Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №2

По дисциплине: “ЯП”

Тема: “Наследование и виртуальные фукции”

Выполнил:

Студент 2-го курса

Группы ПО-7

Смушко О.Р.

Дата выполнения: 15.09.21

Проверил:

Бойко Д.О.

Брест, 2021

**Вариант 8**

**Порядок выполнения работы.**

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

2. Определить в классе статическую компоненту - указатель на нача-

ло связанного списка объектов и статическую функцию для просмотра

списка.

3. Реализовать классы – место, область, город, мегаполис.

4. Написать демонстрационную программу, в которой создаются

объекты различных классов и помещаются в список, после чего список

просматривается.

5. Сделать соответствующие методы не виртуальными и посмотреть,

что будет.

6. Реализовать вариант, когда объект добавляется в список при соз-

дании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

**Определения классов:**

**place.h**

#ifndef LAB\_2\_PLACE\_H  
#define LAB\_2\_PLACE\_H  
  
#include <iostream>  
#include <string>  
#include <list>  
  
using namespace std;  
  
class Place{  
protected:  
 string name;  
public:  
 Place(string name);  
 virtual void show() = 0;  
};  
#endif //LAB\_2\_PLACE\_H

**region.h**

#ifndef LAB\_2\_REGION\_H  
#define LAB\_2\_REGION\_H  
#include "place.h"  
  
class Region: public Place {  
protected:  
 int cityCount;  
public:  
 Region(string name, int cityCount);  
 void show();  
 void set\_count(int cityCount);  
 int get\_count();  
};  
  
#endif //LAB\_2\_REGION\_H

**town.h**

#ifndef LAB\_2\_TOWN\_H  
#define LAB\_2\_TOWN\_H  
#include "place.h"  
  
class Town : public Place{  
protected:  
 float square;  
public:  
 Town(string name, float square);  
 void show();  
 void set\_square(float square);  
 float get\_square();  
};  
  
#endif //LAB\_2\_TOWN\_H

**megapolis.h**

#ifndef LAB\_2\_MEGAPOLIS\_H  
#define LAB\_2\_MEGAPOLIS\_H  
#include "place.h"  
#include "town.h"  
  
class Megapolis: public Town{  
protected:  
 int population;  
public:  
 Megapolis(string name, float square, int population);  
 void show();  
 void set\_population(int population);  
 int get\_population();  
};  
  
#endif //LAB\_2\_MEGAPOLIS\_H

**Определение классов:**

**place.cpp**

#include "place.h"  
  
Place::Place(string name){  
 this->name = name;  
}

**region.cpp**

#include "region.h"  
  
Region::Region(string name, int cityCount) : Place(name) {  
 this->cityCount = cityCount;  
}  
  
void Region::show(){  
 cout << "=================================" << endl << "Region name: " << this->name << endl << "City count: " << this->cityCount << endl<< "=================================" << endl << endl;  
}  
  
void Region::set\_count(int cityCount) {  
 this->cityCount = cityCount;  
}  
  
int Region::get\_count() {  
 return this->cityCount;  
}

**town.cpp**

#include "town.h"  
  
Town::Town(string name, float square) : Place(name){  
 this->square = square;  
}  
  
void Town::show(){  
 cout << "=================================" << endl << "Region name: " << this->name << endl << "Town square: " << this->square << endl<< "=================================" << endl << endl;  
}  
  
void Town::set\_square(float square) {  
 this->square = square;  
}  
  
float Town::get\_square() {  
 return this->square;  
}

**megapolis.cpp**

#include "megapolis.h"  
  
  
Megapolis::Megapolis(string name, float square, int population) : Town(name, square){  
 this->population = population;  
}  
  
void Megapolis::show() {  
 cout << endl << "=================================" << endl << "Region name: " << this->name << endl << "Town square: " << this->square << endl << "Megapolis population: " << this->population << endl << "=================================" << endl << endl;  
}  
  
void Megapolis::set\_population(int population) {  
 this->population = population;  
}  
  
int Megapolis::get\_population() {  
 return this->population;  
}

**Демонстрационная программа:**

**main.cpp**

#include "place.h"  
#include "region.h"  
#include "town.h"  
#include "megapolis.h"  
  
int main(){  
 list<Place\*> places;  
  
 auto \*brestRegion = new Region("Brest", 25);  
 brestRegion->set\_count(58);  
 places.push\_back(brestRegion);  
  
 places.push\_back(new Region("Minsk", 125));  
  
 places.push\_back(new Town("Mogilev", 439.11));  
 places.push\_back(new Town("Gomel", 281.70));  
  
 auto \*vitebskRegion = new Megapolis("Vitebsk", 391.42, 210000);  
  
 places.push\_back(new Megapolis("Grodno", 171.95, 326800));  
 places.push\_back(vitebskRegion);  
  
 std::for\_each(places.begin(), places.end(), [](Place\* place){  
 place->show();  
 });  
  
 cout << "Testing get\_population() : " << vitebskRegion->get\_population() << endl;  
  
 return 0;  
}

**Вывод:** В ходе лабораторной работы научился наследовать классы, создавать абстрактные классы, а также реализовал виртуальную функцию.