Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Компьютерных Технологий и Управления

Лабораторная работа №4

по дисциплине

«Программирование»

Вариант №1116.5

Выполнил:

студент группы R3136

Обудов Владислав Антонович

Преподаватель:

Гаврилов Антон Валерьевич

Санкт-Петербург

1. г.
2. Текст задания.

Изображение выглядит как снимок экрана

Автоматически созданное описание

[*Доктор Пилюлькин сказал, чтоб он вел себя неспокойнее, так как его организм ослаблен болезнью и ему нужно беречь силы. Вскоре радость Незнайки понемногу утихла и сменилась нетерпением. Незнайка ложился в постель, но, полежав минуточку, вскакивал: В те дни в Космическом городке гостили астроном Альфа и лунолог Мемега и приехавшие вместе с ними два физика Квантик и Кантик. Все четверо приехали специально, чтоб познакомиться с устройством космической ракеты и скафандров, так как сами собирались построить ракету и совершить космический полет к Земле. Теперь, кота тайна невесомости была раскрыта, межпланетные полеты стали доступны и для лунатиков. Знайка решил подарить лунным ученым точные чертежи ракеты и велел, чтоб им отдали оставшиеся запасы лунита и антилунита. Альфа сказал, что лунные ученые сохранят Космический городок в порядке и устроят здесь космодром с площадкой для посадки прибывающих на их планету космических кораблей и для запуска ракет на другие планеты. Кода космонавты пришли к решению возвратиться на Землю, Знайка, Фуксия и Селедочка отправились в ангар, чтобы произвести тщательную проверку работы всех узлов и механизмов ракеты. В проверке участвовали и Альфа с Мемегой, а также Кантик и Квантик. Для них зто было чрезвычайно полезно, так как они получили возможность практически ознакомиться с устройством ракеты. К тому же было решено, что Альфа и Мемега совершат полет на ракете вместе с космонавтами. Достигнув поверхности Луны, космонавты пересядут в ракету НИП, а Альфа с Мемегой возвратятся на ракете ФИС обратно в Космический городок. Проверка механизмов ракеты заняла все оставшееся в распоряжении космонавтов время и закончилась только к вечеру.*]

1. Диаграмма классов объектной модели.

Изображение выглядит как текст

Автоматически созданное описание

1. Код программы.

public abstract class ALocation {

private String Name;

public ALocation(String Name) {

this.Name = Name;

}

public String getName() {

return Name;

}

}

public abstract class AThing {

private String Type;

private Shorty Owner;

public AThing(String Type, Shorty Owner) {

this.Type = Type;

this.Owner = Owner;

}

public String getType() {

return Type;

}

public void newOwner(Shorty Owner) {

this.Owner = Owner;

}

}

public class BasicException extends Exception {

private String e;

BasicException(String s){

super(s);

e = s;

}

public String getE() {

return e;

}

}

public class InitException extends NullPointerException {

private String trouble = " не проинициализировано";

InitException(){

trouble = "Непроинициализированная сущность";

}

InitException(String s){

super(s);

trouble = s + trouble;

}

InitException(String s, String s1){

super(s);

trouble = s + s1;

}

public String getE() {

return trouble;

}

}

public interface IDefault {

public int hashCode();

public String toString();

public boolean equals(Object obj);

}

public interface IShorty {

public void LayDown(Furniture f);

public void GetUp();

public void Give(AThing t, Shorty s);

public void MoveTo(ALocation l);

public void MoveTo(ALocation l, String r);

public void Say(String s);

}

enum Planet {

Earth,

Moon,

Other;

}

enum Position {

Astronom,

Lunolog,

Fizik,

Doctor,

None;

}

import java.util.ArrayList;

public class lab3 {

public static void main(String[] args) {

Place MDefaultTown = new Place("другой лунный город", Planet.Moon);

Place MSpaceTown = new Place("Космический городок на Луне", Planet.Moon);

Place MoonSide = new Place("Поверхность Луны", Planet.Moon);

Place ESpaceTown = new Place("Космический городок на Земле", Planet.Earth);

Rocket RocketFIS = new Rocket("ФИС", MSpaceTown);

Rocket.Engine eng = RocketFIS.new Engine();

Rocket RocketNIP = new Rocket("НИП", MoonSide);

Building SpaceStation = new Building("космодром", MSpaceTown);

Building Angar = new Building("ангар", MSpaceTown);

Building DrHouse = new Building("дом Пилюлькина", MSpaceTown);

Shorty DrPilul = new Shorty("Пилюлькин", Position.Doctor, DrHouse);

Shorty Neznaika = new Shorty("Незнайка", Position.None, DrHouse);

Shorty Znaika = new Shorty("Знайка", Position.None, MSpaceTown);

Shorty Alpha = new Shorty("Альфа", Position.Astronom, MDefaultTown);

Shorty Memega = new Shorty("Мемега", Position.Lunolog, MDefaultTown);

Shorty Kwantik = new Shorty("Квантик", Position.Fizik, MDefaultTown);

Shorty Kantik = new Shorty("Кантик", Position.Fizik, MDefaultTown);

Shorty Fuksia = new Shorty("Фуксия", Position.None, SpaceStation);

Shorty Seld = new Shorty("Селедочка", Position.None, SpaceStation);

Thing Lunit = new Thing("лунит", Znaika);

Thing AntiLunit = new Thing("антилунит", Znaika);

Thing Drawings = new Thing("чертежи", Znaika);

Thing SpaceSuits = new Thing("скафандры", Znaika);

Furniture Bed = new Furniture("кровать", DrPilul);

ArrayList<Shorty> MScientists = new ArrayList<>();

MScientists.add(Alpha);

MScientists.add(Memega);

MScientists.add(Kwantik);

MScientists.add(Kantik);

ArrayList<Shorty> EVisiters = new ArrayList<>();

EVisiters.add(Znaika);

EVisiters.add(Neznaika);

ArrayList<Shorty> Mechanics = new ArrayList<>();

Mechanics.add(Znaika);

Mechanics.add(Fuksia);

Mechanics.add(Seld);

for(Shorty s:MScientists) {

s.MoveTo(MSpaceTown, "познакомиться с устройством ракеты и "+SpaceSuits.getType());

}

Neznaika.changeMood("радостное");

DrPilul.Say(Neznaika.getName()+" веди себя спокойнее, твой организм ослаблен болезнью, береги силы.");

Neznaika.changeMood("нетерпеливое");

Neznaika.LayDown(Bed);

Neznaika.GetUp();

Znaika.Give(Lunit, Alpha);

Znaika.Give(AntiLunit, Alpha);

Znaika.Give(Drawings, Alpha);

Alpha.Say("Мы сохраним "+MSpaceTown.getName()+" в порядке и устроим здесь "+SpaceStation.getName()+" с площадкой для посадки ракет.");

for(Shorty s:Mechanics) {

s.MoveTo(Angar, "проверить ракеты");

}

for(Shorty s:MScientists) {

s.MoveTo(Angar, "проверить ракеты");

}

eng.Repair(Znaika);

ArrayList<Shorty> pasFIS = new ArrayList<>();

pasFIS.add(Alpha);

pasFIS.add(Memega);

pasFIS.add(Znaika);

pasFIS.add(Neznaika);

RocketFIS.Fly(MoonSide, pasFIS);

pasFIS.remove(Znaika);

pasFIS.remove(Neznaika);

RocketNIP.Fly(ESpaceTown, EVisiters);

RocketFIS.Fly(MoonSide, pasFIS);

}

}

public class Building extends ALocation {

private Place Location;

public Building(String Name, Place Location) {

super(Name);

try {

if (Location == null) throw new InitException("Локация");

this.Location = Location;

} catch(InitException e) {

e.printStackTrace();

this.Location = new Place("???", Planet.Other);

}

}

public Place getPlace() {

return Location;

}

}

public class Furniture extends AThing {

public Furniture(String Type, Shorty Owner) {

super(Type, Owner);

}

}

public class Place extends ALocation{

private Planet Location;

public Place(String Name, Planet Location) {

super(Name);

try {

if (Location == null) throw new InitException("Планеты"+" не существует");

this.Location = Location;

} catch(InitException e) {

e.printStackTrace();

this.Location = Planet.Other;

}

}

public Planet getPlanet() {

return Location;

}

}

import java.util.ArrayList;

public class Rocket {

private String Name;

private Place Location;

private int flightCount = 0;

private static int maxPasCount = 10;

private boolean status = true;

public Rocket(String Name, Place Location) {

this.Name = Name;

this.Location = Location;

flightCount = 0;

}

public String getName() {

return Name;

}

public static class Drawing {

public static int getMaxPasCount() {

return maxPasCount;

}

}

public class Engine {

public boolean getStatus() {

return status;

}

public void Repair(Shorty s) {

try {

if (s == null) throw new InitException("Коротышка");

if (status == false) {

status = true;

System.out.println("Коротышка "+s.getName()+" починил ракету.");

} else System.out.println(s.getName()+": ракета в норме, ремонт не требуется.");

} catch(InitException e) {

e.printStackTrace();

System.out.println("Никто не приступил к починке ракеты.");

}

}

}

public void Fly(Place pl, ArrayList<Shorty> pas) {

class Flight {

boolean Start() {

try {

if (pl == null) throw new InitException("Точка прибытия"+" не существует.");

if (pas == null) throw new InitException("Список пассажиров ракеты");

if (Location.equals(pl)) throw new BasicException("Ракета "+getName()+" не может полететь туда, где она уже находится.");

if (pas.size()==0) throw new BasicException("Ракета "+getName()+" не может полететь без пассажиров.");

return true;

} catch(InitException eI) {

System.out.println(eI.getE());

System.out.println("Полет отменяется.");

return false;

} catch(BasicException eB) {

System.out.println(eB.getE());

System.out.println("Полет отменяется.");

return false;

}

}

}

Flight \_flight = new Flight();

if (\_flight.Start()) {

System.out.println("Ракета "+Name+" совершила полет к "+pl.getName()+".");

for(Shorty s:pas) {

s.MoveTo(pl);

}

flightCount++;

if (flightCount > 5) {

status = false;

flightCount = 0;

}

}

}

}

public class Shorty implements IDefault, IShorty {

private String Name; //имя

private Position currentP; //должность

private String currentL; //местонахождение

private String status; //положение тела

private String mood; //настроение

private String reason; //причина местонахождения, по умолчанию - ее нет, отвечает на вопрос "зачем?"

public Shorty(String Name, Position p, ALocation pl) {

this.Name = Name;

status = "стоит";

mood = "нормальное";

reason = "";

try {

if (p == null) throw new InitException("Должности"+" не существует");

currentP = p;

} catch(InitException e) {

e.printStackTrace();

currentP = Position.None;

}

try {

if (pl == null) throw new InitException("Место");

currentL = pl.getName();

} catch(InitException e) {

e.printStackTrace();

currentL = "???";

}

}

public String getName() {

return Name;

}

public void changeMood(String s) {

mood = s;

System.out.println("Настроение "+Name+" сменилось на "+s);

}

public String getLocation() {

return currentL;

}

@Override

public void Say(String s) {

System.out.println(Name + " сказал: " + s);

}

@Override

public void LayDown(Furniture f) {

try {

if (f == null) throw new InitException("Мебель");

status = "лежит на " + f.getType();

System.out.println(Name + " лег на " + f.getType());

} catch(InitException e) {

e.printStackTrace();

System.out.println("Коротышка "+Name+" ничего не сделал.");

}

}

@Override

public void GetUp() {

status = "стоит";

System.out.println(Name + " встал");

}

@Override

public void Give(AThing t, Shorty s) {

try {

if (t == null) throw new InitException("Вещь ");

if (s == null) throw new InitException("Коротышка ");

if (s.equals(this)) throw new BasicException("Коротышка "+Name+" не может передать предмет себе");

t.newOwner(s);

System.out.println(Name + " передал(а) " + s.getName() + " предмет " + t.getType());

} catch(InitException e) {

e.printStackTrace();

System.out.println("Передача предмета "+t.getType()+" коротышке "+s.getName()+" не произошла.");

} catch(BasicException e) {

e.printStackTrace();

System.out.println("Передача предмета "+t.getType()+" самому себе не произошла.");

}

}

@Override

public void MoveTo(ALocation l) {

try {

if (l == null) throw new InitException("Локация");

currentL = l.getName();

System.out.println(Name + " приехал в " + l.getName());

} catch(InitException e) {

e.printStackTrace();

}

}

@Override

public void MoveTo(ALocation l, String r) {

try {

if (l == null) throw new InitException("Локация");

currentL = l.getName();

reason = r;

System.out.println(Name + " приехал в " + l.getName() + ", чтобы " + r);

} catch(InitException e) {

e.printStackTrace();

}

}

@Override

public int hashCode() {

return Name.hashCode() + currentP.hashCode() + currentL.hashCode() + status.hashCode() + mood.hashCode() + reason.hashCode();

}

@Override

public String toString() {

return currentP + " " + Name + " " + status + ", настроение: " + mood;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

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

return false;

Shorty other = (Shorty) obj;

if (Name != other.Name)

return false;

if (currentP != other.currentP)

return false;

if (currentL != other.currentL)

return false;

if (status != other.status)

return false;

if (mood != other.mood)

return false;

if (reason != other.reason)

return false;

return true;

}

}

public class Thing extends AThing {

public Thing(String Type, Shorty Owner) {

super(Type, Owner);

}

}

1. Результат работы программы.

Альфа приехал в Космический городок на Луне, чтобы познакомиться с устройством ракеты и скафандры

Мемега приехал в Космический городок на Луне, чтобы познакомиться с устройством ракеты и скафандры

Квантик приехал в Космический городок на Луне, чтобы познакомиться с устройством ракеты и скафандры

Кантик приехал в Космический городок на Луне, чтобы познакомиться с устройством ракеты и скафандры

Настроение Незнайка сменилось на радостное

Пилюлькин сказал: Незнайка веди себя спокойнее, твой организм ослаблен болезнью, береги силы.

Настроение Незнайка сменилось на нетерпеливое

Незнайка лег на кровать

Незнайка встал

Знайка передал(а) Альфа предмет лунит

Знайка передал(а) Альфа предмет антилунит

Знайка передал(а) Альфа предмет чертежи

Альфа сказал: Мы сохраним Космический городок на Луне в порядке и устроим здесь космодром с площадкой для посадки ракет.

Знайка приехал в ангар, чтобы проверить ракеты

Фуксия приехал в ангар, чтобы проверить ракеты

Селедочка приехал в ангар, чтобы проверить ракеты

Альфа приехал в ангар, чтобы проверить ракеты

Мемега приехал в ангар, чтобы проверить ракеты

Квантик приехал в ангар, чтобы проверить ракеты

Кантик приехал в ангар, чтобы проверить ракеты

Знайка: ракета в норме, ремонт не требуется.

Ракета ФИС совершила полет к Поверхность Луны.

Альфа приехал в Поверхность Луны

Мемега приехал в Поверхность Луны

Знайка приехал в Поверхность Луны

Незнайка приехал в Поверхность Луны

Ракета НИП совершила полет к Космический городок на Земле.

Знайка приехал в Космический городок на Земле

Незнайка приехал в Космический городок на Земле

Ракета ФИС совершила полет к Поверхность Луны.

Альфа приехал в Космический городок на Луне

Мемега приехал в Космический городок на Луне

1. Выводы по работе.

В ходе выполнения данной лабораторной работы я применил на практике свои знания об ООП по принципам SOLID, написал код на языке Java по заданной предметной области, реализовал 2 интерфейса IDefault и IShorty, 2 абстрактных класса ALocation и AThing, 2 перечисляемых типа (enum), 2 класса исключений BasicException (checked) и InitException (unchecked), а так же вложенные классы в классе Rocket: Drawing (статический), Flight (локальный) и Engine (внутренний), переопределил методы hashCode(), equals(), toString() в классе Shorty, понял для чего нужны вложенные анонимные классы, но не смог найти им применение в своей работе.