**МИНИСТЕРСТВО**

**ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
 УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине: «**Визуальные средства разработки программных приложений»**

на тему: «Построение модели программной системы»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.  
 Принял: ассистент

Михалевич В.Г.

Гомель 2022

**Цель работы:** научиться создавать модели программных систем и реализовывать их с использованием языка программирования *Java*.

Место, область, мегаполис, город.

1. Определить иерархию классов (в соответствии с вариантом).

2. Определить в классе статическую компоненту-ссылку на начало связанного списка объектов и статическую функцию для просмотра списка.

3. Реализовать классы.

4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаотся в список, после чего список просматривается.

5. Pеализовать вариант, когда объект добавляется в список при создании, в конструкторeе.

Меню приложения предоставляет возможности по добавлению места, области, мегаполиса, города а также выводу списка сущностей и представлено на рисунке 1.

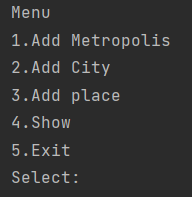


Рисунок 1 – Меню программы

Возможность добавления сущности представлена на рисунке 2.

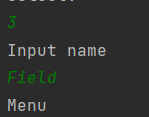


Рисунок 2 – Добавление сущности

Приложение предоставляет возможность по выводу списка добавленных сущностей с детализацией информации и представлена на рисунке 5.

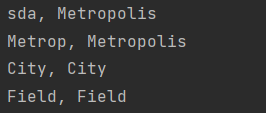


Рисунок 5 – Вывод списка

Листинг программы представлен в приложении А.

**Вывод:** в ходе выполнения лабораторной работы, были изучены построение иерархии и распределение обязанностей между классами.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

package com.company;  
  
import models.\*;  
  
import java.util.Arrays;  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
 private static Scanner \_scanner = new Scanner(System.in);  
 public static void main(String[] args) {  
 while (true) {  
 var choice = getChoice(Arrays.asList("1.Add Metropolis", "2.Add City", "3.Add Field", "4.Add place", "5.Show","6.Exit"), 6);  
 switch (choice) {  
 case 1:  
 var type = "Metropolis";  
 var metropolis = new Metropolis(getString("Input name"),type);  
 metropolis.Add();  
 break;  
 case 2:  
 type = "City";  
 var city = new City(getString("Input name"),type);  
 city.Add();  
 break;  
 case 3:  
 type = "Field";  
 var field = new Field(getString("Input name"),type);  
 field.Add();  
 break;  
 case 4:  
 type = "Place";  
 var place = new Place(getString("Input name"),type);  
 place.Add();  
 break;  
 case 5:  
 Entity.ShowList();  
 break;  
 case 6:  
 return;  
 default:  
 break;  
 }  
 }  
 }  
 private static String getString(String message) {  
 System.out.println(message);  
 return \_scanner.nextLine();  
 }  
  
  
 private static int getChoice(List<String> menuItems, int maxInput) {  
 System.out.println("Menu");  
 for (String item:  
 menuItems) {  
 System.out.println(item);  
 }  
 System.out.println("Select:");  
 int choice;  
 while (true) {  
 try {  
 choice = Integer.parseInt(\_scanner.nextLine());  
 if (choice < 1 || choice > maxInput) throw new Exception("");  
 break;  
 } catch (Exception exception) {  
 System.out.println("Incorrect input");  
 }  
 }  
 return choice;  
 }  
}

package models;  
  
import java.util.LinkedList;  
  
public abstract class Entity {  
 public static LinkedList<Entity> Entities = new LinkedList<>();  
 protected String \_name;  
 protected String \_type;  
  
 public Entity(String name, String type){  
 \_name = name;  
 \_type = type;  
 }  
  
 public void Add(){  
 if(Entities.indexOf(this) == -1) {  
 Entities.add(this);  
 }else{  
 System.out.println("Already added to list");  
 }  
 }  
  
 public void Show(){  
 System.out.println(\_name + ", " + \_type);  
 }  
  
 public static void ShowList(){  
 for (var item:Entities) {  
 item.Show();  
 }  
 }  
  
}

package models;  
  
public class City extends Entity{  
  
 public City(String name, String type){  
 super(name,type);  
 }  
  
 @Override  
 public void Show(){  
 System.out.println(\_name + ", " + \_type);  
 }  
}

package models;  
  
public class Field extends Entity {  
  
 public Field(String name, String type){  
 super(name,type);  
 }  
 @Override  
 public void Show(){  
 System.out.println(\_name + ", " + \_type);  
 }  
}

package models;  
  
public class Metropolis extends Entity{  
 public Metropolis(String name, String type){  
 super(name,type);  
 }  
  
 @Override  
 public void Show(){  
 System.out.println(\_name + ", " + \_type);  
 }  
}

package models;  
  
public class Place extends Entity{  
  
 public Place(String name, String type){  
 super(name,type);  
 }  
  
 @Override  
 public void Show(){  
 System.out.println(\_name + ", " + \_type);  
 }  
}