**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Минибаев Айдар

Группа: 80-201Б-18

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. **Постановка задачи**

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 в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа Position.

1. **Описание программы**

Реализовал класс Position, координаты хранятся в типе double. Реализованы операции сложения, умножения, вычитания, деления. После выбора операции вводятся два числа, над которыми будет производится та или иная операция. Операции реализуются с помощью перегрузки операторов.

Реализовал с помощью перегрузки операторов равенство и неравенство двух точек координат по широте и долготе.

Функции friend std::ostream &operator<<(std::ostream &out, Position const &ob); friend std::istream &operator>>(std::istream &in, Position &ob); используются для ввода и вывода координат точки. Спецификатор friend означает, что функция может использовать закрытые данные из класса Positition.

Функция Position Check() используется для преобразования координат в соответствии с диапозоном: долгота от -180 до 180, широта от -90 до 90.

Функция friend void record(TransNumber &lhs, TransNumber &rhs) используется интерактивных подсказок для пользователя и ввода двух чисел, спецификатор friend означает, что функция может использовать закрытые данные из класса TransNumber.

В случае перехода к блоку 4 в конструкции switch, срабатывают исключения при вводе числа n, равным нулю.

Реализовал просто пример использования пользовательских литералов, для преобразования координат точки согласно диапозонам.

Для удобства использования сделал интерфейс, где можно выбрать операцию над координатами.

1. **Набор testcases**

Тестовые файлы: **test\_\_01.txt**, **test\_\_02.txt**. Тесты представляют из себя набор команд с координатами, которые демонстрируют работу программы. Первое число задает операцию из предложенных в интерактивном меню, затем задается пара чисел типа double для координат первой точки и аналогично для координат второй точки. При умножении вводится число на которое нужно умножить координаты точки и затем сами координаты точки.

**test\_\_01.txt**

1

120.34 54.1

70.11 -49

2

31 43.11

95 43.11

3

4

134.11 12.12

4

4

-120 88.8

5

120.11 54.8989

120.11 54.8989

6

111 23.1

111 29.1

0

Проверка всех операций над числами.

**test\_\_02.txt**

2

22223.333 222112.3

112222222 333

6

112 323

232 22

6

11 10

223 34

4

0

11

123 23

0

Проверка операции неравенства координат и проверка ошибки исключения.

1. **Результат выполнения тестов**

Все тесты успешно пройдены, программа выдает верные результаты.

**test\_\_01.txt**

./oop\_exercise\_02

Hello, World!

-106.79 -85.79

Work with coordinates

Enter a number for action:

1) Addition of coordinates

2) Coordinate difference

3) Multiplication

4) Division

5) Equality test

6) NotEquality test

7) Print Menu

0) Exit

Enter a number for action:

1

Enter the coordinates of the first point separated by space

120.34 54.1

Enter the coordinates of the second point separated by space

70.11 -49

ob1 + ob2:

latitude: -169.550000 longitute: 5.100000

Enter a number for action:

2

Enter the coordinates of the first point separated by space

31 43.11

Enter the coordinates of the second point separated by space

95 43.11

ob1 - ob2:

latitude: -64.000000 longitute: 0.000000

Enter a number for action:

3

Enter number

4

Enter coordinates separated by space

134.11 12.12

Result: latitude: 176.440000 longitute: 48.480000

Enter a number for action:

4

Enter number

4

Enter coordinates separated by space

-120 88.8

Result: latitude: -30.000000 longitute: 22.200000

Enter a number for action:

5

Enter the coordinates of the first point separated by space

120.11 54.8989

Enter the coordinates of the second point separated by space

120.11 54.8989

Сoordinates match

6

111 23.1

111 29.1

Сoordinates do not match

**test\_\_02.txt**

./oop\_exercise\_02

Hello, World!

-106.79 -85.79

Work with coordinates

Enter a number for action:

1) Addition of coordinates

2) Coordinate difference

3) Multiplication

4) Division

5) Equality test

6) NotEquality test

7) Print Menu

0) Exit

Enter a number for action:

2

Enter the coordinates of the first point separated by space

22223.333 222112.3

Enter the coordinates of the second point separated by space

112222222 333

ob1 - ob2:

latitude: 121.333000 longitute: 19.300000

Enter a number for action:

6

112 323

232 22

Сoordinates do not match

Enter a number for action:

1) Addition of coordinates

2) Coordinate difference

3) Multiplication

4) Division

5) Equality test

6) NotEquality test

7) Print Menu

0) Exit

Enter a number for action:

6

11 10

223 34

Сoordinates do not match

Enter a number for action:

1) Addition of coordinates

2) Coordinate difference

3) Multiplication

4) Division

5) Equality test

6) NotEquality test

7) Print Menu

0) Exit

Enter a number for action:

4

Enter number

0

The number must not be zero

Enter n not equal to zero

11

Enter coordinates separated by space

123 23

Result: latitude: 11.181818 longitute: 2.090909

Enter a number for action:

0

Process finished with exit code 0

1. **Листинг программы**

#include <iostream>

#include <cmath>

#include <exception>

/\*

\* Минибаев Айдар М80-201Б

\* 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 в виде перегрузки операторов.

\* Необходимо реализовать пользовательский литерал для работы с константами типа Position.

\*/

class Position {

private:

double latitude;

double longitude;

public:

Position(double lhs, double rhs) : latitude(lhs), longitude(rhs) {}; //инициализация конструктора

Position() : latitude(0), longitude(0) {};

Position Check(Position const ob);

Position operator+(Position const rhs);

Position operator=(Position const rhs);

Position operator-(Position const rhs);

Position operator\*(int n);

Position operator/(int n);

bool operator==(Position const rhs);

bool operator!=(Position const rhs);

friend std::ostream &operator<<(std::ostream &out, Position const &ob);

friend std::istream &operator>>(std::istream &in, Position &ob);

};

Position Position::Check(Position const ob) {

Position c;

double a;

double b;

a = fmod(ob.latitude,360);

if (a > 180){

a = -180 + (fmod(a,180));

} else if(a < -180){

a = 180 + (fmod(a,180));

}

c.latitude = a;

b = fmod(ob.longitude,180);

if (b > 90){

b = -90 + (fmod(b,90));

} else if (b < -90){

b = 90 + (fmod(b,90));

}

c.longitude = b;

return c;

}

Position Position::operator+(Position const rhs) {

Position c;

c.latitude = latitude + rhs.latitude;

c.longitude = longitude + rhs.longitude;

return Check(c);

}

Position Position::operator-(Position const rhs) {

Position c;

c.latitude = latitude - rhs.latitude;

c.longitude = longitude - rhs.longitude;

return Check(c);

}

Position Position::operator\*(int n) {

Position c;

c.latitude = latitude \* n;

c.longitude = longitude \* n;

return Check(c);

}

Position Position::operator/(int n) {

Position c;

c.latitude = latitude / n;

c.longitude = longitude / n;

return Check(c);

}

Position Position::operator=(Position const rhs) {

latitude = rhs.latitude;

longitude = rhs.longitude;

return \*this;

}

bool Position::operator==(Position const rhs) {

return ((latitude == rhs.latitude) && (longitude == rhs.longitude));

}

bool Position::operator!=(Position const rhs) {

return ((latitude!=rhs.latitude) || (longitude!=rhs.longitude));

}

std::ostream &operator<<(std::ostream &out, Position const &ob){

out.precision(6);

out.setf(std::ios::fixed);

out << "latitude: " <<ob.latitude << " longitute: " << ob.longitude << "\n";

out.unsetf(std::ios::fixed);

return out;

}

std::istream &operator>>(std::istream &in, Position &ob){

in >> ob.latitude >> ob.longitude;

return in;

}

long double operator "" \_lat(long double a){

a = fmod(a,360);

if (a > 180){

a = -180 + fmod(a,180);

} else if(a < -180){

a = 180 + fmod(a,180);

}

return a;

}

long double operator "" \_lon(long double a){

a = fmod(a,360);

if (a > 90){

a = -90 + fmod(a,90);

} else if(a < -90){

a = 90 + fmod(a,90);

}

return a;

}

int main() {

std::cout << "Hello, World!" << std::endl;

std::cout << 253.21\_lat << " " << 94.21\_lon << std::endl; // приводим в порядок широту и долготу

Position ob1, ob2;

double n;

int option = 7;

bool check = true;

std::cout << "Work with coordinates" << std::endl;

while(check) {

switch(option) {

case 0: check = false; break;

case 1:

std::cout << "Enter the coordinates of the first point separated by space" << std::endl;

std::cin >> ob1;

std::cout << "Enter the coordinates of the second point separated by space" << std::endl;

std::cin >> ob2;

std::cout << "ob1 + ob2:\n" << ob1 + ob2;

break;

case 2:

std::cout << "Enter the coordinates of the first point separated by space" << std::endl;

std::cin >> ob1;

std::cout << "Enter the coordinates of the second point separated by space" << std::endl;

std::cin >> ob2;

std::cout << "ob1 - ob2:\n" << ob1 - ob2;

break;

case 3:

std::cout << "Enter number" << std::endl;

std::cin >> n;

std::cout << "Enter coordinates separated by space" << std::endl;

std::cin >> ob1;

std::cout << "Result: ";

std::cout << ob1 \* n;

break;

case 4:

std::cout << "Enter number" << std::endl;

std::cin >> n;

try{

if (n==0) throw 0;

} catch(int){

std::cout << "The number must not be zero" << std::endl;

std::cout << "Enter n not equal to zero" << std::endl;

std::cin >> n;

}

std::cout << "Enter coordinates separated by space" << std::endl;

std::cin >> ob1;

std::cout << "Result: ";

std::cout << ob1 / n;

break;

case 5:

std::cout << "Enter the coordinates of the first point separated by space" << std::endl;

std::cin >> ob1;

std::cout << "Enter the coordinates of the second point separated by space" << std::endl;

std::cin >> ob2;

if (ob1 == ob2) std::cout << "Сoordinates match\n";

else std::cout << "Сoordinates do not match\n";

case 6:

std::cin >> ob1 >> ob2;

if (ob1 != ob2) std::cout << "Сoordinates do not match\n";

else std::cout << "Сoordinates match\n";

case 7:

std::cout << "Enter a number for action:" << std::endl;

std::cout << "1) Addition of coordinates" <<

std::endl << "2) Coordinate difference" <<

std::endl << "3) Multiplication" <<

std::endl << "4) Division" <<

std::endl << "5) Equality test" <<

std::endl << "6) Notequality test" <<

std::endl << "6) Print Menu" <<

std::endl << "0) Exit" << std::endl;

break;

}

if(check){

std::cout << "Enter a number for action:" << std::endl;

std::cin >> option;

}

}

return 0;

}

1. **Вывод**

Я научился создавать программу на языке с C++ с использованием перегрузки операторов для работы с координатами. Реализовал перегрузку операторов в соответствии с заданием и пользовательские литералы, которые работают с объектами класса.

**Список литературы**

1. Шилдт, Герберт. С++: базовый курс, 3-е изд. : Пер. с англ. - М. : ООО “И.Д. Вильямс”, 2018. - 624 с. : ил. - Парал. тит. англ.
2. Справочник по языку C++ [Электронный ресурс]. URL <http://www.c-cpp.ru/> (дата обращения 17.09.2019).