**Лабораторна робота №3**

**Тема:** Перевантаження операцій класу

**Мета:** Ознайомитись зі способами перевантаження операцій та навчитись використовувати їх при роботі з об’єктами.

**Завдання 1.**

В класі Int, який розроблений в завданні №1 лабораторної роботи №1, перевизначте чотири цілочисельні арифметичні операції («+», «-», «\*» , «/») так, щоб їх можна було використовувати для операцій з об'єктами класу Int. Якщо результат будь-якої з операцій виходить за межі типу int (в 32-бітній системі), що може мати значення від 2 147 483 648 до -2 147 483 648, то операція повинна послати повідомлення про помилку і завершити програму. Такі типи даних корисні там, де помилки можуть бути викликані арифметичним переповненням, яке неприпустимо. Напишіть програму для перевірки цього класу. **КОД ПРОГРАМИ**

#include <iostream>

*using* *namespace* std;

*class* **Int**

{

*private*:

int a;

*public*:

**Int**(){

*this* -> a = 0;

}

**Int**(int a)

{

*this*->a=a;

};

Int *operator* =(*const* Int&);

Int *operator* +(*const* Int&);

Int *operator* -(*const* Int&);

Int *operator* \*(*const* Int&);

Int *operator* /(*const* Int&);

void **SetInt**(int a)

{

*this*->a=a;

}

void **GetInt**()

{

cout<< *this*->a << endl;

};

};

Int Int::*operator*=(*const* Int &b)

{

*this*->a = b.a;

*return* \**this*;

}

Int Int::*operator*+(*const* Int &b)

{

long double a =(long double) *this* -> a+b.a;

*if* (a<INT\_MIN || a>INT\_MAX)

{

cout<< "Value in not correct";

exit(0);

}

*else* {

*return* Int ((int) a);

}

}

Int Int::*operator*-(*const* Int &b)

{

long double a =(long double) *this* -> a-b.a;

*if* (a<INT\_MIN || a>INT\_MAX)

{

cout<< "Value in not correct";

exit(0);

}

*else* {

*return* Int ((int) a);

}

}

Int Int::*operator*\*(*const* Int &b)

{

long double a =(long double) *this* -> a\*b.a;

*if* (a<INT\_MIN || a>INT\_MAX)

{

cout<< "Value in not correct";

exit(0);

}

*else* {

*return* Int ((int) a);

}

}

Int Int::*operator*/(*const* Int &b)

{

long double a =(long double) *this* -> a/b.a;

*if* (a<INT\_MIN || a>INT\_MAX)

{

cout<< "Value in not correct";

exit(0);

}

*else* {

*return* Int ((int) a);

}

}

int **main**()

{

Int A1(44),A2(11),A3;

A3=A1+A2;

A3.GetInt();

A3=A1-A2;

A3.GetInt();

A3=A1\*A2;

A3.GetInt();

A3=A1/A2;

A3.GetInt();

*return* 0;

}

**Завдання 2.**

Для класу, який розробленого згідно індивідуального завдання лабораторної роботи № 2, визначити операції: - зчитування з потоку вводу sdt::cin; - виводу у потік std::cout.

**КОД ПРОГРАМИ**

#include <iostream>

#include <string.h>

#include <assert.h>

*using* *namespace* std;

*class* **Food** {

char\* Type;

int cal;

*public*:

**Food**(){

Type = *new* char[50];

cal = NULL;

cout << "Konstruktor started" << endl;

}

*friend* istream& *operator* >>(istream&,Food&);

*friend* ostream& *operator* <<(ostream&,Food&);

void **SetType**( char\* \_Type ){

\_Type = Type;

}

char \* **GetType**( ){

cout<< "Type: "<< \*Type << endl;

}

void **SetCalories**( int \_cal ){

\_cal = cal;

}

int **GetCalories**( ){

cout<< "Calories: "<< cal<< endl;

}

~**Food**(){

*delete* [] Type;

cout<< "konstruktor destroyed" << endl;

}

};

istream& *operator* >>(istream& is, Food& x)

{

cout<< "Enter Type: ";

is>>x.Type;

cout<< "Enter cal: ";

is>>x.cal;

}

ostream& *operator* <<(ostream& os, Food& x)

{

os << "Type is = " << x.Type << endl << "Cal is = " << x.cal<< endl;

*return* os;

}

int **main**()

{

Food myObject;

cin>>myObject;

cout<<myObject;

*return* 0;

}