Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно – ориентированное программирование»

**ОТЧЕТ**

По лабораторной работе №5.

Тема: «Перезагрузка операторов»

                                                                       Выполнила: студент группы

ПИ(б)-2-21 Марат кызы Жылдыз

Проверил: Мусабаев Э.Б.

**Бишкек – 2024**

**Лабораторная работа №5.**

**Тема: Перезагрузка операторов.**

**Задания:**

1. Типы данных полезны там, где ошибки могут быть вызваны арифметическим переполнением, которое не допустимо.

Создайте и откомпилируйте класс **Int**. Перегрузите четыре бинарных целочисленных арифметических операции (+, -, \*, /) и унарные операции постфиксной и префиксной форм инкремента с помощью внутренней операторной функции так, чтобы их можно было использовать для операций с объектами класса **Int**.

Если результат какой-либо из них выходит за границы типа **int** (в 32-битной системе), имеющее значения от **2 147 483 648** до **-2 147 483 648**, то операция должна послать сообщение об ошибке и завершить программу. Для выявления ошибки арифметического переполнения используйте концепцию **исключения**.

Для облегчения проверки переполнения выполняйте вычисления с использованием типа **long** **double**. При описании унарных операций используйте указатель **this**.

**Код программы:**

#include <iostream>

#include<Windows.h>

using namespace std;

class Int

{

private:

int a;

public:

Int() :a(0) {} // Коструктор без аргументов

Int(int A) : a(A) {} // Конструктор с одним аргументом

~Int() {} // Деструктор

int show() { // Функция вывода значения. Возвращает целочисленный результат

return a;

}

Int operator+(const Int& val) { // Функция сложения

Int temp;

temp.a = long double(a) + long double(val.a);

return temp;

}

Int operator-(const Int& val) { // Функция разности

Int temp;

temp.a = long double(a) - long double(val.a);

return temp;

}

Int operator\*(const Int& val) { // Функция множителя

Int temp;

temp.a = long double(a) \* long double(val.a);

return temp;

}

Int operator/(const Int& val) { // Функция частности

Int temp;

temp.a = long double(a) / long double(val.a);

return temp;

}

Int operator++() { // Префиксная форма

++a;

return \*this;

}

Int operator++(int) { // Постфиксная форма

a++;

return \*this;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

int x = 0, y = 0, val = 0, itog = 0;

cout << "Введите первое число: " << endl;

cin >> x;

cout << "Введите второе число: " << endl;

cin >> y;

Int ch1(x), ch2(y), rez = ch1 + ch2; // Создаем объект и сразу же вызываем певую функция сложения

while (val != 7) {

system("CLS");

cout << "x= " << x << endl << "y= " << y << endl << endl;

cout << "1) Сложение" << "\n2) Вычитание" << "\n3) Умножение " << "\n4) Деление" << "\n5) Постфиксная форма " << "\n6) Префиксная форма" << endl;

cout << "Выберите действие: ";

cin >> val;

if (val == 1) {

itog = rez.show(); // вывод

if (itog >= 2147483648 || itog <= -2147483648) { // Проверяем. Если значение превышает границы значений в условии, то выводим сообщения

cout << "Переполнения!" << endl;

exit(0);

}

else { // Если нет, то выводим результат

cout << "Сумма: ";

cout << itog << endl;

system("pause");

system("CLS");

}

}

if (val == 2) {

rez = ch1 - ch2;

itog = rez.show();

if (itog > 2147483648 || itog < -2147483648) {

cout << "Переполнения!" << endl;

exit(0);

}

else {

cout << "Разность: ";

cout << itog << endl;

system("pause");

system("CLS");

}

}

if (val == 3) {

rez = ch1 \* ch2;

itog = rez.show();

if (itog >= 2147483648 || itog <= -2147483648) {

cout << "Переполнения!" << endl;

exit(0);

}

else {

cout << "Множитель: ";

cout << itog << endl;

system("pause");

system("CLS");

}

}

if (val == 4) {

rez = ch1 / ch2;

itog = rez.show();

if (itog >= 2147483648 || itog <= -2147483648) {

cout << "Переполнения!" << endl;

exit(0);

}

else {

cout << "Частное: ";

cout << itog << endl;

system("pause");

system("CLS");

}

}

if (val == 5) {

if (itog >= 2147483648 || itog <= -2147483648) {

cout << "Переполнения!" << endl;

exit(0);

}

else {

cout << "Сумма: ";

cout << rez.show() << endl; // Сначало выводим сумму, потом вызываем постфиксную форму.

// Сначало записывается сумма, потом сумма увеличивается на 1

rez++;

cout << "Постфиксная форма: ";

cout << endl << rez.show() << endl;

system("pause");

system("CLS");

}

}

if (val == 6) {

if (itog >= 2147483648 && itog <= -2147483648) {

cout << "Переполнения!" << endl;

exit(0);

}

else {

++rez; // сначало вызываем префиксную форму

// Сумма увеличивается на 1 и потом записывается

cout << "Сумма: ";

cout << rez.show() << endl;

cout << "Постфиксная форма: ";

cout << endl << rez.show() << endl;

system("pause");

system("CLS");

}

}

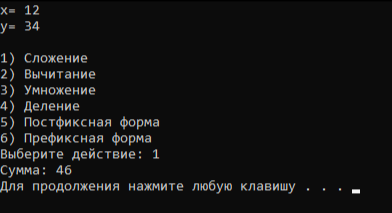
}

cout << "До свидания" << endl;

}

**Результат программы:**

**Рис.1. Сложение.**

****

**Рис.2. Вычитание.**

****

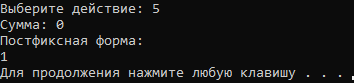
**Рис.3. Умножение.**

****

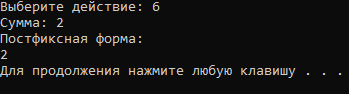
**Рис.4. Деление.**

****

**Рис.5. Постфиксная форма.**

****

**Рис.6. Префиксная форма.**



1. **Создать класс, в котором перегружается метод rect\_area().**

Этот метод возвращает площадь прямоугольника. В этой программе метод **rect\_area()** перегружается двумя способами. В первом — методу передаются оба размера фигуры. Эта версия используется для прямоугольника. Однако, в случае квадрата необходимо задавать только один аргумент, поэтому вызывается вторая версия метода **rect\_area().**

**Код программы:**

#include<iostream>

#include<Windows.h>

using namespace std;

class rectangle {

private:

float a;

float b;

public:

rectangle() :a(0.0), b(0.0) {} // Конструктор по умолчанию

~rectangle() { cout << endl << "Деструктор завершен" << endl; } // Деструктор

// Перегружаемые функции

float rect\_area(float a, float b) { // функция для прямоуголника

return a \* b;

}

float rect\_area(float a) { // функция для квадрата

return pow(a, 2);

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

float st1 = 0, st2 = 0;

int val = 0;

rectangle obj;

while (val != 3 && val < 3) // Пока не равно или пока меньше трех

{

cout << "\tМеню" << endl;

cout << "1) Прямоугольник " << "\n2) Квадрат " << "\n3) Выход" << endl;

cin >> val;

if (val == 1) {

system("CLS");

cout << "Введите сторону a: ";

cin >> st1;

cout << "Введите сторону b: ";

cin >> st2;

cout << endl << "Площадь прямоуголника равна: " << obj.rect\_area(st1, st2) << endl; // Вызов и вывод на экран результата перегружаемой функции

system("pause");

system("CLS");

}

else if (val == 2) {

system("CLS");

cout << "Введите сторону a: ";

cin >> st1;

cout << endl << "Площадь квадрата равна: " << obj.rect\_area(st1) << endl; // Вызов и вывод на экран результата перегружаемой функции

system("pause");

system("CLS");

}

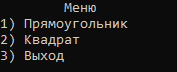
}

cout << "До свидания!!" << endl;

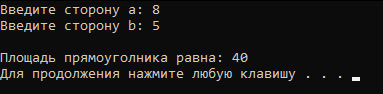
}

**Результат программы:**

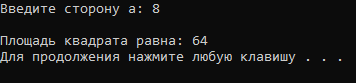
**Рис.1. Меню.**

****

**Рис.2. Прямоугольник.**

****

**Рис.3. Квадрат.**



1. Опишите класс **fraction**, у которого есть одно закрытое целочисленное поле **chislo** типа **double**. Перегрузите для этого класса арифметические операции **сложения**, **вычитания**, **умножения** и **деления** так, чтобы они могли оперировать как с объектами класса, так и с числами (то есть выполнять, например, не только действие 3/4 +2/5, но и 1/2 + 4 или 2\* 5/6). Также перегрузите унарную операцию инкремента в префиксной или постфиксной форме увеличения дроби. Продемонстрируйте работу класса. Используйте конструктор по умолчанию и конструктор с одни аргументом для инициализации поля класса.

**Код программы:**

#include <iostream>

#include<Windows.h>

using namespace std;

class fraction

{

private:

double chislo;

public:

fraction() :chislo(0) {} // Коструктор без аргументов

fraction(int ch) : chislo(ch) {} // Конструктор с одним аргументом

~fraction() {} // Деструктор

void show() { // Функция вывода значения. Возвращает целочисленный результат

cout << endl << chislo << endl;;

}

fraction operator+(const fraction& val) { // Функция сложения

fraction temp;

temp.chislo = chislo + val.chislo;

return temp;

}

fraction operator-(const fraction& val) { // Функция вычитания

fraction temp;

temp.chislo = chislo - val.chislo;

return temp;

}

fraction operator\*(const fraction& val) { // Функция умножения

fraction temp;

temp.chislo = chislo \* val.chislo;

return temp;

}

fraction operator/(const fraction& val) { // Функция деления

fraction temp;

temp.chislo = chislo / val.chislo;

return temp;

}

fraction operator\*(double& zn) { // Функция умножения результата деления на число

fraction temp;

temp.chislo = chislo \* zn;

return temp;

}

fraction operator++() { // Префиксная форма

++chislo;

return \*this;

}

fraction operator++(int) { // Постфиксная форма

chislo++;

return \*this;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

int x = 0, y = 0, val = 0, itog = 0;

double zn = 5.10;

cout << "Введите первое число: " << endl;

cin >> x;

cout << "Введите второе число: " << endl;

cin >> y;

fraction ch1(x), ch2(y), rez = ch1 + ch2, rez2; // Создаем объект и сразу же вызываем певую функция сложения

while (val != 7) {

system("CLS");

cout << "x= " << x << endl << "y= " << y << endl << endl;

cout << "1) Сложение" << "\n2) Вычитание" << "\n3) Умножение " << "\n4) Деление" << "\n5) Постфиксная форма " << "\n6) Префиксная форма" << endl;

cout << "Выберите действие: ";

cin >> val;

if (val == 1) {

rez = ch1 + ch2;

cout << "Сумма: ";

rez.show();

cout << endl;

system("pause");

system("CLS");

}

else if (val == 2) {

rez = ch1 - ch2;

cout << "Разность: ";

rez.show();

cout << endl;

system("pause");

system("CLS");

}

else if (val == 3) {

rez = ch1 \* ch2;

cout << "Множитель: ";

rez.show();

cout << endl;

system("pause");

system("CLS");

}

else if (val == 4) {

rez = ch1 / ch2;

cout << "Частное: ";

rez.show();

cout << endl;

cout << "Умножение результата на число: ";

rez2 = rez \* zn;

rez2.show();

system("pause");

system("CLS");

}

else if (val == 5) {

rez.show();

cout << endl;

rez++;

rez.show();

cout << endl;

system("pause");

system("CLS");

}

else if (val == 6) {

++rez;

rez.show();

cout << endl;

rez.show();

cout << endl;

system("pause");

system("CLS");

}

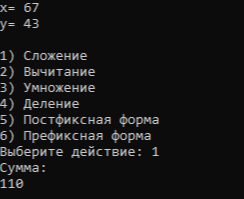
}

cout << "До свидания" << endl;

}

**Результат программы:**

**Рис.1. Сложение.**

****

**Рис.2. Вычитание**

****

**Рис.3. Умножение**

****

**Рис.4. Деление.**

****

**Рис.5. Постфиксная форма.**

****

**Рис.6. Префиксная форма.**

****