МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи № 1

**«ОСНОВИ ЛОГІКИ ВИСЛОВЛЮВАНЬ»**

**Виконав:**

Студент групи ФЕП-14с

Грицюк Максим Олегович

**Перевірив:**

ас. Баран М. О.

Львів 2025

**Мета:** Вивчити основи логіки висловлювань, та закріпити на практиці побудувавши таблицю істинності.

**Обладнання:**

Комп'ютер з встановленим програмним забезпеченням(Intellij IDEA Ultimate).

**Теоретичні відомості**

**Загально про основи логіки висловлювань:**

Висловлюванням називають розповідне речення, про яке можна сказати, що воно є або істинне, або хибне, але не одне й інше одночасно.

Логічні операції:

1. заперечення (логічне „ні”, позначаення „¬”, комп’ютерна

операція NOT);

2. кон’юнкція (логічне „і”, позначення „∧”, комп’ютерна операція

AND);

3. диз’юнкція (логічне „або”, позначення „ ∨”, комп’ютерна операція

OR);

4. імплікація (читають „якщо..., то…”, позначення „→”);

5. еквівалентність (читають „тоді й лише тоді…”, позначення „~”);

6. виключаюче „або” (позначення „⊕”, комп’ютерна операція

XOR).

Семантика – це сукупність правил, за якими формулам надають

значення істинності. Семантику логічних операцій зручно задавати за

допомогою таблиць, які містять значення істинності формул залежно від

значень істинності їх атомів. Такі таблиці називають таблицями

істинності.

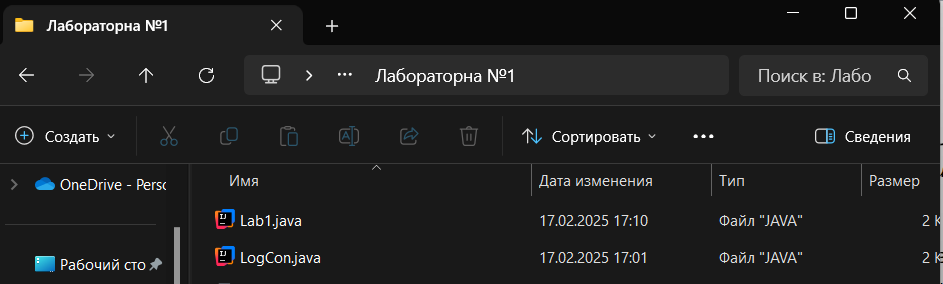
**Хід роботи**

**Частина 1. Побудова логічних зв’язків.**

Завдання №1:*“Створити бібліотеку „Логічна консоль” LogCon (файли LogCon.h,*

*LogCon.cpp).”*

Оскільки я працюю на мові програмування Java в програмному середовищі Intellij IDEA, файли мають розширення відповідно до мови програмування(.java)



Завдання №2:*“У бібліотеці LogCon реалізувати функції, що відповідають логічним зв’язкам:*

*„заперечення” у вигляді функції NOT(a);*

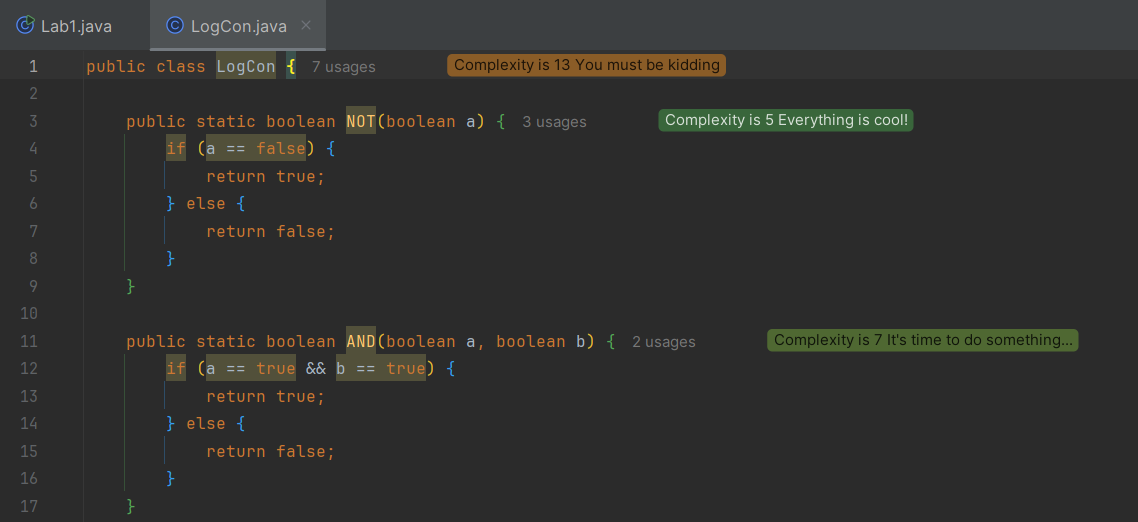
*„кон’юнкція” у вигляді функції AND(a,b)*

*„диз’юнкція” у вигляді функції OR(a,b);*

*„імплікація” у вигляді функції IMP(a,b);*

*„еквівалентність” у вигляді функції EQU(a,b);*

*„виключаюче „або” у вигляді функції XOR(a,b).*



Завдання №3-4-5:*“Створити новий проект Lab\_1 та підключити до нього бібліотеку LogCon.* *У функції main() проекту запрограмувати вивід у консоль*

*таблиці 1.1. з використанням функцій модуля LogCon. Для цього у*

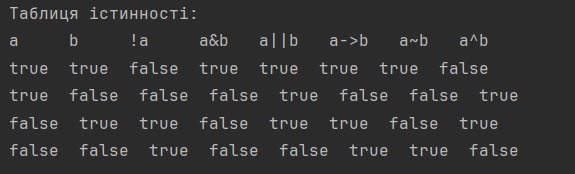
*файлі з функцією main(), окрім бібліотек стандартного вводу/виводу*

*потрібно підключити файл LogCon.h. Відкомпілювати проект та запустити програму на виконання. Продемонструвати результат викладачеві.”*

Код виводу у консоль логічних значень, відповідно до написаного мною раніше модуля, файлу Lab1.



Та кінцевий результат:

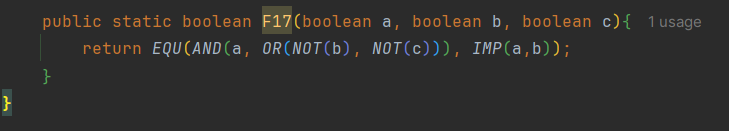


**Частина 2. Побудова таблиць істинності логічних функцій**

Завдання №1: *“У бібліотеці LogCon запрограмувати реалізацію логічної функції,*

*заданої викладачем.”*

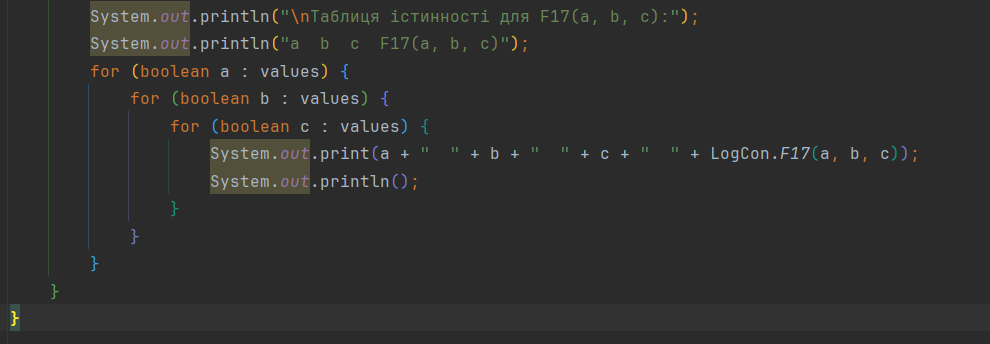
Оскільки в завданні не сказано, що потрібно прописувати умови реалізації логічної функції заново відповідно до необхідної комбінації логічних виразів щоб вирішити мою функцію, я вирішив використати вже наявні, які написав раніше.



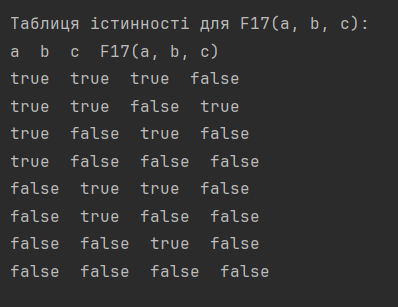
Завдання №2-3: *“* *У функції main() проекту Lab\_1 запрограмувати вивід у консоль*

*таблиці істинності заданої логічної функції.* *Відкомпілювати проект та запустити програму на виконання. Для перевірки результатів побудувати таблицю істинності заданої логічної функції у зошиті. Продемонструвати результат викладачеві.”*

У файлі Lab1.java дописав необхідний функціонал для виведення таблиці істинності, але вже для функції.



Вивід у консолі:

**

**Висновок:** Під час виконання лабораторної роботи я Вивчив побудову логічних зв’язків та основні логічні операції. Опанував створення таблиць істинності для логічних функцій і закріпив отримані знання, розв’язуючи практичні завдання, зокрема для складних висловлювань.

**Додаток:**

*(LogCon.java)*

public class LogCon {  
  
 public static boolean NOT(boolean a) {  
 if (a == false) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean AND(boolean a, boolean b) {  
 if (a == true && b == true) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean OR(boolean a, boolean b) {  
 if (a == true || b == true) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean IMP(boolean a, boolean b) {  
 if (a == false) {  
 return true;  
 } else if (b == true) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean XOR(boolean a, boolean b) {  
 if (a == true && b == false) {  
 return true;  
 } else if (a == false && b == true) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean EQU(boolean a, boolean b) {  
 if (a == b) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 public static boolean F17(boolean a, boolean b, boolean c){  
 return *EQU*(*AND*(a, *OR*(*NOT*(b), *NOT*(c))), *IMP*(a,b));  
 }  
}

*(Lab1.java)*

public class Lab1 {  
 public static void main(String[] args) {  
 System.*out*.println("Таблиця істинності:");  
 System.*out*.println("a b !a a&b a||b a->b a~b a^b");  
 boolean[] values = {true, false};  
 for (boolean a : values) {  
 for (boolean b : values) {  
 System.*out*.print(a + " " + b + " " + LogCon.*NOT*(a) + " " + LogCon.*AND*(a, b) + " " + LogCon.*OR*(a, b) + " " + LogCon.*IMP*(a, b) + " " + LogCon.*EQU*(a, b) + " " + LogCon.*XOR*(a, b));  
 System.*out*.println();  
 }  
 }  
  
 System.*out*.println("\nТаблиця істинності для F17(a, b, c):");  
 System.*out*.println("a b c F17(a, b, c)");  
 for (boolean a : values) {  
 for (boolean b : values) {  
 for (boolean c : values) {  
 System.*out*.print(a + " " + b + " " + c + " " + LogCon.*F17*(a, b, c));  
 System.*out*.println();  
 }  
 }  
 }  
 }  
}