

**Міністерство освіти і науки України**  
**Національний технічний університет України «Київський політехнічний інститут імені**  
**Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи №2 з дисципліни  
«Економіка ІТ-індустрії та підприємництво»  
„ МЕТРИКИ РОЗМІРУ. МЕТРИКА LINES OF CODE”

**Виконав(ла)** \_\_\_\_\_ *ІП-11 Дякунчак Ілона*  
(шифр, прізвище, ім'я, по батькові)

**Перевірів** \_\_\_\_\_ *Родіонов П. Ю.*  
(прізвище, ім'я, по батькові)

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

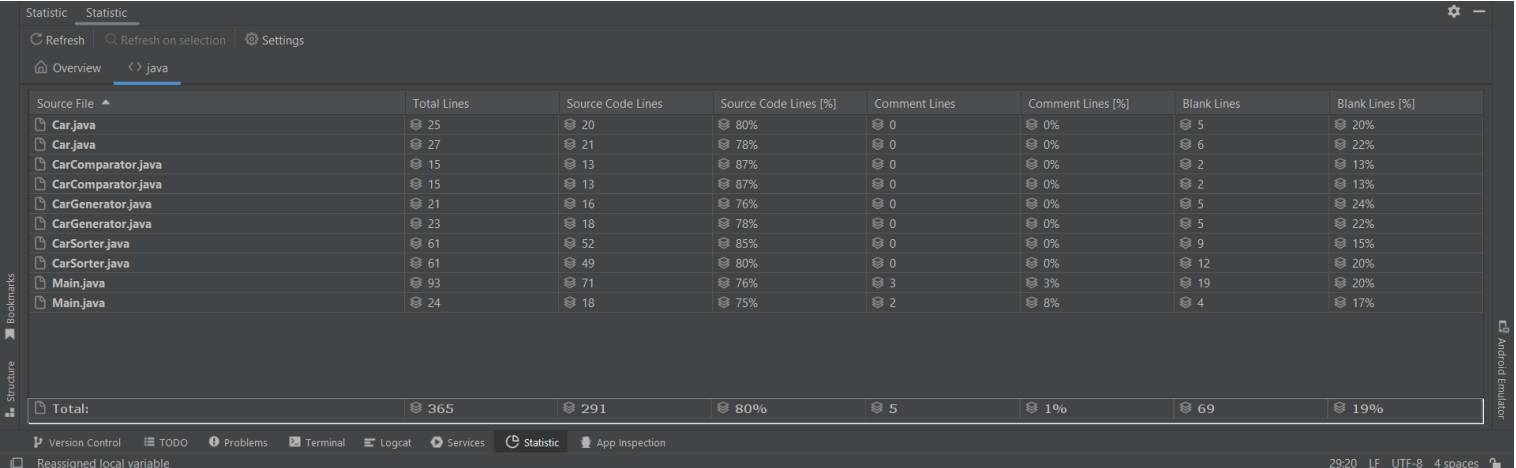
## ЗАВДАННЯ

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.

## ВИКОНАННЯ

1. Вимірювання розміру проекту

Вимірювання розміру я виконувала для проекту своєї минулої курсової роботи, який містить 2 модулі: реалізацію послідовного та паралельного алгоритмів сортування вибором (selection sort) мовою Java. Я розробляла цей проект в середовищі IntelliJ IDEA і для вимірювання використала плагін Statistic. Результати вимірювання наведено на рисунку 1.



Source File	Total Lines	Source Code Lines	Source Code Lines (%)	Comment Lines	Comment Lines (%)	Blank Lines	Blank Lines (%)
Car.java	25	20	80%	0	0%	5	20%
Car.java	27	21	78%	0	0%	6	22%
CarComparator.java	15	13	87%	0	0%	2	13%
CarComparator.java	15	13	87%	0	0%	2	13%
CarGenerator.java	21	16	76%	0	0%	5	24%
CarGenerator.java	23	18	78%	0	0%	5	22%
CarSorter.java	61	52	85%	0	0%	9	15%
CarSorter.java	61	49	80%	0	0%	12	20%
Main.java	93	71	76%	3	3%	19	20%
Main.java	24	18	75%	2	8%	4	17%
<b>Total:</b>	<b>365</b>	<b>291</b>	<b>80%</b>	<b>5</b>	<b>1%</b>	<b>69</b>	<b>19%</b>

Рисунок 1 — Результати вимірювання розміру проекту

Як бачимо, проект містить 10 Java-файлів та 291 рядок коду. З результатів вимірювання можна зробити висновок, що тип проекту — “Органічний”.

2. Економічні розрахунки

Для виконання економічних розрахунків використаємо коефіцієнти з таблиці 1.

Таблиця 1 — Типи проектів

Тип проекту	ab	bb	cb	db
Органічний	2.4	1.05	2.5	0.38
Напіврозділений	3.0	1.12	2.5	0.35
Вбудований	3.6	1.20	2.5	0.32

**Зусилля:**

$$\text{Effort} = ab * \text{size}^{bb} = 2.4 * 0.291^{1.05} = 0,657 \text{ людина/місяць}$$

**Вартість:**

$$\text{Cost} = \text{Effort} * \text{salary} = 0.657 * 24000 = 15768 \text{ грн.}$$

**Час на розробку:**

$$\text{Schedule} = cb * \text{Effort}^{db} = 2.5 * 0.657^{0.38} = 2.13 \text{ місяці}$$

У реальності розробляла проект протягом 2 тижнів лише я. Розбіжності між теоретичними розрахунками та реальністю при оцінці проекту на Java з використанням метрики "lines of code" (LOC) можуть виникати через кілька причин, наприклад через використання фреймворків, які приховують частину логіки за допомогою конфігурацій, що зменшує кількість вручну написаного коду.

3. Дослідження рівнів мов програмування C# та Java

Щоб дослідити рівні мов програмування C# та Java, я створила два класи, які реалізують алгоритм сортування “бульбашкою” та використала сайт <https://godbolt.org/>.

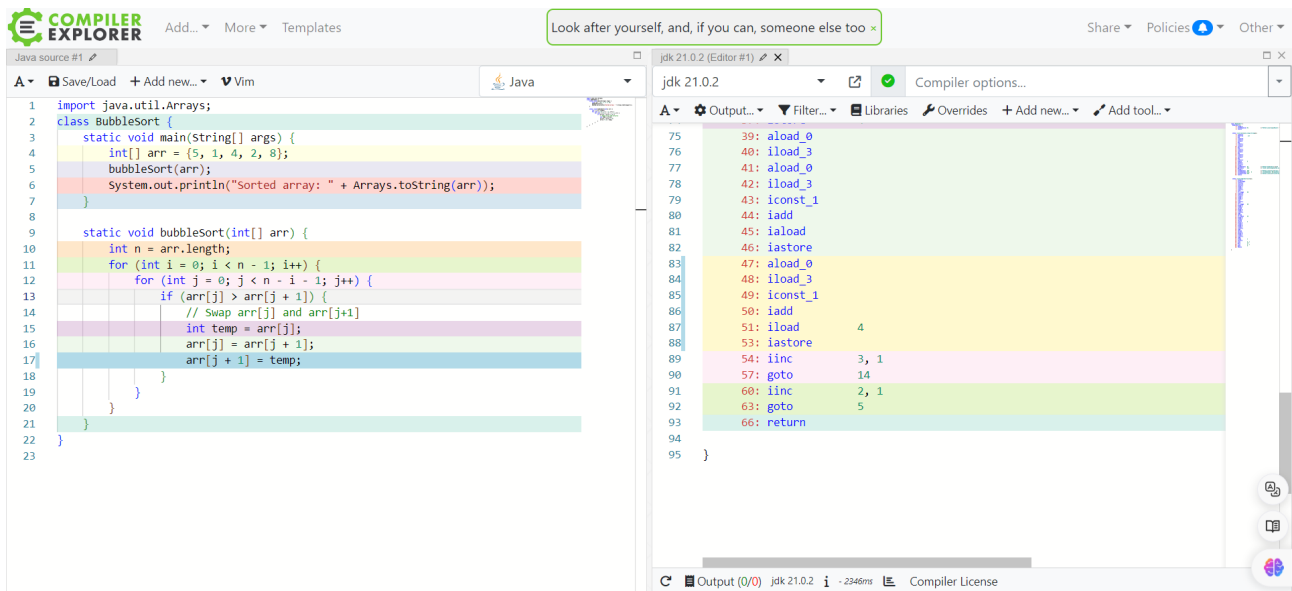


Рисунок 2 — Перетворення Java-коду в байт-код

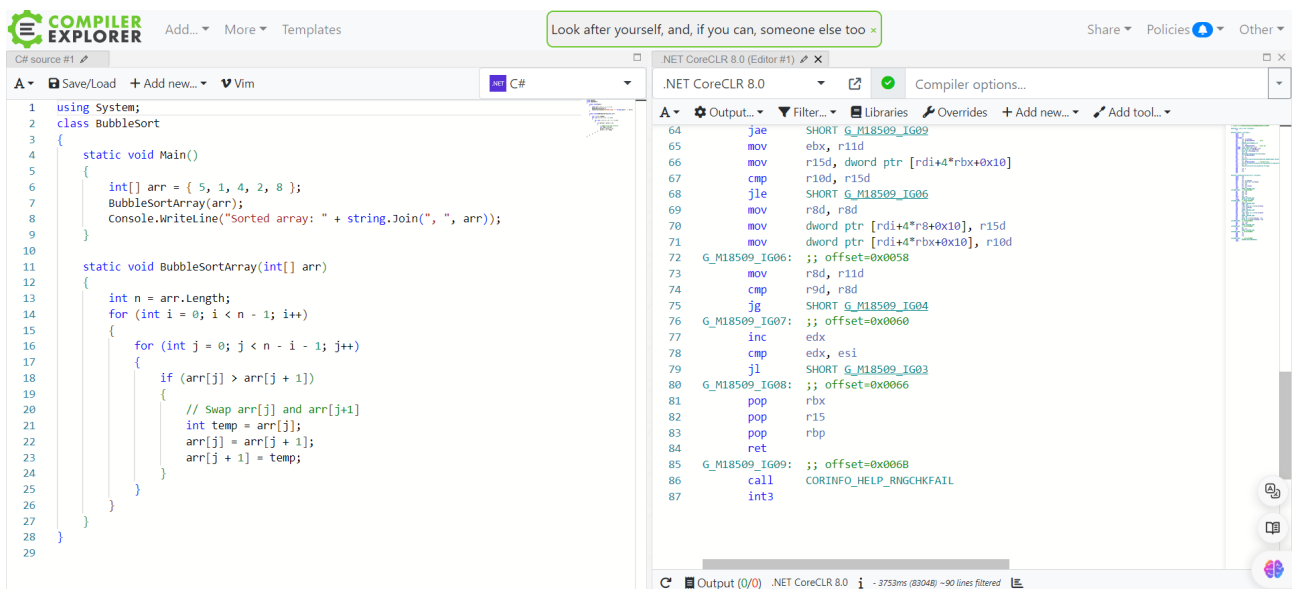


Рисунок 3 — Перетворення C#-коду в код асемблера

На рисунках 2 і 3 показані результати вимірювань. Далі обчислимо відношення кількості рядків асемблера до кількості рядків коду на C# та Java:

Для Java:  $95/23 = 4,13$

Для C#:  $87/29 = 3$

Як бачимо, обчислені значення не значні, проте в більших проектах відношення може бути набагато більшим.

## ВИСНОВОК

Під час виконання практичної роботи я ознайомила з загальними поняттями щодо вимірювань та метрикою розміру Lines of Code. Використовуючи плагін

Statistic в середовищі IntelliJ IDEA, я виконала вимірювання розміру та виконала економічні розрахунки для власного проекту, розробленого на МП Java. Результати розрахунків порівняла з реальністю та дослідила рівні мов C# та Java, обчисливши відношення згенерованого асемблерного коду та байт-коду до звичайного коду. Виконавши дослідження я виявила, що метрика Lines of Code може не враховувати рівень мови програмування, стиль програмування, якість коду. Тому вона не є надійним індикатором продуктивності, якості або складності програмного забезпечення.