МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

ЗВІТ

Про виконання практичної роботи №2

З дисципліни “Економіка ІТ-індустрії та підприємництво”

На тему “Метрики розміру. Міра line of code”

|  |  |
| --- | --- |
| Виконали:  Студенти груп ІП-15  Мєшков А. І. | Перевірила:  пос. Марченко О. І. |

Київ 2024

ПРАКТИЧНА РОБОТА №2

**Мета роботи:** Ознайомитися з загальними поняттями щодо вимірювань та метрикою розміру з мірою Lines of Code. Напрацювати вміння застосування засобів вимірювання метрики. Отримати загальні вміння щодо застосування метрики в економіці програмного забезпечення.

## ЗАВДАННЯ

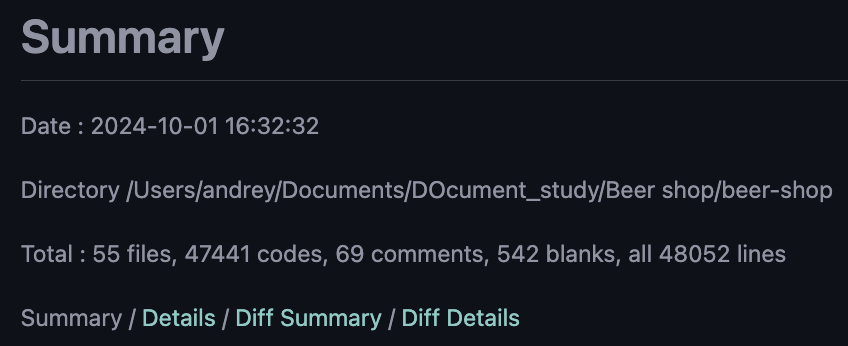
1. Застосовуючи вимірювачі у відповідних середовищах програмування (Visual Studio, Code Counter for Java, CodeCounter, та інші), на прикладі власних програмних текстів виконати вимірювання розміру.
2. Здійснити відповідні економічні розрахунки.
3. Дослідити рівні мов програмування C# та Java.
4. Захистити виконану роботу.

## ХІД РОБОТИ

1. Для початку роботи треба обрати 3 проєкту: Organic, Semi-Detached та Embedded.

Organic – було обрано Інтернет магазин пива, написаний на Javascript. Посилання: <https://github.com/IP15-MieshkovAndrii/beer-shop> .

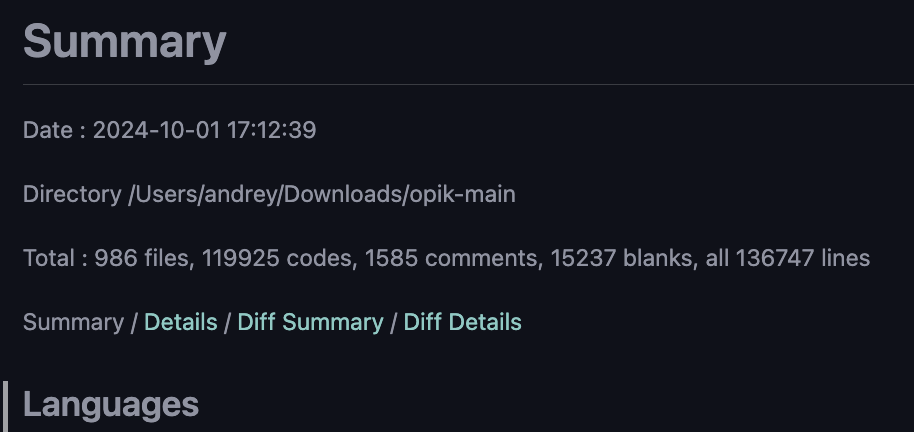
KLOC було виміряно за допомогою VS Code доповнення VS Code Counter. KLOC = 47.



Semi-Detached – Наскрізна платформа розробки LLM з відкритим кодом.

Посилання: <https://github.com/comet-ml/opik> .

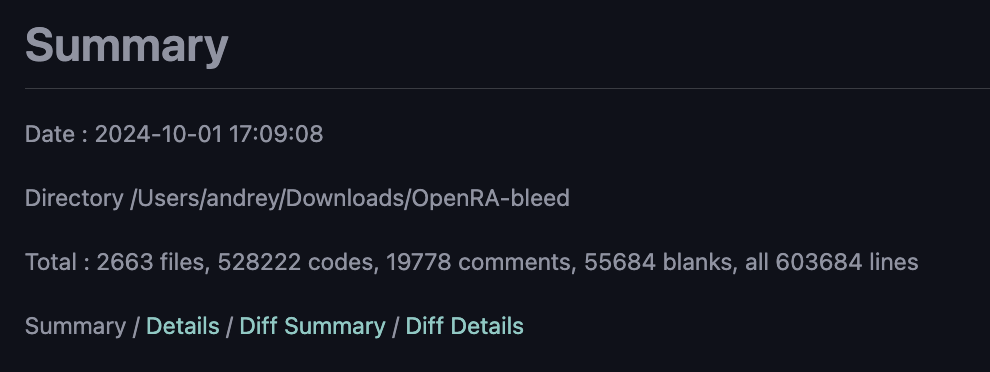
KLOC було виміряно за допомогою VS Code доповнення VS Code Counter. KLOC = 120.



Embedded – Двигун стратегії реального часу з відкритим кодом для ранніх ігор Вествуда, таких як Command & Conquer: Red Alert, написаний на C# з використанням SDL і OpenGL. Працює на Windows, Linux, \*BSD і Mac OS X.

Посилання: <https://github.com/OpenRA/OpenRA> .

KLOC було виміряно за допомогою VS Code доповнення VS Code Counter. KLOC = 528.



1. Виконаємо розрахунки для кожного проєкту.

Effort = ab \*

Cost = Effort \* salary

Schedule = cb \*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип проєкту | аb | bb | cb | db |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Для обраного Organic проєкту size=47. Для цього розрахунку було використано зарплату 29 тис. грн на місяць – середня зарплата Frontend Developer у Києві.

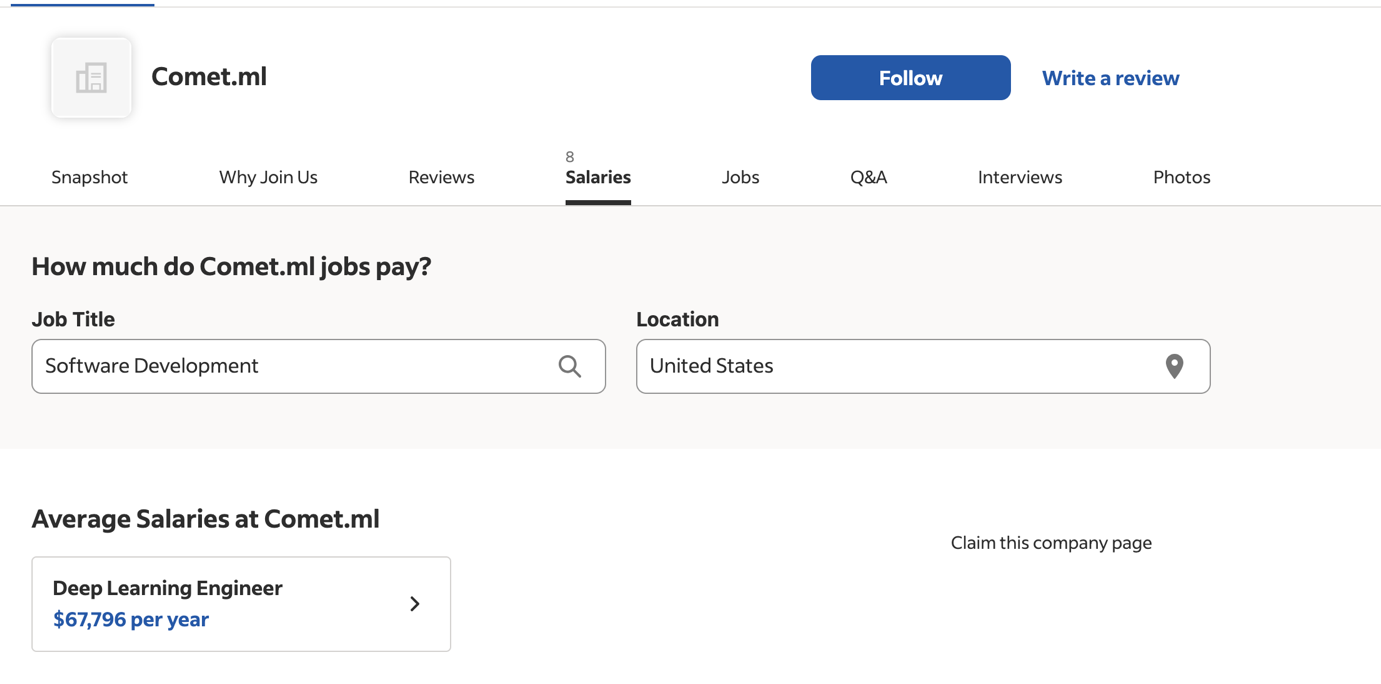


Effort = 2.4\* 137 (людина/місяць)

Cost = 137\* 29,000 =3,973,000(грн)

Schedule = 2.5\* 16(місяців)

Для обраного Semi-Detached проєкту size=120. Для цього розрахунку було використано зарплату $5,650 на місяць – зарплата Deep Learning Engineer.



Effort = 3.0\* 639(людина/місяць)

Cost = 639\* 5,650 =$3,610,350 149,046,187 (грн)

Schedule = 2.5\* 24(місяців)

Для обраного Embedded проєкту size=528. Для цього розрахунку було використано зарплату $12,577 на місяць – середня зарплата Senior Java Software Engineer.



Effort = 3.6\* 6660(людина/місяць)

Cost = 6660\* 12,577 =$83,762,8203,457,983,010 (грн)

Schedule = 2.5\* 42(місяців)

1. Дослідити рівні мов програмування C# та Java.

Для визначення рівнів мов програмування буде використана формула:

Level= BytecodeLOC / LOC,

де BytecodeLOC – кількість рядків коду байткоду, а LOC – кількість рядків програмного коду на мові програмування.

На кожній мові було реалізовано алгоритм сортування бульбашкою. Вимірювання дали такі результати:

C#:

using System;

class BubbleSort {

static void BubbleSortMethod(int[] arr) {

int n = arr.Length;

*for* (int i = 0; i < n-1; i++) {

*for* (int j = 0; j < n-i-1; j++) {

*if* (arr[j] > arr[j+1]) {

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

static void Main(string[] args) {

int[] arr = {64, 34, 25, 12, 22, 11, 90};

BubbleSortMethod(arr);

Console.WriteLine(string.Join(" ", arr));

}

}

Java:

public class BubbleSort {

public static void bubbleSort(int[] arr) {

int n = arr.length;

*for* (int i = 0; i < n-1; i++) {

*for* (int j = 0; j < n-i-1; j++) {

*if* (arr[j] > arr[j+1]) {

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

public static void main(String[] args) {

int[] arr = {64, 34, 25, 12, 22, 11, 90};

bubbleSort(arr);

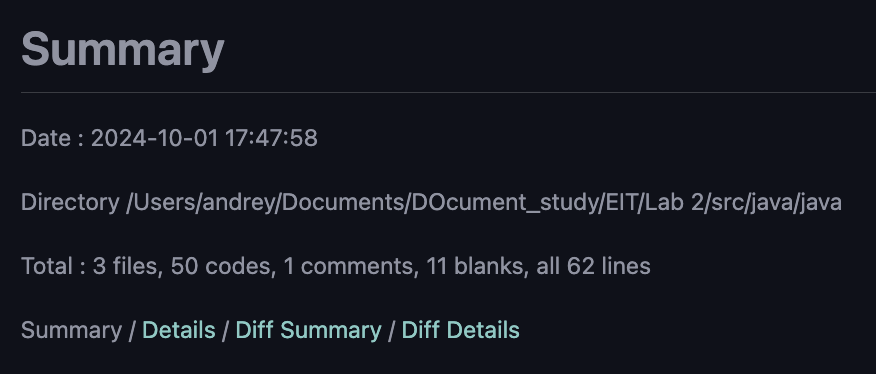
*for* (int i=0; i<arr.length; i++)

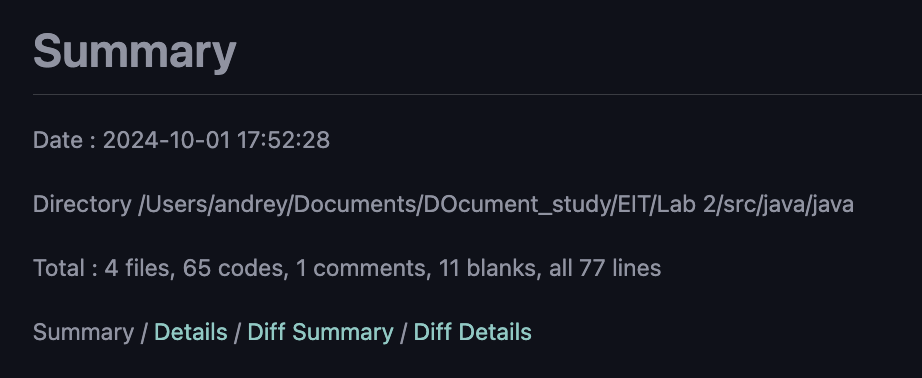
System.out.print(arr[i] + " ");

}

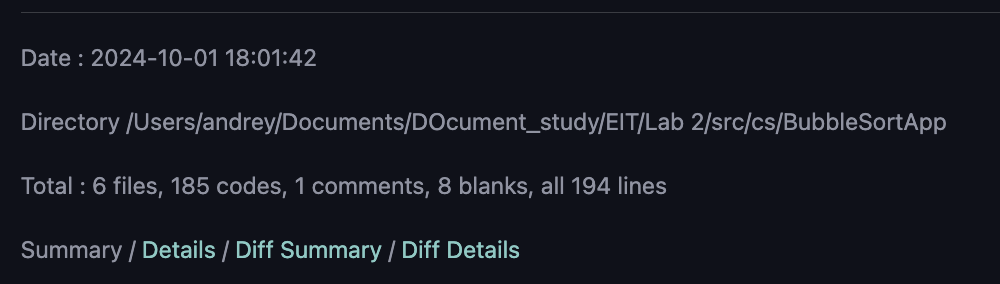
}

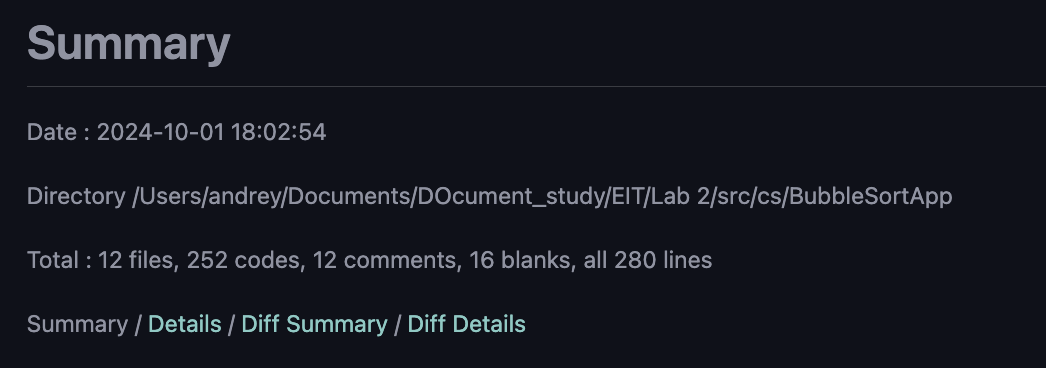
Для Java – до компіляції 50 рядків, а після 65 рядків.





Для С# – до компіляції 185 рядків, а після 252 рядків.





Рівень мови Java становить 50:651:1.3, а для С# 185:2521:1.36.

Результати показують, що С# має більший рівень мови.

## ВИСНОВКИ

У ході виконання практичної роботи було виконано наступні завдання:

1. Ознайомилися з поняттям метрики розміру програмного забезпечення за допомогою вимірювання рядків коду (Lines of Code, LOC).

2. На прикладі трьох проєктів різного типу (Organic, Semi-Detached, Embedded) було проведено вимірювання KLOC та здійснено економічні розрахунки для визначення трудовитрат (Effort), вартості (Cost) та термінів виконання (Schedule). Відповідні формули з використанням параметрів для різних типів проєктів дозволили оцінити витрати на їх розробку.

3. Було визначено рівень мов програмування C# та Java на прикладі реалізації алгоритму сортування бульбашкою. Використовуючи формулу Level = BytecodeLOC / LOC, ми порівняли кількість рядків вихідного коду та байткоду після компіляції для кожної мови.

- Для Java: рівень становить 1:1.3.

- Для C#: рівень становить 1:1.36.

Результати показали, що мова C# має трохи більший рівень у порівнянні з Java, що свідчить про те, що байткод у C# є дещо складнішим і об'ємнішим.

Таким чином, ми отримали практичні навички роботи з метриками коду, а також провели економічні оцінки вартості та термінів виконання програмних проєктів.