

ДИШЛЕВИЙ О.П.

ЗАСТОСУВАННЯ ПРЕДМЕТНО-ОРІЄНТОВАНОГО МЕТОДУ ПОБУДОВИ ЗАЛЕЖНОСТЕЙ МІЖ МЕТРИКАМИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Стаття присвячена описанню застосування предметно-орієнтованого методу побудови залежностей між метриками програмного забезпечення для визначення залежностей між прямими та непрямыми метриками. Розглянута побудова залежності непрямої метрики «розуміння», яка відповідає аналогічній властивості програмного забезпечення, від прямих метрик. Проводиться розрахунок ступеня впливу прямої метрики на метрику «розуміння», будується функція залежностей між найбільш впливовою прямою метрикою та непрямою метрикою.

The article is dedicated to describing the application of subject-oriented method of building dependences between software metrics for determine the dependencies between direct and indirect metrics. It was considered building depends indirect metrics "understanding", which corresponds to similar property of software from a number of direct metrics. Level of impact on the metric "understanding" is calculated for each direct metric, there is determined the most influential metric and there is built functions of dependencies between the most powerful direct and indirect metric.

Вступ

Все більшої актуальності набувають питання дослідження програмного забезпечення. Вирішенням цих питань займається емпірична інженерія програмного забезпечення [1].

Ключовим методом дослідження програмного забезпечення є вимірювання. Головний інструмент, який при цьому застосовується, – метрики. За допомогою метрик оцінюють властивості складових розробки програмного забезпечення (продуктів, процесів). Властивості програмного забезпечення дають його розуміння для аналізу та подальшого використання.

Існують прямі та непрямі метрики [2]. Прямі метрики можна отримати при вимірюванні. На основі прямих метрик важко проводити оцінку властивостей програмного забезпечення, так як вони ніяк не характеризують ці властивості. Для цього використовують непрямі метрики. Непрямі метрики дозволяють отримати числові характеристики програмного забезпечення, які допомагають в розумінні програмного забезпечення. Їх не можна виміряти, але можна отримати аналітичним чи емпіричним шляхом на основі прямих метрик. Головна задача, яка при цьому постає, – як визначити прямі метрики, від яких залежать непрямі, та яка залежність буде існувати між ними [3].

Огляд останніх досліджень

Для побудови залежностей використовують два підходи – статистичний аналіз [4] та нейронні мережі [5]. Вирішення задачі по визначенню залежностей між метриками за допомогою нейромереж потребує знати не тільки значення метрик (вхідних величин), а повинне бути проведено навчання нейромережі [5]. Навчання нейромережі проводиться на уже отриманих раніше даних. Так як питаннями визначення залежностей між метриками займалися тільки в реальному контексті (досліджувалися деякі конкретні метрики), а визначенням залежностей між різними прямими та непрямыми метриками взагалом не займалися, сформулювати правила для навчання нейромережі неможливо. Для вирішення задачі по визначенню залежностей між метриками за допомогою статистичного аналізу достатньо знати тільки значення метрик, а залежності будуються статистичними методами [4]. Тому в даному випадку необхідно застосовувати статистичний аналіз.

Статистичний аналіз проводиться за допомогою відповідних математичних програмних середовищ, до яких належать MatLab, MathCad, Maple, Mathematica, MS Excel. Крім них можна використати статистичні програмні середовища загального призначення Statistica, SPSS, SAS, Systat, Minitab, Statgraphics чи програмні середовища спеціального призначення SYSTAT, S-plus, STATA, PRISM, STADIA, Олимп, Класс-Мастер,

Статистик-Консультант. Досвід їх використання свідчить, що математичні програмні середовища та статистичні програмні середовища загального призначення для вирішення поставленої задачі потребують додаткового програмування з використанням статистичних алгоритмів. Середовища для емпіричних досліджень в програмному забезпеченні немає.

Отже, для побудови залежностей між прямими та непрямыми метриками програмного забезпечення потрібне середовище та метод, який буде використовуватися в середовищі.

Для вирішення головної задачі був розроблений предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення. На основі методу був реалізований засіб [6].

У статті розглядається застосування предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення на прикладі побудови залежності між непрямою метрикою «розуміння» та рядом прямих метрик.

Ціль статті

Основною ціллю статті є дослідження практичного застосування розробленого методу побудови залежностей між метриками через побудову залежності непрямої метрики «розуміння» від прямих метрик та визначення прямих метрик, від яких залежить дана непряма метрика.

Застосування предметно-орієнтованого методу

Сутність предметно-орієнтованого методу побудови залежностей між метриками програмного забезпечення полягає в тому, що побудова залежностей відбувається статистичними методами з врахуванням високої точності вимірювань програмного забезпечення без похибок та гіперболічною спадною залежністю між значеннями метрик та кількістю виміряних програм.

Загальна схема методу представлена на рис. 1.

На основі даного методу будуються залежності між непрямою метрикою «розуміння» та рядом прямих метрик. Метою дослідження є визначення прямих метрик, від яких залежить досліджувана непряма метрика, та яким чином.

Непряма метрика «розуміння» відповідає аналогічній властивості програмного забезпечення. Вона показує зрозумілість компонента програмного забезпечення, яка достатня для його аналізу та використання при розробці нового програмного забезпечення. Вона характеризує зусилля, які потрібно затратити для розуміння логічної структури компонента. В процесах проектування та повторної використання цю метрику слід застосовувати при визначенні доцільності аналізу компонента програмного забезпечення. При малому значенні розуміння доцільніше створити новий компонент, ніж аналізувати та застосовувати готовий. Максимально допустиме значення метрики «розуміння» визначається для кожного компонента індивідуально згідно потреб.

Використання непрямої метрики «розуміння» дозволяє спростити аналіз готового програмного забезпечення та прийняття рішень про можливість його подальшого застосування.

Отримання метрик для дослідження. Прямі метрики для досліджень були отримані шляхом вимірювань програмного забезпечення. Вимірювання прямих метрик програмного забезпечення проходило під час експерименту по визначенню законів розподілів метрик [6]. Було виміряно 50 прямих метрик.

Непряма метрика «розуміння» є деякою функцією від однієї чи декількох прямих метрик. Оскільки для непрямої метрики «розуміння» поки не визначено функції, на основі якої можна було б розрахувати метрику, вона була отримана шляхом експертного оцінювання програмного забезпечення [7]. Оцінку експерт міг вибрати із запропонованих: «погано», «задовільно», «добре», «відмінно». Експертне оцінювання проходило тільки для того програмного забезпечення, яке було до цього виміряне.

Об'єм вибірки. Важливим критерієм при отриманні метрик виступає їх кількість. Необхідний об'єм метрик напряму впливає на результати побудови залежностей. Мінімальна кількість метрик, які необхідні для побудови залежностей між метриками, була визначена експериментальним шляхом.

Для цього було проведено дослідження статистичних характеристик ряду метрик при збільшенні вибірки. Дослідження прово-

дилося в зворотному порядку. З усього об'єму були розраховані статистичні характеристики для декількох вибірок. Ці вибірки були отримані з загальної кількості метрик шляхом зменшення вибірки на 10 %. Наприклад, для метрики WMS було отримано 5000 значень. Для цієї вибірки були розраховані статистичні характеристики. Далі вибірку було зменшено на 10 %, і статистичні характеристики були порашовані для 4500 значень метрики. Далі вибірку було зменшено ще на 10 %, і статистичні характеристики були порашовані для цієї вибірки. Таким чином вибі-

рка зменшувалася до половини початкової вибірки.

Виявилося, що для визначення кількості значень метрик достатньо провести декілька контрольних розрахунків статистичних характеристик, щоб виявити, чи потрібно збільшувати вибірку. Крім цього, було визначено, що найсуттєвішим показником є медіана. Це пов'язано з відсутністю нормального розподілу серед метрик. А також стало зрозуміло, що більше 4-х вибірок брати із запропонованої недоцільно, так як вони в повній мірі відображають загальні закономірності.

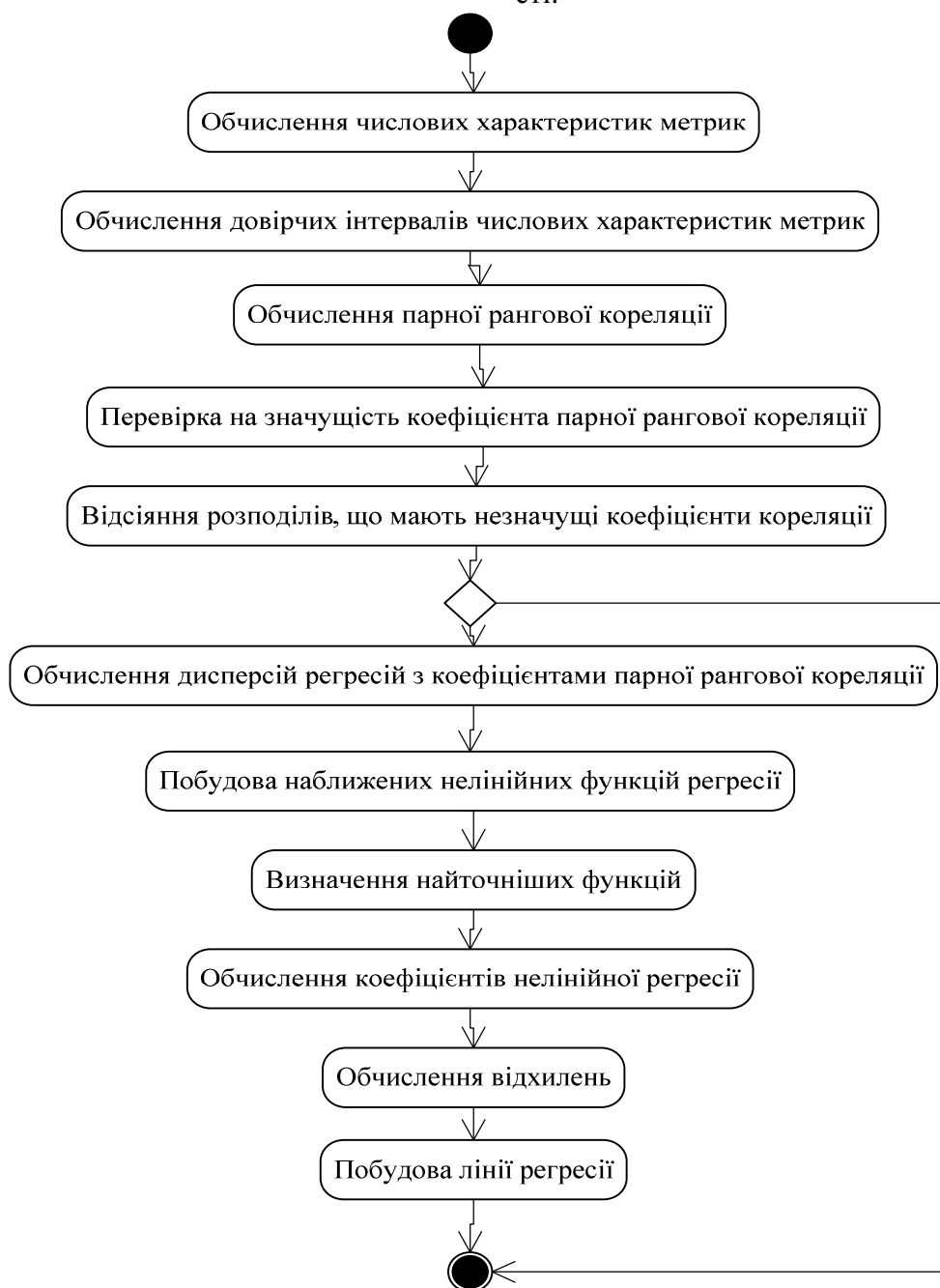


Рис. 1. Предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення

При цьому ключовим показником тут є точність. Для різної заданої точності вибірка

може змінюватися. Тому цей параметр повинен задавати сам дослідник.

Якщо співвідношення між сусідніми замірами не менше, ніж задана точність, то дану вибірку можна вважати повною для побудови залежностей. Тобто збільшувати кількість вимірів непотрібно. Описане співвідношення має вид:

$$\gamma \leq \frac{\tilde{x}_1}{\tilde{x}_2} \approx \frac{\tilde{x}_2}{\tilde{x}_3} \approx \frac{\tilde{x}_3}{\tilde{x}_4} \quad (1)$$

де γ – задана точність, x_i – медіани кожної з вибірок.

Відповідно, якщо медіани відрізняються для кожної з вибірок на задану точність, ви-

міряний об'єм метрик приймається для подальших розрахунків. Якщо ні – пропонується збільшити об'єм вибірки.

Підтвердженням отриманих результатів слугують статистичні характеристики деяких вимірних й досліджених метрик [8].

Обчислення числових характеристик метрик. Відповідно до розробленого методу, спочатку були отримані числові характеристики прямих метрик та непрямой «розуміння» (RA): математичне сподівання, середнє квадратичне відхилення, коефіцієнти ексцесу та асиметрії. Деякі значення приведені в табл.1.

Табл. 1. Початковий та центральний моменти метрик

Познач. метрики	Мат. сподівання	Середнє квадрат. відхилення	Коеф. Асиметрії	Коеф. ексцесу	Пояснення метрики
AVRPF	58.8417	17.9308	-0.429	2.4917	Середнє значення виключень в модулі
TM	2.5140	2.4356	1.7780	6.0558	Загальна кількість функцій (модулів компонента)
TLOC	170.9636	142.7548	1.2357	3.8945	Загальне число непорожніх рядків, включаючи коментарі та рядки заголовків
TNCSS	121.5091	111.8934	1.3023	4.0540	Загальне число вихідних конструкцій. Не включає порожні рядки, рядки коментарю, чи заготовочні рядки
TVG	16.1000	19.7032	2.1608	7.5764	Загальна цикломатична складність у всіх модулях (кількість ключових слів розгалужень в кожному модулі).
TMV	2004.0275	2450.4168	2.0724	7.2958	Загальний об'єм по Хольстеду, обчислений для модулів [9]
AHV	668.0450	577.4597	0.8903	2.9307	Середній об'єм по Хольстеду, обчислений для модулів [9]
TNE	53018.407	72147.877	2.2935	9.3088	Загальні зусилля по Хольстеду [9]
MNE	34660.263	43164.184	0.8843	6.7029	Максимальне значення зусиль по Хольстеду [9]
AVGHE	19865.306	25139.999	1.9345	6.6292	Середнє значення зусиль по Хольстеду [9]
MFO	8.9912	9.3785	1.5897	5.7260	Максимальне число викликів інших функцій в модулі
MFI	1.0000	1.7008	2.4493	10.2992	Максимальне число викликів модуля
AVGFI	0.3838	0.5715	1.5512	5.0540	Загальне число викликів інших функцій в модулях
'RA'	2.9279	1.3823	-0.2189	0.6029	«розуміння»

Обчислення кореляції та відсіання незалежних метрик. Обчислення парної рангової кореляції відбувається шляхом розрахунків коефіцієнтів Спірмена та Кендала [9]. Деякі розраховані коефіцієнти приведені в табл. 2. Крім цього, в таблиці подана значущість отриманих коефіцієнтів, яка говорить про наявність чи відсутність залежності.

Як видно з табл. 2, непряма метрика «розуміння» залежить від більшості прямих метрик, про що свідчить значущість. З загальної кількості 50 прямих метрик було відсіано 5. Для інших 45 метрик будувалися регресії.

Побудова регресій. Побудова регресії проводилася для кожної прямої метрики окремо. Далі буде приведений приклад побудови регресії для непрямой метрики «розуміння» та прямої метрики «кількість непорожніх рядків» (TLOC).

Табл. 2. Коефіцієнти кореляції для непрямой метрики «розуміння» та прямих метрик

Назва метрики	Коефіцієнт кореляції Кендала	Коефіцієнт кореляції Спірмена	Значущість
AVRPF	0.0912	0.1368	+
TM	0.3986	0.5979	+
TLOC	0.0693	0.1040	+
TNCSS	0.0263	0.0395	-
TVG	-0.2482	-0.3724	+
TMV	0.1616	0.2424	+
AHV	0.0926	0.1389	+
TNE	-0.1294	-0.1940	+
MNE	0.1991	0.2987	+
AVGHE	0.0992	0.1489	+
MFO	0.0161	0.0241	-
MFI	0.1941	0.2912	+
AVGFI	0.1327	0.1991	+

Спочатку проводиться розрахунок дисперсій. Дисперсії представлені у графічному

виді на рис. 2. Як видно з нього, значних відхилень по кожному з вимірювань немає, тому наступним етапом буде побудова регресії. Перед побудовою регресії було визначено кореляційне поле для метрик (рис. 3). Кореляційне поле має складну конфігурацію, що свідчить про нелінійну функціональну залежність між метриками.

Далі, як сказано в [10], проводилася побудова декількох наближених регресій, розраховувалися їх коефіцієнти та вибиралася найближча.

Побудовані типові функції регресії та їх відхилення приведені в табл. 3. У функціях: x – математичне сподівання, умовні математичні сподівання експертних оцінок, a , b , c – коефіцієнти.

Табл. 3. Коефіцієнти та відхилення функцій регресії

№ п/п	Функція	Коефіцієнт a	Коефіцієнт b	Коефіцієнт c	Відхилення
1	$y = ax+b$	0.001161	2.83	0	0.2917
2	$y=1/(ax+b)$	-0.0001219	0.352	0	0.2913
3	$y=x/(ax+b)$	0.329	0.4844	0	0.3142
4	$y = a \ln x+b$	1.252	1.127	0	5.7184
5	$y=b \exp(ax)$	0.0003761	2.836	0	0.2914
6	$y = ax^3+b$	0.996e-009	2.918	0	0.2912
7	$y=c+b \exp(ax)$	-11.4	-0.07609	2.938	0.2909
8	$y=ax^2+bx+c$	-6.025e-006	0.002988	2.75	0.3016
9	$y=1/(ax^2+bx+c)$	-0.006006	0.5697	0.376	2.8139
10	$y=x/(ax^2+bx+c)$	-5.971e-005	0.3377	0.3557	0.3210
11	$y=a/(1/x+b)$	2.064	0.6792	0	0.3142
12	$y=1/(a \exp(-x)+b)$	0.009149	0.3403	0	0.2906

За основу для вибору оптимальної функції слугує відхилення.

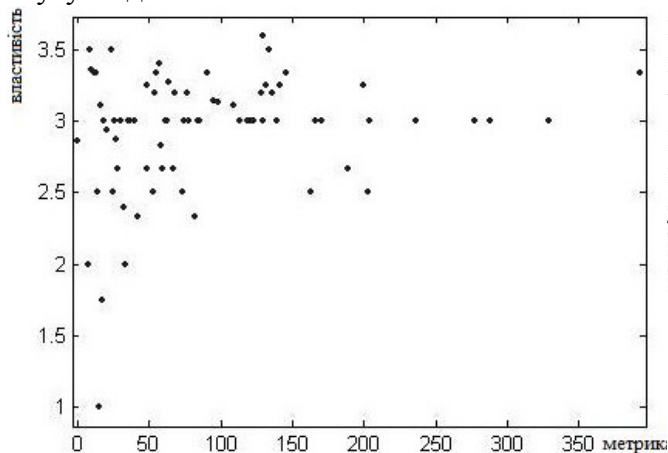


Рис. 3. Кореляційне поле метрик

Проаналізувавши значення відхилень можна зробити висновок, що найменше – 0.2906 для функції $y=1/(a \exp(-x)+b)$ (рис. 4). Тобто це функція, яка найточніше виражає залежність між метрикою «кількість непорожніх рядків» та «розуміння».

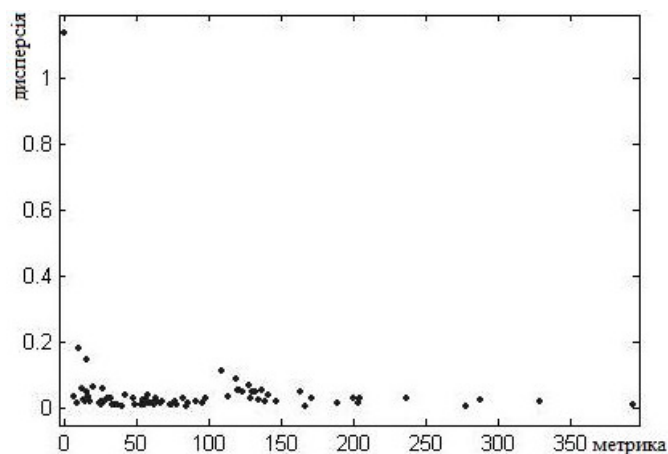


Рис. 2. Значення дисперсій

Висновки

В результаті проведеної роботи була виведена закономірність при визначенні об'єму вибірки метрики. Виявилось, що дану величину можна розрахувати на основі перевірки математичного сподівання декількох менших вибірок.

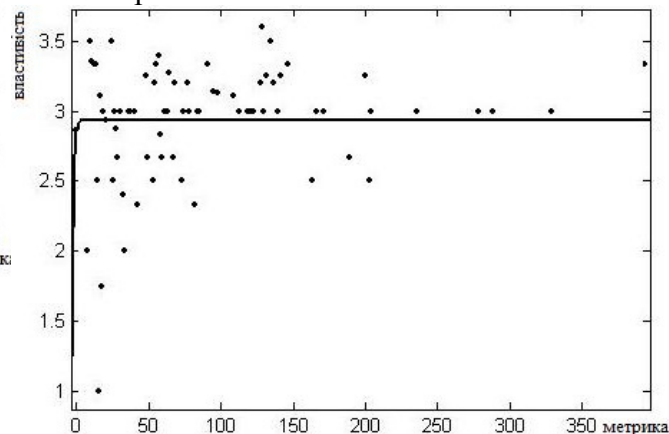


Рис. 4. Функція регресії між метриками

Було визначено, що більшість метрик не мають нормального закону розподілу. Це пов'язано з особливостями даних програм-

ного забезпечення, так як існує великий об'єм програмного коду з невеликими значеннями метрик. Підтвердження цьому є у [8].

Проведені дослідження свідчать про те, що на непряму метрику «розуміння» впливає багато прямих метрик. Серед усіх прямих метрик була вибрана одна, для якої й визначалася функція залежності. Як виявилось, залежності між прямими метриками та непрямою метрикою «розуміння» мають нелінійний характер, про що свідчать коефіцієнти кореляції та побудована регресія між прямою метрикою «кількість непорожніх рядків» та непрямою «розуміння». Для виведення залежності непрямої метрики «розуміння» від прямих метрик, останні отримуються шляхом вимірювання програмного забезпечення, а непряма метрика «розуміння» отримується шляхом експертного оцінювання. В подальшому значення непрямої метрики пропонується отримувати шляхом розрахунків на основі отриманої функції залежності від прямих метрик. Результати досліджень для ряду об'єктно-орієнтованих метрик підтверджуються в [8].

Предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення, приклад застосування якого приведений у даній статті, дозволяє будувати залежності між тими прямими та непрямими метриками, для яких цих залежностей ще не було виведено. В першу чергу це стосується тих непрямих метрик, які безпосередньо характеризують властивості програмного забезпечення та прямими метриками. Це дасть можливість аналізувати властивості програмного забезпечення на основі показників непрямих метрик та приймати управлінські рішення по проекту. Характеризувати властивості програмного забезпечення досить просто при наявності функціональних зв'язків із прямими метриками. Виведені залежності допомагають пов'язати прямі метрики, які можна виміряти, із властивостями програмного забезпечення, які використовуються при прийнятті рішень.

Подальша робота буде направлена на визначення закономірностей між ключовими прямими та непрямими метриками.

Список літератури

1. Koji Torii, Ken-ichi Matsumoto, Kumiyo Nakakoji, Yoshiro Takada, Kaduyuki Shima, Ginger 2: An Enviroment for Computer-Aided Empirical Software Engineering // IEEE Transactions on Software Engineering, vol 25. No 4, July/August 1999, P 475-486.
2. Norman E. Fenton, Shari Lawrence Pfleeger Software Metrics: A Rigorous and Practical Approach.- Cambridge University Press, 1996.-638p.
3. Forrest Shull, Janice Singer, Dag I.K. Sjoberg Guide to Advanced Empirical Software Engineering. – Springer-Verlag London Limited 2008.-394p.
4. Вентцель Е.С. Теория вероятностей: Учеб. для вузов. – 7-е изд. стер. – М.: Высш. шк., 2001. – 575 с.: ил.
5. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - Пер. с англ. – М.: Мир, 1992. - 118 с.
6. Дишлевий О.П. Пакет статистичного аналізу для емпіричної інженерії програмного забезпечення / О.П. Дишлевий // Наука і молодь. Прикладна серія. Збірник наукових праць. – 2009. – № 9. – С. 104–108.
7. Хоменко В.А., Дышлевый А.П. Метод экспертного оценивания свойств повторно используемых компонентов программного обеспечения // Том 1: Материали VIII Міжнародної науково-технічної конференції «Авіа-2007».- Т.1.- К.:НАУ, 2007, с. 13.169-13.172.
8. Michele Lanza, Radu Marinescu. Object-oriented metrics in practice: Using software metrics to characterize, evaluate, and improve the design of object-oriented systems. – Springer- Verlag Berlin Limited 2006.-205p.
9. Software Measurement and Estimation: a practical approach / Linda M. Laird, M. Carol Brennan. John Wiley & Sons, Inc., Hoboken, New Jersey 2006.- 257 p.
10. Бабак В.П., Білецький А.Я., Приставка О.П., Приставка П.О. Статистична обробка даних/ Монографія. – Київ: «МІВВЦ», 2001. – 388 с.