Лабораторная работа №4

по программированию

Вариант 32123

**Выполнила**:

P3115,

Ван Минчжи

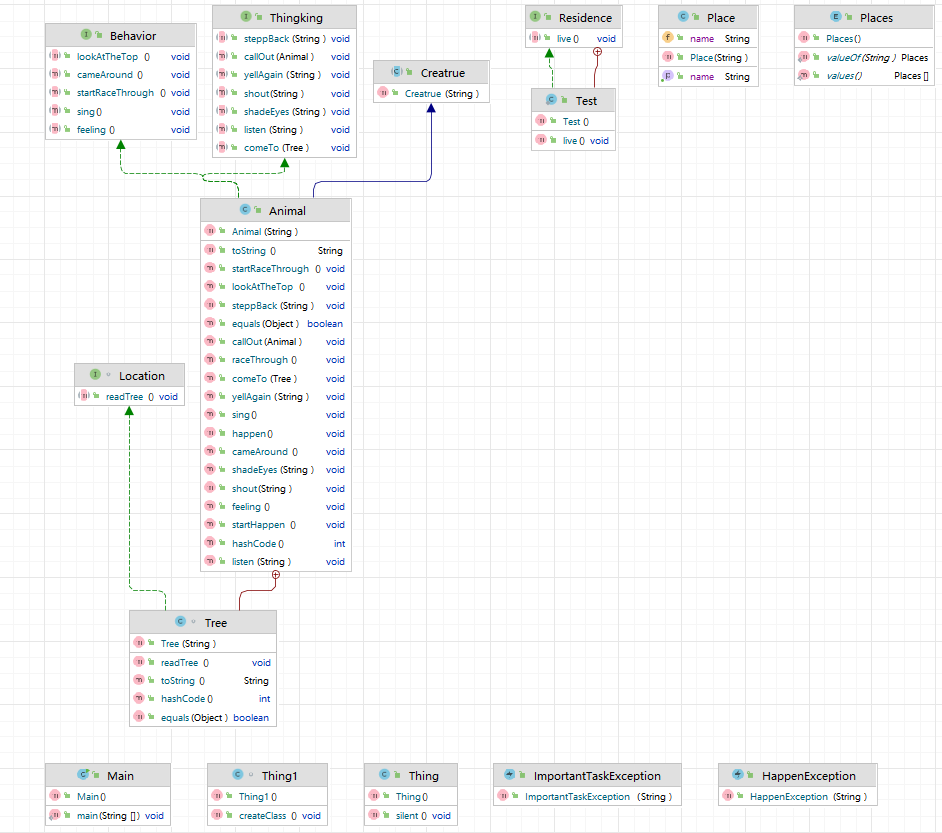
**Преподаватель:**

Усков Иван Владимирович

**Статья**

Кролик несся по опушке Дремучего Леса, с каждой минутой все больше чувствуя важность своей задачи, и наконец он прибежал к дереву, в котором жил Кристофер Робин.Он раза два окликнул Кристофера Робина. Потом он отошел немного назад и, заслонив лапкой глаза от солнца, еще покричал, глядя на верхушку. Потом он зашел с другой стороны и опять покричал: "Эй!" и "Слушай!" и "Это Кролик!", но ничего не произошло. Тогда он замолчал и прислушался, и все замолчало и прислушалось вместе с ним, и в освещенном солнцем Лесу стало тихо-тихо, и потом вдруг где-то в невероятной вышине запел жаворонок.

**Диаграмма классов**

****

**Main**

package l4;

import l4.Animal.Tree;

import l4.Residence.Test;

public class Main {

public static void main(String[] args) throws Exception {

Animal Bunny = new Animal("Bunny");

Animal ChristopherRobin = new Animal("ChristopherRobin");

Test ChristopherRobin1 = new Test();

Animal жаворонок = new Animal("жаронок");

Thing1 th = new Thing1();

Tree tree = Bunny.new Tree("tree");

Bunny.raceThrough();

Bunny.startRaceThrough();

Bunny.feeling();

Bunny.comeTo(tree);

ChristopherRobin1.live();

Bunny.callOut(ChristopherRobin);

Bunny.steppBack(" a little,");

Bunny.shadeEyes("with his paw");

Bunny.shout("some more,");

Bunny.lookAtTheTop();

Bunny.cameAround();

Bunny.yellAgain("\"Hey!Listen!It's Bunny!\"");

try {

Bunny.happen();

} catch (HappenException e) {

System.out.println(e);

e.printStackTrace();

}

Bunny.listen(" was silent and listened,");

th.createClass();

жаворонок.sing();

}

}

**Animal**

package l4;

import java.lang.reflect.Field;

import java.util.Objects;

public class Animal extends Creatrue implements Behavior,Thingking{

public Animal(String name){

super(name);

}

class Tree implements Location{

public String name;

public Tree(String name){

this.name = name;

}

public void readTree(){

}

public String toString(){

return name;

}

public boolean equals(Object otherObject){

if(this==otherObject){

return true;

}

if(otherObject==null){

return false;

}

if (getClass()!=otherObject.getClass()){

return false;

}

Tree tree = (Tree) otherObject;

return Objects.equals(name, tree.name);

}

public int hashCode(){

return name != null ? name.hashCode() : 0;

}

}

public void comeTo(Tree tree) {

System.out.println(name + " come to the " + tree.toString() );

}

public void raceThrough() throws Exception{

Place place = new Place("the edge of the Forest");

Class class1 = place.getClass();

Field f = class1.getDeclaredField("name");

Object value = f.get(place);

System.out.println(name + " raced through " + value );

}

public void callOut(Animal ChristopherRobin){

System.out.println(name +" call out to " + ChristopherRobin.toString() +" twice.");

}

public void steppBack(String content){

System.out.println(name + " stepp back" + content);

}

public void shadeEyes(String content){

String str = " from the sun";

System.out.println("and shading his eyes " + content + str);

}

public void shout(String content){

System.out.println("shouted " + content);

}

public void lookAtTheTop(){

System.out.println("looking up at the top.");

}

public void cameAround(){

System.out.println(name + " came aroud to the other side.");

}

public void yellAgain(String content){

System.out.println(name + " yelled again:" + content);

}

public void listen(String content){

System.out.println(name + content + "\nand the " + Places.sunlit\_Forest + " was silent and quiet.");

}

public void sing(){

System.out.println(Places.Somewhere+" "+name + " was singing");

}

public String toString(){

return name;

}

public boolean equals(Object otherObject){

if(this==otherObject){

return true;

}

if(otherObject==null){

return false;

}

if (getClass()!=otherObject.getClass()){

return false;

}

Animal animal = (Animal) otherObject;

return Objects.equals(name, animal.name);

}

public int hashCode(){

return name != null ? name.hashCode() : 0;

}

boolean isRaceThrough;

boolean isHappen;

public void startRaceThrough(){

System.out.println("feeling the importance of his task.");

this.isRaceThrough = true;

}

public void feeling() throws ImportantTaskException{

if(isRaceThrough){

System.out.println("feeling more and more the importance of his task.");

}else{

throw new ImportantTaskException(name + " walks slowly");

}

}

public void startHappen(){

this.isHappen = false;

}

public void happen() throws HappenException{

if(isHappen == false){

System.out.println("but nothing happened");

}else{

throw new HappenException("Someone answered");

}

}

}

**Behavior**

package l4;

public interface Behavior {

void lookAtTheTop();

void cameAround();

void sing();

void startRaceThrough();

void feeling();

}

**Creatrue**

package l4;

public abstract class Creatrue {

public String name;

public Creatrue(String name){

this.name = name;

}

}

**HappenException**

package l4;

public class HappenException extends Exception{

public HappenException(String message){

super(message);

}

}

**ImportantTaskException**

package l4;

public class ImportantTaskException extends RuntimeException{

public ImportantTaskException(String message){

super(message);

}

}

**Location**

package l4;

interface Location{

public void readTree();

}

**Place**

package l4;

public class Place {

public String name;

public Place(String name){

this.name = name;

}

public String getName(){

return this.name;

}

}

**Places**

package l4;

public enum Places {

sunlit\_Forest,

Somewhere,

edge,

Woods;

}

**Residence**

package l4;

public interface Residence {

void live();

class Test implements Residence {

public void live(){

System.out.println("ChristopherRobin lived in which.");

}

}

}

**Thing**

package l4;

public class Thing {

public void silent(){

System.out.println("Everything was silent and listened");

}

}

class Thing1 {

public void createClass(){

Thing t1 = new Thing(){

public void silent(){

System.out.println("everything was silent and listened with him.");

}

};

t1.silent();

}

}

**Thingking**

package l4;

import l4.Animal.Tree;

public interface Thingking {

void callOut(Animal ChristopherRobin);

void comeTo(Tree tree);

void yellAgain(String content);

void shadeEyes(String content);

void shout(String content);

void listen(String content);

void steppBack(String content);

}

**Результат**

