

Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: Программирование

Лабораторная работа по программированию №4

Выполнил: Конаныхина Антонина
Александровна
Группа: Р3115
Вариант: 311583,4
Преподаватель: Письмак Алексей
Евгеньевич

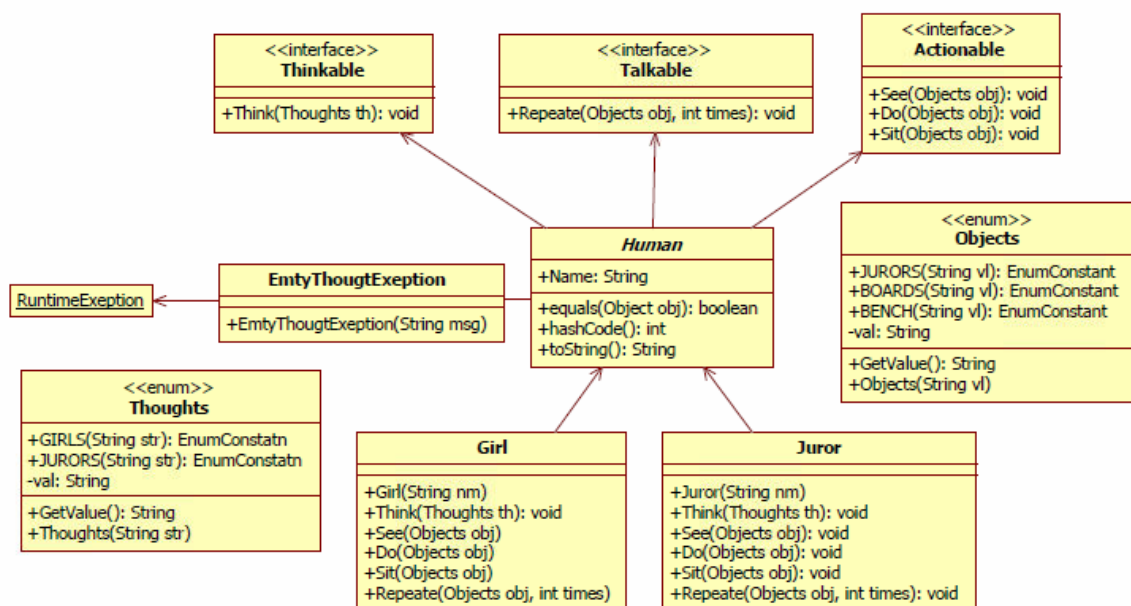
Санкт-Петербург, 2020г

Доработать программу из лабораторной работы #3, обновив реализацию объектной модели в соответствии с новой версией описания предметной области.

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

Судьей, кстати, был сам Король, и так как на парик ему пришлось надеть корону, он очень стеснялся - такой наряд был ему явно не к лицу. "А вот эти двенадцать тварюшек (она не могла найти другого слова - ведь там были и птички и зверьки) - это, наверное, пристяжные... нет, присяжные!" Алиса не без гордости раза два-три повторила это слово. "Вряд ли много найдется девочек в моем возрасте, а то и старше,- подумала она,- которые слыхали такое слово и знают, что оно значит". Пожалуй, она была права, хотя слово "заседатели" было бы ничуть не хуже. Присяжные сидели на большой скамье, стоявшей на возвышении ("Это скамья присяжных. В скамейке то все и дело, кто на нее присел, тот и присяжный",- подумала Алиса). У всех у них были грифельные доски, и все они что-то деловито записывали. - Что это они? - шепнула Алиса Грифону.- Записывать-то нечего, суд еще не начался!

UML-схема:



Код:

```
public class Main
{
    public static void main(String[] args)
    {
        Girl Alice = new Girl("Алиса");
        Juror Jurs = new Juror("Присяжные");
        Actionable King = new Actionable() {
```

```

        @Override
        public void see(Objects obj)
        {
            System.out.println("Король увидел своим пронизательным
взглядом " + obj.getValue());
        }

        public void todo(Objects obj)
        {
            System.out.println("Король " + obj.getValue());
        }
        public void sit(Objects obj)
        {
            System.out.println("Король восседал на " + obj.getValue());
        }
        public void Repeate(Objects obj, int times)
        {
            for (int i = 0; i < times; i++) {
                System.out.println("Король повторили для непонятливых
подданных " + obj.getValue());
            }
        }
    };

    King.todo(Objects.CROWN);

    try {
        if (Math.random() < 0.3)
            Jurs.think(Thoughts.EMPTY);
    }
    catch (RuntimeException e)
    {
        System.out.println(e.getMessage());
    }

    try {
        Alice.think(Thoughts.JURARS);
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
    Alice.repeate(Objects.JURORS_WORDS, 3);

    try {
        Alice.think(Thoughts.GIRLS);
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
    Alice.see(Objects.JURORS);
    Jurs.sit(Objects.BENCH);

    try {
        Alice.think(Thoughts.JURORS);
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }

    try {

```

```

        if (Math.random() < 0.3)
            Jurs.think(Thoughts.EMPTY);
    }
    catch (RuntimeException e)
    {
        System.out.println(e.getMessage());
    }

    Jurs.todo(Objects.BOARDS);

    Alice.todo(Objects.QUITE_SAY);
    //Girl act1 = new Girl("Алис", nar);
    //SeeAction act2 = new SeeAction();
    //System.out.println(nar.equals(Alice));
}

@Override
public boolean equals(Object obj) {
    return super.equals(obj);
}
}

public interface Actionable
{
    public void see(Objects obj);
    public void todo(Objects obj);
    public void sit(Objects obj);
}

public interface Talkable
{
    public void repeate(Objects obj, int times);
}

interface Thinkable
{
    public void think(Thoughts th) throws Exception;
}

public abstract class Human implements Thinkable, Actionable, Talkable
{
    String Name;

    public static class EmptyThoughtExeption extends RuntimeException{
        EmptyThoughtExeption(Thoughts th){
            super("Ошибка возникла из-за передачи в аргумент метода пустой
мысли: "+th.name());
        }
    }

    @Override
    public boolean equals(Object obj) {
        if (this.getClass() != obj.getClass())
            return false;
        Human others = (Human) obj;
        return this.Name == others.Name;
    }

    public int hashCode() {
        return this.Name.hashCode();
    }
}

```

```

    public String toString() {
        return this.Name;
    }
}

public class Juror extends Human
{
    public Juror(String nm)
    {
        Name = nm;
    }

    //String Name;

    @Override

    public void think(Thoughts th) throws RuntimeException
    {
        if(th.getValue().isEmpty())
        {
            throw new EmptyThoughtException(th);
        }
        System.out.println(Name + " подумали " + th.getValue());
    }
    public void see(Objects obj)
    {
        System.out.println(Name + " увидели " + obj.getValue());
    }
    public void todo(Objects obj)
    {
        System.out.println(Name + " " + obj.getValue());
    }
    public void sit(Objects obj)
    {
        System.out.println(Name + " сидели на " + obj.getValue());
    }
    public void repeate(Objects obj, int times)
    {
        for (int i = 0; i < times; i++) {
            System.out.println(Name + " повторили " + obj.getValue());
        }
    }
}

public class Girl extends Human
{
    public Girl(String nm)
    {
        Name = nm;
    }

    //String Name;

    @Override

    public void think(Thoughts th) throws Exception
    {
        if(th.getValue().isEmpty())
        {

```

```

        throw new EmptyThoughtException(th);
    }
    System.out.println(Name + " подумала " + th.getValue());
    if (th == Thoughts.JURARS) {
        class WrongWordExeption extends Exception {
            WrongWordExeption(String msg, Thoughts th){
                super(msg + " передана мысль с ошибкой: " + th.name());
            }
        }
        throw new WrongWordExeption("Правильно: присяжные!", th);
    }
}
public void see(Objects obj)
{
    System.out.println(Name + " увидела " + obj.getValue());
}
public void todo(Objects obj)
{
    System.out.println(Name + " " + obj.getValue());
}
public void sit(Objects obj)
{
    System.out.println(Name + " села на " + obj.getValue());
}
public void repeate(Objects obj, int times)
{
    for (int i = 0; i < times; i++) {
        System.out.println(Name + " повторила слово " + obj.getValue());
    }
}
}

public enum Objects {
    JURORS("присяжных"),
    JURORS_WORDS("присяжные"),
    BOARDS("пишут на грифельных досках"),
    BENCH("скамейке"),
    CROWN("надел корону на парик, он очень стеснялся - такой наряд был ему явно не к лицу."),
    QUITE_SAY("шепнула Грифону: Что это они? Записывать-то нечего, суд еще не начался!");

    private String val;
    Objects(String vl)
    {
        val = vl;
    }
    public String getValue()
    {
        return val;
    }
}

enum Thoughts
{
    GIRLS("Вряд ли много найдется девочек в моем возрасте, а то и старше, которые слышали такое слово и знают, что оно значит"),
    JURORS("Это скамья присяжных. В скамейке то все и дело, кто на нее присел, тот и присяжный"),
    JURARS("А вот эти двенадцать тварюшек (она не могла найти другого слова - ведь там были и птички и зверьки) - это, наверное, пристяжные... "),
    EMPTY("");
}

```

```
private String val;  
Thoughts(String str)  
{  
    val = str;  
}  
String getValue()  
{  
    return val;  
}  
}
```

Вывод программы:

Король надел корону на парик, он очень стеснялся - такой наряд был ему явно не к лицу.

Ошибка возникла из-за передачи в аргумент метода пустой мысли: EMPTY

Алиса подумала А вот эти двенадцать тварюшек (она не могла найти другого слова - ведь там были и птички и зверьки) - это, наверное, присяжные...

Правильно: присяжные! передана мысль с ошибкой: JURARS

Алиса повторила слово присяжные

Алиса повторила слово присяжные

Алиса повторила слово присяжные

Алиса подумала Вряд ли много найдется девочек в моем возрасте, а то и старше, которые слышали такое слово и знают, что оно значит

Алиса увидела присяжных

Присяжные сидели на скамейке

Алиса подумала Это скамья присяжных. В скамейке то все и дело, кто на нее присел, тот и присяжный

Ошибка возникла из-за передачи в аргумент метода пустой мысли: EMPTY

Присяжные пишут на грифельных досках

Алиса шепнула Грифону: Что это они? Записывать-то нечего, суд еще не начался!

Вывод:

В процессе выполнения лабораторной работы я научилась дорабатывать программу до новой объектной модели, создавать свои исключения, а также работать с анонимными, локальными и вложенными классами.