

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕХНІЧНИЙ КОЛЕДЖ ТНТУ ІМЕНІ ІВАНА ПУЛЮЯ
ВІДДІЛЕННЯ ЕЛЕКТРОННИХ АПАРАТІВ
Циклова комісія програмних систем і комплексів

ЗВІТ

про виконання лабораторних робіт з дисципліни:
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Студента 3 курсу групи КН-321
спеціальності 122 «Комп'ютерні науки»

Попович О.В.

(прізвище та ініціали)

Перевірив: Р.О. Слободян
(підпис)

Тернопіль – 2020

Лабораторна робота №3

Тема: Перевантаження операцій класу

Мета: ознайомитись зі способами перевантаження операцій та навчитись використовувати їх при роботі з об'єктами.

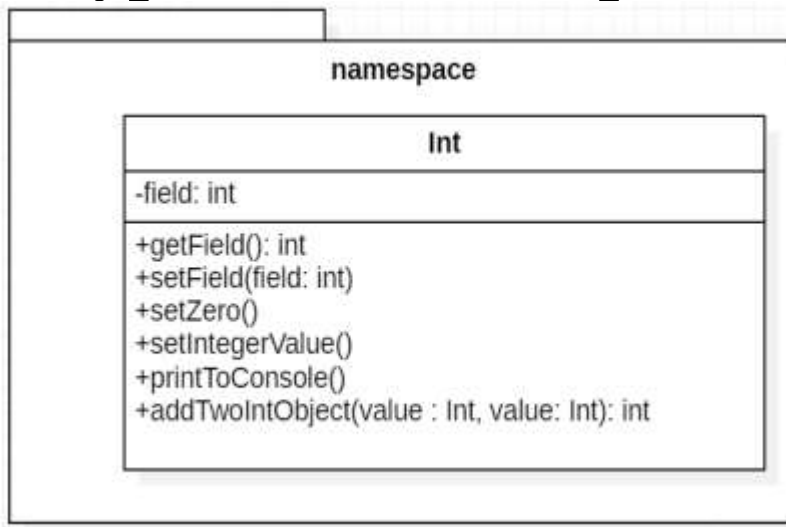
Завдання 1.

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.

Підказка: для полегшення перевірки переповнення виконуйте обчислення з використанням типу `long double`.

Структурна UML-діаграма класу



КОД ПРОГРАМИ

```
#include <iostream>
#include <string>
```

```

using namespace std;

class Int
{
private:
    int field;
public:

    int getField(){return field;}

    void setField(int field){this->field = field;}

    void setZero(){this->field = 0;}

    void setIntegerValue()
    {
        cout << "Enter the number" << endl;
        int value;
        cin >> value; cout << endl;
        this->field = value;
    }

    void printToConsole()
    {
        cout << "Field = " << field << endl;
    }

    int addTwoIntObject(Int object1, Int object2)
    {
        return object1.getField() + object2.getField();
    }

    Int operator +(const Int& other)
    {
        Int temp;
        long double number = (long double)this->field + (long double)other.field;

        if (number > INT32_MAX)
        {
            cout << "ERROR!!!" << endl;
            exit(0);
        }
        else {
            temp.field = this->field + other.field;
            return temp;
        }
    }

    Int operator -(const Int& other)
    {
        Int temp;
        long double number = (long double)this->field - (long double)other.field;

        if (number < INT32_MIN)
        {
            cout << "ERROR!!!" << endl;
            exit(0);
        }
        else {

```

```

        temp.field = this->field - other.field;
        return temp;
    }
}
Int operator *(const Int& other)
{
    Int temp;
    long double number = (long double)this->field * (long double)other.field;

    if (number > INT32_MAX)
    {
        cout << "ERROR!!!" << endl;
        exit(0);
    }
    else {
        temp.field = this->field * other.field;
        return temp;
    }
}
Int operator /(const Int& other)
{
    Int temp;
    temp.field = this->field / other.field;
    return temp;
}
};

int main()
{
    Int object1;
    object1.setIntegerValue();

    Int object2;
    object2.setIntegerValue();

    Int suma = object1 + object2;
    cout << "object1 + object2 = " << suma.getField() << endl;

    Int minus = object1 - object2;
    cout << "object1 - object2 = " << minus.getField() << endl;

    Int multiply = object1 * object2;
    cout << "object1 * object2 = " << multiply.getField() << endl;

    Int division = object1 / object2;
    cout << "object1 / object2 = " << division.getField() << endl;

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

1)

```
Microsoft Visual Studio Debug Console
Enter the number
200000

Enter the number
500

object1 + object2 = 200500
object1 - object2 = 199500
object1 * object2 = 100000000
object1 / object2 = 400

C:\Users\ASUS\source\repos\ConsoleApplication23\Debug\
```

2)

```
Microsoft Visual Studio Debug Console
Enter the number
1501500

Enter the number
40000

object1 + object2 = 1541500
object1 - object2 = 1461500
ERROR!!!

C:\Users\ASUS\source\repos\ConsoleApplication23\Debug\
```

Завдання 2.

Завдання 2. Для класу, який розроблено згідно індивідуального завдання лабораторної роботи № 2, визначити операції:

- зчитування з потоку вводу `std::cin`;
- виводу у потік `std::cout`.

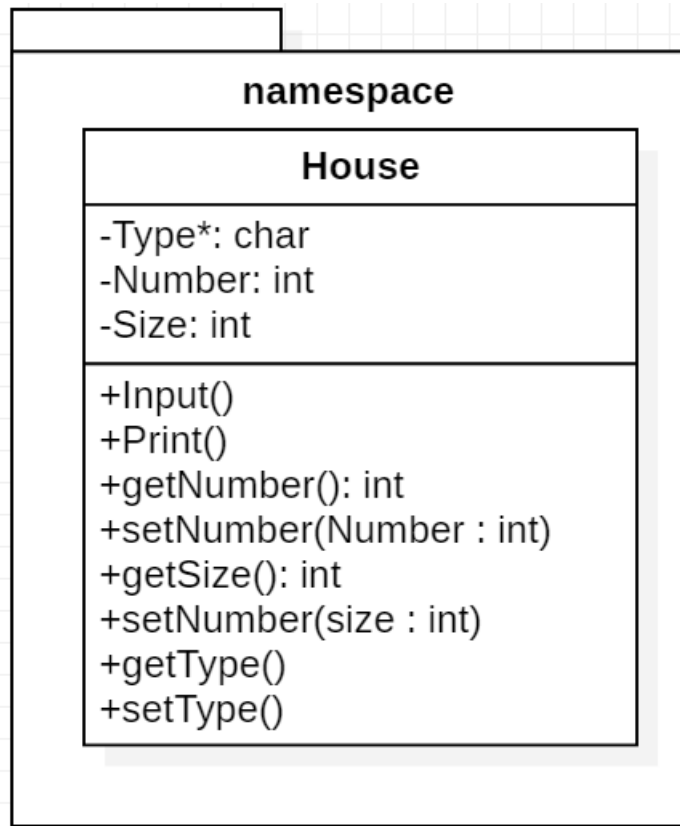
Перевірити роботу перевизначених функцій у функції `main()` за допомогою коду:

НазваКласуЗгідноВаріанта `myObject`;

`std::cin >> myObject`;

`std::cout << myObject`;

Структурна UML-діаграма класу



КОД ПРОГРАММЫ

```
#include <iostream>
using namespace std;

class House
{
private:
    char *Type;
    int Number;
    int Size;
public:
    House() {};

    House(char * Type_, int Number, int Size) {

        this->Number = Number;
        this->Size = Size;

        Type = new char[Size];

        for (int i = 0; i < Size; i++)
        {
            this->Type[i] = Type_[i];
        }
    };

    House(const House& other) {
```

```

        this->Type = new char[other.Number];

        for (int i = 0; i < other.Size; i++)
        {
            this->Type[i] = other.Type[i];
        }
    };

    void Input() {
        cout << "-----" << endl << "Enter the
Number" << endl;
        cin >> this->Number;

        cout << "Enter the size of the string array" << endl;
        cin >> this->Size;

        Type = new char[Size];

        for (int i = 0; i < Size; i++)
        {
            cin >> Type[i];
        }
        cout << "-----";
    };

    void Print() const {
        cout << "-----" << endl << "char*-->";
        for (int i = 0; i < this->Size; i++)
        {
            cout << Type[i];
        }
        cout << endl;
        cout << "Number --> " << this->Number << endl;
        cout << "-----" << endl;;
    };

    int getNumber() { return Number; }
    void setNumber(int Number) { this->Number = Number; }

    int getSize() { return Size; }
    void setSize(int size) { this->Size = size; }

    void getType()
    {
        cout << endl;
        for (int i = 0; i < this->Size; i++)
            cout << Type[i];
    }

    void setType()
    {
        cout << endl << "Enter the size of the string array" << endl;
        int size; cin >> size;
        Type = new char[size];
        for (int i = 0; i < size; i++)
        {
            cin >> Type[i];
        }
    }

```

```

    }
}

~House() {
    delete[]Type;
};

void operator >>(istream & s, House &h)
{
    h.Input();
}

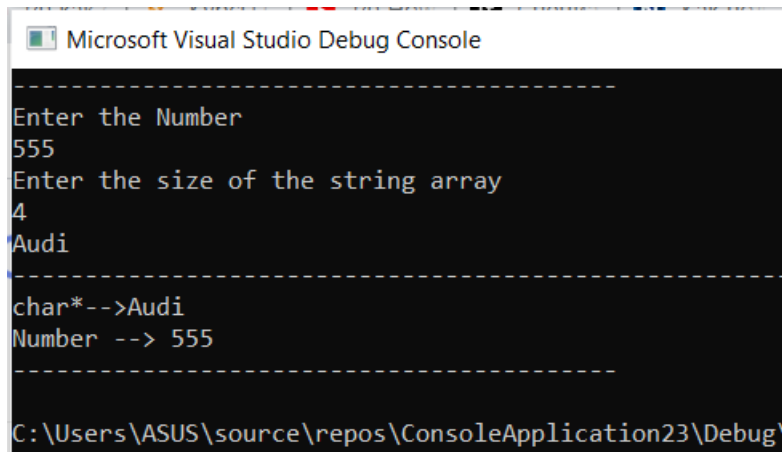
void operator <<(ostream & s, House &h)
{
    h.Print();
}

int main()
{
    House h;
    cin >> h;
    cout << h;

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ



```

Microsoft Visual Studio Debug Console
-----
Enter the Number
555
Enter the size of the string array
4
Audi
-----
char*-->Audi
Number --> 555
-----
C:\Users\ASUS\source\repos\ConsoleApplication23\Debug\

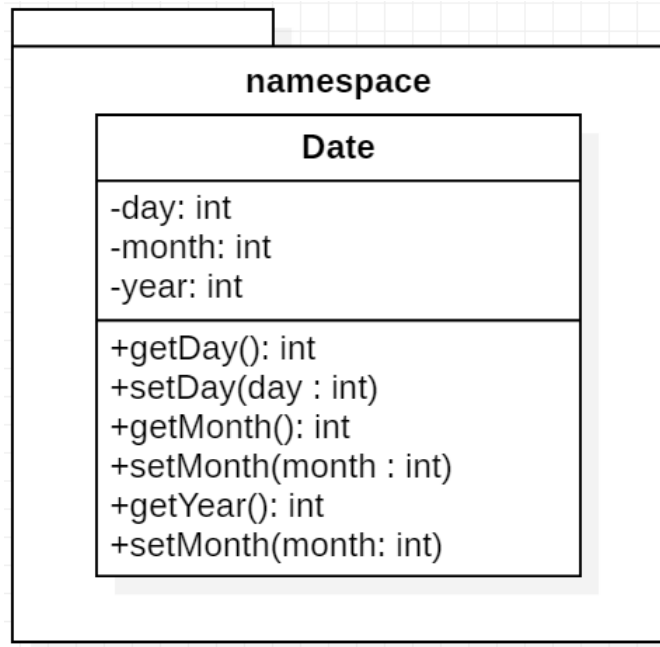
```

Завдання 3.

Завдання 3.* Для заданого варіанта індивідуального завдання виконати перевантаження операцій для зручності роботи з об'єктами. При необхідності оголосить певні операторні функції друзями класу.

3. Створити клас – дата з полями у закритій частині: день (1-31), місяць (1-12), рік (ціле число). Визначити необхідні конструктори, методи доступу, деструктор. Перевантажити потокові операції введення і виведення, операції + (збільшення на n днів), – (різниця між двома датами), < та >= .

Структурна UML-діаграма класу



КОД ПРОГРАМИ

```
#include <iostream>
#include <math.h>
using namespace std;

class Date
{
private:
    int day, month, year;

public:
    Date() {}
    Date(int day, int month, int year)
    {
        this->day = day;
        this->month = month;
        this->year = year;
    }

    int getDay() { return day; }
    int getMonth() { return month; }
    int getYear() { return year; }
```

```

void setDay(int day) { this->day = day; }
void setMonth(int month) { this->month = month; }
void setYear(int year) { this->year = year; }

Date operator +(Date& date)
{
    Date temp;
    temp.setDay(this->getDay() + date.getDay());
    temp.setMonth(this->getMonth() + date.getMonth());
    temp.setYear(this->getYear() + date.getYear());
    return temp;
}
Date &operator -(Date& date)
{
    Date temp;
    temp.setDay(abs(this->getDay() - date.getDay()));
    temp.setMonth(abs(this->getMonth() - date.getMonth()));
    temp.setYear(abs(this->getYear() - date.getYear()));

    return temp;
}
bool operator <(Date& date)
{
    if (this->getDay() + this->getMonth() * 31 + this->getYear() * 365 <
date.getDay() + date.getMonth() * 31 + date.getYear() * 365)
        return true;
    else
        return false;
}

bool operator >=(Date& date)
{
    if (this->getDay() + this->getMonth() * 31 + this->getYear() * 365 >=
date.getDay() + date.getMonth() * 31 + date.getYear() * 365)
        return true;
    else
        return false;
}
~Date() {}
};

ostream & operator <<(ostream &s, Date &date)
{
    s << date.getDay() << ":" << date.getMonth() << ":" << date.getYear() << endl;
    return s;
}

istream & operator >>(istream &i, Date &date)
{
    int day, month, year;
    cin >> day; date.setDay(day);
    cin >> month; date.setMonth(month);
    cin >> year; date.setYear(year);
    return i;
}

int main()

```

```

{
    cout << "(date_1)  cin>> ";
    Date date1;
    cin >> date1; cout << "(date_1)  cout<< ";
    cout << date1; cout << endl;

    Date date2;
    cout << "(date_2)  cin>> ";
    cin >> date2; cout << "(date_2)  cout << ";
        cout << date2;

    Date date3 = date1 + date2;
    cout << endl << "date_1+date_2--> " << date3 << endl;

    Date date4 = date1 - date2;
    cout << endl << "date_1-date_2--> " << date4 << endl;
    bool l = date1 < date2;
    bool m = date1 >= date2;

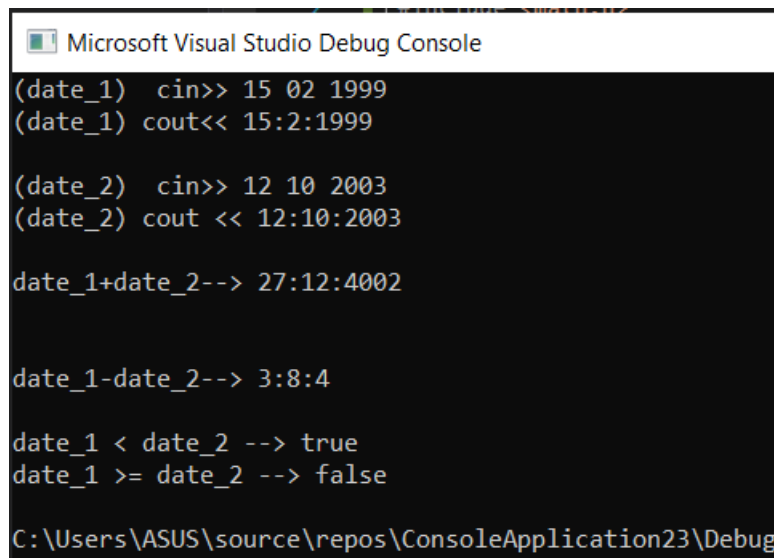
    if (l == 1) cout << "date_1 < date_2 --> true" << endl;
    else cout << "date_1 < date_2 --> false" << endl;

    if (m == 1) cout << "date_1 >= date_2 --> true" << endl;
    else cout << "date_1 >= date_2 --> false" << endl;

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ



```

Microsoft Visual Studio Debug Console
(date_1)  cin>> 15 02 1999
(date_1)  cout<< 15:2:1999

(date_2)  cin>> 12 10 2003
(date_2)  cout << 12:10:2003

date_1+date_2--> 27:12:4002

date_1-date_2--> 3:8:4

date_1 < date_2 --> true
date_1 >= date_2 --> false

C:\Users\ASUS\source\repos\ConsoleApplication23\Debug

```

Висновок: ознайомився зі способами перевантаження операцій та навчився використовувати їх при роботі з об'єктами.