

Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

Факультет ПИиКТ

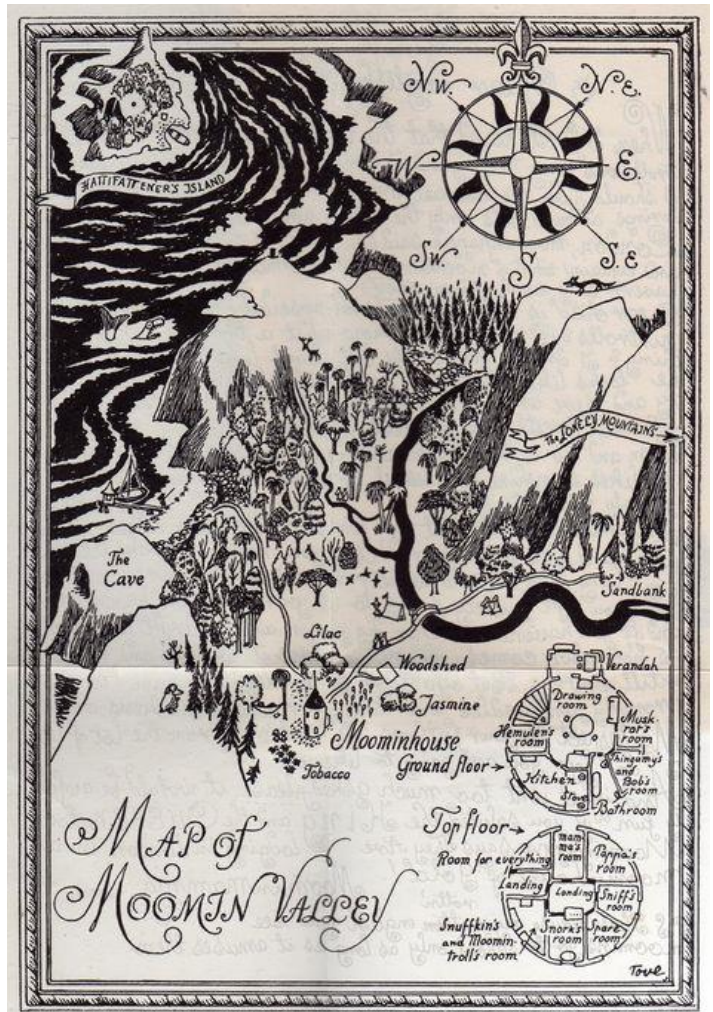
Лабораторная работа №4
по дисциплине
«Программирование»

Вариант 971,5

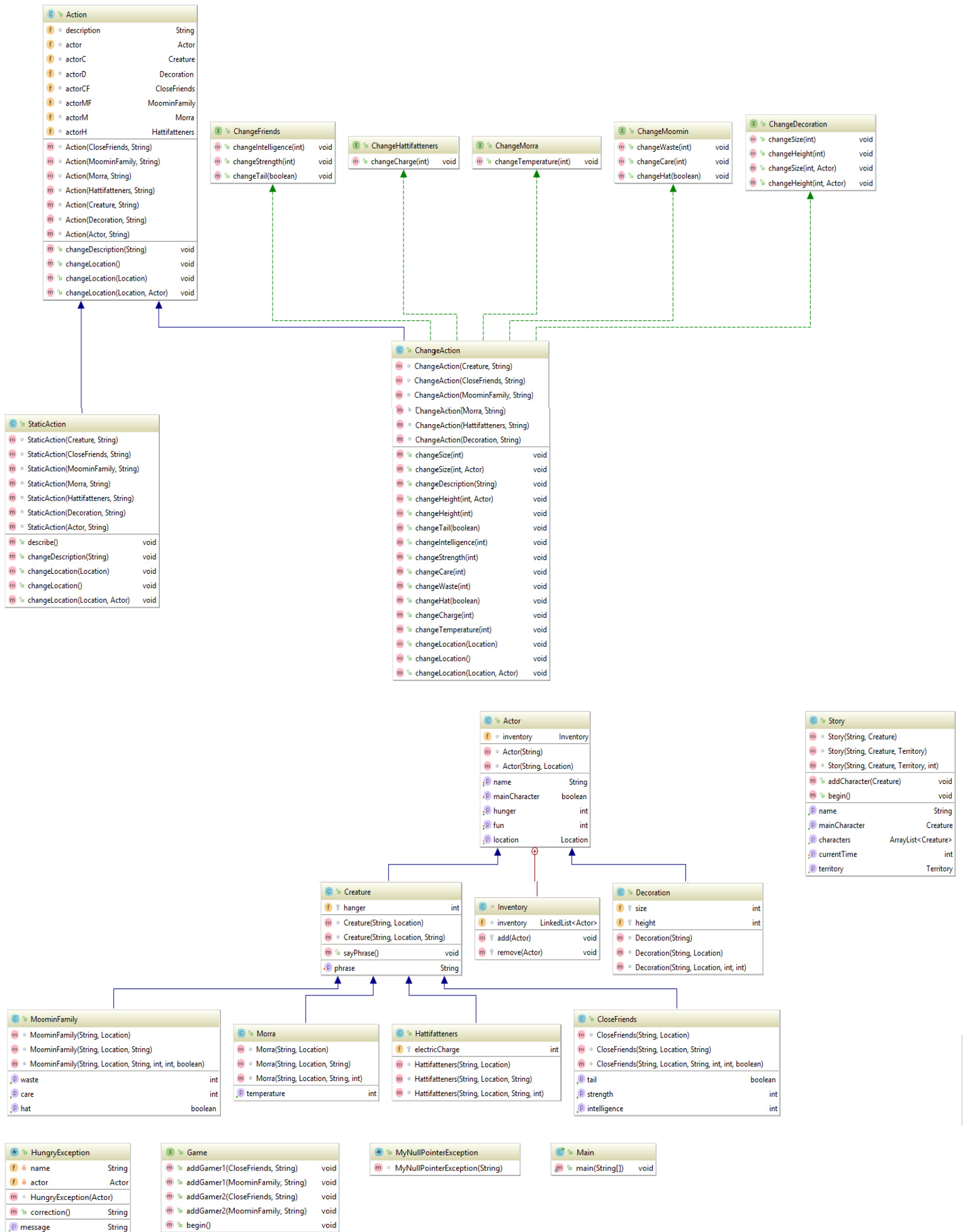
Выполнил: Киселёв Сергей Владимирович
Преподаватель: Письмак Алексей Евгеньевич

Задание

Это был исключительный день. Затеялась игра в девственный лес. Муми-тролль был Тарзаном, фрекен Снорк была Джейн. Сниффу разрешили быть сыном Тарзана, а Снусмумрик взял на себя роль шимпанзе Читы. Снорк ползал в подлеске с вставными челюстями из апельсиновых корок [Спроси у мамы: она знает, как они делаются!] и изображал врага вообще. Веселье продолжалось до самого вечера. Никого не тревожило, что вход в погреб может зарости, и все думать забыли о бедняге Хемуле. А он торчал на дворе в мокром платье, прилипавшем к ногам, и караулил Панталошку. Время от времени он съедал яблочко или считал тычинки в каком-нибудь цветке джунглей, но больше вздыхал. Дождь перестал, близились сумерки. И едва только зашло солнце, что-то случилось с зеленым холмом, заключившим в себя Муми-дом. Он стал увядать так же быстро, как вырос. Плоды сморщились и попадали на землю. Цветы поникли, а их листья свернулись трубочками. Дом наполнился шорохом и потрескиванием. Хемуль глядел, глядел, потом подошел и легонько потянул к себе ветку. Она была сухая, как трут, и сразу же отломилась: Тут Хемуля осенило. Он собрал огромную кучу хвороста, сходил в дровяной сарай за спичками -- и на садовой дорожке запалил костер. Веселый и довольный, Хемуль подсел к огню и просушил свое платье. А немного погодя его еще раз осенило, и он с нехемульской силой затащил в огонь хвост Панталошки: больше всего на свете Хемуль любил жареную рыбу.



UML



Story.java

```
package com.kamikadze328.vk;

import java.util.ArrayList;

public class Story {
    private String name;
    private ArrayList<Creature> characters = new ArrayList<>();
    public Creature mainCharacter;
    private Territory territory;
    private int currentTime;

    Story(String n, Creature M) {
        try {
            name = n;
            mainCharacter = M;
            M.setMainCharacter(true);
            M.setHunger(10);
            characters.add(M);
            territory = Territory.Moominhouse;
            currentTime = 12;
        } catch (NullPointerException e) {
            throw new MyNullPointerException("Похоже вы не задали главного героя!!!");
        }
    }

    Story(String n, Creature M, Territory t) {
        try {
            name = n;
            M.setMainCharacter(true);
            M.setHunger(10);
            mainCharacter = M;
            characters.add(M);
            territory = t;
            currentTime = 12;
        } catch (NullPointerException e) {
            throw new MyNullPointerException("Похоже вы не задали главного героя!!!");
        }
    }

    Story(String n, Creature M, Territory t, int ct) {
        try {
            name = n;
            M.setMainCharacter(true);
            M.setHunger(10);
            mainCharacter = M;
            characters.add(M);
            territory = t;
            currentTime = ct;
        } catch (NullPointerException e) {
            throw new MyNullPointerException("Похоже вы не задали главного героя!!!");
        }
    }

    public void setCurrentTime(int currentTime) {
        this.currentTime = currentTime;
        System.out.println(currentTime + " часов");
    }

    public void addCharacter(Creature creature) {
        characters.add(creature);
    }

    public ArrayList<Creature> getCharacters() {
```

```

        return characters;
    }

    public String getName() {
        return name;
    }

    public Creature getMainCharacter() {
        return mainCharacter;
    }

    public int getCurrentTime() {
        return currentTime;
    }

    public Territory getTerritory() {
        return territory;
    }

    public void begin() {
        System.out.println("\nВ далёкой-далёкой галактике жил-был " +
mainCharacter.getName());
        System.out.println("Было у него много друзей");
        for (int i = 0; i < characters.size(); i++) {
            if (characters.get(i).getName() != mainCharacter.getName()) {
                System.out.println(characters.get(i).getName());
            }
        }
        System.out.println("И вот однажды в " + currentTime + " часов" + " в " +
territory + "...\\n");
    }
}

```

Action.java

```
package com.kamikadze328.vk;
```

```

public abstract class Action {
    String description;
    Actor actor;
    Creature actorC;
    Decoration actorD;
    CloseFriends actorCF;
    MoominFamily actorMF;
    Morra actorM;
    Hattifatteners actorH;

    Action(CloseFriends a, String d) {
        actorCF = a;
        actor = a;
        description = d;
    }
    Action(MoominFamily a, String d){
        actorMF = a;
        actor = a;
        description = d;
    }
    Action(Morra a, String d){
        actorM = a;
        actor = a;
        description = d;
    }
    Action(Hattifatteners a, String d){
        actorH = a;
        actor = a;
        description = d;
    }
    Action(Creature a, String d){

```

```

        actorC = a;
        actor = a;
        description = d;
    }
    Action(Decoration a, String d){
        actorD = a;
        actor = a;
        description = d;
    }
    Action(Actor a, String d){
        actor = a;
        description =d;
    }
    public void changeDescription(String s){
        description = s;
    }
    public abstract void changeLocation();
    public abstract void changeLocation(Location l) throws HungryException;
    public abstract void changeLocation(Location l, Actor a) throws HungryException;
}

```

StaticAction.java

```
package com.kamikadze328.vk;
```

```

public class StaticAction extends Action{
    StaticAction(Creature a, String d){
        super(a, d);
    }
    StaticAction(CloseFriends a, String d){
        super(a, d);
    }
    StaticAction(MoominFamily a, String d){
        super(a, d);
    }
    StaticAction(Morra a, String d){
        super(a, d);
    }
    StaticAction(Hattifatteners a, String d){
        super(a, d);
    }
    StaticAction(Decoration a, String d){
        super(a, d);
    }
    StaticAction(Actor a, String d){
        super(a,d);
    }
    public void describe() {
        System.out.println(actor.getName() + " " + description);
    }
    @Override
    public void changeDescription(String s) {
        super.changeDescription(s);
    }

    @Override
    public void changeLocation(Location l) {
        if(actor.getLocation()==l){
            System.out.println(actor.getName() + " " + description + " Б " + l);
        }
    }

    @Override
    public void changeLocation() {
        System.out.println(actor.getName() + " " + description + " Б " +
actor.getLocation());
    }
}

```

```

@Override
public void changeLocation(Location l, Actor a) {
    if (actor.getLocation() == l) {
        System.out.println(a.getName() + " " + description + " " + actor.getName()
+ " в " + l);
    }
}
}

```

ChangeAction.java

```
package com.kamikadze328.vk;
```

```

public class ChangeAction extends Action implements ChangeFriends,
ChangeHattifatteners, ChangeMoomin, ChangeMorra, ChangeDecoration{
    ChangeAction(Creature a, String d){
        super(a, d);
    }
    ChangeAction(CloseFriends a, String d) {
        super(a, d);
    }
    ChangeAction(MoominFamily a, String d){
        super(a, d);
    }
    ChangeAction(Morra a, String d){
        super(a, d);
    }
    ChangeAction(Hattifatteners a, String d){
        super(a, d);
    }
    ChangeAction(Decoration a, String d){
        super(a, d);
    }

    @Override
    public void changeSize(int s) {
        System.out.printf(actor.getName() + " " + description +" до размера %d \n",
s);
        actorD.size = s;
    }

    @Override
    public void changeSize(int s, Actor a) {
        System.out.println(a.getName() + " " + description +" до размера "+ s + "
предмет "+actor.getName());
    }

    @Override
    public void changeDescription(String s) {
        super.changeDescription(s);
    }

    @Override
    public void changeHeight(int h, Actor a) {
        if(h>actorD.height){
            System.out.printf(a.getName() + " поднимает " + actor.getName() + " на
%d\n", h-actorD.height);
        }
        if (h<actorD.height&&h==0){
            System.out.println(a.getName() + " кладёт " + actor.getName() + " на
землю");
        }
        else{
            System.out.printf(a.getName() + " опускает " + actor.getName() + " на
%d\n", -h+actorD.height);
        }
    }
}

```

```

@Override
public void changeHeight(int h) {
    if(h>actorD.height){
        System.out.printf(actor.getName() + " поднялся на %d\n", h-actorD.height);
    }
    if(h<actorD.height&&h==0){
        System.out.println(actor.getName() + " упал на землю");
    }
    else{
        System.out.printf(actor.getName() + " опустился на %d\n", -
h+actorD.height);
    }
    actorD.height = h;
}

@Override
public void changeTail(boolean t) {
    actorCF.tail = t;
}

@Override
public void changeIntelligence(int i) {
    actorCF.intelligence = i;
}

@Override
public void changeStrength(int s){
    actorCF.strength = s;
}

@Override
public void changeCare(int c) {
    actorMF.care = c;
}

@Override
public void changeWaste(int w) {
    actorMF.waste = w;
}

@Override
public void changeHat(boolean h) {
    actorMF.hat = h;
}

@Override
public void changeCharge(int e) {
    actorH.electricCharge = e;
}

@Override
public void changeTemperature(int t) {
    actorM.temperature = t; }

@Override
public void changeLocation(Location l) throws HungryException {
    actor.setLocation(l);
    System.out.print(actor.getName() + " " + description + " в " + l + "\n");
    actor.setHunger(actor.getHunger()-1);
    if(actor.getHunger()< 8) throw new HungryException(actor);
}

@Override
public void changeLocation() {
    System.out.print(actor.getName() + " " + description +"\n");
}
@Override

```



```

    public void changeLocation(Location l, Actor a) throws HungryException {
        a.inventory.add(actor);
        System.out.print(a.getName() + " " + description + " " + actor.getName() + " в "
+ l + "\n");
        actor.setLocation(l);
        a.setLocation(l);
        a.inventory.remove(actor);
        a.setHunger(a.getHunger()-1);
        if(a.getHunger()< 8) throw new HungryException(actor);
    }
}

```

Actor.java

```
package com.kamikadze328.vk;
```

```
import java.util.LinkedList;
```

```

public class Actor {
    private String name;
    private int fun;
    private boolean isMainCharacter;
    protected Location location;
    private int hunger;
    class Inventory{
        LinkedList<Actor> inventory= new LinkedList<>();
        protected void add(Actor a){
            inventory.add(a);
            System.out.println(Actor.this.getName() + " поднимает " + a.getName());
        }
        protected void remove(Actor a){
            inventory.remove(a);
            System.out.println(Actor.this.getName() + " опускает " + a.getName());
        }
    }
    Inventory inventory = new Inventory();
    Actor(String n){
        name = n;
        fun = 1;
    }
    Actor(String n, Location l){
        name = n;
        location = l;
        fun = 1;
    }

    public String getName(){
        return name;
    }

    public int getHunger() {
        return hunger;
    }

    public void setHunger(int hunger) {
        this.hunger = hunger;
    }

    public Location getLocation(){
        return location;
    }
    public void setLocation(Location l){location = l;}

    public void setMainCharacter(boolean is){
        isMainCharacter= is;
    }

    public void setName(String name) {

```

```

        this.name = name;
    }

    public int getFun() {
        return fun;
    }

    public void setFun(int fun) {
        this.fun = fun;
    }
}

```

Decoration.java

```

package com.kamikadze328.vk;

public class Decoration extends Actor {
    protected int size;
    protected int height;
    Decoration(String n){
        super(n);
    }
    Decoration(String n, Location l){
        super(n,l);
    }
    Decoration(String n, Location l, int s, int h){
        super(n,l);
        size = s;
        height = h;
    }
}

```

Creature.java

```

package com.kamikadze328.vk;

public class Creature extends Actor {
    private String phrase;
    protected int hanger;
    Creature(String n, Location l){
        super(n,l);
    }
    Creature(String n, Location l, String p){
        super(n,l);
        phrase = p;
    }
    public void setPhrase(String aPhrase){
        phrase = aPhrase;
    }
    public void sayPhrase(){
        System.out.print(this.getName() + ": " + phrase + "\n");
    }
}

```

CloseFriends.java

```

package com.kamikadze328.vk;

public class CloseFriends extends Creature {
    protected int intelligence;
    protected int strength;
    protected boolean tail;
    CloseFriends(String n, Location l){
        super(n,l);
    }
    CloseFriends(String n, Location l, String p) {
        super(n, l, p);
    }
}

```

```

    CloseFriends(String n, Location l, String p, int i, int s, boolean t){
        super(n,l,p);
        intelligence = i;
        strength = s;
        tail = t;
    }

    public int getIntelligence() {
        return intelligence;
    }

    public int getStrength() {
        return strength;
    }
    public boolean getTail(){
        return tail;
    }
}

```

ChangeFriends.java

```

package com.kamikadze328.vk;

public interface ChangeFriends {
    void changeIntelligence(int i);
    void changeStrength(int s);
    void changeTail(boolean t);
}

```

Location.java

```

package com.kamikadze328.vk;

public enum Location {
    TopFloor,
    GroudFloor,
    Yard,
    Hill,
    Hangar,
    FootPath,
    Seamark,
    NaN
}

```

Territory.java

```

package com.kamikadze328.vk;

public enum Territory {
    TheCave,
    Moominhouse,
    TheLonelyMountain,
    TheForrest,
    TheSea,
    TheIslandOfDad
}

```

Game.java

```

package com.kamikadze328.vk;

public interface Game {
    void addGamer1(CloseFriends a, String name);
    void addGamer1(MoominFamily a, String name);
    void addGamer2(CloseFriends a, String name);
    void addGamer2(MoominFamily a, String name);
    void begin();
}

```

HungryException.java

```
package com.kamikadze328.vk;
```

```
public class HungryException extends Throwable {
    private String name;
    private Actor actor;

    HungryException(Actor a) {
        super(a.getName());
        name = a.getName();
        actor = a;
    }

    @Override
    public String getMessage() {
        return super.getMessage() + " голодный";
    }

    public String correction(){
        actor.setHunger(actor.getHunger() + 1);
        return name + " достаёт из кормана булочку и пожирает её";
    }
}
```

MyNullPointerException.java

```
package com.kamikadze328.vk;
```

```
public class MyNullPointerException extends java.lang.NullPointerException {
    MyNullPointerException(String message){
        super(message);
    }
}
```

Main.java

```
package com.kamikadze328.vk;
```

```
import java.util.ArrayList;
```

```
public class Main {
    public static void main(String[] args) {
        try{
            CloseFriends SnorkMaiden = new CloseFriends("Фрекен Снорк",
Location.TopFloor);
            CloseFriends Sniff = new CloseFriends("Снифф", Location.TopFloor);
            CloseFriends Snusmumrik = new CloseFriends("Снусмумрик",
Location.TopFloor);
            MoominFamily Moomintroll = new MoominFamily("Мумми-тролль",
Location.TopFloor);
            Game game = new Game(){
                ArrayList<String> gamers1Name = new ArrayList<>();
                ArrayList<Actor> gamers1 = new ArrayList<>();
                ArrayList<Actor> gamers2 = new ArrayList<>();
                ArrayList<String> gamers2Name= new ArrayList<>();
                public void addGamer1(CloseFriends a, String name){
                    gamers1Name.add(name);
                    gamers1.add(a);
                }
                public void addGamer1(MoominFamily a, String name){
                    gamers1Name.add(name);
                    gamers1.add(a);
                }
                public void addGamer2(CloseFriends a, String name){
```

```

        gamers2Name.add(name);
        gamers2.add(a);
    }
    public void addGamer2(MoominFamily a, String name){
        gamers2Name.add(name);
        gamers2.add(a);
    }
    public void begin(){
        for(int i=0; i<gamers1.size(); i++){
            System.out.print(gamers1Name.get(i) + ", ");
            System.out.print("начинают сражение против ");
            for(int i=0; i<gamers2.size(); i++){
                System.out.print(gamers2Name.get(i) + ", ");
                System.out.print("\nПобеждают ");
                if(Math.random()>0.5) {
                    for(int i=0; i<gamers1.size(); i++) {
                        System.out.print(gamers1.get(i).getName() + ", ");
                        gamers1.get(i).setFun(gamers1.get(i).getFun()+10);
                    }
                }
                else{
                    for(int i=0; i<gamers2.size(); i++) {
                        System.out.print(gamers2.get(i).getName() + ", ");
                        gamers2.get(i).setFun(gamers1.get(i).getFun()+10);
                    }
                }
            }
            System.out.print("\n\n");
        }
    };
    ((Game) game).addGamer1(SnorkMaiden, "Джейн");
    ((Game) game).addGamer1(Moomintroll, "Тарзан");
    ((Game) game).addGamer2(Sniff, "сын Тарзана");
    ((Game) game).addGamer2(Snusmumrik, "шимпанзе Чита");

    CloseFriends Khemul = new CloseFriends("Хемуль", Location.Hill, "*вздых*",
499, 50, false);
    StaticAction karaul = new StaticAction(Khemul, "караулит Панталашку");
    StaticAction yabloko = new StaticAction(Khemul, "ест яблоко");
    StaticAction schitat = new StaticAction(Khemul, "считает тычинки в цветке
джунглей");
    StaticAction smotret = new StaticAction(Khemul, "смотрит");
    StaticAction osenit = new StaticAction(Khemul, "выполняет мыслительные
процессы");
    StaticAction dozhd = new StaticAction(new Decoration("Дождь"), "перестал");
    ChangeAction sun = new ChangeAction(new Decoration("Солнце"), "зашло");
    Story moomin = new Story("kek", Khemul, Territory.Moominhouse, 18);
    ChangeAction holm = new ChangeAction(new Decoration("холм", Location.Hill,
10, 0), "увядает");
    ChangeAction plod = new ChangeAction(new Decoration("плод", Location.Hill,
3, 7), "сморщивается");
    ChangeAction tsvetok = new ChangeAction(new Decoration("цветок",
Location.Hill, 2, 2), "пониж");
    StaticAction list = new StaticAction(new Decoration("лист"), "сворачивается
трубочкой");
    Creature dom = new Creature("дом", Location.Hill, "*шорох и
потрескивание*");
    ChangeAction hodit = new ChangeAction(Khemul, "подходит к кусту");
    ChangeAction skhodit = new ChangeAction(Khemul, "идёт");
    Decoration vetka = new Decoration("ветка", Location.Hill, 1, 7);
    ChangeAction vetkaC = new ChangeAction(vetka, "");
    StaticAction vetkaS = new StaticAction(vetka, "ломается");
    Decoration vetki = new Decoration("ветки");
    ChangeAction sobrat = new ChangeAction(vetki, "собирает");

    StaticAction brat = new StaticAction(Khemul, "берёт спички");
    StaticAction koster = new StaticAction(new Decoration("костёр",
Location.FootPath, 6, 0), "пылает");

```

```

5, 0);

Decoration Khvost = new Decoration("хвост Панталошки", Location.FootPath,

ChangeAction sila = new ChangeAction(Khemul, "увеличивает");
ChangeAction tashit = new ChangeAction(Khvost, "тащит");
moomin.begin();
((Game) game).begin();
Khemul.sayPhrase();
if (Math.random() > 0.5) {
    yabloko.describe();
}
else schitat.describe();
moomin.setCurrentTime(19);
dozhd.describe();
sun.changeLocation();
holm.changeSize(5);
plod.changeSize(2);
plod.changeHeight(0);
tsvetok.changeSize(1);
list.describe();
dom.sayPhrase();
smotret.describe();
hodit.changeLocation();
vetkaC.changeHeight(5, Khemul);
vetkaS.describe();
osenit.describe();
sobrat.changeLocation(Location.Yard, Khemul);
skhodit.changeLocation(Location.Hangar);
brat.describe();
skhodit.changeLocation(Location.FootPath);
koster.describe();
sila.changeStrength(10000);
tashit.changeLocation(Location.Yard, Khemul);
} catch(HungryException e){
    System.out.println(e.getMessage());
    System.out.println(e.correction());
} finally{
    System.out.println("Плотно покушав, " + moomin.mainCharacter.getName() + " лёг спать");
    System.out.println("Вот и сказочке конец, а кто слушал молодец");
}
}
}

```

Вывод программы

В далёкой-далёкой галактике жил-был Хемуль
 Было у него много друзей
 И вот однажды в 18 часов в Moominhouse...

Джейн, Тарзан, начинают сражение против сын Тарзана, шимпанзе Чита,
 Побеждают Снифф, Снусмумрик,

Хемуль: *вздых*
 Хемуль считает тычинки в цветке джунглей
 19 часов
 Дождь перестал
 Солнце зашло
 холм увядает до размера 5
 плод сморщивается до размера 2
 плод упал на землю
 цветок поник до размера 1
 лист сворачивается трубочкой
 дом: *шорох и потрескивание*
 Хемуль смотрит
 Хемуль подходит к кусту
 Хемуль опускает ветка на 2
 ветка ломается
 Хемуль выполняет мыслительные процессы

Хемуль поднимает ветки
Хемуль собирает ветки в Yard
Хемуль опускает ветки
Хемуль идёт в Hangar
Хемуль берёт спички
Хемуль идёт в FootPath
Хемуль голодный
Хемуль достаёт из кормана булочку и пожирает её

Вывод: В далёкой-далёкой галактике жил-был мальчик, который сдавал все лабы вовремя...
Однако история умалчивает, что с ним случилось.