МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Кафедра «Програмна інженерія та інформаційні технології управління»

РЕФЕРАТ

з дисципліни «Економіка організації виробництва»

на тему: «Набір метрик Чидамбера і Кемерера»

Виконав:

ст. гр. КН-416A

Антоненко В.В.

Перевірила:

Єршова С.І.

Харків 2020

**ЗМІСТ**

**ВСТУП3**

1. **Набір метрик Чидамбера і Кемерера 6**
   1. Метрика 1: Зважені методи на клас WMC 6
   2. Метрика 2: Висота дерева спадкування DIT 7
   3. Метрика 3: Кількість дітей NOC 8
   4. Метрика 4: Зчеплення між класами об'єктів СВО 9
   5. Метрика 5: Відгук для класу RFC 10
   6. Метрика 6: Недолік зв'язності в методах LСOM11
2. **Використання метрик Чидамбера-Кемерера  15**
3. **Висновок16**
4. **Перелік літератури17**

ВСТУП

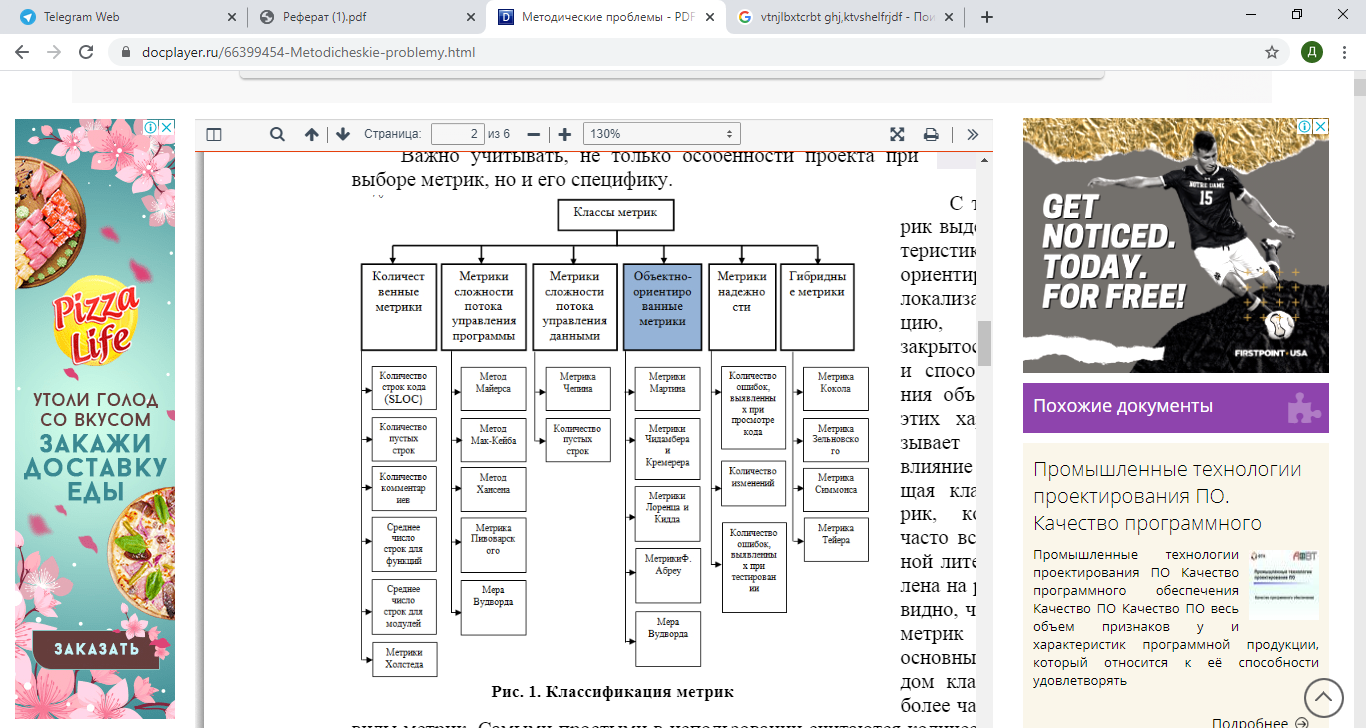
На сьогоднішній день ситуація в світі інформаційних технологій складається таким чином, що більшість розробляються об'єктно великих проектів на орієнтованих мовах програмування. Забезпечення якості проекту є головним завданням, як керівників проекту, так і самих розробників. Для ефективної оцінки програмного забезпечення використовуються метрики.

Визначення метрики в контексті інформаційних технологій, відповідно до стандарту ISO 14598 звучить як кількісний масштаб і метод, який може використовуватися для вимірювання. Інше визначення метрики - захід, що дозволяє отримати чисельне значення деякої властивості програмного забезпечення[1].  Об'єктно-орієнтовані метрики вводяться з метою: поліпшити розуміння якості продукту оцінити ефективність процесу конструювання, поліпшити якість роботи на етапі проектування.

Для будь-якої інформаційної системи визначаються набори метрик повинні орієнтуватися на особливості і унікальні характеристики. Більшість метрик, були отримані досвідченим пут м в ході роботи над проектами інформаційних систем.

Важливо враховувати, не тільки особливості проекту при виборі метрик, але і його специфіку.

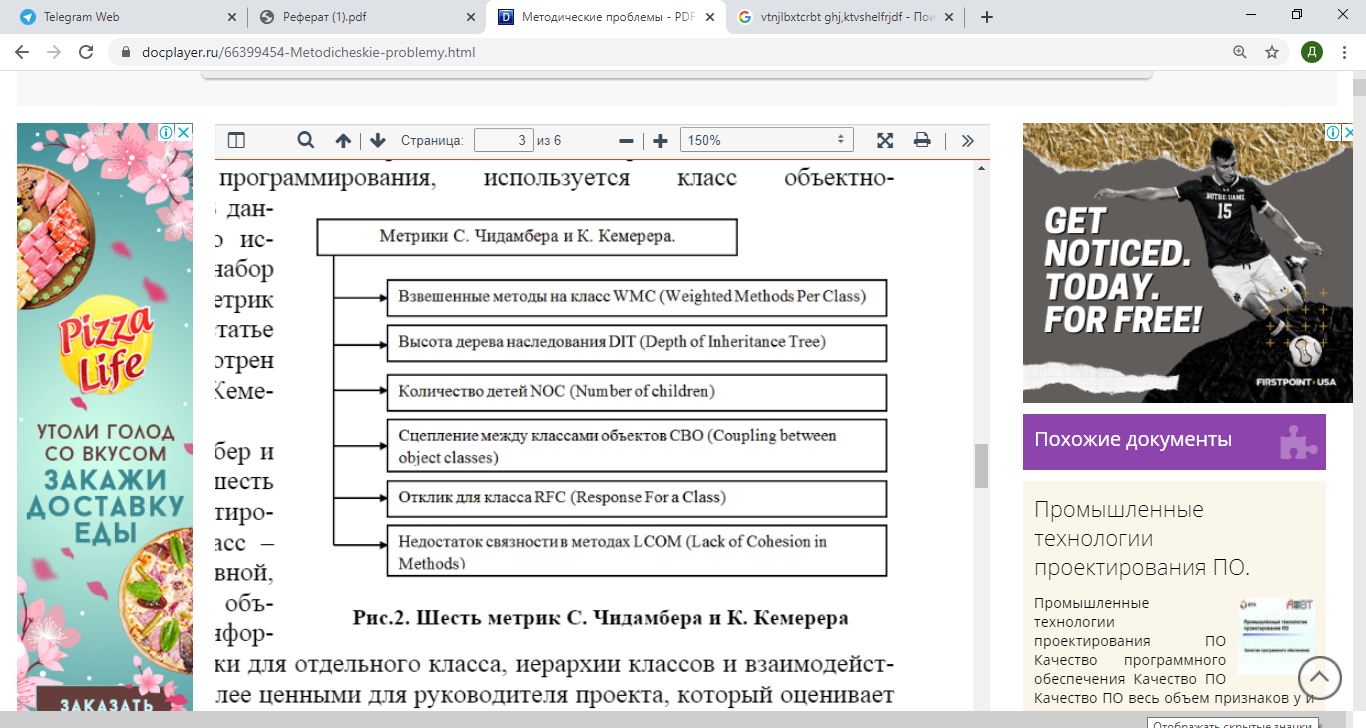
З точки зору метрик виділяють п'ять характеристик об'єктно-орієнтованих систем: локалізацію, інкапсуляцію , інформаційну закритість , успадкування та способи абстрагування об'етов.Каждая з цих характеристик надає безпосереднє вплив на метрики. Загальна класифікація мет рик, яка найбільш часто зустрічається в науковій літературі, представлена на рис. 1. З малюнка видно, що класифікація метрик складається з семи основних класів. У кожному класі вказані найбільш часто використовуваних . Сам ими простими у використанні вважаються кількісні метрики. В даний клас входять такі метрики як: кількість порожніх рядків, кількість кому м ентаріев; відсоток кому м ентаріев (відношення числа рядків, утримуючи щ їх коментарі до про щ йому кількості рядків, виражене у відсотках); середнє число рядків для функцій (класів, файлів); середнє число рядків, утримуючи щ їх вихідний код для функцій (класів, файлів), середнє число рядків для модулів.



Мал. 1.Классифікація метрик

На схемі, представленій на мал. 1 бачимо, що кожен клас має набір метрик. Метрик, що входять у визначений клас , існує велика кількість, тому в схемі відображені найбільш часто зустрічаються набори метрик, проте далеко не всі.

Для аналізу і якісної оцінки проектів, створених за допомогою мов об 'ектно-орієнтованого програмування, використовується клас  об'єктно-орієнтованих метрик. В даному класі найбільш використовувані є набір метрик Мартіна, набір метрик  Чидамбера і Кемерера. В рефераті докладно буде розглянуто набір метрик Чидамбера і Кемерера [2] .



Мал. 2. Шість метрик Чидамбера і Кемерера

1. **Набір метрик Чидамбера і Кемерера**

У 1994 році С. Чидамбер і К. Кемерер ( Chidamber і Кетегег ) запропонували шість проектних метрик, орієнтованих на класи [ 1 ]. Клас - фундаментальний елемент об'єктно-орієнтованої (ОО) системи. Тому вимірювання і метрики для окремого класу, ієрархії класів і співпраці класів безцінні для програмного інженера, який повинен оцінити якість проекту.

Набір Чидамбера-Кемерера найбільш часто цитується в програмній індустрії і наукових дослідженнях. Розглянемо кожну з метрик набору.

**1.1. Метрика 1: Зважені методи на клас WMC ( Weighted Methods Per Class )**

Припустимо, що в класі *С*визначені *п*методів зі складністю Для оцінки складності може бути обрана будь-яка метрика складності (наприклад, цикломатическая складність). Головне - нормалізувати цю метрику так, щоб номінальна складність для методу приймала значення 1. У цьому випадку  



Кількість методів і їх складність є індикатором витрат на реалізацію і тестування класів. Крім того, чим більше методів, тим складніше дерево спадкування (всі підкласи успадковують методи їх батьків). З ростом кількості методів в класі його застосування стає все більш специфічним, тим самим обмежується можливість багаторазового використання. З цих причин метрика WMC повинна мати розумно низьке значення.

Дуже часто застосовують спрощену версію метрики. При цьому вважають і тоді WMC-кількість методів в класі.

Виявляється, що підраховувати кількість методів в класі досить складно. Можливі два протилежні варіанти обліку.

1. Підраховуються тільки методи поточного класу. Успадковані методи ігноруються. Обгрунтування - успадковані методи вже підраховані в тих класах, де вони визначалися. Таким чином, інкрементного класу - кращий показник його функціональних можливостей, який відображає його право на існування. Найбільш важливим джерелом інформації для розуміння того, що робить клас, є його власні операції. Якщо клас не може відреагувати на повідомлення (наприклад, в ньому відсутній власний метод), тоді він пошле повідомлення батькові.

2. Підраховуються методи, визначені в поточному класі, і все успадковані методи. Цей підхід підкреслює важливість простору станів в розумінні класу (а не інкрементного класу).

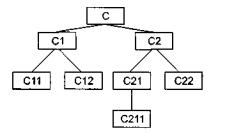
Існує ряд проміжних варіантів. Наприклад, підраховуються поточні методи і методи, прямо успадковані від батьків. Аргумент на користь даного підходу - на поведінку дочірнього класу найбільш сильно впливає спеціалізація батьківських класів.

На практиці прийнятний будь-який з описаних варіантів. Головне - не міняти варіант обліку від проекту до проекту [1] . Тільки в цьому випадку забезпечується коректний збір метричних даних.

Метрика WMC дає відносну міру складності класу. Якщо вважати, що всі методи мають однакову складність, то це буде просто кількість методів в класі. Існують рекомендації по складності методів. Наприклад, М. Лоренц вважає, що середня довжина методу повинна обмежуватися 8 рядками для Smalltalk і 24 рядками для C ++ [ 2 ]. Взагалі, клас, який має максимальну кількість методів серед класів одного з них рівня, є найбільш складним; швидше за все, він специфічний для цього додатка і містить найбільшу кількість помилок.

**1.2. Метрика 2: Висота дерева спадкування DIT ( Depth of Inheritance Tree )**

*DIT*визначається як максимальна довжина шляху від листа до кореня дерева успадкування класів. Для показаної на мал. 3 ієрархії класів метрика DIT дорівнює 3.



Мал. 3 . Дерево успадкування класів

Відповідно, для окремого класу DIT, це довжина максимального шляху від даного класу до кореневого класу в ієрархії класів.

У міру зростання DIT ймовірно, що класи нижнього рівня будуть наслідувати багато методів. Це призводить до труднощів в прогнозі поведінки класу. Висока ієрархія класів (велике значення DIT) призводить до більшої складності проекту, так як означає залучення більшої кількості методів і класів.

Разом з тим, велике значення DIT має на увазі, що багато методів можуть використовуватися багаторазово.

**1.3. Метрика 3: Кількість дітей NOC**

**( Number of children )**

Підкласи, які безпосередньо підпорядковані суперкласу, називаються його дітьми. Значення NOC дорівнює кількості дітей, тобто кількості безпосередніх спадкоємців класу в ієрархії класів. На рис. 3 клас *С*2 має двох дітей - підкласи  *С*21 і *С*22.

З збільшенням NOC зростає багаторазовість використання, так як спадкування - це форма повторного використання.

Однак при зростанні NOC послаблюється абстракція батьківського класу. Це означає, що в дійсності деякі з дітей вже не є членами батьківського класу і можуть бути неправильно використані.

Крім того, кількість дітей характеризує потенційний вплив класу на проект. У міру зростання NOC зростає кількість тестів, необхідних для перевірки кожної дитини.

Метрики DIT і NOC - кількісні характеристики форми і розміру структури класів. Добре структурована об'єктно-орієнтована система частіше буває організована як ліс класів, ніж як надвисока дерево. На думку Г. Буча, слід будувати збалансовані по висоті і ширині структури успадкування: зазвичай не вище, ніж 7 ± 2 рівня, і не ширше, ніж 7 + 2 гілки [ 3 ].

**1.4. Метрика 4: Зчеплення між класами об'єктів СВО ( Coupling between object classes )**

*СВО -*це кількість співробітництв, передбачених для класу, тобто кількість класів, з якими він з'єднаний. З'єднання означає, що методи даного класу використовують методи або екземплярні змінні іншого класу.

Інше визначення метрики має наступний вигляд: СВО дорівнює кількості зчеплень класу; зчеплення утворює виклик методу або властивості в іншому класі.

Дана метрика характеризує статичну складову зовнішніх зв'язків класів.

З ростом СВО багаторазовість використання класу, ймовірно, зменшується. Очевидно, що чим більша незалежність класу, тим легше його повторно використовувати в інших програмах.

Високе значення СВО ускладнює модифікацію і тестування, яке слід за виконанням модифікації. Зрозуміло, що, чим більше кількість зчеплень, тим вище чутливість всього проекту до змін в окремих його частинах. Мінімізація межоб'ектних зчеплень покращує модульність і сприяє інкапсуляції проекту.

*СВО*для кожного класу повинно мати розумно низьке значення. Це узгоджується з рекомендаціями щодо зменшення зчеплення стандартного програмного забезпечення.

**1.5. Метрика 5: Відгук для класу RFC**

**( Response For a Class )**

Введемо допоміжний визначення. Безліч відгуку класу RS - це безліч методів, які можуть виконуватися у відповідь на прибуття повідомлень в об'єкт цього класу. Формула для визначення RS має вигляд

**

де *—*безліч методів, що викликаються методом г, *{М} -*безліч всіх методів в класі.  

Метрика RFC дорівнює кількості методів у безлічі відгуку, тобто дорівнює потужності цього безлічі:



Наведемо інше визначення метрики: RFC - це кількість методів класу плюс кількість методів інших класів, що викликаються з даного класу.

Метрика RFC є мірою потенційної взаємодії даного класу з іншими класами, дозволяє судити про динаміку поведінки відповідного об'єкта в системі. Дана метрика характеризує динамічну складову зовнішніх зв'язків класів.

Якщо у відповідь на повідомлення може бути викликано велику кількість методів, то ускладнюються тестування і налагодження класу, так як від розробника тестів потрібен більший рівень розуміння класу, зростає довжина тестової послідовності.

З ростом RFC збільшується складність класу. Найгірша величина відгуку може використовуватися при визначенні часу тестування.

**1.6. Метрика 6: Недолік зв'язності в методах LСOM ( Lack of Cohesion in Methods )**

Кожен метод всередині класу звертається до одного або декількох властивостей ( екземплярність змінним). Метрика *LCOM*показує, наскільки методи не пов'язані один з одним через властивості (змінні). Якщо все методи звертаються до однакових властивостях, то LCOM = 0. 

Введемо позначення:

НЕ ПОВ'ЯЗАНІ - кількість пар методів без загальних  екземплярність змінних;

ПОВ'ЯЗАНІ - кількість пар методів із загальними екземплярність змінними.

- набір екземплярність змінних, використовуваних методом

Бачимо, що





Тоді формула для обчислення нестачі зв'язності в методах набуде вигляду



Можна визначити метрику по-іншому: LCOM - це кількість пар методів, не пов'язаних з властивостями класу, мінус кількість пар методів, що мають такий зв'язок.

Розглянемо приклади застосування метрики LCOM.

**Приклад 1:**У класі є методи: *M1, M2, М3, М4.*Кожен метод працює зі своїм набором екземплярність змінних: 

**

В цьому випадку







**Приклад 2:**В класі використовуються методи: *M1, M2, М3.*Для кожного методу заданий свій набір екземплярність змінних: 

**







Можливості підключення методів всередині класу повинна бути високою, так як це сприяє інкапсуляції. Якщо LCOM має високе значення, то методи слабо пов'язані один з одним через властивості. Це збільшує складність, в зв'язку з чим зростає ймовірність помилок при проектуванні.

Високі значення LCOM означають, що клас, ймовірно, треба спроектувати краще (розбиттям на два або більше окремих класу). Будь-яке обчислення LCOM допомагає визначити недоліки в проектуванні класів, так як ця метрика характеризує якість упаковки даних і методів в оболонку класу.

Висновок: зв'язність в класі бажано зберігати високу, тобто слід домагатися найнижчого значення LCOM.

Набір метрик Чідамбера-Кемерера - одна з піонерських робіт по комплексній оцінці якості ОО-проектування. Відомі численні пропозиції щодо вдосконалення, розвитку даного набору. Розглянемо деякі з них.

Недоліком метрики WMC є залежність від реалізації. Наведемо приклад.

Розглянемо клас, що пропонує операцію інтегрування. Можливі дві реалізації:

1) кілька простих операцій:

Set\_interval ( min . Max ) Setjnethod ( method ) Set\_precision ( precision )

Set\_function\_to\_integrate ( function ) Integrate ;

2) одна складна операція:

Integrate ( function , min , max . Method , precision )

Для забезпечення незалежності від цих реалізацій можна визначити метрику WMC2:



Для нашого прикладу WMC2 = 5 і для першої, і для другої реалізації. Зауважимо, для першої реалізації WMC = 5, а для другої реалізації WMC = 1.

Додатково можна визначити метрику *Середнє число аргументів методу ANAM*( Average Number of Arguments per Method ): 

ANAM = WMC2 / WMC.

Корисність метрики ANAM пояснити ще легше. Вона орієнтована на прийняті в проектуванні рішення - застосовувати прості операції з малою кількістю аргументів, а нескладні операції - з численними аргументами.

Ще одна пропозиція - ввести метрику, симетричну метриці LCOM. У той час як формула метрики LCOM має вигляд:



симетрична їй метрика *Нормалізована NLСОМ*обчислюється за формулою:



Діапазон значень цієї метрики: 0 ≤ NLCOM ≤ 1, причому чим ближче NLCOM до 1, тим вище зв'язаність класу.

У наборі Чідамбера-Кемерера відсутня метрика для прямого виміру інформаційної закритості класу. В силу цього була запропонована метрика *Поведінкова закритість інформації BIH*( Behaviourial Information Hiding ): 

**

*де WEOC - зважені зовнішні операції на клас (фактично це WMC); WIEOC - зважені внутрішні і зовнішні операції на клас.*

WIEОС обчислюється так само, як і WMC, але враховує повний набір операцій, що реалізуються класом. Якщо BIH = 1, клас показує іншим класам всі свої можливості. Чим менше ВIM, тим менше мабуть поведінку класу. BIH може розглядатися і як міра складності. Складні класи, ймовірно, будуть мати малі значення BIH, а прості класи - значення, близькі до 1. Якщо клас з високою WMC має значення BIH, близьке до 1, слід з'ясувати, чому він настільки бачимо ззовні.

**2. Використання метрик Чидамбера-Кемерера**

Оскільки основу логічного представлення ПО утворює структура класів, для оцінки її якості зручно використовувати метрики Чідамбера-Кемерера . Для обчислення метрики LCOM треба визначити кількість пар методів класу. Воно розраховується по формулі



де *т -*кількість методів класу. 

Оскільки в класі три методу, можливі три пари: op\_ al () & ор\_а2 (), op\_al () & ор\_а3 () і ор\_а2 () & ор\_а3 (). Перша і друга пари не мають загальних властивостей, третя пара має загальну властивість ( pal ). Таким чином, кількість непов'язаних пар дорівнює 2, кількість пов'язаних пар дорівнює 1, і LCOM = 2-1 = 1.

Відзначимо також, що для класу Class D метрика СВО дорівнює 2, так як тут використовуються властивість pal і метод op\_ f ( ) з інших класів. Метрика LCOM в цьому класі дорівнює 0, оскільки методи op\_ dl ( ) і op\_d2 () пов'язані з властивості pdl , а негативне значення заборонено.

**Висновок:**

Використання наборів метрик Чидамбера - Кемерера дозволяє НЕ тільки визначати складність проекту, але модифікувавши певні характеристики, врахувати особливості розроблюваного проекту, провести ефективний аналіз інформаційної системи, виявити вразливі місця.

В даний час при оцінці проектів, реалізованих на мовах  об'єктно-орієнтованого програмування НЕ досить використання одного набору метрик

 Чидамбера -Кемерера.

**ПЕРЕЛІК ЛІТЕРАТУРИ:**

1. Черноножкін С.К. Методи і інструменти метричної підтримки розробки якісних програм: автореферат, - Новосибірськ, 1998.
2. Орлов С.А. Технології розробки ПО. / - СП б . : Пітер, 2002 - 464с.
3. Буч Граді, Про 'ектно -орієнтований аналіз і проектування з прикладами додатків С ++

[http : // khpi - iip . mipk . kharkiv . edu / library / case / buch / index . html](https://translate.google.com/translate?hl=ru&prev=_t&sl=ru&tl=uk&u=http://khpi-iip.mipk.kharkiv.edu/library/case/buch/index.html)