



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

## **Практична робота №5**

### **Економіка ІТ-індустрії**

**Тема:** Оцінювання вартості і інших характеристик програмного забезпечення за методом аналогії

Виконав  
студент групи ІП-11:  
Панченко С. В.

Перевірив:  
Родіонов П. В.

Київ 2024

## ЗМІСТ

1 Мета.....	7
2 Завдання.....	8
3 Виконання.....	9
3.1 Аналіз функціональних точок RWLock.....	9
3.1.1 Функції даних.....	9
3.1.2 Транзакційні функції.....	9
3.1.3 Розрахунок нескоригованих функціональних точок (UFP).....	10
3.1.4 Оцінка загальних характеристик системи (GSC).....	10
3.1.5 Сума значень GSC (TDI).....	11
3.1.6 Розрахунок фактору коригування значень (VAF).....	11
3.1.7 Розрахунок скоригованих функціональних точок (AFP).....	11
3.1.8 Висновок.....	11
3.2 Аналіз функціональних точок NBUCurrencyTracker.....	11
3.2.1 Функції даних.....	11
3.2.2 Транзакційні функції.....	12
3.2.3 Розрахунок нескоригованих функціональних точок (UFP).....	13
3.2.4 Оцінка загальних характеристик системи (GSC).....	13
3.2.5 Сума значень GSC (TDI).....	13
3.2.6 Розрахунок фактору коригування значень (VAF).....	14
3.2.7 Розрахунок скоригованих функціональних точок (AFP).....	14
3.2.8 Висновок.....	14
3.3 Аналіз функціональних точок CppFuse.....	14
3.3.1 Функції даних.....	14
3.3.2 Транзакційні функції.....	15
3.3.3 Розрахунок нескоригованих функціональних точок (UFP).....	15
3.3.4 Оцінка загальних характеристик системи (GSC).....	16
3.3.5 Сума значень GSC (TDI).....	16
3.3.6 Розрахунок фактору коригування значень (VAF).....	16

3.3.7 Розрахунок скоригованих функціональних точок (AFP).....	16
3.3.8 Висновок.....	16
3.4 Аналіз функціональних точок MCTS Checkers.....	17
3.4.1 Функції даних.....	17
3.4.2 Транзакційні функції.....	17
3.4.3 Розрахунок нескоригованих функціональних точок (UFP).....	18
3.4.4 Оцінка загальних характеристик системи (GSC).....	18
3.4.5 Сума значень GSC (TDI).....	19
3.4.6 Розрахунок фактору коригування значень (VAF).....	19
3.4.7 Розрахунок скоригованих функціональних точок (AFP).....	19
3.4.8 Висновок.....	19
3.5 Аналіз функціональних точок Flask Project.....	19
3.5.1 Функції даних.....	19
3.5.2 Транзакційні функції.....	20
3.5.3 Розрахунок нескоригованих функціональних точок (UFP).....	21
3.5.4 Оцінка загальних характеристик системи (GSC).....	21
3.5.5 Сума значень GSC (TDI).....	21
3.5.6 Розрахунок фактору коригування значень (VAF).....	21
3.5.7 Розрахунок скоригованих функціональних точок (AFP).....	22
3.5.8 Висновок.....	22
3.6 Аналіз функціональних точок Functional Programming Haskell.....	22
3.6.1 Функції даних.....	22
3.6.2 Транзакційні функції.....	22
3.6.3 Розрахунок нескоригованих функціональних точок (UFP).....	23
3.6.4 Оцінка загальних характеристик системи (GSC).....	23
3.6.5 Сума значень GSC (TDI).....	24
3.6.6 Розрахунок фактору коригування значень (VAF).....	24
3.6.7 Розрахунок скоригованих функціональних точок (AFP).....	24
3.6.8 Висновок.....	24
3.7 Аналіз функціональних точок Network Programming.....	24

3.7.1 Функції даних.....	24
3.7.2 Транзакційні функції.....	25
3.7.3 Розрахунок нескоригованих функціональних точок (UFP).....	26
3.7.4 Оцінка загальних характеристик системи (GSC).....	26
3.7.5 Сума значень GSC (TDI).....	26
3.7.6 Розрахунок фактору коригування значень (VAF).....	26
3.7.7 Розрахунок скоригованих функціональних точок (AFP).....	27
3.7.8 Висновок.....	27
3.8 Аналіз функціональних точок System Programming.....	27
3.8.1 Функції даних.....	27
3.8.2 Транзакційні функції.....	27
3.8.3 Розрахунок нескоригованих функціональних точок (UFP).....	28
3.8.4 Оцінка загальних характеристик системи (GSC).....	28
3.8.5 Сума значень GSC (TDI).....	29
3.8.6 Розрахунок фактору коригування значень (VAF).....	29
3.8.7 Розрахунок скоригованих функціональних точок (AFP).....	29
3.8.8 Висновок.....	29
3.9 Аналіз функціональних точок Petri Object Model Paint.....	29
3.9.1 Функції даних.....	29
3.9.2 Транзакційні функції.....	30
3.9.3 Розрахунок нескоригованих функціональних точок (UFP).....	31
3.9.4 Оцінка загальних характеристик системи (GSC).....	31
3.9.5 Сума значень GSC (TDI).....	31
3.9.6 Розрахунок фактору коригування значень (VAF).....	31
3.9.7 Розрахунок скоригованих функціональних точок (AFP).....	32
3.9.8 Висновок.....	32
3.10 Аналіз функціональних точок Console Chess.....	32
3.10.1 Функції даних.....	32
3.10.2 Транзакційні функції.....	32
3.10.3 Розрахунок нескоригованих функціональних точок (UFP).....	33

3.10.4 Оцінка загальних характеристик системи (GSC).....	33
3.10.5 Сума значень GSC (TDI).....	34
3.10.6 Розрахунок фактору коригування значень (VAF).....	34
3.10.7 Розрахунок скоригованих функціональних точок (AFP).....	34
3.10.8 Висновок.....	34
3.11 Аналіз функціональних точок CPong.....	34
3.11.1 Функції даних.....	34
3.11.2 Транзакційні функції.....	35
3.11.3 Розрахунок нескоригованих функціональних точок (UFP).....	36
3.11.4 Оцінка загальних характеристик системи (GSC).....	36
3.11.5 Сума значень GSC (TDI).....	36
3.11.6 Розрахунок фактору коригування значень (VAF).....	36
3.11.7 Розрахунок скоригованих функціональних точок (AFP).....	37
3.11.8 Висновок.....	37
3.12 Аналіз функціональних точок Personal Website.....	37
3.12.1 Функції даних.....	37
3.12.2 Транзакційні функції.....	37
3.12.3 Розрахунок нескоригованих функціональних точок (UFP).....	38
3.12.4 Оцінка загальних характеристик системи (GSC).....	38
3.12.5 Сума значень GSC (TDI).....	39
3.12.6 Розрахунок фактору коригування значень (VAF).....	39
3.12.7 Розрахунок скоригованих функціональних точок (AFP).....	39
3.12.8 Висновок.....	39
3.13 Аналіз функціональних точок Software Security Course.....	39
3.13.1 Функції даних.....	39
3.13.2 Транзакційні функції.....	40
3.13.3 Розрахунок нескоригованих функціональних точок (UFP).....	41
3.13.4 Оцінка загальних характеристик системи (GSC).....	41
3.13.5 Сума значень GSC (TDI).....	41
3.13.6 Розрахунок фактору коригування значень (VAF).....	41

3.13.7 Розрахунок скоригованих функціональних точок (AFP).....	42
3.13.8 Висновок.....	42
3.14 Аналіз функціональних точок Multithreaded Programming Course.....	42
3.14.1 Функції даних.....	42
3.14.2 Транзакційні функції.....	42
3.14.3 Розрахунок нескоригованих функціональних точок (UFP).....	43
3.14.4 Оцінка загальних характеристик системи (GSC).....	43
3.14.5 Сума значень GSC (TDI).....	44
3.14.6 Розрахунок фактору коригування значень (VAF).....	44
3.14.7 Розрахунок скоригованих функціональних точок (AFP).....	44
3.14.8 Висновок.....	44
3.15 Оцінка проекту.....	45
Висновок.....	48

## **1 МЕТА**

Навчитися здійснювати оцінку проєкту за рахунок використання методу аналогій.

## 2 ЗАВДАННЯ

Використовуючи власні наробки, або веб сервіси для спільної розробки програмного забезпечення (наприклад GitHub), або проєктуючи з завданням викладача відповідні застосунки виконати наступне:

1. Вибрати проєкти та створити базу даних проєктів. Навести посилання на вибрані проєкти, для можливості викладачем перегляду проєктів і оцінювання вірності результатів що будуть отримані студентом під час виконання розрахунків.
2. Вибрати проєкт, який відкладаємо в сторону – він буде як «новий проєкт» що використовується для оцінки зусиль проєкту програмного забезпечення за аналогією.
3. Визначити необхідні для застосування методу аналогії атрибути (характеристики) «нового проєкту», не враховувати в ці атрибути size, Effort та мову програмування проєкту. Обчислити значення відповідні характеристик для всіх проєктів.
4. Внести всі дані у власний Database.
5. Застосовуючи одну або декілька метрик відстані, обчислити відстані усіх проєктів до «нового» проєкту, внести у свій data set. Виявити три найближчі проєкти. Навести розрахунки, найближчі проєкти позначити – наприклад іншим кольором.
6. Оцінити економічні показники нового проєкту. При оцінці враховувати мову програмування аналогічних проєктів (оскільки «новий» проєкт має розроблятися на певній мові і відповідно для застосування методу аналогій повинна бути вибрана ця мова або близька).



## 3 ВИКОНАННЯ

### 3.1 Аналіз функціональних точок RWLock

Репозиторій [SideShowBoBGOT/RwLock](#) містить реалізацію механізму блокування читання-запису (Read-Write Lock) на C++. Для визначення нескоригованих функціональних точок (UFP) проаналізуємо функції даних та транзакційні функції, представлені в коді.

#### 3.1.1 Функції даних

Внутрішні логічні файли (ILF):

- Клас RWLock зберігає стан блокування та інформацію про потоки, що взаємодіють з ним.
- Кількість елементів даних (DET): 5
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з іншими системами або зовнішніми джерелами даних відсутня.
- Кількість EIF: 0

#### 3.1.2 Транзакційні функції

Зовнішні введення (EI):

- Методи `lock_read()`, `lock_write()`, `unlock_read()` та `unlock_write()` приймають вхідні дані для встановлення або зняття блокування.
- Кількість EI: 4
- Кількість елементів даних (DET) на кожен метод: 3
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

Зовнішні виведення (EO):

- Метод `get_status()` повертає інформацію про поточний стан блокування.
- Кількість EO: 1
- Кількість елементів даних (DET): 5
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 4

Зовнішні запити (EQ):

- Метод `is_locked()` перевіряє, чи встановлено блокування, і повертає відповідь без зміни стану системи.
- Кількість EQ: 1
- Кількість елементів даних (DET): 3
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

### 3.1.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EI \times 3) + (EO \times 4) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (4 \times 3) + (1 \times 4) + (1 \times 3) = 7 + 12 + 4 + 3 = 26$$

### 3.1.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 0
- Розподілена обробка даних: 0
- Продуктивність: 3
- Обмеження апаратних ресурсів: 0
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 0
- Ергономіка: 0
- Інтенсивність зміни даних: 1
- Складність обробки: 2

- Повторне використання: 3
- Зручність інсталяції: 2
- Зручність адміністрування: 1
- Можливість портування: 4
- Гнучкість: 3

### 3.1.5 Сума значень GSC (TDI)

$$TDI = 0 + 0 + 3 + 0 + 2 + 0 + 0 + 1 + 2 + 3 + 2 + 1 + 4 + 3 = 21$$

### 3.1.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 21) + 0.65 = 0.86$$

### 3.1.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 26 \times 0.86 \approx 22.36 \approx 22$$

### 3.1.8 Висновок

- Нескориговані функціональні точки (UFP): 26
- Скориговані функціональні точки (AFP): 22

## 3.2 Аналіз функціональних точок NBUCurrencyTracker

Репозиторій [SideShowBoBGOT/NBUCurrencyTracker](https://github.com/SideShowBoBGOT/NBUCurrencyTracker) містить C++ застосунок для отримання валютних курсів з API Національного банку України (НБУ). Застосунок використовує кілька зовнішніх бібліотек, таких як CLI11, cpr-http-lib, fmt, FTXUI та інші, для створення командного інтерфейсу, здійснення HTTP-запитів, форматування рядків та побудови текстового інтерфейсу користувача.

### 3.2.1 Функції даних

Внутрішні логічні файли (ILF):

- Класи, що зберігають конфігураційні параметри та отримані дані про валютні курси.
  - Кількість елементів даних (DET): 10
  - Кількість типів записів (RET): 2
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 10
- Зовнішні інтерфейсні файли (EIF):
- Конфігураційний файл config.toml, що зберігає налаштування застосунку.
  - Кількість елементів даних (DET): 5
  - Кількість типів записів (RET): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 5

### 3.2.2 Транзакційні функції

Зовнішні введення (EI):

- Методи для зчитування конфігураційних параметрів та введення команд користувача.
- Кількість EI: 3
- Кількість елементів даних (DET) на кожен метод: 5
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

Зовнішні виведення (EO):

- Методи для виведення отриманих валютних курсів та повідомлень про помилки.
- Кількість EO: 2
- Кількість елементів даних (DET): 7
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 5

Зовнішні запити (EQ):

- Методи для запиту поточних налаштувань та статусу застосунку.
- Кількість EQ: 2
- Кількість елементів даних (DET): 4
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

### 3.2.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 10) + (EIF \times 5) + (EI \times 4) + (EO \times 5) + (EQ \times 3)$$

$$UFP = (1 \times 10) + (1 \times 5) + (3 \times 4) + (2 \times 5) + (2 \times 3) = 10 + 5 + 12 + 10 + 6 =$$

43

### 3.2.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 2
- Продуктивність: 3
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 3
- Ергономіка: 2
- Інтенсивність зміни даних: 2
- Складність обробки: 3
- Повторне використання: 3
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 3
- Гнучкість: 3

### 3.2.5 Сума значень GSC (TDI)

$$TDI = 3 + 2 + 3 + 1 + 2 + 3 + 2 + 2 + 3 + 3 + 2 + 2 + 3 + 3 = 34$$

### 3.2.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 34) + 0.65 = 0.99$$

### 3.2.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 43 \times 0.99 \approx 42.57 \approx 43$$

### 3.2.8 Висновок

- Нескориговані функціональні точки (UFP): 43
- Скориговані функціональні точки (AFP): 43

## 3.3 Аналіз функціональних точок CppFuse

Репозиторій [SideShowBoBGOT/CppFuse](https://github.com/SideShowBoBGOT/CppFuse) містить реалізацію файлової системи на основі бібліотеки FUSE (Filesystem in Userspace) з підтримкою багатопотоковості. Ця система надає стандартні файлові операції, такі як getattr, readlink, mknod, mkdir, unlink, rmdir, symlink, chmod, read, write та readdir.

### 3.3.1 Функції даних

Внутрішні логічні файли (ILF):

- Класи, що зберігають метадані файлів та директорій, а також інформацію про стан блокувань.
- Кількість елементів даних (DET): 15
- Кількість типів записів (RET): 3
- Рівень складності: Високий
- Ваговий коефіцієнт: 15

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з іншими системами або зовнішніми джерелами даних відсутня.
- Кількість EIF: 0

### 3.3.2 Транзакційні функції

Зовнішні введення (EI):

- Методи для створення файлів (mknod), створення директорій (mkdir), видалення файлів (unlink), видалення директорій (rmdir), створення символічних посилань (symlink), зміни прав доступу (chmod).
- Кількість EI: 6
- Кількість елементів даних (DET) на кожен метод: 7
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Високий
- Ваговий коефіцієнт: 6

Зовнішні виведення (EO):

- Методи для читання файлів (read), отримання атрибутів файлів (getattr), читання символічних посилань (readlink).
- Кількість EO: 3
- Кількість елементів даних (DET): 10
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Високий
- Ваговий коефіцієнт: 7

Зовнішні запити (EQ):

- Методи для перерахунку вмісту директорій (readdir), перевірки доступу (access).
- Кількість EQ: 2
- Кількість елементів даних (DET): 5
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

### 3.3.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 15) + (EIF \times 5) + (EI \times 6) + (EO \times 7) + (EQ \times 4)$$

$$UFP = (1 \times 15) + (0 \times 5) + (6 \times 6) + (3 \times 7) + (2 \times 4) = 15 + 0 + 36 + 21 + 8 =$$

80

### 3.3.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 4
- Продуктивність: 4
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 3
- Інтенсивність взаємодії з користувачем: 2
- Ергономіка: 2
- Інтенсивність зміни даних: 3
- Складність обробки: 4
- Повторне використання: 3
- Зручність інсталяції: 3
- Зручність адміністрування: 3
- Можливість портування: 4
- Гнучкість: 4

### 3.3.5 Сума значень GSC (TDI)

$$TDI = 3 + 4 + 4 + 2 + 3 + 2 + 2 + 3 + 4 + 3 + 3 + 3 + 4 + 4 = 44$$

### 3.3.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 44) + 0.65 = 1.09$$

### 3.3.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 80 \times 1.09 \approx 87.2 \approx 87$$

### 3.3.8 Висновок

- Нескориговані функціональні точки (UFP): 80



- Скориговані функціональні точки (AFP): 87

### 3.4 Аналіз функціональних точок MCTS Checkers

Репозиторій [SideShowBoBGOT/mcts\\_checkers](https://github.com/SideShowBoBGOT/mcts_checkers) містить реалізацію гри в шашки з використанням алгоритму Монте-Карло Tree Search (MCTS). Ця програма дозволяє гравцеві змагатися з комп'ютером, який приймає рішення на основі MCTS.

#### 3.4.1 Функції даних

Внутрішні логічні файли (ILF):

- Клас, що представляє стан ігрової дошки та зберігає інформацію про розташування шашок.
- Кількість елементів даних (DET): 64 (по одному для кожної клітинки дошки)
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з зовнішніми файлами або системами не передбачена.
- Кількість EIF: 0

#### 3.4.2 Транзакційні функції

Зовнішні введення (EI):

- Методи для введення ходів гравця та налаштувань гри.
- Кількість EI: 2
- Кількість елементів даних (DET) на кожен метод: 5
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

Зовнішні виведення (EO):

- Методи для відображення стану дошки та результатів гри.

- Кількість EO: 2
  - Кількість елементів даних (DET): 10
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 5
- Зовнішні запити (EQ):
- Методи для отримання поточного стану гри та можливих ходів.
  - Кількість EQ: 2
  - Кількість елементів даних (DET): 8
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 4

### **3.4.3 Розрахунок нескоригованих функціональних точок (UFP)**

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 3) + (EO \times 5) + (EQ \times 4)$$

$$UFP = (1 \times 7) + (0 \times 5) + (2 \times 3) + (2 \times 5) + (2 \times 4) = 7 + 0 + 6 + 10 + 8 = 31$$

### **3.4.4 Оцінка загальних характеристик системи (GSC)**

- Обмін даними: 2
- Розподілена обробка даних: 1
- Продуктивність: 3
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 3
- Ергономіка: 3
- Інтенсивність зміни даних: 2
- Складність обробки: 4
- Повторне використання: 2
- Зручність інсталяції: 3
- Зручність адміністрування: 2

- Можливість портування: 3
- Гнучкість: 3

#### 3.4.5 Сума значень GSC (TDI)

$$TDI = 2 + 1 + 3 + 2 + 2 + 3 + 3 + 2 + 4 + 2 + 3 + 2 + 3 + 3 = 35$$

#### 3.4.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 35) + 0.65 = 1.00$$

#### 3.4.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 31 \times 1.00 = 31$$

#### 3.4.8 Висновок

- Нескориговані функціональні точки (UFP): 31
- Скориговані функціональні точки (AFP): 31

### 3.5 Аналіз функціональних точок Flask Project

Репозиторій [SideShowBoBGOT/FlaskProject](https://github.com/SideShowBoBGOT/FlaskProject) містить веб-застосунок, розроблений на основі Flask, який забезпечує аутентифікацію користувачів та управління базою даних користувачів, відділів і співробітників. Застосунок підтримує функції додавання, редагування та видалення записів, а також надає веб-інтерфейс для взаємодії з цими даними.

#### 3.5.1 Функції даних

Внутрішні логічні файли (ILF):

- Таблиці бази даних для зберігання інформації про користувачів, відділи та співробітників.
- Кількість елементів даних (DET): приблизно 20 (наприклад, ім'я, прізвище, електронна пошта, відділ, посада тощо)
- Кількість типів записів (RET): 3 (користувачі, відділи, співробітники)

- Рівень складності: Середній
- Ваговий коефіцієнт: 10
- Зовнішні інтерфейсні файли (EIF):
- Конфігураційні файли, такі як config.py, які зберігають налаштування застосунку.
- Кількість елементів даних (DET): 5
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 5

### 3.5.2 Транзакційні функції

Зовнішні введення (EI):

- Форми для введення даних користувачів, відділів та співробітників.
- Кількість EI: 3
- Кількість елементів даних (DET) на кожну форму: 7
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

Зовнішні виведення (EO):

- Сторінки для відображення списків користувачів, відділів та співробітників.
- Кількість EO: 3
- Кількість елементів даних (DET): 10
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Середній
- Ваговий коефіцієнт: 5

Зовнішні запити (EQ):

- Пошукові функції для фільтрації співробітників за датою народження або іншими параметрами.
- Кількість EQ: 2

- Кількість елементів даних (DET): 5
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

### 3.5.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 10) + (EIF \times 5) + (EI \times 4) + (EO \times 5) + (EQ \times 4)$$

$$UFP = (1 \times 10) + (1 \times 5) + (3 \times 4) + (3 \times 5) + (2 \times 4) = 10 + 5 + 12 + 15 + 8 =$$

50

### 3.5.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 2
- Розподілена обробка даних: 1
- Продуктивність: 3
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 3
- Ергономіка: 3
- Інтенсивність зміни даних: 2
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 3
- Гнучкість: 3

### 3.5.5 Сума значень GSC (TDI)

$$TDI = 2 + 1 + 3 + 1 + 2 + 3 + 3 + 2 + 3 + 2 + 2 + 2 + 3 + 3 = 32$$

### 3.5.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 32) + 0.65 = 0.97$$

### 3.5.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 50 \times 0.97 = 48.5 \approx 49$$

### 3.5.8 Висновок

- Нескориговані функціональні точки (UFP): 50
- Скориговані функціональні точки (AFP): 49

## 3.6 Аналіз функціональних точок Functional Programming Haskell

Репозиторій [SideShowBoBGOT/FunctionalProgrammingHaskell](https://github.com/SideShowBoBGOT/FunctionalProgrammingHaskell) містить лабораторні роботи з функціонального програмування на мові Haskell. Кожна лабораторна робота представляє окремий модуль з певною кількістю функцій та операцій.

### 3.6.1 Функції даних

Внутрішні логічні файли (ILF):

- Кожен модуль (Lab1.hs, Lab2.hs, тощо) містить визначення типів даних та функцій.
- Кількість елементів даних (DET): В середньому 10 на модуль
- Кількість типів записів (RET): 1 на модуль
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з зовнішніми системами або файлами не передбачена.
- Кількість EIF: 0

### 3.6.2 Транзакційні функції

Зовнішні введення (EI):

- Функції, що приймають вхідні параметри від користувача.
- Кількість EI: В середньому 5 на модуль

- Кількість елементів даних (DET) на кожну функцію: 3
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

Зовнішні виведення (EO):

- Функції, що повертають результати обчислень.
- Кількість EO: В середньому 5 на модуль
- Кількість елементів даних (DET): 3
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 4

Зовнішні запити (EQ):

- Функції, що здійснюють запити без зміни стану системи.
- Кількість EQ: В середньому 3 на модуль
- Кількість елементів даних (DET): 3
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

### 3.6.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 3) + (EO \times 4) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (0 \times 5) + (5 \times 3) + (5 \times 4) + (3 \times 3) = 7 + 0 + 15 + 20 + 9 = 51$$

### 3.6.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 1
- Розподілена обробка даних: 1
- Продуктивність: 2
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 1
- Інтенсивність взаємодії з користувачем: 2

- Ергономіка: 2
- Інтенсивність зміни даних: 1
- Складність обробки: 2
- Повторне використання: 2
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 2
- Гнучкість: 2

### 3.6.5 Сума значень GSC (TDI)

$$TDI = 1 + 1 + 2 + 1 + 1 + 2 + 2 + 1 + 2 + 2 + 2 + 2 + 2 + 2 = 23$$

### 3.6.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 23) + 0.65 = 0.88$$

### 3.6.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 51 \times 0.88 \approx 44.88 \approx 45$$

### 3.6.8 Висновок

- Нескориговані функціональні точки (UFP): 51
- Скориговані функціональні точки (AFP): 45

## 3.7 Аналіз функціональних точок Network Programming

Репозиторій [SideShowBoBGOT/network\\_programming\\_eighth\\_semester](https://github.com/SideShowBoBGOT/network_programming_eighth_semester) містить матеріали з мережевого програмування, включаючи лекції, завдання та лабораторні роботи, написані переважно мовою C.

### 3.7.1 Функції даних

Внутрішні логічні файли (ILF):



- Код містить структури даних для обробки мережеских з'єднань та повідомлень.
  - Кількість елементів даних (DET): приблизно 15
  - Кількість типів записів (RET): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 7
- Зовнішні інтерфейсні файли (EIF):
- Взаємодія з зовнішніми конфігураційними файлами або бібліотеками.
  - Кількість елементів даних (DET): приблизно 10
  - Кількість типів записів (RET): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 5

### 3.7.2 Транзакційні функції

Зовнішні введення (EI):

- Функції для прийому даних від користувача або мережі.
- Кількість EI: 5
- Кількість елементів даних (DET): 8
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

Зовнішні виведення (EO):

- Функції для відправки даних користувачу або в мережу.
- Кількість EO: 5
- Кількість елементів даних (DET): 8
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 5

Зовнішні запити (EQ):

- Функції для запиту стану з'єднання або даних.

- Кількість EQ: 3
- Кількість елементів даних (DET): 5
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

### 3.7.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 4) + (EO \times 5) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (1 \times 5) + (5 \times 4) + (5 \times 5) + (3 \times 3) = 7 + 5 + 20 + 25 + 9 = 66$$

### 3.7.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 3
- Продуктивність: 4
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 3
- Інтенсивність взаємодії з користувачем: 2
- Ергономіка: 2
- Інтенсивність зміни даних: 2
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 3
- Гнучкість: 3

### 3.7.5 Сума значень GSC (TDI)

$$TDI = 3 + 3 + 4 + 2 + 3 + 2 + 2 + 2 + 3 + 2 + 2 + 2 + 3 + 3 = 36$$

### 3.7.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 36) + 0.65 = 1.01$$

### 3.7.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 66 \times 1.01 \approx 66.66 \approx 67$$

### 3.7.8 Висновок

- Нескориговані функціональні точки (UFP): 66
- Скориговані функціональні точки (AFP): 67

## 3.8 Аналіз функціональних точок System Programming

Репозиторій [SideShowBoBGOT/SystemProgrammingFourthSemester](https://github.com/SideShowBoBGOT/SystemProgrammingFourthSemester)

містить матеріали з системного програмування, включаючи лабораторні роботи та приклади коду, написані мовами C та Assembly.

### 3.8.1 Функції даних

Внутрішні логічні файли (ILF):

- Код містить структури даних для обробки системних викликів та керування пам'яттю.
- Кількість елементів даних (DET): приблизно 20
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з зовнішніми бібліотеками та системними файлами.
- Кількість елементів даних (DET): приблизно 15
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 5

### 3.8.2 Транзакційні функції

Зовнішні введення (EI):

- Функції для прийому даних від користувача або системи.
  - Кількість EI: 4
  - Кількість елементів даних (DET): 6
  - Кількість типів файлів, що використовуються (FTR): 2
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 4
- Зовнішні виведення (EO):
- Функції для виведення результатів обробки.
  - Кількість EO: 4
  - Кількість елементів даних (DET): 6
  - Кількість типів файлів, що використовуються (FTR): 2
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 5
- Зовнішні запити (EQ):
- Функції для отримання системної інформації.
  - Кількість EQ: 3
  - Кількість елементів даних (DET): 5
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 3

### 3.8.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 4) + (EO \times 5) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (1 \times 5) + (4 \times 4) + (4 \times 5) + (3 \times 3) = 7 + 5 + 16 + 20 + 9 = 57$$

### 3.8.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 2
- Розподілена обробка даних: 2
- Продуктивність: 3
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 2

- Інтенсивність взаємодії з користувачем: 2
- Ергономіка: 2
- Інтенсивність зміни даних: 2
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 2
- Гнучкість: 2

### 3.8.5 Сума значень GSC (TDI)

$$TDI = 2 + 2 + 3 + 2 + 2 + 2 + 2 + 2 + 3 + 2 + 2 + 2 + 2 + 2 = 32$$

### 3.8.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 32) + 0.65 = 0.97$$

### 3.8.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 57 \times 0.97 \approx 55.29 \approx 55$$

### 3.8.8 Висновок

- Нескориговані функціональні точки (UFP): 57
- Скориговані функціональні точки (AFP): 55

## 3.9 Аналіз функціональних точок Petri Object Model Paint

Репозиторій [SideShowBoBGOT/PetriObjModelPaint](https://github.com/SideShowBoBGOT/PetriObjModelPaint) реалізує техніку моделювання об'єктів Петрі, що дозволяє швидко та гнучко створювати моделі складних дискретних систем подій.

### 3.9.1 Функції даних

Внутрішні логічні файли (ILF):

- Класи, що представляють мережі Петрі та їх компоненти.
  - Кількість елементів даних (DET): приблизно 25
  - Кількість типів записів (RET): 2
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 10
- Зовнішні інтерфейсні файли (EIF):
- Взаємодія з бібліотеками для графічного відображення моделей.
  - Кількість елементів даних (DET): приблизно 15
  - Кількість типів записів (RET): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 5

### 3.9.2 Транзакційні функції

Зовнішні введення (EI):

- Функції для створення та редагування елементів моделі.
- Кількість EI: 6
- Кількість елементів даних (DET): 10
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

Зовнішні виведення (EO):

- Функції для відображення результатів моделювання.
- Кількість EO: 5
- Кількість елементів даних (DET): 12
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Середній
- Ваговий коефіцієнт: 5

Зовнішні запити (EQ):

- Функції для отримання інформації про стан моделі.
- Кількість EQ: 4

- Кількість елементів даних (DET): 8
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

### 3.9.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 10) + (EIF \times 5) + (EI \times 4) + (EO \times 5) + (EQ \times 4)$$

$$UFP = (1 \times 10) + (1 \times 5) + (6 \times 4) + (5 \times 5) + (4 \times 4) = 10 + 5 + 24 + 25 + 16 = 80$$

### 3.9.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 3
- Продуктивність: 4
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 3
- Інтенсивність взаємодії з користувачем: 3
- Ергономіка: 3
- Інтенсивність зміни даних: 2
- Складність обробки: 4
- Повторне використання: 3
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 3
- Гнучкість: 3

### 3.9.5 Сума значень GSC (TDI)

$$TDI = 3 + 3 + 4 + 2 + 3 + 3 + 3 + 2 + 4 + 3 + 2 + 2 + 3 + 3 = 38$$

### 3.9.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 38) + 0.65 = 1.03$$

### 3.9.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 80 \times 1.03 \approx 82.4 \approx 82$$

### 3.9.8 Висновок

- Нескориговані функціональні точки (UFP): 80
- Скориговані функціональні точки (AFP): 82

## 3.10 Аналіз функціональних точок Console Chess

Репозиторій [SideShowBoBGOT/ConsoleChess](https://github.com/SideShowBoBGOT/ConsoleChess) містить консольну реалізацію гри в шахи, написану мовою C++.

### 3.10.1 Функції даних

Внутрішні логічні файли (ILF):

- Класи, що представляють шахову дошку та фігури.
- Кількість елементів даних (DET): приблизно 20
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з бібліотеками для обробки введення користувача.
- Кількість елементів даних (DET): приблизно 10
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 5

### 3.10.2 Транзакційні функції

Зовнішні введення (EI):

- Функції для введення ходів гравців.
- Кількість EI: 4



- Кількість елементів даних (DET): 6
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 3
- Зовнішні виведення (EO):
- Функції для відображення шахової дошки та результатів гри.
  - Кількість EO: 4
  - Кількість елементів даних (DET): 8
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 4
- Зовнішні запити (EQ):
- Функції для перевірки стану гри.
  - Кількість EQ: 3
  - Кількість елементів даних (DET): 5
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 3

### 3.10.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 3) + (EO \times 4) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (1 \times 5) + (4 \times 3) + (4 \times 4) + (3 \times 3) = 7 + 5 + 12 + 16 + 9 = 49$$

### 3.10.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 2
- Розподілена обробка даних: 1
- Продуктивність: 3
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 3

- Ергономіка: 2
- Інтенсивність зміни даних: 2
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 2
- Гнучкість: 2

### 3.10.5 Сума значень GSC (TDI)

$$TDI = 2 + 1 + 3 + 1 + 2 + 3 + 2 + 2 + 3 + 2 + 2 + 2 + 2 + 2 = 31$$

### 3.10.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 31) + 0.65 = 0.96$$

### 3.10.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 49 \times 0.96 \approx 47.04 \approx 47$$

### 3.10.8 Висновок

- Нескориговані функціональні точки (UFP): 49
- Скориговані функціональні точки (AFP): 47

## 3.11 Аналіз функціональних точок CPong

Репозиторій [SideShowBoBGOT/CPong](https://github.com/SideShowBoBGOT/CPong) містить консольну реалізацію гри Pong, написану мовою C.

### 3.11.1 Функції даних

Внутрішні логічні файли (ILF):

- Класи або структури, що представляють ігрові об'єкти, такі як ракетки та м'яч.

- Кількість елементів даних (DET): приблизно 10
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Взаємодія з бібліотеками для обробки введення користувача та відображення графіки.
- Кількість елементів даних (DET): приблизно 8
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 5

### 3.11.2 Транзакційні функції

Зовнішні введення (EI):

- Функції для введення команд гравця (рух ракетки вгору/вниз).
- Кількість EI: 2
- Кількість елементів даних (DET): 5
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

Зовнішні виведення (EO):

- Функції для відображення ігрового стану (позиції ракеток, м'яча, рахунок).
- Кількість EO: 3
- Кількість елементів даних (DET): 7
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 4

Зовнішні запити (EQ):

- Функції для отримання інформації про поточний стан гри.

- Кількість EQ: 2
- Кількість елементів даних (DET): 4
- Кількість типів файлів, що використовуються (FTR): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 3

### 3.11.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 3) + (EO \times 4) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (1 \times 5) + (2 \times 3) + (3 \times 4) + (2 \times 3) = 7 + 5 + 6 + 12 + 6 = 36$$

### 3.11.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 1
- Розподілена обробка даних: 1
- Продуктивність: 2
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 2
- Ергономіка: 2
- Інтенсивність зміни даних: 1
- Складність обробки: 2
- Повторне використання: 1
- Зручність інсталяції: 2
- Зручність адміністрування: 1
- Можливість портування: 2
- Гнучкість: 2

### 3.11.5 Сума значень GSC (TDI)

$$TDI = 1 + 1 + 2 + 1 + 2 + 2 + 2 + 1 + 2 + 1 + 2 + 1 + 2 + 2 = 22$$

### 3.11.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 22) + 0.65 = 0.87$$

### 3.11.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 36 \times 0.87 \approx 31.32 \approx 31$$

### 3.11.8 Висновок

- Нескориговані функціональні точки (UFP): 36
- Скориговані функціональні точки (AFP): 31

## 3.12 Аналіз функціональних точок Personal Website

Репозиторій [SideShowBoBGOT/SideShowBoBGOT.github.io](https://SideShowBoBGOT/SideShowBoBGOT.github.io) містить персональний вебсайт, створений за допомогою HTML, CSS та JavaScript.

### 3.12.1 Функції даних

Внутрішні логічні файли (ILF):

- Статичні сторінки вебсайту, що містять контент.
- Кількість елементів даних (DET): приблизно 30
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 7

Зовнішні інтерфейсні файли (EIF):

- Підключені зовнішні бібліотеки або API.
- Кількість елементів даних (DET): приблизно 15
- Кількість типів записів (RET): 1
- Рівень складності: Низький
- Ваговий коефіцієнт: 5

### 3.12.2 Транзакційні функції

Зовнішні введення (EI):

- Форми зворотного зв'язку або інші інтерактивні елементи.
- Кількість EI: 2

- Кількість елементів даних (DET): 5
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 3
- Зовнішні виведення (EO):
- Відображення динамічного контенту або повідомлень.
  - Кількість EO: 3
  - Кількість елементів даних (DET): 7
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 4
- Зовнішні запити (EQ):
- Запити для отримання інформації з серверу.
  - Кількість EQ: 2
  - Кількість елементів даних (DET): 6
  - Кількість типів файлів, що використовуються (FTR): 1
  - Рівень складності: Низький
  - Ваговий коефіцієнт: 3

### 3.12.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 7) + (EIF \times 5) + (EI \times 3) + (EO \times 4) + (EQ \times 3)$$

$$UFP = (1 \times 7) + (1 \times 5) + (2 \times 3) + (3 \times 4) + (2 \times 3) = 7 + 5 + 6 + 12 + 6 = 36$$

### 3.12.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 2
- Розподілена обробка даних: 1
- Продуктивність: 2
- Обмеження апаратних ресурсів: 1
- Транзакційна навантаженість: 2
- Інтенсивність взаємодії з користувачем: 3

- Ергономіка: 3
- Інтенсивність зміни даних: 2
- Складність обробки: 2
- Повторне використання: 1
- Зручність інсталяції: 2
- Зручність адміністрування: 2
- Можливість портування: 2
- Гнучкість: 2

### 3.12.5 Сума значень GSC (TDI)

$$TDI = 2 + 1 + 2 + 1 + 2 + 3 + 3 + 2 + 2 + 1 + 2 + 2 + 2 + 2 = 29$$

### 3.12.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 29) + 0.65 = 0.94$$

### 3.12.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 36 \times 0.94 \approx 33.84 \approx 34$$

### 3.12.8 Висновок

- Нескориговані функціональні точки (UFP): 36
- Скориговані функціональні точки (AFP): 34

## 3.13 Аналіз функціональних точок Software Security Course

Репозиторій [SideShowBoBGOT/software\\_security\\_seventh\\_semester](https://github.com/SideShowBoBGOT/software_security_seventh_semester) містить матеріали, пов'язані з курсом "Безпека програмного забезпечення" за сьомий семестр.

### 3.13.1 Функції даних

Внутрішні логічні файли (ILF):

- Матеріали лекцій, лабораторні роботи та інші навчальні ресурси.

- Кількість елементів даних (DET): приблизно 50
  - Кількість типів записів (RET): 5
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 10
- Зовнішні інтерфейсні файли (EIF):
- Посилання на зовнішні ресурси або бібліотеки, що використовуються в курсі.
  - Кількість елементів даних (DET): приблизно 20
  - Кількість типів записів (RET): 2
  - Рівень складності: Середній
  - Ваговий коефіцієнт: 7

### 3.13.2 Транзакційні функції

Зовнішні введення (EI):

- Форми для подання завдань або зворотного зв'язку.
- Кількість EI: 3
- Кількість елементів даних (DET): 10
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

Зовнішні виведення (EO):

- Генерація звітів про успішність або результати тестів.
- Кількість EO: 4
- Кількість елементів даних (DET): 15
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Середній
- Ваговий коефіцієнт: 5

Зовнішні запити (EQ):

- Запити для отримання інформації про розклад занять або дедлайни.
- Кількість EQ: 2



- Кількість елементів даних (DET): 8
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

### 3.13.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 10) + (EIF \times 7) + (EI \times 4) + (EO \times 5) + (EQ \times 4)$$

$$UFP = (1 \times 10) + (1 \times 7) + (3 \times 4) + (4 \times 5) + (2 \times 4) = 10 + 7 + 12 + 20 + 8 =$$

57

### 3.13.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 2
- Продуктивність: 3
- Обмеження апаратних ресурсів: 2
- Транзакційна навантаженість: 3
- Інтенсивність взаємодії з користувачем: 4
- Ергономіка: 4
- Інтенсивність зміни даних: 3
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 3
- Зручність адміністрування: 3
- Можливість портування: 3
- Гнучкість: 3

### 3.13.5 Сума значень GSC (TDI)

$$TDI = 3 + 2 + 3 + 2 + 3 + 4 + 4 + 3 + 3 + 2 + 3 + 3 + 3 + 3 = 39$$

### 3.13.6 Розрахунок фактору коригування значень (VAF)

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 39) + 0.65 = 1.04$$

### 3.13.7 Розрахунок скоригованих функціональних точок (AFP)

$$AFP = UFP \times VAF$$

$$AFP = 57 \times 1.04 \approx 59.28 \approx 59$$

### 3.13.8 Висновок

- Нескориговані функціональні точки (UFP): 57
- Скориговані функціональні точки (AFP): 59

## 3.14 Аналіз функціональних точок Multithreaded Programming Course

Репозиторій [SideShowBoBGOT/MultithreadedProgrammingSixthSemester](https://github.com/SideShowBoBGOT/MultithreadedProgrammingSixthSemester) містить матеріали курсу з багатопотокового програмування, включаючи лабораторні роботи, курсові проєкти та лекції.

### 3.14.1 Функції даних

Внутрішні логічні файли (ILF):

- Лабораторні роботи та курсові проєкти, що містять код та супровідну документацію.
- Кількість елементів даних (DET): приблизно 50
- Кількість типів записів (RET): 5
- Рівень складності: Середній
- Ваговий коефіцієнт: 10

Зовнішні інтерфейсні файли (EIF):

- Зовнішні бібліотеки або API, що використовуються в проєктах.
- Кількість елементів даних (DET): приблизно 20
- Кількість типів записів (RET): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 7

### 3.14.2 Транзакційні функції

Зовнішні введення (EI):

- Форми введення даних для налаштування параметрів програм.
- Кількість EI: 3
- Кількість елементів даних (DET): 10
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4
- Зовнішні виведення (EO):
- Генерація звітів про результати виконання програм.
- Кількість EO: 4
- Кількість елементів даних (DET): 15
- Кількість типів файлів, що використовуються (FTR): 3
- Рівень складності: Середній
- Ваговий коефіцієнт: 5
- Зовнішні запити (EQ):
- Запити для отримання інформації про стан виконання програм.
- Кількість EQ: 2
- Кількість елементів даних (DET): 8
- Кількість типів файлів, що використовуються (FTR): 2
- Рівень складності: Середній
- Ваговий коефіцієнт: 4

### 3.14.3 Розрахунок нескоригованих функціональних точок (UFP)

$$UFP = (ILF \times 10) + (EIF \times 7) + (EI \times 4) + (EO \times 5) + (EQ \times 4)$$

$$UFP = (1 \times 10) + (1 \times 7) + (3 \times 4) + (4 \times 5) + (2 \times 4) = 10 + 7 + 12 + 20 + 8 =$$

57

### 3.14.4 Оцінка загальних характеристик системи (GSC)

- Обмін даними: 3
- Розподілена обробка даних: 2
- Продуктивність: 3
- Обмеження апаратних ресурсів: 2

- Транзакційна навантаженість: 3
- Інтенсивність взаємодії з користувачем: 4
- Ергономіка: 4
- Інтенсивність зміни даних: 3
- Складність обробки: 3
- Повторне використання: 2
- Зручність інсталяції: 3
- Зручність адміністрування: 3
- Можливість портування: 3
- Гнучкість: 3

#### **3.14.5 Сума значень GSC (TDI)**

$$TDI = 3 + 2 + 3 + 2 + 3 + 4 + 4 + 3 + 3 + 2 + 3 + 3 + 3 + 3 = 39$$

#### **3.14.6 Розрахунок фактору коригування значень (VAF)**

$$VAF = (0.01 \times TDI) + 0.65$$

$$VAF = (0.01 \times 39) + 0.65 = 1.04$$

#### **3.14.7 Розрахунок скоригованих функціональних точок (AFP)**

$$AFP = UFP \times VAF$$

$$AFP = 57 \times 1.04 \approx 59.28 \approx 59$$

#### **3.14.8 Висновок**

- Нескориговані функціональні точки (UFP): 57
- Скориговані функціональні точки (AFP): 59

### 3.15 Оцінка проекту

Побудуємо таблицю 3.1 проектів.

Таблиця 3.1 Прокети

Назва проекту	Нескоригована кількість функціональних точок	Скоригована кількість функціональних точок	Мова програмуван ня	Тисячі рядків коду	Зусилл я
RwLock	26	22	0	0.215	0.48
NBUCurrencyTracker	43	43	0	1.135	2.74
CppFuse	80	87	0	1.775	4.38
mcts_checkers	31	31	0	2.803	7.08
FlaskProject	50	49	2	3.241	8.25
FunctionalProgrammingHaskell	51	45	3	2.632	6.63
network_programming_eighth_se mester	66	67	0	1.359	3.31

<b>Назва проекту</b>	<b>Нескоригована кількість функціональних точок</b>	<b>Скоригована кількість функціональних точок</b>	<b>Мова програмуван ня</b>	<b>Тисячі рядків коду</b>	<b>Зусилл я</b>
SystemProgrammingFourthSemester	57	55	1	4.652	12.06
PetriObjModelPaint	80	82	4	14.292	39.18
ConsoleChess	49	47	0	1.107	2.67
CPong	36	31	0	0.245	0.55
SideShowBoBGOT.github.io	36	34	5	6.395	16.84
software_security_seventh_semester	57	59	5	1.784	4.41
MultithreadedProgrammingSixthSemester	57	59	4	10.099	27.21

Візьмемо проект CPPFuse і порахуємо відстані відносно нього у таблиці 3.2.

Таблиця 3.2 Дистанції

Назва проекту	Дистанція
mcts_checkers	0.0
CPong	8.61
RwLock	12.5
SideShowBoBGOT.github.io	12.93
NBUCurrencyTracker	17.6
ConsoleChess	24.54
FunctionalProgrammingHaskell	24.6
FlaskProject	26.28
SystemProgrammingFourthSemester	35.79
software_security_seventh_semester	38.64
MultithreadedProgrammingSixthSemester	43.98
network_programming_eighth_semester	50.37
CppFuse	74.47
PetriObjModelPaint	78.62

## ВИСНОВОК

Аналіз проєктів показує значні варіації в складності, функціональності та обсязі виконаної роботи. Проєкт **mcts\_checkers** служить базовою точкою порівняння. Він має 31 нескориговану функціональну точку, 31 скориговану функціональну точку, обсяг коду в 2.803 тисячі рядків та зусилля в 7.08 людино-місяців. Його характеристики дозволяють оцінювати інші проєкти за близькістю до цієї точки, що показує їхню схожість або відмінність.

Найближчий до **mcts\_checkers** проєкт — **CPong**. Його дистанція становить лише 8.61, що свідчить про відносно схожу функціональність, але значно менший обсяг коду (0.245 тисячі рядків) та мінімальні зусилля (0.55 людино-місяців). Це вказує на те, що CPong є спрощеним проєктом, реалізованим з меншими ресурсами, але зберігає деяку подібність у функціональних точках.

На більшій дистанції, 12.5, знаходиться **RwLock**, який також має менший обсяг функціональних точок (22 скоригованих), але відрізняється спрощеною реалізацією з використанням лише 0.215 тисячі рядків коду. Незважаючи на меншу складність, цей проєкт демонструє зменшені зусилля (0.48 людино-місяців), що робить його цікавим прикладом оптимізації.

**SideShowBoBGOT.github.io** і **NBUCurrencyTracker** мають дистанції 12.93 і 17.6 відповідно. Перший виділяється великою кількістю коду (6.395 тисячі рядків), тоді як другий демонструє помірний обсяг коду (1.135 тисячі рядків) і витрачених зусиль (2.74 людино-місяці). Вони показують, як масштабність веб-проєктів або API інтеграцій впливає на загальний обсяг роботи.

Віддаленіші проєкти, такі як **SystemProgrammingFourthSemester** (35.79) і **MultithreadedProgrammingSixthSemester** (43.98), мають значно більший обсяг коду (понад 10 тисяч рядків у другому) та високі зусилля (12.06 і 27.21 людино-місяців відповідно). Ці проєкти демонструють складність системних і багатопотокових завдань, які потребують більше ресурсів.



Найбільш віддаленим від **mcts\_checkers** є проєкт **PetriObjModelPaint** з дистанцією 78.62. Цей проєкт має найбільшу кількість скоригованих функціональних точок (82) та найбільший обсяг коду (14.292 тисячі рядків), що пояснює високі витрати зусиль (39.18 людино-місяців). Це свідчить про його комплексність і значну функціональність, яка перевищує всі інші аналізовані проєкти.

Таким чином, **mcts\_checkers** займає проміжну позицію серед проєктів, надаючи орієнтир для порівняння. Віддалені проєкти демонструють, як зростання функціональних точок, обсягу коду та зусиль ускладнює реалізацію, але водночас підвищує загальну функціональність і масштабність.