Міністерство освіти і науки України

Львівський національний університет ім. Івана Франка

Кафедра радіофізики і компю’терних технологій

Звіт

Про виконання лабораторної роботи №1

Основи логіки висловлювань

Студента факультету електроніки та

комп’ютерних технологій

Бика Володимира

Група Феі-12

Викладач Шмигельсткий Я.А.

Львів-2019

**Постановка задачі:** створити програму, яка виводить стандартну таблиця істинності логічних зв’язків та таблицю істинності заданої функції.

**Завдання:**

**1.** Створити власну бібліотеку “Логічна консоль”, в якій реалізувати функції, що відповідають наступним логічним операціям: заперечення, кон’юнкція, диз’юнкція, імплікація, еквівалентність, виключаюче або. Створити проект, в якому за допомогою функцій з власної бібліотеки реалізувати вивід стандартної таблиці істинності.

**2.** В бібліотеці запрограмувати реалізацію власної логічної функції. Додати в проект вивід таблиці істинності для даної функції.

**3.** Перевірити правильність результатів побудувавши таблицю істинності даної функції в зошиті.

Всі функції реалізовано за допомогою функцій логічного типу-bool, а також оператора умовного переходу if…else…

**Код заголовного файлу власної бібліотеки (Log1.h)**

#pragma once

bool NOT(bool a);

bool AND(bool a, bool b);

bool OR(bool a, bool b);

bool IMP(bool a, bool b);

bool EQU(bool a, bool b);

bool XOR(bool a, bool b);

bool F2(bool a, bool b, bool c);

**Код реалізації функцій у власній бібліотеці (Lib1.cpp)**

#include "Lib1.h"

bool NOT(bool a)

{

return a ? false : true;

//return !a;

}

bool AND(bool a, bool b)

{

if (a)

if (b)

return true;

return false;

//return a && b;

}

bool OR(bool a, bool b)

{

return a ? a : b;

//return a || b;

}

bool IMP(bool a, bool b)

{

return OR(NOT(a), b);

//return !a || b;

}

bool EQU(bool a, bool b)

{

return OR(AND(a, b), AND(NOT(a), NOT(b)));

//return a == b;

}

bool XOR(bool a, bool b)

{

return EQU(a, b) ? false : OR(a, b);

}

bool F2(bool a, bool b, bool c)

{

return EQU(IMP(a, AND(b, c)), AND(IMP(a, b), IMP(a, c)));

}

**Код основного файлу проекту (Source.cpp)**

#include <iostream>

#include "Lib1.h"

int main()

{

bool a1, b1;

char answer;

int a, b;

bool arr[8][3] = {

{true, true, true},

{false, false, false},

{false, true, true},

{false, false, true},

{true, false, true},

{true, false, false},

{true, true, false},

{false, true, false},

};

while (true) {

std::cout << "Type a and b: ";

std::cin >> a >> b;

a1 = static\_cast<bool>(a);

b1 = static\_cast<bool>(b);

std::cout << "#1 First part (a, b)\n";

std::cout << "NOT(a) : " << NOT(a1) << std::endl;

std::cout << "NOT(b) : " << NOT(b1) << std::endl;

std::cout << "AND: " << AND(a1, b1) << std::endl;

std::cout << "OR " << OR(a1, b1) << std::endl;

std::cout << "IMP " << IMP(a1, b1) << std::endl;

std::cout << "EQU " << EQU(a1, b1) << std::endl;

std::cout << "XOR " << XOR(a1, b1) << std::endl;

// SPECIAL TASK

std::cout << "#2 Second part (a, b, c)\n";

for (auto& elem : arr) {

std::cout << "F2("

<< elem[0] << elem[1] << elem[2] << ") -> "

<< F2(elem[0], elem[1], elem[2]) << std::endl;

}

// CONTINUE ?

std::cout << "Do you wanna continue ? (y/n)\n";

std::cin >> answer;

if (answer != 'y' && answer != 'Y')

return 1;

}

return 1;

}

В даній програмі вивід таблиці істинності представлений у вигляді символів 0 та 1: 0-false, 1-true.

**Скріншот роботи програми**



**Висновки**

У цій лабораторній роботі я реалізував функції логічних операцій та створив програму для виведення на екран таблиць істинності. Працюючи над даною програмою я використовував функції логічного типу(bool) та умовний оператор if…else... Також в цій програмі я реалізував метод створення власних бібліотек. Результатом моєї роботи є програма, яка будує вірні таблиці істинності.