Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики.

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3
предмет «Программирование» вариант 311256

ФИО: Фам Мань Туан

Группа: P3112

Преподаватель: Максимов Андрей Николаевич

Перцев Тимофей Сергеевич

Санкт-Петербург 2020 г.

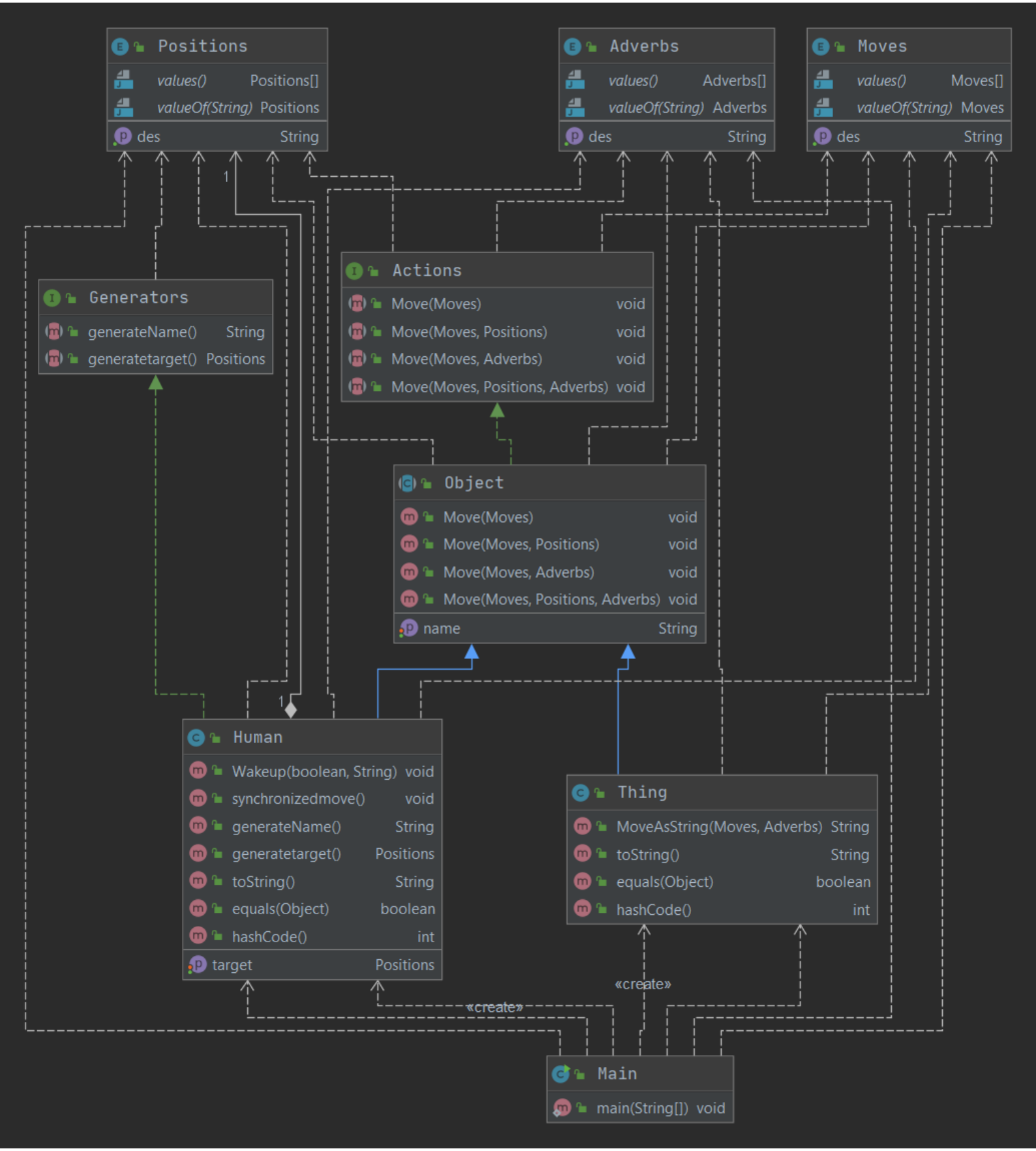
Задание:

1. Доработать объектную модель приложения.
2. Перерисовать диаграмму классов в соответствии с внесёнными в модель изменениями.
3. Согласовать с преподавателем изменения, внесённые в модель.
4. Модифицировать программу в соответствии с внесёнными в модель изменениями.

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

Шкатулка с нитками плыла и плыла. Ее занесло в заливчик, где дом сел на мель. Покачавшись в прибрежных камышах, шкатулка наконец увязла в иле. Но малышка Мю не проснулась. Она не проснулась даже тогда, когда рыболовный крючок взвился над ней и заДорогой читатель! Приготовься к неожиданности. цепился за шкатулку. Крючок дернулся разок-другой, леска натянулась, и шкатулка осторожно поплыла. Случайности и совпадения творят чудеса. Ничего не зная друг о друге и о приключениях друг друга, семейство муми-троллей и Снусмумрик случайно оказались в одном и том же заливе в самый вечер летнего солнцестояния. Это и в самом деле был Снусмумрик. В своей старой зеленой шляпе он стоял на берегу и таращился на шкатулку.

Диаграмма класс



Исходный код

Main.java

```
public class Main {
    public static void main(String[] args) {
        Thing Box = new Thing("The thread box");
        Box.Move(Moves.FLOAT);
        Box.Move(Moves.CARRIED,Positions.BAY);
        Thing House = new Thing("The house");
        House.Move(Moves.LOCATED,Positions.BAY);
        Human Mu = new Human("Mu");
        Thing Fishhook = new Thing("The Fishhook");
        Mu.Wakeup(Math.random()<0.5,Fishhook.MoveAsString(Moves.HOVER,Adverbs.GIRL));
        Fishhook.Move(Moves.CLUNG,Positions.BOX);
        Box.Move(Moves.FLOAT,Adverbs.CAREFULLY);
        Human Moonmin = new Human("Moomin's family",Positions.BAY);
        Human Snusmumrik = new Human("Snusmumrik",Positions.BAY);
        int numberofhumans = (int)(Math.round(Math.random()*10)+5);
        Human[] humans = new Human[numberofhumans];
        humans[0] = Moonmin; humans[1]=Snusmumrik;
        for (int i=2;i<numberofhumans;i++){
            humans[i] = new Human();
        }
        humans[0].synchronizedmove();
        Snusmumrik.Move(Moves.STAND,Positions.BANK,Adverbs.GREENHAT);
        Snusmumrik.Move(Moves.STARE,Positions.BOX);
    }
}
```

Object.java

```
public abstract class Object implements Actions {
    private String Name;
    @Override
    public void Move(Moves move){
        System.out.println(this.getName()+move.getDes());
    }
    @Override
    public void Move(Moves move, Positions position){
        System.out.println(this.getName()+move.getDes()+position.getDes());
    }
    @Override
    public void Move(Moves move, Adverbs adverb){
        System.out.println(this.getName()+move.getDes()+adverb.getDes());
    }
    @Override
    public void Move(Moves move, Positions position, Adverbs adverb){
        System.out.println(this.getName()+move.getDes()+position.getDes()+adverb.getDes());
    }
    public String getName() {
        return Name;
    }
    public void setName(String name) {
        Name = name;
    }
}
```

Human.java

```
import java.util.Objects;
import java.util.Random;

public class Human extends Object implements Generators{
    public static Human[][] ListofTargets = new Human[Positions.values().length][20];
```

```
    private Positions Target;
    public Human(){
        generateName();
        generatetarget();
        //System.out.println(this.getName()+" "+this.getTarget()+" "+this);
    }
    public Human(String name, Positions target) {
        this.setName(name);
        this.setTarget(target);
    }
    public Human(String name) {
        this.setName(name);
        generatetarget();
    }
    public void Wakeup(boolean did, String when) {
        System.out.println(this.getName()+((did)?"did":"didn't even") +" wake up when " + when);
    }
    private void append(int index, Human value){
        int i;
        for (i=0;i<ListofTargets[index].length;i++){
            if (ListofTargets[index][i] == null) {break;}
        }
        ListofTargets[index][i] = value;
    }
    public void synchronizedmove(){
        System.out.println("Accidents and coincidences work wonders. Knowing nothing about each
other and about each other's adventures");
        for (int i=0;i<ListofTargets.length;i++) {
            StringBuilder result = new StringBuilder();
            for (int j = 0; j < ListofTargets[i].length; j++) {
                if (ListofTargets[i][j] != null) {
                    result.append((j == 0) ? "" : " and ").append(ListofTargets[i][j].getName());
                }
            }
            if (Positions.values()[i]==Positions.BAY) {
                System.out.println(result + Moves.BE.getDes() +
Positions.values()[i].getDes()+Adverbs.EVENING.getDes());
            }
            else if (!result.toString().equals("")) {
                System.out.println(result + Moves.BE.getDes() + Positions.values()[i].getDes());
            }
        }
    }
    final Random rand = new Random();
    @Override
    public  void generateName() {
        this.setName(Beginning[rand.nextInt(Beginning.length)] +
Middle[rand.nextInt(Middle.length)]+End[rand.nextInt(End.length)]);
    }
    @Override
    public void generatetarget() {
        this.setTarget(Positions.values()[(int)(Math.random()*Positions.values().length)]);
    }
    public Positions getTarget(){
        return Target;
    }
    public void setTarget(Positions target) {
        Target = target;
        append(Positions.valueOf(this.getTarget().name()).ordinal(),this);
    }
    @Override
    public String toString() {
        return "Name: " + this.getName() + "Target: " + this.getTarget();
    }
```

```
    @Override
    public boolean equals(java.lang.Object o) {
        if (this == o) return true;
        if (!(o instanceof Human)) return false;
        Human human = (Human) o;
        return Objects.equals(this.getName(), human.getName()) &&
Objects.equals(this.getTarget(),human.getTarget());
    }
    @Override
    public int hashCode() {
        return Objects.hash(this.getName(), this.getTarget());
    }
}
```

Thing.java

```
import java.util.Objects;

public class Thing extends Object{
    public Thing(String name){
        this.setName(name);
    }
    public String MoveAsString(Moves move, Adverbs adverb){
        return (this.getName()+move.getDes()+adverb.getDes());
    }
    @Override
    public String toString() {
        return "Name: " + this.getName();
    }
    @Override
    public boolean equals(java.lang.Object o) {
        if (this == o) return true;
        if (!(o instanceof Thing)) return false;
        Thing thing = (Thing) o;
        return Objects.equals(this.getName(), thing.getName());
    }
    @Override
    public int hashCode() {
        return Objects.hash(this.getName());
    }
}
```

Adverbs.java

```
public enum Adverbs {
    CAREFULLY(" carefully"),
    GREENHAT(" with an old green hat"),
    EVENING(" in the evening of the summer solstice"),
    GIRL(" over her");
    final private String Des;
    Adverbs(String setDes) {
        this.Des = setDes;
    }
    public String getDes() {
        return Des;
    }
}
```

Moves.java

```
public enum Moves {
    FLOAT(" floated"),
    CARRIED(" was carried"),
    LOCATED(" is located"),
    BOGGED(" was bogged down"),
    HOVER(" hovered"),
    CLUNG(" clunged"),
    HOOK(" hooked"),
```

```
    BE(" were"),
    STAND(" stood"),
    SWAY(" was swaying"),
    TWITCH(" twiched"),
    PULL(" pulled"),
    STARE(" stared");

    private String Des;

    Moves(String setDes) {
        this.Des = setDes;
    }

    public String getDes() {
        return Des;
    }
}
```

Positions.java

```
public enum Positions {
    BAY(" to the bay"),
    COASTALREEDS(" in the coastal reeds"),
    MUD(" in the mud"),
    BOX(" to the box"),
    BANK(" on the bank");

    private String Des;

    Positions(String setDes) {
        this.Des = setDes;
    }

    public String getDes() {
            return Des;
    }
}
```

Actions.java

```
public interface Actions {
    void Move(Moves move);
    void Move(Moves move, Positions position);
    void Move(Moves move, Adverbs adverb);
    void Move(Moves move, Positions position, Adverbs adverb);
}
```

Generators.java

```
public interface Generators {
     String[] Beginning = { "Kr", "Ca", "Ra", "Mrok", "Cru",
            "Ray", "Bre", "Zed", "Drak", "Mor", "Jag", "Mer", "Jar", "Mjol",
            "Zork", "Mad", "Cry", "Zur", "Creo", "Azak", "Azur", "Rei", "Cro",
            "Mar", "Luk" };
     String[] Middle = { "air", "ir", "mi", "sor", "mee", "clo",
            "red", "cra", "ark", "arc", "miri", "lori", "cres", "mur", "zer",
            "marac", "zoir", "slamar", "salmar", "urak" };
     String[] End = { "d", "ed", "ark", "arc", "es", "er", "der",
            "tron", "med", "ure", "zur", "cred", "mur" };
     void generateName();
     void generatetarget();
}
```

Результат работы

The thread box floated

The thread box was carried to the bay

The house is located to the bay

The thread box was swaying in the coastal reeds

The thread box was bogged down in the mud

Mudidn't even wake up when The Fishhook hovered over her

The Fishhook clunged to the box

The Fishhook twiched once or tight

The line pulled tight

The thread box floated carefully

Accidents and coincidences work wonders. Knowing nothing about each other and about each other's adventures

Moomin's family and Snusmumrik were to the bay in the evening of the summer solstice

Zedmaraczur and Mrokmiriark and Marmeeark were in the mud

Creomirizur and Azaksormed and Zorksalmararc were to the box

Zorksorer and Mjolzoirarc and Jarredder were on the bank

Snusmumrik stood on the bank

Snusmumrik stared to the box

Вывод:

После выполнения этой лабораторной работы я понял, что такое SOLID и STUPID. Я знал, как разрабатывать программу.