### Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Центр программной инженерии

ОТЧЕТ

к лабораторной работе №4 по дисциплине:

|  |
| --- |
| «Объектно-ориентированное программирование» |
| Разработка классов с использованием механизмов  наследования, полиморфизма и инкапсуляции. |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб-19-2 |  |  |  | Качалин Д.В. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Маланова Т.В. |
|  | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2020 г.

Оглавление

[1 Постановка задачи 3](#_Toc32783644)

[2 Алгоритм 4](#_Toc32783645)

[3 Таблица спецификации 5](#_Toc32783646)

[4 Таблица тестов 7](#_Toc32783647)

[5 Результаты тестирования 10](#_Toc32783648)

[6 Исходный код 11](#_Toc32783649)

[Список использованных источников 16](#_Toc32783650)

1 Постановка задачи

В соответствии с индивидуальным заданием описать иерархию классов, для каждого класса описать поля и соответствующие методы доступа к ним. В зависимости от задания некоторые из этих классов (как минимум один) являются абстрактными и служат для выделения общих данных и поведения для других классов. Абстрактный класс должен содержать как минимум один абстрактный метод, реализация которого у его наследников должна различаться. Помимо этого, в общую часть задания входит разработка класса, группирующего объекты описанных в соответствии с заданием классов. Для первого варианта индивидуального задания это может быть класс «кафедра». Для выполнения задания необходимо создать некоторое количество объектов, добавить их в группу используя предусмотренные методы класса-«группы» и для каждого из них вызвать унаследованный метод.Вариант 8.

место, область, город, столица;

2 Блок-схема алгоритма

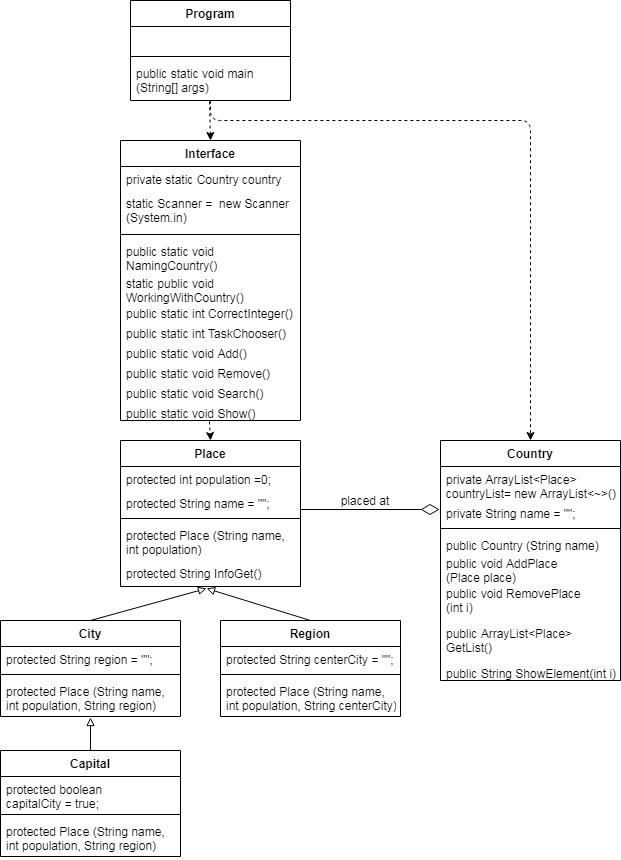


Рисунок 1.1 – Иерархия классов

3 Таблица спецификации

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Описание переменных | Тип данных |
| **Класс Program** | | | |
| **Метод main – основная программа** | | | |
| 1 | I | Экземпляр класса Interface | Interface |
| **Класс Place** | | | |
| 1 | name | Наименование места | String |
| 2 | population | Количество населения, живущего в данном месте | int |
| **Класс Region (наследует Place)** | | | |
| 3 | centerCity | Центральный город региона | String |
| **Класс City (наследует Place)** | | | |
| 3 | region | Название региона, где находится данный город | String |
| **Класс Capital (наследует City)** | | | |
| 4 | capitalCity | Показывает, является ли город столицей | boolean |
| **Класс Country** | | | |
| 1 | countryList | Список мест, которые находятся в данной стране | ArrayList <Place> |
| 2 | name | Название страны | String |
| **Метод AddPlace (Place place) – метод, добавляющий место в список** | | | |
| 3 | place | Место для добавления в список | Place |
| **Класс Interface** | | | |
| 1 | country | Экземпляр класса Country с которой будет работать пользователь программы | Country |
| 2 | console | Экземпляр класс Scanner, для считывания информации, введенной в консоли | Scanner |
| **Метод Repopulation() – метод, меняющий количество населения места** | | | |
| 1 | name | Имя место, по которому будет осуществляться поиск | String |
| 2 | b | Показывает, нашла ли программа нужное место | boolean |
| **Метод CorrectInteger() – метод для корректного ввода целого числа** | | | |
| 1 | c | Целое число, вводимое пользователем | int |
| **Метод TaskChooser(int d) – метод для определения операции, которую выбрал пользователь** | | | |
| 1 | d | Количество операций для выбора пользователем | int |
| 2 | c | Операция, которую выберет пользователь | int |
| **Метод Add() – метод, добавляющий место в список (страну)** | | | |
| 1 | place | Мест, которое будем добавлять в список | Place |
| 2 | name | Название нового места | String |
| 3 | population | Количество населения в новом месте | int |
| 4 | regionName | Название региона для места (только для экземпляров класс City и Capital)  Продолжение таблицы 1 | String |
| **Метод Remove() – метод, удаляющий место из списка (страны)** | | | |
| 1 | name | Имя места, по которому будет осуществляться поиск | String |
| **Метод Search() – метод для поиска места в списке (стране) и выводе всей информации о нем** | | | |
| 1 | name | Имя места, по которому будет осуществляться поиск | String |

4 Таблица тестов

Таблица 2

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| 1 | Country.name = Страна1 | Name your country  Страна1  What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation |
| 2 | Выбор задания:  TaskChooser =1 (Добавление места)  Выбор какое место вводить  TaskChooser =1 (Регион)  name = регион1  population = 546789  centerCity = город1 | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  1  What do you want to add?  1 - region  2 - city  1  Enter a name  регион1  Enter a number of population  546789  Enter a center city name  город1 |
| 3 | Выбор задания:  TaskChooser =1 (Добавление места)  Выбор какое место вводить  TaskChooser =2 (город)  Выбор между столицей и обычным городом  TaskChooser = 1 (Столица)  name = город2  population = 654321  region = регион2 | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  1  What do you want to add?  1 - region  2 - city  2  Enter a name  город2  Enter a number of population  654321  Enter a region name  регион2  Is this city central city of region?  1 - yes  2 - no  1 |
| 4  Продолжение таблицы 2 | Выбор задания:  TaskChooser =1 (Добавление места)  Выбор какое место вводить  TaskChooser =2 (город)  Выбор между столицей и обычным городом  TaskChooser = 1 (Столица)  name = город3  population = 864322  region = регион3 | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  1  What do you want to add?  1 - region  2 - city  2  Enter a name  город3  Enter a number of population  864322  Enter a region name  регион3  Is this city central city of region?  1 - yes  2 - no  2 |
| 5 | Выбор задания:  TaskChooser =4 (показать список) | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  4  -регион1  -город2  -город3 |
| 6 | Выбор задания:  TaskChooser =3 (поиск)  name = город2 | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  3  Enter place name  город2  name - город2  population - 654321  region - регион2  true |
| 7  Продолжение таблицы 2 | Выбор задания:  TaskChooser =6 (перенаселение)  name = город2  population = 1111111  Выбор задания:  TaskChooser =3 (поиск)  name = город2 | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  6  Enter place name  город2  Enter new population  1111111  What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  3  Enter place name  город2  name - город2  population - 1111111  region - регион2  true |
| 8 | Выбор задания:  TaskChooser =2 (удаление)  name = регион1  Выбор задания:  TaskChooser =4 (показать список) | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  2  Enter name of place that you want to delete from your country  регион1  Deleted successful  What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  4  -город2  -город3 |
| 9 | Выбор задания:  TaskChooser =5 (завершить работу программы) | What do you want to do?  1 - add  2 - remove  3 - search  4 - show all list  5 - close  6 - repopulation  5  Process finished with exit code 0 |

5 Результаты тестирования

Продолжение таблицы 2

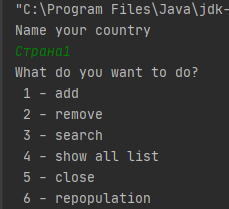


Рисунок 2.1 – Тест 1 (создание страны-списка)

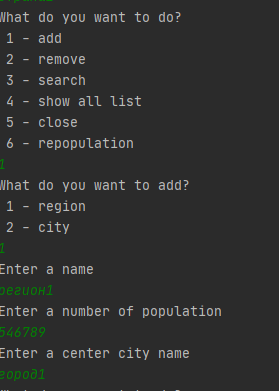


Рисунок 2.2 – Тест 2 (создание и добавление региона в список)

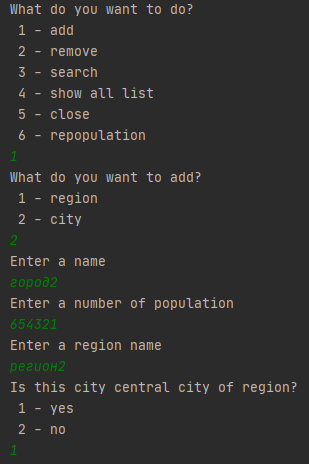


Рисунок 2.3 – Тест 3 (создание и добавление столицы в список)

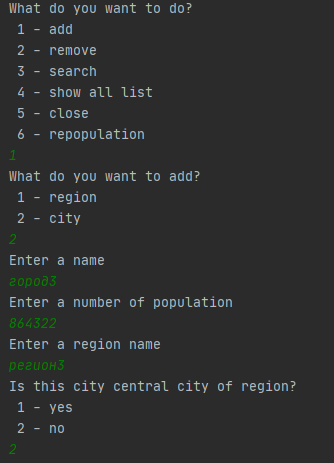


Рисунок 2.4 – Тест 4 (создание и добавление города в список)

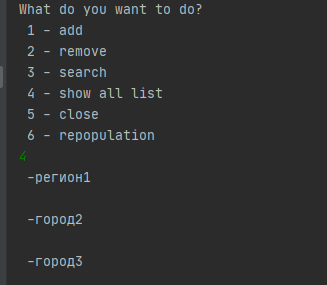


Рисунок 2.5 – Тест 5 (Показ всего списка)

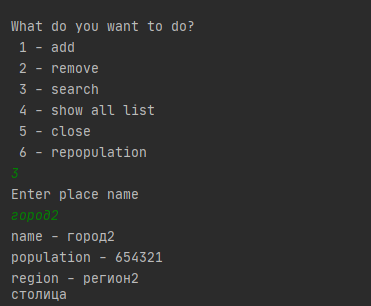


Рисунок 2.6 – Тест 6 (поиск и вывод информации о месте)

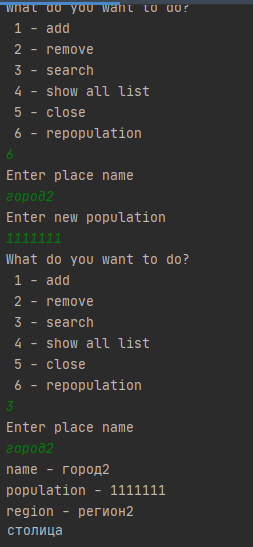


Рисунок 2.7 – Тест 7(изменение населения для места)

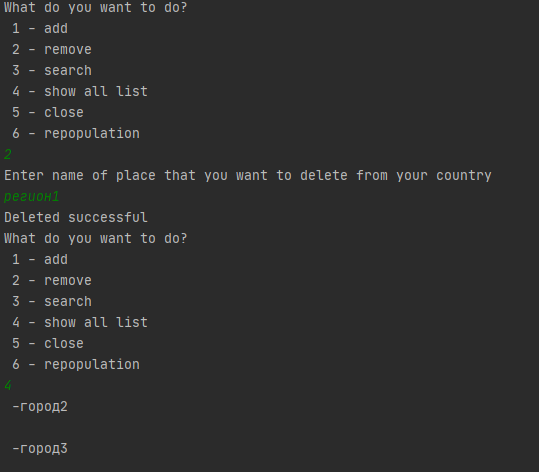


Рисунок 2.8 – Тест 8 (удаление места из списка)

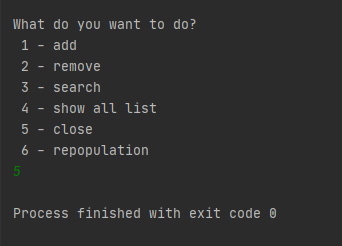


Рисунок 2.9 – Тест 9 (завершение работы программы)

6 Исходный код

Class Program

public class JavaApplication8 {

public static void main(String[] args) {

Interface I = new Interface();

I.NamingCountry();

}}

Class Place

public abstract class Place {

protected int population = 0;

protected String name = "";

protected Place (String name, int population){

this.name = name;

this.population = population;

}

protected String InfoGet(){

return String.format("name - %s\npopulation - %s",this.name,this.population);

}

protected void Repopulation (int population){

this.population = population;

}

Class Region

public class Region extends Place {

protected String centerCity = "";

protected Region(String name, int population, String centerCity){

super(name, population);

this.centerCity = centerCity;

}

@Override

protected String InfoGet(){

return super.InfoGet() + String.format("\ncentral city - %s",this.centerCity);

}}

Class City

public class City extends Place {

protected String region = "";

protected City (String name, int population, String region){

super(name, population);

this.region = region;

}

@Override

protected String InfoGet(){

return super.InfoGet() + String.format("\nregion - %s",this.region);

}}

Class Capital

public class Capital extends City{

protected boolean capitalCity = true;

protected Capital (String name, int population, String region){

super(name, population, region);

}

@Override

protected String InfoGet(){

return super.InfoGet() + "\n" + "Столица";

}}}

Class Country

public class Country {

private ArrayList<Place> countryList= new ArrayList<Place>();

private String name;

public Country(String name){

this.name = name;

}

public void AddPlace(Place place){

countryList.add(place);

}

public void RemovePlace(int i){

countryList.remove(i);

}

public ArrayList<Place> GetList(){

return this.countryList;

}

public String ShowElement(int i){

return this.countryList.get(i).InfoGet();

}}

Class Interface

import java.util.ArrayList;

import java.util.Scanner;

public class Interface {

private static Country country;

static Scanner console = new Scanner(System.in);

public static void NamingCountry() {

System.out.println("Name your country");

country = new Country(console.nextLine());

WorkingWithCountry();

}

public static void WorkingWithCountry(){

System.out.println("What do you want to do?\n 1 - add\n 2 - remove\n 3 - search\n 4 - show all list\n 5 - close\n 6 - repopulation");

switch(TaskChooser(6)){

case 1:

Add();

break;

case 2:

Remove();

break;

case 3:

Search();

break;

case 4:

Show();

break;

case 5:

return;

case 6:

Repopulation();

}

WorkingWithCountry();

return;

}

public static void Repopulation(){

System.out.println("Enter place name");

String name = console.nextLine();

boolean b = false;

for (int i = 0; i < country.GetList().size(); i ++){

if (name.equals(country.GetList().get(i).name)){

System.out.println("Enter new population");

int population = CorrectInteger();

country.GetList().get(i).Repopulation(population);

b = true;

}}

if (!b){

System.out.println("Nothing was found");

}}

public static int CorrectInteger(){

try {

int c = Integer.parseInt(console.nextLine());

return c;

}

catch (NumberFormatException e){

System.out.println("Incorrect enter, repeat");

return CorrectInteger();

}}

public static int TaskChooser(int d){

int c;

c = CorrectInteger();

if (c > d){

System.out.println("Incorrect enter, repeat");

return TaskChooser(d);

}

if (c < 1){

System.out.println("Incorrect enter, repeat");

return TaskChooser(d);

}

return c;

}

public static void Add(){

Place place = new Region("", 0, "");

System.out.println("What do you want to add?\n 1 - region\n 2 - city");

String name;

int population;

switch (TaskChooser(2)) {

case 1:

System.out.println("Enter a name");

name = console.nextLine();

System.out.println("Enter a number of population");

population = CorrectInteger();

System.out.println("Enter a center city name");

place = new Region(name, population, console.nextLine());

break;

case 2:

System.out.println("Enter a name");

name = console.nextLine();

System.out.println("Enter a number of population");

population = CorrectInteger();

System.out.println("Enter a region name");

String regionName = console.nextLine();

System.out.println("Is this city central city of region?\n 1 - yes\n 2 - no");

int z = TaskChooser(2);

if (z == 1) {

place = new Capital(name, population, regionName);

} else if (z == 2) {

place = new City(name, population, regionName);

}

break;

}

country.AddPlace(place);

}

public static void Remove(){

System.out.println("Enter name of place that you want to delete from your country");

String name = console.nextLine();

for (int i = 0; i < country.GetList().size(); i++){

if (name.equals(country.GetList().get(i).name)){

country.RemovePlace(i);

System.out.println("Deleted successful");

}}}

public static void Search(){

System.out.println("Enter place name");

String name = console.nextLine();

boolean b = false;

for (int i = 0; i < country.GetList().size(); i ++){

if (name.equals(country.GetList().get(i).name)){

System.out.println(country.ShowElement(i));

b = true;

}}

if (!b){

System.out.println("Nothing was found");

}}

public static void Show (){

for (int i = 0; i < country.GetList().size(); i++){

System.out.println(" -" + country.GetList().get(i).name + "\n");

}}

Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во

ИРНИТУ, 2017. – 24 c.

1. CyberForum.ru – Информация о языке Java
2. ProgLand.su – Информация о языке Java