МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СХІДНОУКРАЇНСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

імені ВОЛОДИМИРА ДАЛЯ

Звіт

з лабораторної роботи № 2

**З ТЕМИ “ПЕРЕГРУЗКА ОПЕРАТОРОВ ДЛЯ КЛАССОВ”**

Виконав студент:

Група КН-18д

Скороход С.Г.

Дата 29.04.2020

**м. Северодонецьк**

2020

## Цель работы

Получение навыков перегрузки арифметических операторов, операторов присваивания и ввода/вывода для пользовательских классов.

**Виконання роботи**

Створення глобальної константної змінної для зберігання розміру масивів

const int Size= 5;

Створення шаблону класу

template<class T> class inpu {

public:

private:

T \*nums = new T[Size];

};

Створення конструктору класу ( при створенні об’єкту масив ініціалізуеться нулями)

inpu() {

for (int i = 0; i < Size; i++)

{

nums[i] = 0;

}

}

Створення конструктору класу ( при створенні об’єкту всі члени масиву ініціалізуеться заданим числом)

inpu(T num)

{

for (int i = 0; i < Size; i++)

{

nums[i] = num;

}

}

Створення деструктору класу

~inpu() {

delete[] nums;

}

Створення перегрузки оператора «-»

inpu operator-(const inpu &Mon)

{

inpu<T> Temp;

for (int i = 0; i < Size; i++)

{

Temp.nums[i] = this->nums[i] - Mon.nums[i];

}

return Temp;

}

Створення перегрузки оператора «+»

inpu operator+(const inpu &Mon)

{

inpu<T> Temp;

for (int i = 0; i < Size; i++)

{

Temp.nums[i] = this->nums[i] + Mon.nums[i];

}

return Temp;

}

Створення перегрузки оператора «+=»

void operator+=(const inpu &Mon)

{

for (int i = 0; i < Size; i++)

{

this->nums[i] += Mon.nums[i];

}

}

Створення перегрузки оператора «-=»

void operator-=(const inpu &Mon)

{

for (int i = 0; i < Size; i++)

{

this->nums[i] -= Mon.nums[i];

}

}

Створення перегрузки оператора індексу масива **«[]»**

T operator[](const int index)

{

return this->nums[index];

}

Створення перегрузки оператора виведення «<<»

friend ostream & operator<< (ostream &stream, inpu ob)

{

for (int i = 0; i < 5; i++)

{

stream << ob.nums[i]<<"\t";

}

stream << "\n";

return stream;

}

Створення перегрузки оператора введення «>>»

friend istream & operator>> (istream &stream, inpu &ob)

{

cout << "Введите 5 чисел" << endl;

for (int i = 0; i < 5; i++) {

stream >> ob.nums[i];

}

return stream;

}

Наповнення функції main()

setlocale(LC\_ALL, "rus"); // підтримка кирилиці

inpu<double> P; // створення об’єкту класу inpu з типом даних double

cin >> P; // наповнення масиву

inpu<double> S; // створення об’єкту класу inpu з типом даних double

cin >> S; // наповнення масиву

inpu<double> Z; // створення об’єкту класу inpu з типом даних double

cout << P; //вивід масиву певного класу

cout << S; //вивід масиву певного класу

Z = P + S; //заповнення нового масиву шляхом додавання двух попередник

P+=S; // збільшення даних масиву об’єкта P на данні масиву об’єкта S

cout << Z; //вивід масиву певного класу

cout << Z[2]; // вивід відповідного елемента масиву певного класу

ЗАГАЛЬНИЙ ВИД КОДУ

#include <iostream>

using namespace std;

const int Size = 5;

template<class T> class inpu {

public:

inpu() {

for (int i = 0; i < Size; i++)

{

nums[i] = 0;

}

}

inpu(T num)

{

for (int i = 0; i < Size; i++)

{

nums[i] = num;

}

}

~inpu() {

delete[] nums;

}

inpu operator-(const inpu &Mon)

{

inpu<T> Temp;

for (int i = 0; i < Size; i++)

{

Temp.nums[i] = this->nums[i] - Mon.nums[i];

}

return Temp;

}

inpu operator+(const inpu &Mon)

{

inpu<T> Temp;

for (int i = 0; i < Size; i++)

{

Temp.nums[i] = this->nums[i] + Mon.nums[i];

}

return Temp;

}

void operator+=(const inpu &Mon)

{

for (int i = 0; i < Size; i++)

{

this->nums[i] += Mon.nums[i];

}

}

void operator-=(const inpu &Mon)

{

for (int i = 0; i < Size; i++)

{

this->nums[i] -= Mon.nums[i];

}

}

T operator[](const int index)

{

return this->nums[index];

}

friend ostream & operator<< (ostream &stream, inpu ob)

{

for (int i = 0; i < 5; i++)

{

stream << ob.nums[i]<<"\t";

}

stream << "\n";

return stream;

}

friend istream & operator>> (istream &stream, inpu &ob)

{

cout << "Введите 5 чисел" << endl;

for (int i = 0; i < 5; i++) {

stream >> ob.nums[i];

}

return stream;

}

private:

T \*nums = new T[Size];

};

int main()

{

inpu<double> P

cin >> P;

inpu<double> S;

cin >> S;

inpu<double> Z;

cout << P;

cout << S;

Z = P + S;

cout << Z;

cout << Z[2];

P+=S;

cout << P;

}