Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа

Дисциплина: «Объектно-ориентированное программирование»

III семестр

Задание 1: «Простые классы»

|  |  |
| --- | --- |
| Группа: | М8О-206Б-18, №16 |
| Студент: | Павлова Ксения Андреевна |
| Преподаватель: | Журавлёв Андрей Андреевич |
| Оценка: |  |
| Дата: | 27.09.2019 |

Москва, 2019

1. **Задание**

(*вариант № 16*): Создать класс Position для работы с географическими координатами. Координаты задаются двумя числами: широта и долгота. Долгота находится в диапазоне от -180 до 180 градусов. Широта находится в диапазоне от -90 до 90 градусов. Реализовать арифметические операции сложения, вычитания, умножения и деления, а также операции сравнения.

1. **Адрес репозитория на GitHub**

<https://github.com/KsuPav/oop_exercise_01>

1. **Код программы на С++**

CMakeLists.txt

cmake\_minimum\_required(VERSION 2.8)

project(lab1) #Название проекта

set(SOURCE\_EXE main.cpp)

set(SOURCE\_LIB Position.cpp)

add\_library(Position STATIC ${SOURCE\_LIB}) #Создание статической библиотеки

add\_executable(main ${SOURCE\_EXE}) #Создает исполняемый файл с именем main

target\_link\_libraries(main Position) # Линковка программы с библиотекой

Position.cpp

#include "Position.h"

#include <iostream>

#include <inttypes.h>

#include <cmath>

#define PI 3.14159265

Position::Position() {

this->W = 0;

this->L = 0;

}

Position::Position(double W, double L) {

this->W = W;

this->L = L;

}

Position Position::add(Position B) { //Сумма

Position C {(this->W)+(B.W), (this->L)+(B.L)};

return C;

}

Position Position::inc(Position B) { //Разность

Position C {(this->W)-(B.W), (this->L)-(B.L)};

return C;

}

Position Position::mult(Position B) { //Произведение

Position C {(this->W)\*(B.W), (this->L)\*(B.L)};

return C;

}

Position Position::div(Position B) { //Деление

Position C {(this->W)/(B.W), (this->L)/(B.L)};

return C;

}

double Position::comp(Position MAI) { //Подсчёт расстояния

double S;

int R = 6371; //Среднее значение радиуса Земли

S = R \* acos(sin((this->W) \* PI / 180) \* sin((MAI.W) \* PI / 180) + cos((this->W) \* PI / 180) \* cos((MAI.W) \* PI / 180) \* cos(fabs(((this->L) - MAI.L) \* PI / 180)));

//Формулка для нахождения расстояния между точками при наличии их координат в градусах S=R\*arccos(sin(W1)sin(W2)+cos(W1)cos(W2)cos(|L1-L2|))

return S;

}

void Position::print() {

std::cout << this->W << "; " << this->L << std::endl;

}

Position.h

#ifndef Position\_h

#define Position\_h

#include <inttypes.h>

class Position {

private:

double W;

double L;

public:

Position();

Position(double W, double L);

Position add(Position B);

Position inc(Position B);

Position mult(Position B);

Position div(Position B);

double comp(Position B);

void print();

};

#endif

main.cpp

#include <iostream>

#include "Position.h"

#include <cmath>

int main() {

double W, L; //Широта и долгота

double S, S1, S2;

int R;

W = 55.811152; //Координаты МАИ

L = 37.501312;

Position MAI = Position (W, L);

std::cout << "Введите значения широты (|x|<91) и долготы (|x|<181) первой точки\n";

std::cin >> W >> L;

Position A = Position(W, L);

std::cout << "Введите значения широты (|x|<91) и долготы (|x|<181) второй точки\n";

std::cin >> W >> L;

Position B = Position(W, L);

Position C;

std::cout << "Результаты сложения:\n";

C = A.add(B);

C.print();

std::cout << "Результаты вычитания:\n";

C = A.inc(B);

C.print();

std::cout << "Результаты умножения:\n";

C = A.mult(B);

C.print();

std::cout << "Результаты деления:\n";

C = A.div(B);

C.print();

std::cout << "Результаты сравнения расстояний от МАИ до точек:\n";

S1 = A.comp(MAI);

std::cout << "Расстояние до первой точки: " << S1 << " км" << std::endl;

S2 = B.comp(MAI);

std::cout << "Расстояние до второй точки: " << S2 << " км" << std::endl;

if (S1 > S2) {

std::cout << "Вердикт: до второй точки ближе.\n";

} else {

if (S2 > S1) {

std::cout << "Вердикт: до первой точки ближе.\n";

} else {

std::cout << "Вердикт: одинаково.\n";

}

}

return 0;

}

test\_01.txt

0 0

55.811152 37.501312

test\_02.txt

12 -179

-44 0

1. **Результаты выполнения тестов**

test\_01\_result.txt

Результаты сложения:

55.8112; 37.5013

Результаты вычитания:

-55.8112; -37.5013

Результаты умножения:

0; 0

Результаты деления:

0; 0

Результаты сравнения расстояний от МАИ до точек:

Расстояние до первой точки: 7063.74 км

Расстояние до второй точки: 0 км

Вердикт: до второй точки ближе.

test\_02\_result.txt

Результаты сложения:

-32; -179

Результаты вычитания:

56; -179

Результаты умножения:

-528; -0

Результаты деления:

-0.272727; -inf

Результаты сравнения расстояний от МАИ до точек:

Расстояние до первой точки: 11748.3 км

Расстояние до второй точки: 11643.3 км

Вердикт: до второй точки ближе.

1. **Объяснение результатов работы программы**

Программа получает на вход координаты двух точек, происходит выполнение арифметических операций. Далее, для придания заданию практичности, было принято решение модифицировать его. Таким образом, в программе осуществляется подсчёт расстояний от заданной точки (в данном случае выбраны координаты МАИ) до точек, полученных на вход, и сравнение расстояний.

1. **Вывод**

Были изучены основы объектно-ориентированного программирования, методы, классы, был написан простой класс Position.