**

**}**

**return false;**

**}**

**//HashCode**

**@Override**

**public int hashCode() {**

**return type.hashCode();**

**}**

**@Override**

**public String toString(){**

**HouseAddress houseAdress = null;**

**final String sType = type.getTmpName();**

**String sAddress = null;**

**if (type == HouseType.BIG){**

**sAddress = HouseAddress.BIG.getDirectionCode();**

**} else {**

**if (type == HouseType.BEAUTIFULL){**

**sAddress = HouseAddress.BEAUTIFULL.getDirectionCode();**

**} else {**

**if (type == HouseType.SMALL){**

**sAddress = HouseAddress.SMALL.getDirectionCode();**

**} else {**

**if (type == HouseType.UGLY) {**

**sAddress = HouseAddress.UGLY.getDirectionCode();**

**}**

**}**

**}**

**}**

**return sType + sAddress;**

**}**

**}**

***Файл* HouseAddress*.java:***

**package Pack;**

**public enum HouseAddress {**

**BIG ("No 12\_Lombart Street\_Francisco Town\_USA "),**

**SMALL ("No 11\_Great Ocean Street\_Victoria Town\_Australia "),**

**BEAUTIFULL("No 10\_Las Vegas Boulevard South Street\_Nevada Town\_USA "),**

**UGLY("No 15\_Stelvio Street\_Sondrio Town\_ITALIA ");**

**private final String shortCode;**

**HouseAddress(String code) {**

**this.shortCode = code;**

**}**

**public String getDirectionCode(){**

**return this.shortCode;**

**}**

**}**

***Файл HouseType.java***

**package Pack;**

**public enum HouseType {**

**BIG("LIVE IN BIG HOUSE ON "),**

**SMALL("LIVE IN SMALL HOUSE ON "),**

**BEAUTIFULL("LIVE IN BEAUTIFUL HOUSE ON "),**

**UGLY("LIVE IN UGLY HOUSE ON ");**

**private final String tmpName;**

**HouseType(String name) {**

**this.tmpName = name;**

**}**

**public String getTmpName() {**

**return this.tmpName;**

**}**

**}**

***Файл Human.java:***

**package Pack;**

**public abstract class Human extends Animal{**

**public Human(String name, int age, Position position, int volum, int speed) {**

**super(name, age, position, volum, speed);**

**}**

**abstract void skill();**

**abstract void run();**

**abstract void jump();**

**abstract void punch();**

**@Override**

**public String toString() {**

**return "\*" + name + " " + age + " YEARS OLD ";**

**}**

**}**

***Файл Position.java:***

**package Pack;**

**import java.awt.Point;**

**import java.util.Objects;**

**public class Position extends Point{**

**Point point;**

**public Position(int x, int y){**

**point = new Point(x, y);**

**}**

**public Point getPoint() {**

**return point;**

**}**

**@Override**

**public int hashCode(){**

**return point.hashCode();**

**}**

**@Override**

**public boolean equals(Object obj) {**

**if (this == obj) {**

**return true;**

**}**

**if (obj == null) {**

**return false;**

**}**

**if (getClass() != obj.getClass()) {**

**return false;**

**}**

**final Position other = (Position) obj;**

**if (Objects.equals(this.point, other.point)) {**

**return false;**

**}**

**return true;**

**}**

**}**

***Файл SpeakSkill.java:***

**package Pack;**

**public interface SpeakSkill {**

**public void Speak();**

**}**

***Файл UnfamiliarMan.java:***

**package Pack;**

**public class UnfamiliarMan extends Human implements SpeakSkill{**

**public UnfamiliarMan(String name, int age, Position position, int volum, int speed) {**

**super(name, age, position, volum, speed);**

**}**

**@Override**

**public void skill() {**

**System.out.println("SKILL: Punch, Run, Jump");**

**}**

**@Override**

**public void punch(){**

**System.out.println("PUNCH WITH " + speed\*1.5 + "N");**

**}**

**@Override**

**public void jump() {**

**System.out.println("JUMP TO " + speed\*0.8 + "M");**

**}**

**@Override**

**public void run() {**

**System.out.println("RUN WITH THE SPEED OF " + speed + "KM/H" );**

**}**

**@Override**

**public void Speak(){**

**System.out.println("PLEASE HELP ME TO FIND MY FRIEND");**

**}**

**}**

***Файл WorkMan.java:***

**package Pack;**

**public class WorkMan extends Human implements WorkSkill, SpeakSkill{**

**String job;**

**boolean love;**

**public WorkMan(String name, int age, Position position, int volum, int speed, String job, boolean love) {**

**super(name, age, position, volum, speed);**

**this.job = job;**

**this.love = love;**

**}**

**@Override**

**public void skill() {**

**System.out.println("SKILL: Punch, Run, Jump");**

**}**

**@Override**

**public void punch(){**

**System.out.println("PUNCH WITH " + speed + "N");**

**}**

**@Override**

**public void jump() {**

**System.out.println("JUMP TO " + speed\*0.4 + "M");**

**}**

**@Override**

**public void run() {**

**System.out.println("RUN WITH THE SPEED OF " + speed\*0.8 + "KM/H" );**

**}**

**@Override**

**public void WorkAs(){**

**System.out.println(name + " IS WORKING AS " + job);**

**}**

**@Override**

**public void Speak(){**

**if (love) System.out.println(name + " LOVES WORKING AS " + job); else**

**System.out.println(name + " HATES WORKING AS " + job);**

**}**

**}**

***Файл WorkSkill.java:***

**package Pack;**

**public interface WorkSkill {**

**public void WorkAs();**

**}**

**Вывод программы:**

\*Nick 23 YEARS OLD LIVE IN BEAUTIFUL HOUSE ON No 10\_Las Vegas Boulevard South Street\_Nevada Town\_USA

SKILL: Punch, Run, Jump

RUN WITH THE SPEED OF 25KM/H

PUNCH WITH 37.5N

JUMP TO 20.0M

PLEASE HELP ME TO FIND MY FRIEND

\*Nick 23 YEARS OLD LIVE IN BIG HOUSE ON No 12\_Lombart Street\_Francisco Town\_USA

SKILL: Punch, Run, Jump

RUN WITH THE SPEED OF 12.0KM/H

PUNCH WITH 15N

JUMP TO 6.0M

Phil IS WORKING AS BUSINESSMAN

Phil LOVES WORKING AS BUSINESSMAN

\*Carnation 27 YEARS OLD LIVE IN SMALL HOUSE ON No 11\_Great Ocean Street\_Victoria Town\_Australia

SKILL: Punch, Run, Jump

RUN WITH THE SPEED OF 12.0KM/H

PUNCH WITH 15N

JUMP TO 6.0M

Carnation IS WORKING AS BAKER

Carnation HATES WORKING AS BAKER

Nick AND Carnation ARE AT THE SAME PLACE

Nick AND Phil ARE NOT AT THE SAME PLACE

\*Draco IS AT Green City

SKILL: Fly, Firebreath

FLY WITH 15KM/H

FIRE BREATH OF 300oC

**Вывод:** В ходе данной лабораторной работы я изучил Простое и множественное наследование, abstract class, inteface, enum