

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1. Означення задачі добування опінії	10
1.2. Історія галузі добування опінії	12
1.3. Порівняльний аналіз систем добування та оцінювання опінії	15
1.4. Обґрунтування використання вагових коефіцієнтів при формуванні інтегральної оцінки об'єкту	17
1.5. Методи організації API	22
1.6. Висновки до першого розділу	24
РОЗДІЛ 2 ОПИС ТЕХНОЛОГІЙ ВИКОРИСТАНИХ ДЛЯ РОЗРОБКИ ..	25
2.1. Мова програмування PHP	25
2.2. Розширювана мова розмітки	31
2.3. Adobe Dreamweaver CS 5.5	35
2.4. Загальні відомості про HTTP сервер Apache	38
2.5. Конфігурування HTTP серверу Apache	40
2.6. Висновки до другого розділу	43
РОЗДІЛ 3 ПРОЕКТУВАННЯ СИСТЕМИ	44
3.1. Розробка алгоритму системи	44
3.2. Розробка прикладного програмного забезпечення	49
3.3. Тестування системи	52
3.3.1. Тестування швидкості виконання	52
3.3.2. Тестування правильності написання коду	54
3.3.3. Тестування обрахунків	56
3.4. Висновки до третього розділу	57
РОЗДІЛ 4 ОБґРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ	58
4.1. Розрахунок норм часу на виконання науково-дослідної роботи	59

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи	59
4.3. Розрахунок матеріальних витрат	62
4.4. Розрахунок витрат на електроенергію	62
4.5. Визначення транспортних затрат	63
4.6. Розрахунок суми амортизаційних відрахувань	63
4.7. Обчислення накладних витрат	64
4.8. Складання кошторису витрат та визначення собівартості НДР	65
4.9. Розрахунок ціни НДР	65
4.10. Визначення економічної ефективності і терміну окупності капітальних вкладень	66
4.11. Висновки до четвертого розділу.....	67
РОЗДІЛ 5 Охорона праці та Безпека в надзвичайних ситуаціях	68
5.1. Охорона праці	68
5.2. Визначення Безпека в надзвичайних ситуаціях	71
РОЗДІЛ 6 ЕКОЛОГІЯ.....	76
6.1. Актуальність охорони навколишнього середовища.....	76
6.2. Статистичне оцінювання техногенних впливів	76
6.2. Статистичний аналіз екологічності виробництва	79
ВИСНОВКИ	82
АНОТАЦІЯ	84
ANOTATION	85
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	86
Додаток А Акт впровадження	90
Додаток Б Текст наукової публікації дипломної роботи магістра	91
Додаток В Вміст wslд файлу	92
Додаток Г Лістинг програми	95

ВСТУП

Актуальність теми. У зв'язку зі стрімким розвитком інформаційних технологій та інтернет-ресурсів зокрема, виникла потреба у аналізі надвеликих масивів текстової інформації. На сьогоднішній день все більше і більше користувачів мережі залишають відгуки стосовно певних програмних продуктів чи послуг на сторінках соціальних мереж, форумів чи блогів. Емоційне забарвлення (опінія) притаманне більшості висловлювань користувачів в тому числі і коментарям інтернет-магазинів програмних додатків.

Засоби opinion mining роблять крок назустріч користувачам, котрі прагнуть вибрати кращу послугу, з-поміж тисячі інших більш релевантний програмний продукт чи виробника та оцінити якість свого ПЗ чи послуги, здійснити маркетингове дослідження чи провести бренд-аудит, відповідно до відгуків користувачів. Використання засобів opinion mining дозволить суттєво зменшити час на пошук бажаного продукту для покупців, полегшить вибір послуги для споживачів, а також уможливить аналіз відгуків щодо власної продукції виробників, що полегшить роботу в сфері соціології при опитуванні користувачів мережі Інтернет. Тому тема дослідження є актуальною.

До провідних науковців, котрі займаються розв'язанням проблем в області opinion mining можна віднести: Bing Liu, Ph.D., професора кафедри комп'ютерних наук Іллінойського університету в Чикаго; Lillian Lee, Ph.D., професора кафедри комп'ютерних наук Корнельського університету. В Україні питаннями в цій галузі почали цікавитися досить недавно. Одними із небагатьох можна назвати компанії SemanticForce (semanticforce.net) та Юскан (youscan.com.ua).

Мета і задачі дослідження. Метою роботи є обґрунтування методу організації та створення API для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів.

Для досягнення вказаної мети в роботі поставлено наступні задачі:

- здійснити аналіз галузі Opinion Mining;
- провести порівняльний аналіз сучасних систем добування та оцінки opinii;
- проаналізувати методи побудови API для інформаційної системи;
- створити алгоритм інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів;
- здійснити програмну реалізацію інформаційної системи;
- провести тестування створеної інформаційної системи.

Об'єкт дослідження: процес дослідження методів організації API.

Предмет дослідження: методи організації API для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів.

Методи дослідження. Для вирішення поставлених задач використано наступні методи: аналіз та узагальнення – при проведенні аналізу існуючих методів добування та оцінювання opinii об'єктів і методів організації API; проектування та програмування – при створенні програмного засобу обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів.

Наукова новизна отриманих результатів:

- вперше запропоновано алгоритм інформаційної системи з ідентифікацією користувачів, обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів.
- вперше використано запропонований алгоритм інформаційної системи для організації API, що дозволило розширити функціональні можливості системи.

Практичне значення отриманих результатів. Інформаційна система дасть змогу оцінити опінію за кількістю позитивних, негативних та нейтральних коментарів по відношенню до компонент об'єктів, й в майбутньому покращити характеристики об'єктів.

Розроблений програмний комплекс впроваджено у ПП Брайляк В.С.
Акт впровадження наведено у додатку А.

Публікації. Результати роботи апробовано на VI студентській науково-технічній конференції ТНТУ ім. І. Пулюя “Природничі та гуманітарні науки. Актуальні питання” м. Тернопіль 2013р.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Означення задачі добування opinii

Емоційне забарвлення, або opinia, властиве майже усім висловлюванням людей, особливо в Інтернеті, де висловлювати свою думку дуже просто – така можливість є в Інтернет-магазинах, на блогах та форумах. Опінія показує уявлення людини про якість, характер, значення когось або чогось [1, 5].

Для більш чіткого розуміння термінів, дамо означення таким поняттям як добування opinii та оцінювання opinii.

Добування opinii (Opinion Mining) – галузь, що знаходиться на перетині пошуку інформації (Informational Retrieval), видобування даних (Data Mining) та комп'ютерної лінгвістики (Computer Linguistic), котра намагається виявити opinii, які містяться в текстових даних, написаних на природних мовах. Основною задачею Opinion Mining є виявлення емоційного забарвлення тексту та емоційної оцінки автора по відношенню до об'єктів, про які йде мова [9].

Opinion Extraction – підобласть Opinion Mining, котра включає в себе методи отримання вхідних даних для OM [9].

Sentiment Analysis та Sentiment Classification – підобласті Opinion Extraction та Opinion Mining. Задачею Sentiment Analysis та Sentiment Classification є аналіз емоційного забарвлення та класифікація емоційного забарвлення тексту відповідно [9].

Співвідношення понять Opinion Mining, Opinion Extraction, Sentiment Analysis та Sentiment Classification зображено у вигляді діаграми Вена на рисунку 1.1.

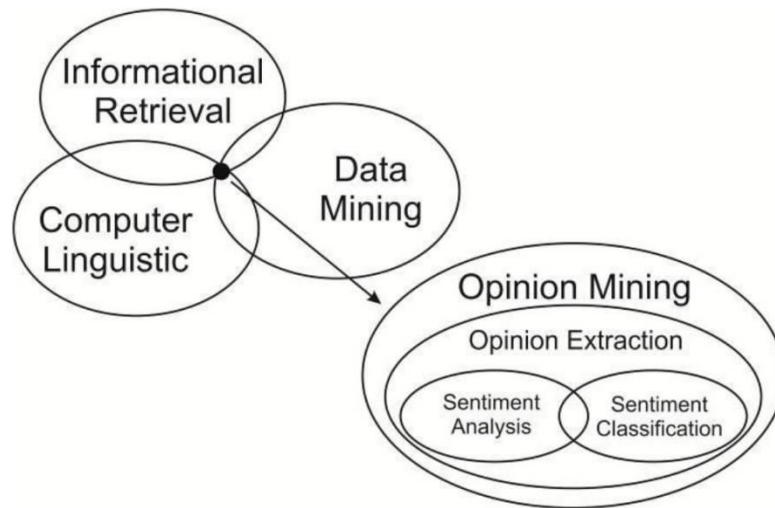


Рис. 1.1 Співвідношення Opinion Mining, Opinion Extraction, Sentiment Analysis та Sentiment Classification

Основними задачами добування опінії з текстових даних Інтернет ресурсів є:

1. Пошук релевантних документів, в котрих йде мова про об'єкт дослідження;
2. Семантичний аналіз речень, в яких згадується об'єкт дослідження (тобто пошук об'єкта, його компонент/характеристик та прикметників, котрі описують ці компоненти/характеристики);
3. Визначення емоційного забарвлення опінії (позитивного, негативного чи нейтрального) в реченні та речень у всьому тексті загалом;
4. Резюмування результатів та вивід їх у потрібній формі.

Добування опінії з web-сторінок Інтернет-ресурсів відбувається в декілька основних етапів (попередня підготовка даних, аналіз даних, класифікація опінії, вивід результату), які зображені на рисунку 2.2.

Попередня підготовка даних полягає в зборі релевантних web-документів, котрі містять опінії про шуканий об'єкт та виокремлення з них корисного тексту. Під корисним текстом розуміється текст без html-тегів. Така процедура забезпечується html-парсером, котрий видаляє html-теги з web-документу та формує чистий текст для наступного його аналізу.



Рис. 1.2. Структурна схема процедури добування та оцінювання опінії

Наступним етапом є аналіз даних. На цьому етапі відбувається підготовка текстових даних до класифікації опінії, зокрема: виокремлення потрібних речень та їхній подальший POS-аналіз (POS – part of speech tagging, розподіл речення на частини мови). POS-парсер формує XML-документ, в якому кожному слову, відповідно до частини мови, присвоюється окремий тег. Після POS-процедури, як правило, здійснюється пошук об'єкта, його характеристик та прикметників, котрі, власне і є опінією цього об'єкта.

Оцінювання опінії відбувається згідно певних математичних методів класифікації текстів. Найчастіше використовується одновимірний емотивний простір: позитив-негатив, або добре-погано. Але відомі й успішні випадки використання багатовимірних просторів [6].

Після всіх вищеописаних етапів здійснюється вивід результату оцінювання опінії текстових даних, котрий може бути відображений як в графічному, так і в текстовому вигляді.

1.2. Історія галузі добування опінії

Вперше поняття «opinion» використав Квірк (Quirk) у своїй книзі у 1985 році [2], де він визначив опінію, як закритий стан (стан, який не відкритий для об'єктивного спостереження або верифікації).

Однією із перших робіт, які були присвячені аналізу опінії в текстовій інформації є праці Subjective Understanding: Computer Models of

Belief Systems [3] та Beliefs, points of view and multiple environments [4]. Згодом, наукові дослідження стосовно добування opinii, в основному, були зосереджені на інтерпретації метафор, коментарях, точках зору та споріднених областях.

2001 рік можна вважати початком підвищення широкої поінформованості про наукові проблеми і можливості пов'язані із добуванням opinii та аналізу емоційного забарвлення текстової інформації, згодом було опубліковано близько сотні робіт, присвячених цим питанням.

До основних факторів, що стимулювали розвиток наукових досліджень в області Opinion Mining слід віднести:

- удосконалення методів машинного навчання в обробці природної мови та пошуку інформації;
- створення наборів даних для алгоритмів машинного навчання у мережі World Wide Web, зокрема, розвиток процесу коментування на веб-сайтах;
- реалізація інтелектуальних завдань та інтелектуальних комерційних додатків для області Opinion Mining.

До провідних науковців, котрі займаються розв'язанням проблем в області opinion mining можна віднести:

- Bing Liu, Ph.D., професора кафедри комп'ютерних наук Іллінойського університету в Чикаго, автора книги Web Data Mining – Exploring Hyperlinks, Contents and Usage Data та його наукових співробітників;
- Lillian Lee, Ph.D., професора кафедри комп'ютерних наук Корнельського університету, Bo Pang, аспірантки Ph.D., кафедри комп'ютерних наук Корнельського університету, співробітниці Yahoo Research, авторів книги Opinion mining and sentiment analysis;

- розробників інформаційних систем Google Products (Sasha Blair Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, Jeff Reynar), Bing Shopping, QuarkShop.

Серед російських науковців можна виділити:

- розробників системи RCO (Ермаков А.Е., Поляков П.Ю., Плешко В.В., Киселев С.Л., Харламов А.А., Рябов Г.Н.), котрими загалом опубліковано близько 50 робіт в області комп'ютерної лінгвістики та штучного інтелекту;

- спеціалістів проекту «VAAL», котрі розробили ряд методик контент-аналізу тексту, зокрема, кандидат філософських наук, старший науковий співробітник сектору логіки Інституту філософії РАН, Шалака Володимира Івановича та Димшіца Михайла.

Також варто відзначити фундаторів ідеї створення та реалізації системи OPTIMISM (Opinion Text Mining from Social Meshes), провідних науковців міждисциплінарних груп зі спільними інтересами та колишніми подібними дослідженнями XLDB@LASIGE, LIACC та ICS. Команда науковців XLDB: Mário J. Silva (провідний науковий співробітник), Paula Carvalho, Carlos Costa. З ICS та LIACC/FEUP варто виділити Eugénio Oliveira, Luís Sarmiento, Pedro Magalhães.

В Україні питаннями пов'язаними з добуванням та оцінкою опіній почали цікавитися досить недавно. Одними із небагатьох можна назвати компанії SemanticForce (semanticforce.net) та Юскан (youscan.com.ua), котрі розробили ряд методів збору та аналізу текстової інформації із блогів, форумів та соціальних мереж, зокрема для аналізу емоційного забарвлення висловлювань в мережі Інтернет.

1.3. Порівняльний аналіз систем добування та оцінювання опінії

Найбільш відомими програмними системами із використанням Opinion Mining на даний час є:

- система VAAL – базується на побудові частотного словника та його подальшого аналізу на присутність певних слів, що дають змогу визначити з деякою ймовірністю психолінгвістичні категорії;
- система InfoStream – ґрунтується на статистичних закономірностях, пов'язаних з присутністю певних лексем в текстах, байесовському підході та нейронних мережах;
- розробки компанії RCO – характеризуються широким спектром оригінальних алгоритмів і технологій інтелектуальної обробки текстів на природній мові;
- система Opinion Observer – розроблена для візуального порівняння двох об'єктів по їхніх характеристиках згідно відгуків користувачів;
- веб-сервіси TweetFeel та Twitrratr – забезпечують пошук ключового слова по повідомленнях твіттера та визначення їхнього емоційного забарвлення;
- веб-сервіс TextMap – система визначення вживаності ключового слова в засобах масової інформації (однією з функцій, є визначення емоційного забарвлення повідомлень);
- веб-сервіс QuarkShop – класифікує зібрані відгуки користувачів таким чином, щоб дізнатися, що інші люди говорять про той чи інший продукт;
- веб-сервіс SentiMetrix – фреймворк, котрий базується на інноваційних технологіях, для вимірювання емоційного забарвлення коментарів висловлених в електронних засобах масової інформації;

- проекти Bing Shopping та Google Products – дозволяють здійснювати пошук об'єктів купівлі, ранжують результати пошуку згідно відгуків користувачів в інтернет-магазинах.

Як видно з таблиці, жодна з зараз існуючих систем не використовує вагові коефіцієнти між компонентами об'єкта.

1.4. Обґрунтування використання вагових коефіцієнтів при формуванні інтегральної оцінки об'єкту

Як відомо [12], опінія текстової інформації в коментарях інтернет-магазинів для оцінки об'єктів подається п'ятіркою:

$$O = (e_j, a_{j,k}, so_{j,k,i,l}, h_i, t_l), \quad (1.1)$$

де, e_j – цільовий об'єкт (сутність);

$a_{j,k}$ – аспект/компонента/властивість цільового об'єкта e_j ;

$so_{j,k,i,l}$ – значення емоційного забарвлення опінії, висловленого власником опінії h_i стосовно компоненти $a_{j,k}$ цільового об'єкта e_j в час t_l ($so_{j,k,i,l}$ може набувати таких числових значень: +1 – позитивне висловлювання, -1 – негативне висловлювання, 0 – нейтральне висловлювання, або приймати інші числові значення в залежності від вибраної шкали).

Запропоновано використовувати уточнену модель опінії [12], а саме:

- Введення ваги $w_{j,k}$ для компоненти об'єкта $a_{j,k}$, у зв'язку з тим, що різні властивості/компоненти/характеристики можуть бути більш чи менш важливими;
- Запровадження лінгвістичної змінної для подання емоційного забарвлення $so_{j,k,i,l}$, що дозволить отримувати більш точніше числове значення опінії висловлювань;
- Введення ваги w_i для власника опінії h_i , в зв'язку з тим, що різні люди можуть мати різні пріоритети. Наприклад, висловлювання експерта в даній галузі повинно бути важливішим, ніж інші висловлювання;
- Введення ваги w_l для часу t_l , коли була висловлена опінія, в зв'язку з тим, що в певний період часу висловлювання може бути більш актуальним, ніж в інший.

Таким чином, враховуючи наведені вище міркування, модель (1.1) для формалізованого опису opinii об'єкта можна зобразити у вигляді:

$$Opinion = (O_j, (F_{j,k}, W_{j,k}), SO_{j,k,i,l}(H_i, W_i), (T_l, W_l)), \quad (1.2)$$

де $\{O_j, j = \overline{1, J}\}$ – множина об'єктів. O_j – об'єкт (сутність), J – кількість об'єктів;

$\{F_{j,k}, j = \overline{1, J}, k = \overline{1, K_j}\}$ – множина компонент. $F_{j,k}$ – аспект/компонента/властивість цільового об'єкта O_j , K_j – кількість компонент j -го об'єкта. Якщо $K_j=1$, то $O_j=F_{j,l}$ – у випадку, коли у висловлюванні відсутнє згадування певної компоненти об'єкта, проте описується об'єкт загалом;

$\{W_{j,k}, j = \overline{1, J}, k = \overline{1, K_j}\}$ – вага компоненти $F_{j,k}$;

$\{SO_{j,k,i,l}, j = \overline{1, J}, k = \overline{1, K_j}, i = \overline{1, I_{j,k}}, l = \overline{1, L_{j,k,i}}\}$ – множина емоційного забарвлення висловлювань. $SO_{j,k,i,l}$ – значення емоційного забарвлення opinii, висловленого власником opinii H_i стосовно компоненти $F_{j,k}$ цільового об'єкта O_j в час T_l . $SO_{j,k,i,l}$ представляється у вигляді лінгвістичної змінної та може набувати числових значень відповідно до її множини;

$\{W_i, i = \overline{1, I_{j,k}}\}$ – вага власника opinii H_i . $I_{j,k}$ – кількість власників opinii, що прокоментували k -й компонент j -го об'єкта;

$\{W_l, l = \overline{1, L_{j,k,i}}\}$ – вага часової мітки T_l . $L_{j,k,i}$ – кількість часових міток, в які i -й власник opinii прокоментував k -й компонент j -го об'єкта.

Введення таких уточнень для математичної формалізації opinii дасть змогу отримати більш достовірніший результат при узагальненні емоційного забарвлення висловлювань.

При інтегральному оцінюванні opinii об'єкта «application» його характеристики (компоненти), такі як «Functionality» та «Efficiency» у моделі (1) матимуть однакову вагу, що є не у повній мірі виправданим та коректним

з точки зору покупця. Проте, якщо ввести вагові коефіцієнти між такими компонентами як «Reliability», «Usability», «Efficiency», «Portability», «Maintainability» та «Functionality», то може виявитися, що компонента «Usability» важливіша за компоненту «Portability», а компонента «Functionality» важливіша за компоненту «Efficiency» при порівнянні двох, чи більше об'єктів, що дасть змогу більш достовірніше та точніше оцінити та порівняти ці об'єкти.

Важливою задачею є врахування вагових коефіцієнтів $W_{k,j}$ між компонентами $F_{j,k}$, W_i між власниками opinii H_i та W_l відносно часової мітки T_l в математичній моделі opinii (1.2) з урахуванням відгуків користувачів для покращення систем підтримки прийняття рішень при ранжуванні чи порівнянні об'єктів.

Розгляньмо для прикладу такий випадок: нехай є два об'єкти – Application1 та Application2 з компонентами «Functionality», «Reliability», «Usability», «Efficiency», «Maintainability» та «Portability». На рисунку 1.3 показано кількість позитивних, негативних, нейтральних відгуків користувачів.

1	2	3	4	5	6	7	8	9	10	11
E	A	SO _{pos}	SO _{neg}	SO _{neu}	SO	SO	SO	SO	SO	SO
app 1		поз (+1)	нег (-1)	нейтр (0)	всього	заг. к-ть	рейтинг	%	коеф.	% * коеф
	Functionality	125	25	3	153	513	100	6,08643	11,667046	71,0106
	Reliability	243	69	2	314	706	174	10,5904	16,056402	170,043
	Usability	500	34	1	535	807	466	28,3628	18,353423	520,554
	Efficiency	232	53	4	289	530	179	10,8947	12,053673	131,321
	Maintainability	421	42	3	466	1149	379	23,0676	26,131453	602,789
	Portability	369	24	7	400	692	345	20,9982	15,738003	330,469
				всього	2157	4397	1643	100	100	1826,19
app 2		поз (+1)	нег (-1)	нейтр (0)	всього	заг. к-ть	рейтинг	%	коеф.	% * коеф
	Functionality	350	8	2	360	513	342	18	11,667046	210,007
	Reliability	367	21	4	392	706	346	18,2105	16,056402	292,396
	Usability	223	42	7	272	807	181	9,52632	18,353423	174,841
	Efficiency	179	51	11	241	530	128	6,73684	12,053673	81,2037
	Maintainability	680	2	1	683	1149	678	35,6842	26,131453	932,48
	Portability	256	31	5	292	692	225	11,8421	15,738003	186,371
				всього	2240	4397	1900	100	100	1877,3

Рис. 1.3 Приклад даних та обчислень

На рисунку 1.3 обчислено загальну кількість відгуків покомпонентно (стовбець 7), рейтинг компоненти об'єкта (стовбець 8) та відсоткове співвідношення рейтингу компоненти (стовбець 9). Вагові коефіцієнти (стовбець 10), обчислюються за формулою (1.3):

$$W_{j,k} = \frac{\sum_{j=1}^J \sum_{k=1}^{K_j} \sum_{i=1}^{I_{j,k}} L_{j,k,i}}{\sum_{j=1}^J \sum_{i=1}^{I_{j,k}} L_{j,k,i}}, \quad (1.3)$$

де $L_{j,k,i}$ – кількість часових міток, в які i -й власник опінії прокоментував k -й компонент j -го об'єкта.

На рисунку 1.4 зображено діаграму відгуків користувачів, що побудована згідно даних на рисунку 1.3. Виникає запитання, який з програмних продуктів вважати кращим? Теоретично можна вважати кращим Application 1, тому що згідно діаграми відсоткового співвідношення емоційного забарвлення коментарів в неї переважають компоненти «Usability», «Efficiency» та «Portability».

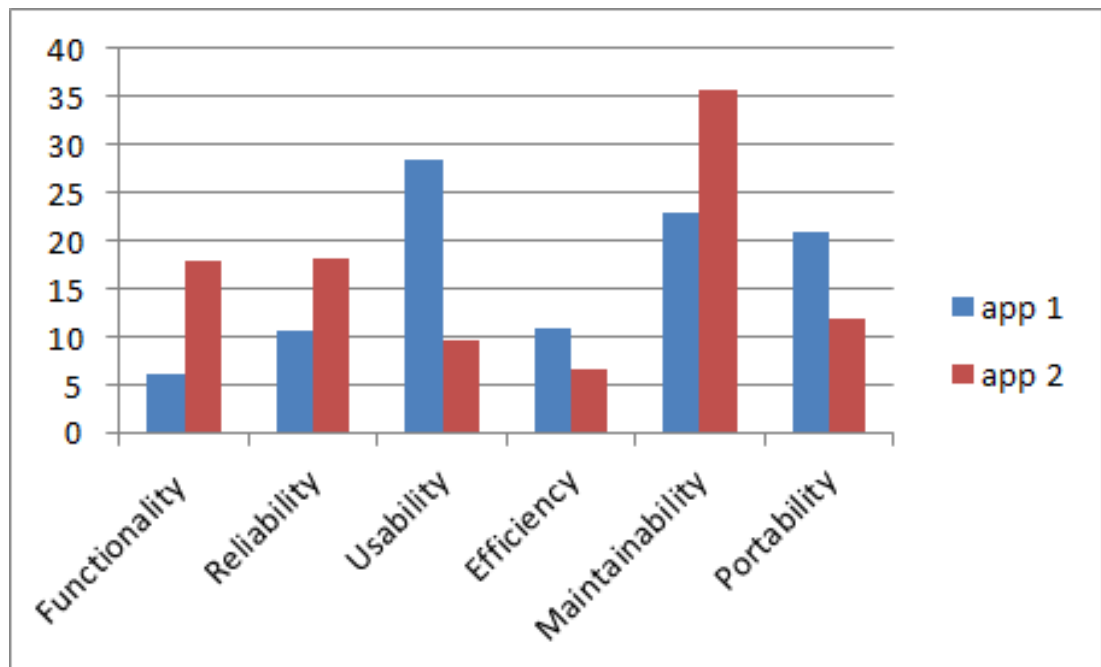


Рис 1.4 Діаграма співвідношення відгуків користувачів

На рисунку 1.5 зображено діаграму співвідношення кількості відгуків користувачів із застосуванням вагових коефіцієнтів.

Як зображено рисунку 1.5, якщо ввести вагові коефіцієнти між такими компонентами як «Reliability», «Usability», «Efficiency», «Portability», «Maintainability» та «Functionality», то може виявитися, що компонента «Maintainability» важливіша за компоненту «Portability», а компонента «Reliability» важливіша за компоненту «Usability» та взагалішому результаті отримуємо, що продукт Application2 кращий, ніж Application 1.

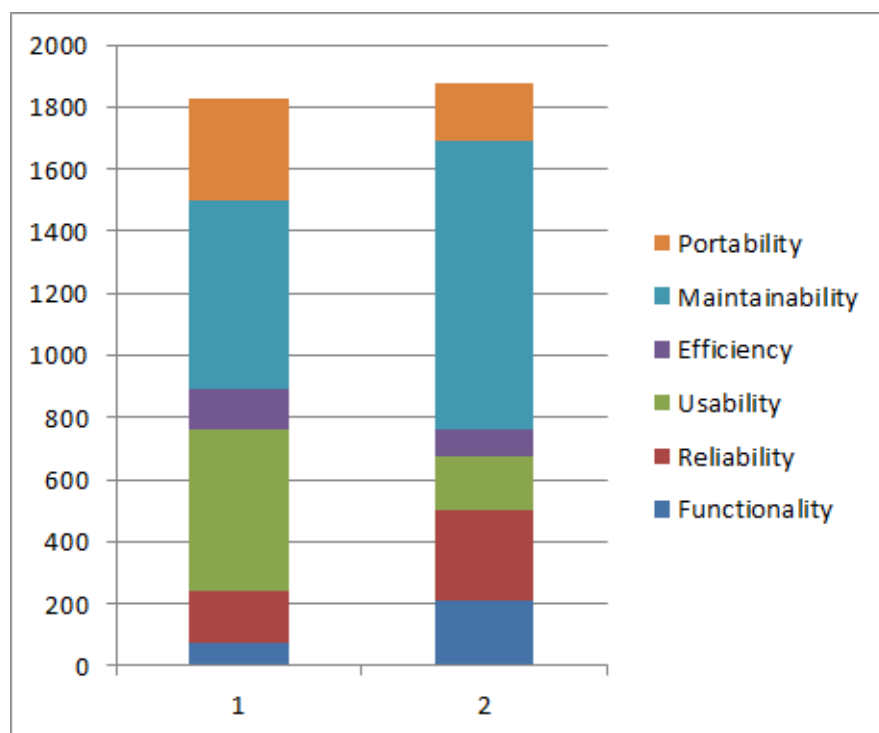


Рис 1.5 Діаграма співвідношення відгуків користувачів з застосуванням вагових коефіцієнтів

З робіт [5-9] слідує, що значення емоційного забарвлення $SO_{j,k,l,l}$, як правило, характеризується лише трьома значеннями (+1 – позитивне висловлювання, -1 – негативне висловлювання, 0 – нейтральне висловлювання, хоча можуть бути й інші варіації, напр. від -5 до 5), що в свою чергу, не в повній мірі відображає opiniю користувача щодо певного об'єкта.

У роботі $SO_{j,k,i,l}$ описується відповідно до (1.2) у вигляді лінгвістичної змінної, що дає змогу представити емоційне забарвлення висловлювань за допомогою нечітких значень відповідно до її терм-множини.

Таким чином може бути подане у вигляді набору:

$$SO_{j,k,i,l} = (b, P, X, G, M), \quad (1.4)$$

де b – найменування лінгвістичної змінної;

P – множина її значень (терм-множина), що представляють собою імена нечітких змінних, областю визначення, кожної з яких є множина X ;

G – синтаксична процедура, що дає змогу оперувати елементами терм-множини T , зокрема, генерувати нові терми (значення);

M – семантична процедура, що дає змогу перетворити кожне нове значення лінгвістичної змінної, утвореною процедурою G , у нечітку змінну, тобто сформувати відповідну нечітку множину.

Таким чином власник opinii H_i може виступати в ролі експерта, а емоційне забарвлення $SO_{j,k,i,l}$ можна описати у вигляді лінгвістичної змінної.

1.5. Методи організації API

API в контексті веб-розробки, зазвичай визначається як набір HTTP запитів, а також визначення структури повідомлення відповіді, які використовують Extensible Markup Language (XML) або JavaScript Object Notation (JSON) формат.

SOA (service-oriented architecture) – архітектурний шаблон програмного забезпечення, модульний підхід до розробки програмного забезпечення, заснований на використанні розподілених, слабо пов'язаних заміняних компонентів, оснащених стандартизованими інтерфейсами для взаємодії за стандартизованими протоколами.

За словами Кліва Фінкельштейна (Clive Finkelstein), автора інформаційної інженерії (information engineering) [7], до появи концепції SOA при розробці систем в якості відправного моменту для програмування бізнес-логіки використовувалися діаграми робочих потоків і блок-схеми систем. Розроблені вручну програми ретельно тестувалися, після чого впроваджувалися. Сьогодні ситуація змінилася докорінно: сучасні інструменти управління бізнес-процесами дозволяють обійтися без ручної розробки та тестування. Так, за допомогою методів моделювання можна перевіряти коректність виконання бізнес-логіки, представленої в діаграмах, а потім автоматично отримувати описи цих діаграм на XML-мовах управління бізнес-процесами. На думку Кліва Фінкельштейна, така технологія управління бізнес-процесами є великим кроком вперед з точки зору підвищення ефективності розробки систем; за значимістю її можна порівняти із створенням в кінці 50-х років компіляторів мови високого рівня. Дійсно, даний підхід дозволяє спростити виклик Web-сервісів з будь-якого місця розташування і їх виконання на основі бізнес-правил. Крім того, при зміні цих правил, коригується відповідна логіка в діаграмах: діаграми автоматично генеруються заново. Таким чином, закладаються передумови для переходу від повільного ручного кодування, використовуваного зараз при створенні систем, до автоматизованого. Завдяки цьому компанії зможуть реалізовувати зміни бізнес-правил за хвилини або години, а не за місяць або рік.

Для організації запитів для надсилання і отримання даних використовують такі найпоширеніші методи:

- SOAP – протокол обміну структурованими повідомленнями в розподілених обчислювальних системах, базується на форматі XML. Однією з переваг даного протоколу є його гнучкість, що дозволяє використовувати різні транспортні протоколи. Стандартні реалізації використовують HTTP, як транспортний протокол, однак можливо використовувати JMS чи SMTP.

Не зважаючи на те, що SOAP є стандартом, різні програми часто генерують повідомлення в несумісному форматі. Наприклад, запит згенерований Axis-клієнтом, не буде розпізнано сервером WebLogic.

- REST (Representational State Transfer) – підхід до архітектури протоколів, які забезпечують доступ до інформаційних ресурсів. У REST-сервісах акцент зроблений на доступ до ресурсів, а не на виконання віддалених сервісів, в цьому й кардинальна відмінність від SOAP-сервісів. Якщо SOAP-клієнти запитують виконання дії на сервері, то REST-клієнти просто вимагають сам ресурс. Наприклад, замість того щоб запитувати віддалене виконання функції для знаходження потрібного замовлення, просто запитуються дані про замовлення.

1.6. Висновки до першого розділу

В першому розділі проаналізовано предметну область галузі Opinion Mining, визначено завдання цієї галузі та описано структурну схему цього процесу. Проаналізовано існуючі системи для видобування opinii, кожна з яких має свої переваги та недоліки. Обґрунтовано використання в роботі вагових коефіцієнтів при обчисленні інтегрального показника opinii об'єктів. В результаті використання вагових коефіцієнтів система матиме підвищену точність аналізу вхідних даних. Проведено аналіз методів організації API для інформаційної системи.

РОЗДІЛ 2

ОПИС ТЕХНОЛОГІЙ ВИКОРИСТАНИХ ДЛЯ РОЗРОБКИ

2.1. Мова програмування PHP

PHP – це мова програмування, створена для генерації HTML-сторінок на Web-сервері і роботи з базами даних. В даний час підтримується переважною більшістю хостинг-провайдерів. Входить в LAMP – «стандартний» набір для створення Web-сайтів (Linux, Apache, MySQL) [10].

Назва PHP – аббревіатура, що означає «PHP: HypertextPreprocessor» (раніше розшифровувалось як «PersonalHomePageTools»). Спочатку PHP створювався як надбудова над Perl для полегшення розробки Web-сторінок.

У області програмування для мережі, PHP – одна з найпопулярніших скриптових мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності і розповсюдженню початкових кодів на основі ліцензії PHP. PHP відрізняється наявністю ядра і модулів, що підключаються, «розширень»: для роботи з базами даних, динамічною графікою, криптографічними бібліотеками, документами формату PDF. Існують сотні розширень, проте в стандартне постачання входить лише декілька десятків добре зарекомендували себе. Інтерпретатор PHP підключається до Web-серверу або через модуль, створений спеціально для цього сервера (наприклад, для Apache або IIS), або як CGI програми. Крім цього, він може використовуватися для вирішення адміністративних завдань в операційних системах UNIX, GNU / Linux, Microsoft Windows, Mac OS X і AmigaOS. Проте в такій якості він не набув поширення, віддаючи пальму першості Perl, Python і VBScript.

В даний час PHP використовується сотнями тисяч розробників. Порядку 20 мільйонів сайтів повідомляють про роботу з PHP, що складає більше п'ятої частки доменів Інтернету.

Код, створений для більш ранніх версій мови, часто не працює або працює некоректно з більш пізніми версіями мови. У більш пізніх версіях виключаються конструкції, методики, функції, що застосовувалися раніше. У результаті, додатки, створені кілька років тому, практично втрачають працездатність для сучасних версій мови і вимагають значної модифікації. Такі зміни обумовлені двома факторами: усунення неузгодженого синтаксису та усунення конструкцій, які заохочують створення небезпечного коду. Справедливості заради, слід зазначити, що відсутність зворотної сумісності взагалі характерно для сучасних різних мов.

PHP є мовою програмування з динамічною типізацією, що не вимагає вказівки типу при оголошенні змінних. Перетворення між скалярними типами часто здійснюється неявно без додаткових зусиль (втім PHP надає широкі можливості і для явного перетворення типів).

Масиви (array) підтримують числові і рядкові ключі і є гетерогенними. Масиви можуть містити значення будь-яких типів, включаючи інші масиви.

Порядок елементів і їх ключів зберігається.

PHP підтримує широкі об'єктно-орієнтовані можливості, повна підтримка яких була введена в п'ятій версії мови.

Об'єктно-орієнтоване програмування (ООП) — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. В ній використано декілька технологій від попередніх парадигм, зокрема успадкування, модульність, поліморфізм та інкапсуляцію. Незважаючи на те, що ця парадигма з'явилась в 1960-тих роках, вона не мала широкого застосування до 1990-тих. Сьогодні багато мов програмування (зокрема, Java, C#, C++, Python, PHP, Ruby та Objective-C, ActionScript 3) підтримують ООП.

Об'єктно-орієнтоване програмування сягає своїм корінням до створення мови програмування Симула в 1960-тих роках, одночасно з

посиленням дискусій про кризу програмного забезпечення. Разом із тим, як ускладнювалось апаратне та програмне забезпечення, було дуже важко зберегти якість програм. Об'єктно-орієнтоване програмування частково розв'язує цю проблему шляхом наголошення на модульності програми.

На відміну від традиційних поглядів, коли програму розглядали як набір підпрограм, або як перелік інструкцій комп'ютеру, ООП програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожний об'єкт здатний отримувати повідомлення, обробляти дані, та надсилати повідомлення іншим об'єктам. Кожен об'єкт — своєрідний незалежний автомат з окремим призначенням та відповідальністю.

Об'єктно-орієнтоване програмування - це метод програмування, оснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування.

На думку Алана Кея [12], розробника мови Smalltalk, якого вважають одним з «батьків-засновників» ООП, об'єктно-орієнтований підхід полягає в наступному наборі основних принципів:

- Все є об'єктами.
- Всі дії та розрахунки виконуються шляхом взаємодії (обміну даними) між об'єктами, при якій один об'єкт потребує, щоб інший об'єкт виконав деяку дію. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення. Повідомлення — це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися при виконанні дії.
- Кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів.
- Кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів.

- У класі задається поведінка (функціональність) об'єкта. Таким чином усі об'єкти, які є екземплярами одного класу, можуть виконувати одні й ті ж самі дії.

- Класи організовані у єдину деревовидну структуру з загальним корінням, яка називається ієрархією успадкування. Пам'ять та поведінка, зв'язані з екземплярами деякого класу, автоматично доступні будь-якому класу, розташованому нижче в ієрархічному дереві.

Таким чином, програма являє собою набір об'єктів, що мають стан та поведінку. Об'єкти взаємодіють використовуючи повідомлення. Будується ієрархія об'єктів: програма в цілому — це об'єкт, для виконання своїх функцій вона звертається до об'єктів що містяться у ньому, які у свою чергу виконують запит шляхом звернення до інших об'єктів програми. Звісно, щоб уникнути безкінечної рекурсії у зверненнях, на якомусь етапі об'єкт трансформує запит у повідомлення до стандартних системних об'єктів, що даються мовою та середовищем програмування. Стійкість та керованість системи забезпечуються за рахунок чіткого розподілення відповідальності об'єктів (за кожен дію відповідає певний об'єкт), однозначного означення інтерфейсів міжоб'єктної взаємодії та повної ізоляваності внутрішньої структури об'єкта від зовнішнього середовища (інкапсуляції).

В результаті дослідження Дебори Дж. Армстронг (англ. Deborah J. Armstrong) комп'ютерної літератури можна відокремити фундаментальні поняття (принципи), використані у переважній більшості визначень об'єктно-орієнтованого програмування. До них належить:

- Клас визначає абстрактні характеристики деякої сутності, включаючи характеристики самої сутності (її атрибути або властивості) та дії, які вона здатна виконувати (її поведінки, методи або можливості). Наприклад, клас Собака може характеризуватись рисами, притаманними всім

собакам, зокрема: порода, колір хутра, здатність гавкати. Класи вносять модульність та структурованість в об'єктно-орієнтовану програму. Як правило, клас має бути зрозумілим для не-програмістів, що знаються на предметній області, що, у свою чергу, значить, що клас повинен мати значення в контексті. Також, код реалізації класу має бути досить самодостатнім. Властивості та методи класу, разом називаються його членами.

- Об'єкт окремий екземпляр класу. Клас Собака відповідає всім собакам шляхом опису їхніх спільних рис; об'єкт Сірко є одним окремим собакою, окремим варіантом значень характеристик. Собака має хутро; Сірко має коричнево-біле хутро. Об'єкт Сірко є екземпляром (примірником) класу Собака. Сукупність значень атрибутів окремого об'єкта називається станом.

- Метод – можливості об'єкта. Оскільки Сірко — Собака, він може гавкати. Тому гавкати() є одним із методів об'єкта Сірко. Він може мати й інші методи, зокрема: місце(), або їсти(). В межах програми, використання методу має впливати лише на один об'єкт; всі Собаки можуть гавкати, але треба щоб гавкав лише один окремий собака.

- Абстрагування спрощення складної дійсності шляхом моделювання класів, що відповідають проблемі, та використання найприйнятнішого рівня деталізації окремих аспектів проблеми. Наприклад Собака Сірко більшу частину часу може розглядатись як Собака, а коли потрібно отримати доступ до інформації специфічної для собак породи коллі — як Коллі і як Тварина (можливо, батьківський клас Собака) при підрахунку тварин Петра.

- Успадкування. Клас може мати «підкласи», спеціалізовані, розширені версії надкласу. Можуть навіть утворюватись цілі дерева успадкування. Наприклад, клас Собака може мати підкласи Коллі, Пекінес,

Вівчарка і т.п. Так, Сірко може бути екземпляром класу Вівчарка. Підкласи успадковують атрибути та поведінку своїх батьківських класів, і можуть вводити свої власні. Успадкування може бути одиничне (один безпосередній батьківський клас) та множинне (кілька батьківських класів). Це залежить від вибору програміста, який реалізовує клас та мови програмування. Так, наприклад, в Java дозволене лише одинарне успадкування, а в C++ і те і інше.

- Приховування інформації (інкапсуляція) – приховування деталей про роботу класів від об'єктів, що їх використовують або надсилають їм повідомлення. Так, наприклад, клас Собака має метод гавкати(). Реалізація цього методу описує як саме повинно відбуватись гавкання (приміром, спочатку вдихнути() а потім видихнути() на обраній частоті та гучності). Петро, хазяїн пса Сірка, не повинен знати як він гавкає. Інкапсуляція досягається шляхом вказування, які класи можуть звертатися до членів об'єкта. Як наслідок, кожен об'єкт представляє кожному іншому класу певний інтерфейс — члени, доступні іншим класам. Інкапсуляція потрібна для того, аби запобігти використанню користувачами інтерфейсу тих частин реалізації, які, швидше за все, будуть змінюватись. Це дозволить полегшити внесення змін, тобто, без потреби змінювати і користувачів інтерфейсу. Наприклад, інтерфейс може гарантувати, що щенята можуть додаватись лише до об'єктів класу Собака кодом самого класу. Часто, члени класу позначаються як публічні (англ. public), захищені (англ. protected) та приватні (англ. private), визначаючи, чи доступні вони всім класам, підкласам, або лише до класу в якому їх визначено. Деякі мови програмування йдуть ще далі: Java використовує ключове слово private для обмеження доступу, що буде дозволений лише з методів того самого класу, protected — лише з методів того самого класу і його нащадків та з класів із того ж самого пакету, C# та VB.NET відкривають деякі члени лише для класів із тієї ж збірки

шляхом використання ключового слова `internal` (C#) або `Friend` (VB.NET), а `Eiffel` дозволяє вказувати які класи мають доступ до будь-яких членів.

- Поліморфізм означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по різному на однакові повідомлення. Наприклад, якщо Собака отримує команду `голос()`, то у відповідь можна отримати Гав; якщо Свиня отримує команду `голос ()`, то у відповідь можна отримати Рох-рох. На практиці - це реалізовується шляхом реалізації ряду підпрограм (функцій, процедур, методи тощо) з однаковими іменами, але з різними параметрами. В залежності від того, що передається і вибирається відповідна підпрограма.

2.2. Розширювана мова розмітки

Розширювана мова розмітки (англ. Extensible Markup Language, скорочено XML) — запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними програмними додатками, зокрема, через Інтернет. Є спрощеною підмножиною мови розмітки SGML. XML документ складається із текстових знаків, і придатний до читання людиною. Стандарт XML (перше видання від 10 лютого 1998, останнє, четверте видання 29 вересня 2006) визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої, шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних. Ці обмеження описуються мовами схем (англ. Schema), таких як XML Schema (XSD), DTD або RELAX NG. Прикладами мов, основаних на XML є: XSLT, XAML, XUL, RSS, MathML, GraphML, XHTML, SVG, і також XML Schema.

Коректний документ відповідає всім синтаксичним правилам XML. Документ, що не є коректним, не може називатись XML-документом. Сумісний синтаксичний аналізатор не повинен обробляти такі документи. Зокрема, коректний XML документ має:

- Лише один елемент у корені.
- Непорожні елементи розмічено початковим та кінцевим тегам (наприклад, `<пункт>Пункт 1</пункт>`). Порожні елементи можуть помічатись «закритим» тегом, наприклад `<IAmEmpty />`. Така пара еквівалентна `<IAmEmpty></IAmEmpty>`.
- Один елемент не може мати декілька атрибутів з однаковим іменем. Значення атрибутів знаходяться або в одинарних ('), або у подвійних (") лапках.
- Теги можуть бути вкладені, але не можуть перекриватись. Кожен некореневий елемент мусить повністю знаходитись в іншому елементі.
- Документ має складатися тільки з правильно закодованих дозволених символів множини Юнікоду. Єдиними кодуваннями, які обов'язково має розуміти XML-процесор, є UTF-16 та UTF-8. Фактичне та задеклароване кодування (англ. *character encoding*) документа мають збігатись. Кодування може бути задекларовано ззовні, як у заголовку «Content-Type» при передачі по протоколу HTTP, або в самому документі використанням явної розмітки на самому початку документа. У разі відсутності інформації про кодування, документ має бути в кодуванні UTF-8 (або його підмножині ASCII).

Приклад коректного документу який відповідає всі вище описаним вимогам наведено в лістингу 2.1.

Документ називається валідним, якщо він є коректним, містить посилання на граматичні правила та повністю відповідає обмеженням, вказаним у цих правилах (DTD або XML Schema або іншому подібному документі).

Лістинг 2.1 – Приклад XML-документу

```
<?xml version="1.0" encoding="utf-8"
xmlns="http://www.w3.org/2001/XMLSchema ?">
<Container ContainerType="Stack"
xmlns="http://schemas.microsoft.com/BusinessApplications/Layout"
>
  <CustomProperties>
  </CustomProperties>
  <Children>
    <OBPart PartType="List">
      <CustomProperties >
        <CustomProperty Name="Source" Value="AssociationList" />
        <CustomProperty Name="Data" Value="GetOrders" />
        <CustomProperty Name="MaximumVisibleRow" Value="12" />
        <CustomProperty Name="DefaultActionName" Value="Edit" />
        <CustomProperty Name="DefaultActionSource" Value="Targ"
      />
        <CustomProperty Name="ConsumerPartName" Value="IPOBP" />
      </CustomProperties>
      <ActionNames>
      </ActionNames>
    </OBPart>
    <OBPart PartType="InfoPath">
      <CustomProperties>
        <CustomProperty Name="Name" Value="IPOBP" />
        <CustomProperty Name="FormLocation" Value="OrderHeader"
      />
        <CustomProperty Name="IsReadOnly" Value="True" />
        <CustomProperty Name="Height" Value="300" />
      </CustomProperties>
    </OBPart>
  </Children>
</Container>
```

XML документ має ієрархічну логічну структуру, і може представлятись у вигляді дерева. Вузлами цього дерева можуть бути:

- елементи, фізична структура яких складається із:
 - а) коректної пари відкриваючого та закриваючого тегів (<Назва-тега>) та (</Назва-тега>);
 - б) тега порожнього елемента (<Назва-тега />),

- атрибути, що мають вигляд пар ключ-значення (назва атрибута="значення атрибута") і знаходяться або у відкриваючому, або у порожньому тезі (подібно до метаданих);

- вказівки щодо обробки документа (англ. Processing Instruction) (<?Обробник параметр ?>);
- коментарі (<!-- Текст коментаря -->);
- текст, або у вигляді простого тексту, або фрагментів CDATA (<![CDATA[довільний текст]]>).

З точки зору активного аналізу XML документ розглядається як послідовність елементів, які зчитуються послідовно використовуючи шаблон проектування ітератор. Такий підхід дозволяє створення рекурсивних аналізаторів, у яких структура коду відображає структуру аналізованих XML документів, проміжні результати аналізу можуть бути використані і розміщені у вигляді локальних змінних в підпрограмах, що виконують аналіз, передані як параметри до підпрограм нижчого рівня або повернені до підпрограм вищого рівня. До прикладів активних аналізаторів належать StAX у мові програмування Java, SimpleXML у PHP та System.Xml.XmlReader у .NET.

Активний аналізатор створює ітератор, що послідовно обходить різні елементи, атрибути та дані в XML документі. Код, що використовує цей «ітератор», може перевіряти поточний елемент (аби дізнатись, наприклад, чи є цей елемент стартовим, кінцевим або текстовим), та дізнаватись про його атрибути (локальна назва, простір назв, значення XML атрибутів зміст тексту тощо) і може пересунути ітератор на наступний елемент. Таким чином, код аналізатора може зчитувати інформацію із документа під час обходу. Підхід рекурсивного спуску сприяє зберіганню даних у вигляді типізованих локальних змінних в коді аналізатора, в той час як SAX, наприклад, зазвичай вимагає, аби аналізатор явно зберігав проміжні дані в стеку елементів, що є вищими елементами від того, який зараз аналізується. Код активного аналізатора може бути більш прямолінійним та зрозумілим і простішим для підтримки, аніж код SAX аналізатора.

2.3. Adobe Dreamweaver CS 5.5

AdobeDreamweaver (попередня назва Macromedia Dreamweaver) – текстовий редактор, що відноситься до категорії WYSIWYG-редакторів, має дуже багато переваг: зручний інтерфейс, налаштування функцій, підтримка великих проектів і ShockWave-технологій, можливість завантаження файлів через FTP, підтримка SSI і багато чого іншого. Для роботи в цій програмі не потрібно досконально знати HTML (у цьому і полягає перевага технології WYSIWYG (англ. WhatYouSeeIsWhatYouGet) –що бачиш, те й отримуєш).

Проте Dreamweaver на кілька кроків випереджає інші редактори, що використовують технологію WYSIWYG, у першу чергу завдяки тому, що генерує «чистий» HTML-код. Dreamweaver дозволяє позбутися однотипної роботи при створенні сторінок (наприклад, верстка тексту) за допомогою використання опції «Запис послідовності команд» (записується послідовність вироблених вами команд, після натискання, наприклад, CTRL+P, Dreamweaver відтворює їх усі у тій самій послідовності).

Від конкурентів Adobe Dreamweaver CS5.5 відрізняє в тому числі і те, що якщо ви вже працювали в якійсь програмі від Adobe, то розберетеся і в інших, в інтерфейсі багато схожих пунктів.

«Недружність» по відношенню до початківців є особливо прикрою, тому компанія Adobe дійсно намагається зробити продукт більш привабливим для новачків. Вибір, що надається користувачеві, змінений, щоб зменшити структурну складність, полегшити розуміння і забезпечити докладними коментарями власне вихідний текст, спростивши завдання потребуючим допомоги. Ймовірно, це вдалий хід, але в іншому Dreamweaver - величезний і страхітливий продукт, за освоєння якого візьмуться лише найвідважніші, які не потребують великої допомоги.

В програмі є можливість перевірки посилань на читабельність, причому не посторінково, а всього сайту цілком, що може заощадити чимало

часу розробнику. Наявна величезна кількість різних дизайнів, втім, при необхідності можна створити дизайн самостійно. Можливість включати і відключати властивості CSS (вони відключаються символами коментарів, але не видаляються зовсім) полегшує діагностику та експерименти. Крім того, в режимі інспектування (InspectMode) простіше розрізняти компоненти блокової моделі (межі, поля, в тому числі оточуючі текст, і інші елементи HTML), що нелегко зробити в попередніх версіях програми. Увімкнувши цей режим кожен з різних компонентів буде відтворюватися особливим кольором, що полегшить пошук.

При використанні разом з видом LiveView режим інспектування являє собою відмінний спосіб швидко коректувати компоновку і вигляд сторінок, при цьому зменшується число обмежень на перегляд елементів. Наприклад, в режимі інспектування можна переміщатися по сторінках, які виглядають так само, як для користувачів (переглядаючи при цьому зміни в програмному коді, що відбуваються в процесі взаємодії з динамічними даними та серверними додатками).

Один з найкращих компонентів Dreamweaver CS5, який напевно сподобається як досвідченим, так і починаючим це BrowserLab. BrowserLab – нова служба CS Live, за допомогою якої можна переглядати проєктовані матеріали в різних браузерях та операційних системах, не покидаючи Dreamweaver. Коло сумісних браузерів досить широке – в даний час в нього входять Firefox, Internet Explorer, Safari і Google Chrome – щоб легко переконатися в тому, що сторінки сайту правильно відображаються в цих браузерах.

Є можливість писати сайти, заточені під екран мобільного телефону, за допомогою JQuery, в тому числі під Android і IOS. Величезна кількість тегів – HTML, CFML, JSP, ASP, PHP, WHL. Можна вставляти в «тіло» сайту SWF, FLV, ActiveX, Shockwave, різні виконувані програми. Присутня велика

кількість шрифтів, стилів і кольорів оформлення, можна підібрати собі будь-який, який подобається, забезпечивши максимальну зручність для відвідувачів і унікальність дизайну сторінки – все це надає справжню свободу творчості. За системними вимогам програма не дуже вимоглива, їй вистачить п'ятисот мегабайт на жорсткому диску, звичайного процесора IntelPentium 4 або Athlon 64, п'ятисот дванадцяти мегабайт ОЗУ (ну, тут звичайно чим більше – тим краще), працює під усіма ОС Windows, починаючи з XP і закінчуючи Windows 7, є можливість запустити під Macintosh. Для онлайн-підтримки, зрозуміло, знадобиться підключення до Інтернету.

Серед найбільш корисних, кваліфікованим користувачам, функцій – розширена сумісність з відкритою системою контролю версій Subversion (більше можливостей для локальної роботи з файлами, синхронізація з віддаленим репозитарієм SVN, скасування змін, можливість приховати непотрібні файли), інтеграція (через окремий модуль розширення) з AdobeBusinessCatalyst для розробки сайтів з мінімумом програмування внутрішніх (backend) компонентів, поглиблена взаємодія з системами управління контентом, зокрема які базуються PHP. Користувачам таких інфраструктур, як WordPress, Drupal та Joomla, пропонуються підказки для кодування в Dreamweaver CS5, завдяки яким процес організації взаємодії між різними компонентами стає значно зрозумілішим.

Крім того, функція динамічно пов'язаних файлів (DynamicallyRelatedFiles) відображає всі зовнішні файли й сценарії, які використовуються на сторінках, і дозволяє вести пошук по типу файлів або виразу з шаблонами, якщо елемент знаходиться у віддаленому сховищі. Це дуже корисний інструмент, якщо потрібно обробити багато файлів.

2.4. Загальні відомості про HTTP сервер Apache

Apache HTTP – відкритий веб-сервер для UNIX-подібних, Microsoft Windows, NovellNetWare та інших операційних систем. На сьогодні є найуживанішим веб-сервером мережі Інтернет.

Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation.

Сервер Apache створений співтовариством незалежних розробників «ApacheGroup», члени якої у свій час брали участь у проєкті з побудови перших Web-серверів у NCSA (National Center for Supercomputer Applications, USA). «Apache Group» пропонує Web-сервери, сумісні з будь-якою UNIX-системою, встановленої на будь-якій апаратній платформі.

Web-сервер Apache, як і всі інші Web-сервери, базується на ідеях і частині коду, реалізованих у першому по-справжньому популярному Web-Сервері– NCSA httpd 1.3.

Версія 2 веб-сервера Apache була істотним переписом великої частини коду програми версії 1.x, з сильним нахилом на подальшу модульність та портативність. Версія 2.2 має гнучкіший API авторизації. Вона також включає поліпшені модулі кешу й проксі сервера.

Web-сервер Apache є самостійним, некомерційним, вільно розповсюджуваним продуктом. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Вони різняться від серверної підтримки мов програмування до схем аутентифікації. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl PHP.

Популярні методи архівації на Apache включають зовнішній модуль `mod_gzip`, створений для зменшення розміру веб-сторінок, переданих по HTTP.

Функції віртуального хостингу дозволяють одній інсталяції Apache обслуговувати різні веб-сайти. Наприклад, одна машина, з однією інсталяцією Apache може одночасно містити `www.example.com`, `www.test.com`, `test47.test-server.test.com` і так далі.

Apache передусім використовується для передачі через HTTP статичних та динамічних веб-сторінок у всесвітній павутині. Багато веб-застосунків спроектовано, зважаючи на середовище і можливості, які надає цей веб-сервер.

Продукт може працювати в якості кешувального проксі-сервера, що дозволяє істотно підвищити продуктивність роботи користувачів локальної мережі при роботі з документами, розташованими в Інтернет. Можна задавати такі параметри і налаштування проксі-сервера:

- типи файлів, які необхідно кешувати або навпаки, не включати в кеш;
- максимальний обсяг дискового простору, відведений під кеш;
- періодичний перегляд і індексування бази даних кеша з метою вивільнення дискового простору шляхом видалення застарілих.

Apache зіграв ключову роль у початковому зростанні всесвітньої павутини, і продовжує бути найпопулярнішим у світі веб-сервером, де-факто платформою, на яку орієнтуються інші веб-сервери.

Відповідно до статистики Netcraft за червень 2010 року, Apache є найпоширенішим серверним програмним забезпеченням в Мережі: на цей веб-сервер припадала частка близько 49 % відповідного сегменту ринку (майже 85 мільйонів сайтів). Друге місце за популярністю займають програмні платформи Microsoft – 35,4 % (61 мільйон сайтів).

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache

Software Foundation, без участі сторонніх програмістів, на мові програмування C.

Теоретично, ядро Apache може функціонувати в чистому вигляді, без використання модулів. Однак, функціональність такого рішення вкрай обмежена.

2.5. Конфігурування HTTP серверу Apache

Apache установлює свої файли в наступні каталоги:

1. `/etc/httpd/conf` – цей каталог містить всі файли конфігурації сервера;
2. `/etc/red/` – у цьому каталозі знаходяться інші каталоги, які містять сценарії запуску системи. Apache установлює їхній повний набір для Web-сервера. Ці сценарії можуть бути використані для запуску й зупинки сервера у ручному режимі, а також будуть автоматично запускати й зупиняти сервер, коли робоча станція зупинена, запускається або перезавантажується;
3. `/home/httpd` – у цей каталог установлює стандартний набір піктограм для сервера й CGI-сценаріїв;
4. `/usr/doc` й `/usr/man` – у цих каталогах містяться керівництво `manual pages` і файли `read me`. Більшість пакетів розміщає файли `readme` й іншу аналогічну документацію в спеціально названому для версії сервера каталозі `/usr/doc`;
5. `/usr/sbin` – у цей каталог розміщено всі виконувані програми.

Apache зчитує конфігураційні установки із трьох файлів: `access.conf`, `httpd.conf` й `srn.conf`. Спочатку така організація була розроблена для підтримки зворотної сумісності із сервером NCSA. Файли конфігурації знаходяться у підкаталозі `conf` сервера. Резервні копії цих файлів також включено в програмний дистрибутив. Вони відповідно названі `access.conf-dist`, `httpd.conf-dist` й `srn.conf-dist`.

Налаштування роботи сервера відбувається за допомогою конфігураційних директив. Директиви – це спеціальні команди, що встановлюють деякі опції. Ними потрібно скористатися для включення таких необхідних опцій роботи сервера, як розміщення файлів, важливих для конфігурації й роботи сервера.

Директиви розміщуються по одній у рядку. Одні директиви встановлюють тільки значення імені файлу, інші дають можливість визначити кілька опцій. Деякі спеціальні директиви, названо секціями (section) і вони дуже схожі на теги HTML. Секції розміщено в кутових дужках. Після секції звичайно йде група директив, дія яких поширюється на зміст каталогу, відповідного до секції.

Всі секції закриваються спеціальним секційним виразом, що має вигляд закриття тегу HTML. Деякі із цих конструкцій знаходяться у файлах `conf/access.conf` й `conf/httpd.conf`. Спеціальні секційні вирази схожі на інші директиви й визначаються по одному в рядку.

Файл `httpd.conf` містить конфігураційні директиви, що управляють роботою сервера та встановлюють місце розташування файлів реєстрації, ID(UUID) користувача, під яким він працює, номер використовуваного порту й багато чого іншого. Необхідно відредагувати значення початкових установок, що важливі для конфігурації сайту. Зазвичай залишають без зміни більшість установок, за винятком перерахованих нижче.

Директиву `ServerAdmin` варто настроїти на адресу Web-майстра, що керує сервером. Це повинна бути діюча адреса e-mail або псевдонім, наприклад, `webmaster@Ваш.домен`.

Директиви `User` й `Group` встановлюють UID і груповий ID(GID), які будуть використані сервером при формуванні запитів. Краще застосовувати параметри, установлені по замовчуванню, тобто `nobody` та `nogroup`. Імена `nobody` й `nogroup` існують відповідно у файлах `/etc/passwd` й `/etc/group`. Повноваження для цих UID й GID повинні бути обмежені, тому що у

випадку виникнення «дір системного захисту», ці програми будуть продовжувати виконуватися із призначеним UID. Якщо сервер запущений у режимі супер-користувача, хтось може скористатися виниклими «дірами» в системному захисті.

ServerName встановлює ім'я вузла, яке буде повертати сервер. Якщо це значення не визначене, сервер спробує визначити його сам й установить директиву ServerName з використанням імені в канонічному виді. Директива ServerName повинна бути дійсним ім'ям, визначеним службою DomainNameSystem (DNS) для мережі. ServerRoot. Ця директива встановлює абсолютний шлях до каталогу сервера. Вона вказує серверу, де шукати всі файли ресурсів і конфігурації. Більшість цих ресурсів визначено у файлі конфігурації, що відносяться до каталогу ServerRoot.

Файл srm.conf – це файл конфігурації ресурсів. Його керуючі установки віднесено до місця розташування дерева Web-документів, до каталогів CGI-програм і до задач конфігурації інших ресурсів, які мають відношення до Web-сайту. Зазвичай всі установки у файлі srm.conf залишають за замовчуванням. Найбільш важливими директивами в цьому файлі конфігурації є наступні:

- DocumentRoot – встановлює абсолютний шлях до дерева документів. Дерево документів – це головний каталог, з якого Apache буде обслуговувати файли. За умовчуванням він установлений як /home/httpd/html;

- UserDir – визначає вихідний каталог локального користувача, де він буде поміщати загальні HTML-документи. Тобто, кожен користувач буде мати власний HTML-каталог. Стандартне значення для цієї директиви – public_html, так що кожен користувач може створити каталог з ім'ям public_html у вихідному каталозі, і HTML-документи, розміщені в цьому каталозі, будуть доступні за адресою http:// ім'я_сервера/~username, де username – зареєстроване ім'я звичайного користувача.

Глобальний файл керування доступом – `access.conf`. У ньому задається тип доступу користувачів до сайту й документів, якими наповнюється його, визначаючи рівень системного захисту й іноді дозволяючи іншим користувачам змінювати деякі параметри системного захисту. За замовчуванням система встановлює необмежений доступ до документів у `DocumentRoot`.

Опції, які встановлюються в глобальних файлах конфігурації, можуть бути скасовані за допомогою файлу `.htaccess`. Файл `.htaccess` дає можливість установити директиви сервера на кожен каталог. Ця можливість зручна для каталогів, звідки користувач не повинен мати доступу до головних файлів конфігурації сервера. За умовчуванням ця директива дозволяє всі можливості.

WEB-сервер Apache зазвичай працює або в автономному режимі, або в режимі `daemon`. Робота сервера залежить від значення директиви `ServerType` у файлі `conf/httpd.conf`. Автономний сервер краще справляється зі своєю роботою, ніж сервери, на яких виконується `inetd`, тому що його процес зазвичай завжди готовий обслужити запит.

2.6. Висновки до другого розділу

В другому розділі проведено аналіз основних технологій, які використовувалися в для розробки інформаційної системи. Обґрунтовано використання мови програмування РНР, що дозволить досягнути гнучкості в роботі та швидкодії. Використано розширювану мову розмітки, що спростила процес передачі інформації. Використання редактора Adobe Dreamweaver CS 5.5. оптимізувало процес розробки інформаційної системи. Описано властивості та способи конфігурування HTTP серверу Apache.

РОЗДІЛ 3

ПРОЕКТУВАННЯ СИСТЕМИ

3.1. Розробка алгоритму систем

При розробці алгоритму необхідно врахувати, що розроблювана система повинна приймати вхідні дані про користувача і дані по кількостях оціночних коментарів з так званого зовнішнього середовища. Система ідентифікує користувача і у разі успішної ідентифікації опрацьовує вхідні дані піднімаючись по структурній ієрархії – починаючи з коментарів (сентименти: позитивні, негативні, нейтральні), переходячи до характеристик («Reliability», «Usability», «Efficiency», «Portability», «Maintainability» та «Functionality»), закінчуючи об'єктами. В системі реалізовано можливість роботи з більш, ніж двома об'єктами одночасно. Після проведення обчислень, система поітераційно виводить вихідні дані. Алгоритм роботи системи зображений у вигляді блок-схеми на рисунку 3.1.

Окремої уваги потребують алгоритми ідентифікації користувачів та обрахування вагового коефіцієнта.

При підключенні користувача до сервера для надсилання і отримання даних потрібно пройти процедуру реєстрації і ідентифікації.

Процес реєстрації проходять користувачі для присвоєння їм певного номеру (ідентифікатора) і надання їм відповідних прав доступу.

Ідентифікація – процес розпізнавання користувача в системі, як правило, за допомогою наперед визначеного ідентифікатора або іншої інформації про нього, яку може сприймати система, з метою обліку і контролювання дій користувача в системі.

Процес реєстрації можна пройти на веб-сторінці заповнивши спеціальні поля вводу і відправивши дані на сервер. Заповнена форма розглядається адміністратором і у випадку правильно заповнення,

адміністратор генерує спеціальний ключ, який використовується для ідентифікації. Також генерується спеціальний секретний ключ який використовується для створення підпису.

Отриманий ідентифікатор потрібно використовувати при надсиланні даних до сервера API вказавши як один з параметрів запиту.

Підпис забезпечує захист даних від їх підміни під час надсилання.

Для створення підпису потрібно виконати наступні кроки.

1. Відсортувати всі параметри які передаються в алфавітному порядку.
2. Провести конкатенацію відсортованих параметрів і секретного ключа, який видається після реєстрації.
3. Створити хеш використовуючи хеш-функцію MD5. В якості параметра в хеш-функцію потрібно передати результат конкатенації з попереднього кроку.

Результатом виконання хеш-функції буде послідовність букв і цифр які відіграють роль підпису.

Сервер отримавши запит від користувача ідентифікує його за допомогою ідентифікатора який міститься в запиті і отримує секретний ключ який потрібний для перевірки підпису. Дальше на основі переданих параметрів і секретного ключа створюється підпис по кроках описаних вище. Наступним етапом є порівняння створеного підпису і підпису який надійшов в запиті. У випадку якщо підписи відрізняються сервер припиняє подальше виконання і видає відповідне повідомлення про помилку. В іншому випадку сервер продовжується виконання програми згідно даних в запиті.

Алгоритм ідентифікації показано у вигляді блок схемі на рисунку 3.2.

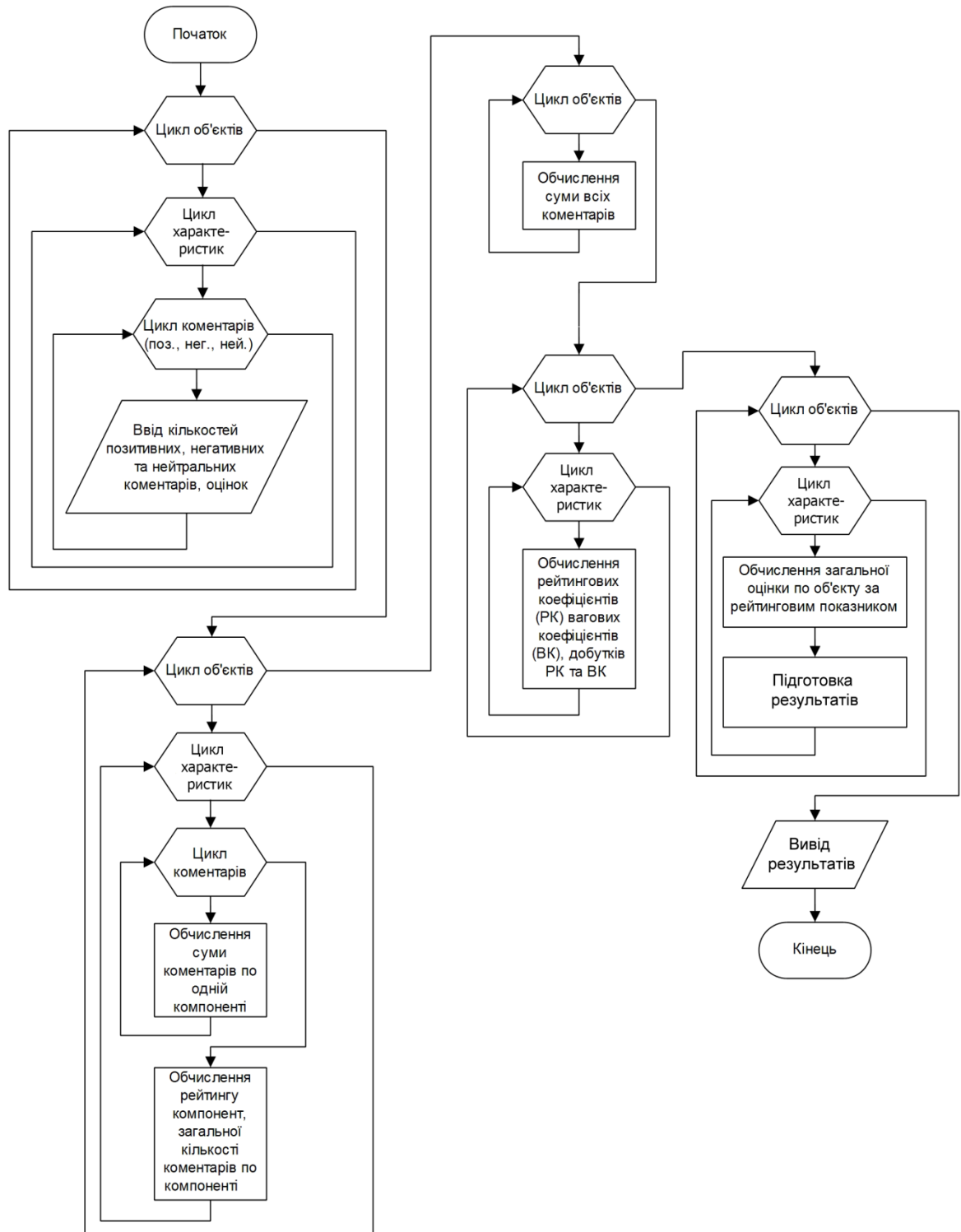


Рис. 3.1 Блок схема алгоритму роботи системи

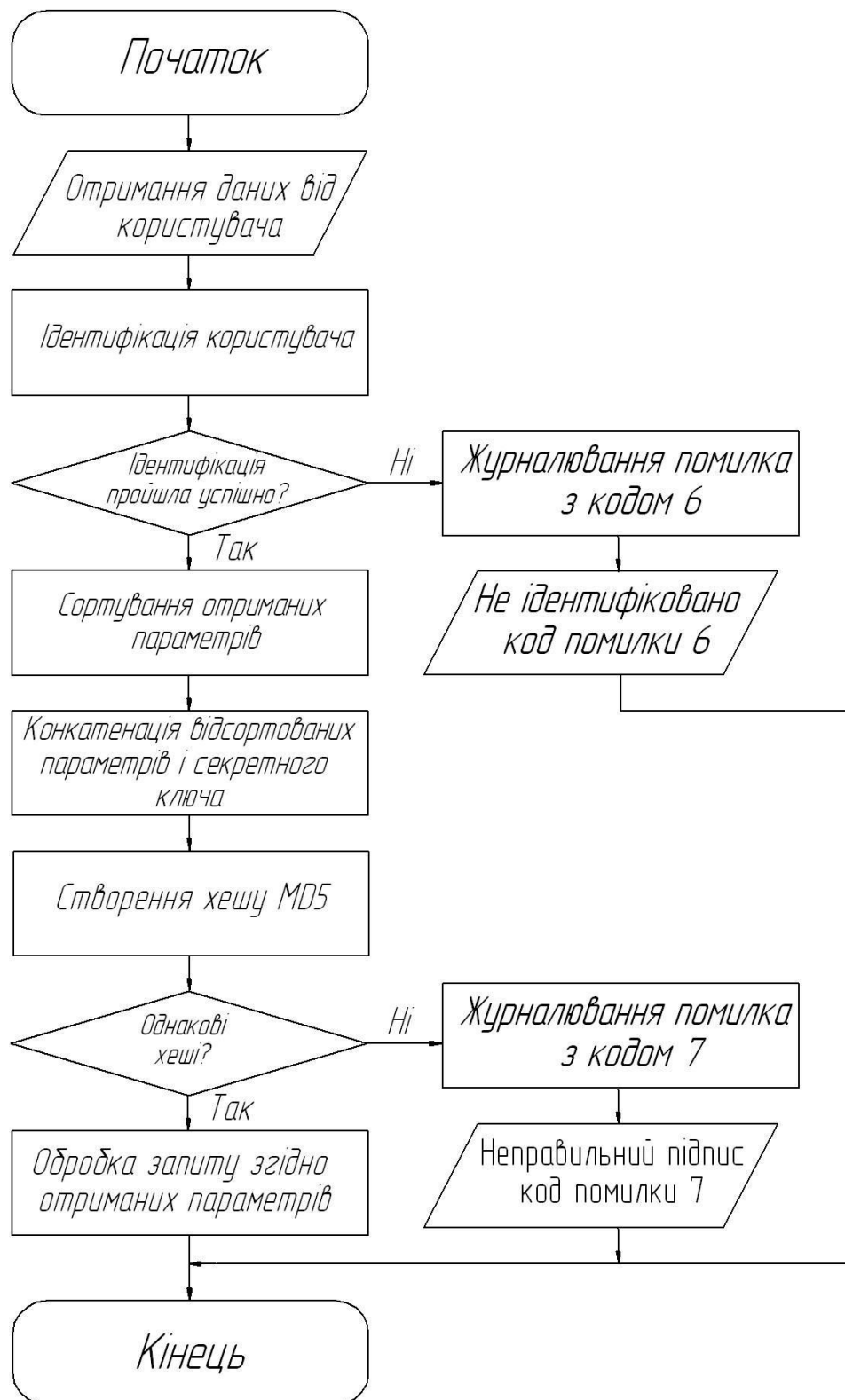


Рис. 3.2 Алгоритм ідентифікації користувача

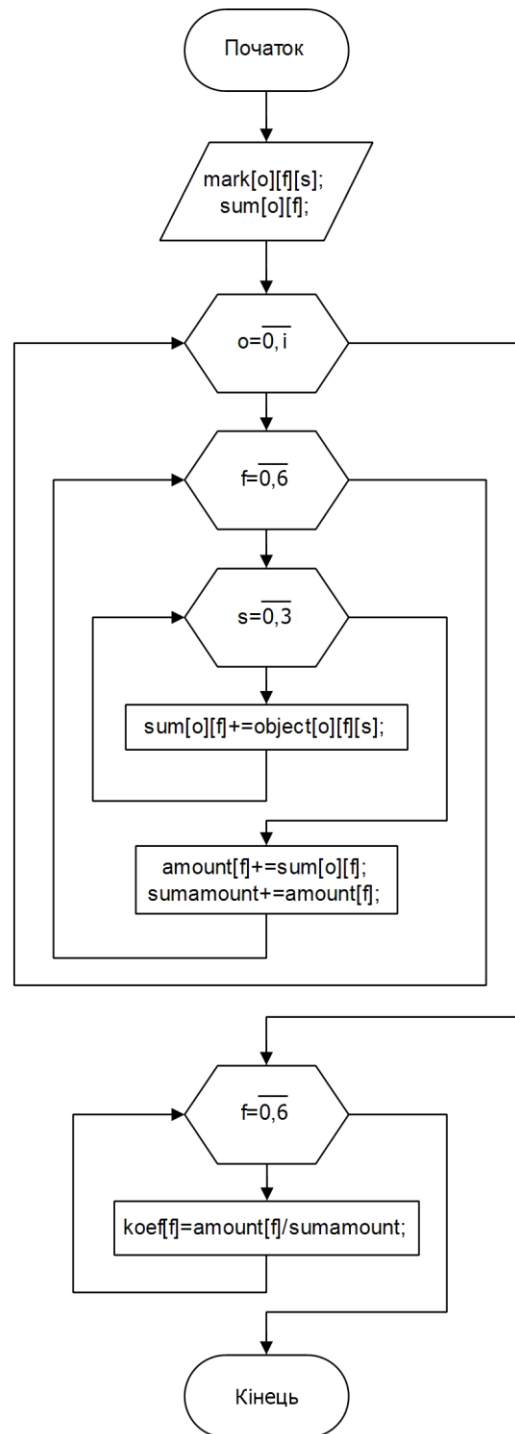


Рис. 3.3 Алгоритм знаходження вагового коефіцієнта

Зображена блок-схема зчитує отриману кількість коментарів по всіх об'єктах, характеристиках і сентиментах, після чого знаходиться кількість коментарів по кожній характеристиці кожного об'єкта, сума коментарів по кожній характеристиці всіх об'єктів, загальна кількість коментарів та ваговий коефіцієнт.

3.2. Розробка прикладного програмного забезпечення

Для опису функцій і роботи веб-сервісу використовується мова опису WSDL (Web Services Description Language).

Документ WSDL можна розбити на наступні логічні частини:

1. визначення типів даних (types) - визначення виду відправляються і одержуваних сервісом XML-повідомлень
2. елементи даних (message) - повідомлення, використовувані web-сервісом
3. абстрактні операції (portType) - список операцій, які можуть бути виконані з повідомленнями
4. зв'язування сервісів (binding) - спосіб, яким повідомлення буде доставлено.

Вміст wsdl файлу який використовується для запуску SOAP сервера наведено в додатку В.

Код програми який відповідає за ідентифікацію користувача і перевірку підпису наведено в лістингу 3.1

Лістинг 3.1 – Код ідентифікації користувача і перевірки підпису

```
function identifie($userID, $params, $signature){
    global $db;
    if($userID !== ''){
        $res = $db->query("SELECT * FROM users WHERE userID=".
(int)$userID);
        if($res->numRows() == 0){
            error_log(array('code'=>'6', 'msg'=>'user ID not
found '));
            throw new SoapFault("6", "userID not found");
            return false;
        }
        $userData = $db->fetch($res);
```

```

        //chech signature
        sort($params);
        $cocatent = '';
        foreach($prams as $value){
            $cocatent .= $value;
        }
        $cocatent .= $userData['secretKey'];
        $signature1 = MD5($cocatent);
        if($signature == $signature1){
            return true;
        }else{
            error_log(array('code'=>'7','msg'=>'Sign fail'));
            throw new SoapFault("7","Signature fail");
            return false;
        }
    }else{
        error_log(array('code'=>'6','msg'=>'Reuire userID'));
        throw new SoapFault("6","Reuire userID");
        return false;
    }
}

```

В лістингу 3.1 описана функція, яка приймає три параметри: ідентифікатор користувача, параметри запиту і підпис. В даній функції використовується глобальна змінна `$db` яка містить об'єкт для роботи з базою даних де містяться дані про зареєстрованих користувачів. Якщо ідентифікація і перевірка підпису пройшли успішно функція повертає логічне значення `true`. В іншому випадку функція викидає виключення `SoapFault` з текстом помилки.

Код обчислення суми коментарів по одній характеристиці окремих об'єктів, рейтингу характеристик, загальної кількості коментарів по одній характеристиці всіх об'єктів, загального рейтингу, суми всіх коментарів,

рейтингового відсотку, вагового коефіцієнта, добутку рейтингового відсотку та вагового коефіцієнта наведено у лістингу 3.2.

Лістинг 3.2 – Код основних обчислень

```
for($o=0;$o<$i;$o++) {
    for($f=0;$f<6;$f++) {
        for($s=0;$s<3;$s++) {
            $sum[$o][$f]+=$object[$o][$f][$s];
        }
        $rat[$o][$f]=$object[$o][$f][0]-$object[$o][$f][1];
        $amount[$f]+=$sum[$o][$f];
        $sumrat[$o]+=$rat[$o][$f];
    }
}
for($f=0;$f<6;$f++)
    $sumamount+=$amount[$f];
for($o=0;$o<$i;$o++) {
    for($f=0;$f<6;$f++) {
        $percent[$o][$f]=$rat[$o][$f]/($sumrat[$o]/100);
        $kcoef[$f]= number_format($amount[$f]/($sumamount/100),
2, '.', '');
        $pk[$o][$f]= number_format($percent[$o][$f]*$kcoef[$f], 2,
 '.', '');
        $nlvk[$o][$f]= number_format($nlv[$o][$f]*$kcoef[$f],2, '.',
 '');
    }
}
```

В лістингу 3.2 масив змінних `$sum[$o][$f]` – сума коментарів по кожній характеристиці кожного об’єкту, `$rat[$o][$f]` – рейтинг характеристик, `$amount[$f]` – кількість коментарів по кожній характеристиці всі об’єктів, `$sumrat[$o]` – сума рейтингів по кожному об’єкту, `$sumamount` – загальна сума коментарів, `$percent[$o][$f]` – рейтинговий коефіцієнт, `$kcoef[$f]` – ваговий коефіцієнт, `$pk[$o][$f]` – добуток рейтингового відсотку та вагового коефіцієнту. Коли всі обрахунки закінчені, відбувається виведення результатів. Результати виводяться в звичайному.

3.3. Тестування системи

Підчас тестування системи враховуються такі фактори:

- швидкість виконання PHP скрипта;
- правильність написання коду;
- тестування обчислення.

3.3.1. Тестування швидкості виконання

Швидкодія виконання PHP скрипта залежить від особливостей інтерпретування коду самою мовою програмування, кількості виконуваних функцій та операторів в скрипті, та від об'ємів даних, які обраховуються в програмному коді.

Для тестування швидкості виконання PHP скрипта можна використати простий клас, наведений у лістингу 3.3.

Лістинг 3.3 – Клас тестування швидкодії PHP скрипта

```
<?php
set_time_limit(0);
class timer
{
    private $start_time;

    privatefunctionget_time()
    {
        list($usec, $seconds) = explode(" ", microtime());
        return ((float)$usec + (float)$seconds);
    }
    functionstart_timer()
    {
        $this->start_time = $this->get_time();
    }
    functionend_timer()
    {
        return ($this->get_time() - $this->start_time);
    }
}
```

?>

Першою функцією є `set_time_limit(0)`, яка визначає максимальний час роботи нашого скрипта. Так як час обмежувати нам не потрібно, цій функції передається значення «0». В цьому класі є оголошення трьох методів класу:

- `get_time`;
- `start_timer`;
- `end_timer`.

Перший метод повертає значення поточного часу, другий – записує поточний час в змінну `$start_time`, а третій – повертає різницю між поточним часом та `$start_time`.

Тестування швидкодії коду відбуватиметься тричі, для порівняння швидкості з різними вхідними величинами.

Час, затрачений на виконання коду з першим набором вхідних даних становив 276 мікросекунд.

Час, затрачений на виконання коду з другим набором вхідних даних становив 298 мікросекунд, тобто різниця майже непомітна.

Час, затрачений на виконання коду з третім набором вхідних даних становив 418 мікросекунд.

Проаналізувавши результати тестів, можна зробити висновки, що час, потрібен на виконання коду з більшими об'ємами вхідних даних не сильно відрізняється від часу з меншими об'ємами вхідних даних, тобто швидкість виконання коду не є сильно залежною від вхідних даних. Швидкодія коду системи знаходиться на високому рівні, так як всі обчислення при максимальних вхідних даних тривають менше половини секунди.

Але в разі необхідності оптимізувати швидкодію PHP скрипта можна наступними способами:

- виносити «\$змінні» з «текстових стрічок» – прискорення 25-40%;
- короткі змінні, не більше 7 символів – прискорення 15%;

- уникнення використання доступу до елементів асоціативного масиву по імені не взятому в лапки – прискорення 30%;
- використання регулярних виразів відповідних до тексту – коли в тексті є спеціальні символи, необхідно використовувати PERL-орієнтований стандарт регулярних виразів. Коли ж в тексті відсутні спеціальні символи, то використання POSIX-орієнтованого стандарту ефективніше. Підбір неправильного стандарту регулярного виразу сповільнить частину алгоритму на 60-200%;
- використання циклів `for`, `while` ефективніше ніж `foreach` на 15-30% та використання функції `sizeof ()` швидше, ніж використання функції `count ()` на 15%;
- читання файлів за допомогою функції `file()` на 40% ефективніше, ніж з використанням функції `forep` в циклі.

Оптимізуючи код, потрібно пам'ятати про те, що його покращення не повинно впливати на його початкову функціональність. Як мінімум, після переробки додаток не повинен мати більше помилок на одиницю рядків коду, ніж до початку цього процесу.

3.3.2. Тестування правильності написання коду

Тестування коду на мові РНР для програмістів, які звикли до строгого коду на Java, Pascal, C++, де компілятор бере на себе контроль і виявляє багато типів помилок при компіляції, може бути проблематичним. РНР не потребує обов'язкової ініціалізації змінних, допускає досить вільний опис полів в класах та ін. Такі приклади свободи приводять до великої кількості помилок в коді, які не завжди просто виявити. Ряд помилок може проявлятися тільки при певних вхідних значеннях змінних. До того ж при еволюції програмного забезпечення часто стається так, що певні старі

функції працюють некоректно. Швидко виявити помилки таких типів дозволяють засоби автоматичного тестування, такі як PHP SimpleTest.

Проект SimpleTest розповсюджується як OpenSource і доступний для завантаження з SourceForge.net. Також деякі середовища розробки мають вбудований або розширюваний функціонал такого типу. Adobe Dreamweaver CS 5.5, який був використаний для написання додатку, перевіряє правильність написання коду «на ходу».

Результати тестування коду за допомогою PHPSimpleTest зображені на рисунку 3.4.

phpsimpletest

1/1 test cases complete: 4 passes, 0 fails and 0 exceptions.

Рис 3.4 Результати тестування правильності коду

Крім того, досягнути відсутності помилок в коді можна за допомогою вбудованих в додаток процедур по перевірці правильності роботи функцій програми. Якщо прийняти вміст функції за чорний ящик (який має визначену програмістом функціональність), то по реакції функції на набір вхідних значень можна судити про правильність її роботи. Даний процес відбувається шляхом аналізу відповідності масивів вхідних значень з вихідними. Це означає, що якщо в нас є масив вхідних даних, результат опрацювання програмою яких нам відомий, то перевірка коду не становить проблем – варто запустити цей тестуючий масив вхідних даних та порівняти результати роботи програми з тими результатами, які в нас є.

В разі передбаченої оптимізації та покращення коду в майбутньому, хорошим тоном буде створення «самотестуючого» коду з самого початку розробки програми, або певної функції. Після внесення певних змін в код, програмісту достатньо запустити тестуючий код, який видасть результати

порівняння еталонних даних з поточними. Для простоти реалізації найкраще використати log-файл, в який будуть заноситися результати тестування

Також не рекомендується виводити результати тестування безпосередньо в користувацький додаток, оскільки таким чином він може знайти розміщення скриптів на сервері, що недопустимо по нормах безпеки. Але потрібно пам'ятати про те, що тестування правильності коду не повинне впливати на загальну швидкодію програми.

3.3.3. Тестування обчислення

Тестування обчислення націлене на виявлення в програмному коді таких операцій, які призводять до неправильних результатів та помилок.

Перший тест передбачає введення більшої кількості позитивних, ніж негативних коментарів. Приклад результуючих обчислень наведено на рисунку 3.5. Другий тест полягає у введенні більшої кількості негативних, ніж позитивних коментарів. Результат обчислення з вхідними даними другого тесту зображено на рисунку 3.6

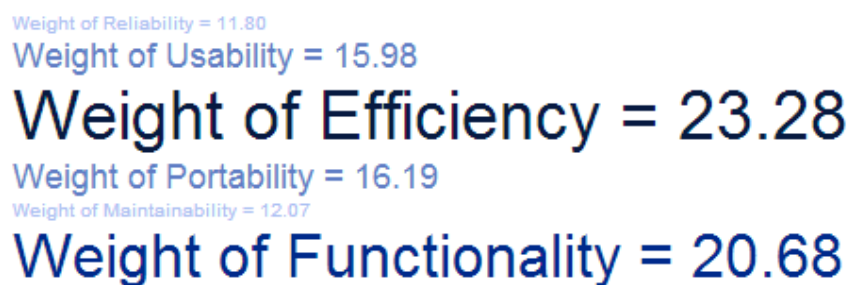


Рис. 3.5 Приклад результуючих обчислень

Як видно з результатів першого тесту, обрахування з більшістю позитивних коментарів відбувається правильно.

Weight of Reliability = 16.37
Weight of Usability = 13.21
Weight of Efficiency = 22.88
Weight of Portability = 21.17
Weight of Maintainability = 8.81
Weight of Functionality = 17.55

Рис. 3.6 Результат обрахування другого тесту

З результатів другого тесту можна зробити висновок, що підрахунок з більшістю негативних коментарів теж відбувається вірно.

3.4. Висновки до третього розділу

В третьому розділі розроблено алгоритм ідентифікації користувачів, обчислення вагових коефіцієнтів та повний алгоритми інформаційної системи. Написано код програми та протестованого його на швидкодію, стабільність та правильність виконання обчислень. Результуюча програма пройшла всі тести успішно. Запропоновано шляхи оптимізації програми для підвищення швидкодії виконання РНР скрипта.

РОЗДІЛ 4

ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Метою економічної частини дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності науково-дослідної роботи (НДР), і прийняття рішення про її подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки або робіт в даному напрямку.

Об'єктом розробки є програмний інтерфейс для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника опії об'єктів.

Розрахунку вартості НДР виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- визначити суму матеріальних затрат;
- обчислити витрати на електроенергію;
- розрахувати транспортні витрати;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності .

4.1. Розрахунок норм часу на виконання науково-дослідної роботи

Для визначення загальної тривалості проведення НДР доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

Таблиця 4.1

Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Постановка задачі та збір інформації	Керівник проекту	5
2	Розробка структурної схеми	Інженер	3
3	Написання програмного коду	Інженер	70
4	Тестування програми	Лаборант	5
5	Підготовка документації	Інженер	6
6	Від лагодження програми	Інженер	8
Разом		-	97

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_r, \quad (4.1)$$

де, T_c – тарифна ставка, грн.;

K_r – кількість відпрацьованих годин.

Тарифні ставки: керівник проекту – 12,0грн./год., інженер – 10,0 грн./год., лаборант – 5,0 грн./год.

Отже основна заробітна плата для:

керівника проекту $Z_{осн1} = 5 \cdot 12 = 60$ грн.

інженера $Z_{осн2} = 87 \cdot 10 = 870$ грн.

лаборанта $Z_{осн3} = 5 \cdot 5 = 25$ грн.

Сумарна основна заробітна плата становить $Z_{осн} = 60 + 870 + 25 = 955$ грн.

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$Z_{дод.} = Z_{осн.} \cdot K_{допл.}, \quad (4.2)$$

де, $K_{допл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту $Z_{дод2} = 60 \cdot 0,12 = 7,2$ грн.

Інженера $Z_{дод3} = 870 \cdot 0,12 = 104,4$ грн.

Лаборанта $Z_{дод4} = 25 \cdot 0,12 = 3$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{дод} = 7,2 + 104,4 + 3 = 114,6 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{о.п.}$) визначаються за формулою:

$$B_{о.п.} = Z_{осн.} + Z_{дод.}, \quad (4.3)$$

$$B_{о.п.} = 955,00 + 114,6 = 1069,6 \text{ грн}$$

Крім того, слід визначити відрахування на соціальні заходи:

- 1) фонд страхування на випадок безробіття – 1,6 %;
- 2) фонд по тимчасовій втраті працездатності – 1,4 %;
- 3) пенсійний фонд – 33,2 %;
- 4) Внески на страхування від нещасного випадку на виробництві та професійні захворювання – 1,4%.

У сумі зазначені відрахування становлять 37,6 %.

Отже, сума відрахувань на соціальні заходи буде становити:

$$B_{с.з.} = \Phi ОП \cdot 0,376, \quad (4.4)$$

$$B_{с.з.} = 1069,6 \cdot 0,376 = 402,17 \text{ грн.}$$

де, ФОП – фонд оплати праці, грн.

Проведені розрахунки витрат на оплату праці зведемо у наступну таблицю 4.2.

Таблиця 4.2

Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на оплату праці, грн. $6=3+4+5$
		Тарифна ставка, грн.	К-сть від- працьов. год.	Фактично нарах. з/пл., грн.			
<i>A</i>	<i>B</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
1	Кер.проекту	10	5	60	7,2	-	-
2	Інженер	10	87	870	104,4	-	-
3	Лаборант	3	5	25	3	-	-
Разом				955	114,60	402,17	1471,77

Отже, загальні витрати на оплату праці становлять 1471,77 грн.

4.3. Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{Bi} = q_i \cdot p_i, \quad (4.5)$$

де, q_i – кількість витраченого матеріалу i -го виду;

p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$Z_{м.в.} = \sum M_{Bi} \quad (4.6)$$

Проведені розрахунки занесемо у таблицю 4.3.

Таблиця 4.3

Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна 1- ці, грн.	Загальна сума витрат, грн.
1	CD диски	шт	2	2,00	5,00
2	Папір для друку	листів	50	0,10	5,00
Р а з о м					10,00

Отже, загальна сума матеріальних витрат на написання програми становить 10,00грн.

4.4. Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (4.7)$$

де, W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Час роботи ПК над даним проектом становить 97 годин, споживана потужність - 0,5 кВт/год.

$$З_e = 0,5 * 97 * 1,08 = 52,38 \text{ грн.}$$

4.5. Визначення транспортних затрат

Транспортні витрати слід прогнозувати у розмірі 8–10 % від загальної суми матеріальних затрат і сума затрат для перевезення працівників.

$$T_{\text{в}} = Z_{\text{м.в.}} \cdot 0,08 \dots 0,1 + T_n, \quad (4.8)$$

де, $T_{\text{в}}$ – транспортні витрати.

T_n – витрати на транспортування працівника;

Сума для перевезення працівників визначається за формулою:

$$T_n = D \cdot 4, \quad (4.9)$$

де D – кількість робочих днів.

Над проектом керівник працює 1 день, лаборант – 1, інженер – 11.

$$T_{n1} = 1 * 4 = 4 \text{ грн.}$$

$$T_{n2} = 1 * 4 = 4 \text{ грн.}$$

$$T_{n3} = 11 * 4 = 44 \text{ грн.}$$

Загальні витрати для транспортування всіх працівників становить:

$$T_n = 4 + 4 + 44 = 52 \text{ грн.}$$

$$\text{Отже, } T_{\text{в}} = 10,00 * 0,1 + 52 = 53,00 \text{ грн.}$$

4.6. Розрахунок суми амортизаційних відрахувань

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60% (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B \cdot H_A}{100\%} \quad (4.10)$$

де, A – амортизаційні відрахування за звітний період, грн.;

B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.;

H_A – норма амортизації, %.

Оскільки для написання програми та її тестування використовується один ПК вартістю 2560 грн, то квартальна сума амортизаційних відрахувань становитиме:

$$A_{пк} = \frac{2560 \cdot 15}{100} = 384 \text{ грн.}$$

Враховуючи, що ПК працює над даним проектом не весь квартал, а тільки 97 год, то $A_{пк} = (384/588) \cdot 97 = 63,35$ грн

4.7. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_v = B_{o.n.} \cdot 0,2 \dots 0,6 \quad (4.11)$$

де, H_v – накладні витрати.

$$H_v = 1069,6 \cdot 0,5 = 534,8 \text{ грн.}$$

4.8. Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у таблицю 4.4.

Таблиця 4.4

Кошторис витрат

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	1069,60	48,97
Відрахування на соціальні заходи	402,17	18,41
Матеріальні витрати	10	0,46
Витрати на електроенергію	52,38	2,40
Транспортні витрати	53	2,43
Амортизаційні відрахування	63,35	2,90
Накладні витрати	534,80	24,44
Собівартість	2185,30	100

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_e + T_e + A + H_e \quad (4.12)$$

Отже, собівартість дорівнює $C_B = 2185,30$ грн.

4.9. Розрахунок ціни НДР

Ціну робіт можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен.}) \cdot K + B_{i.n.}}{K} \cdot (1 + ПДВ) \quad (4.13)$$

де, $P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень, од. ;

$B_{i.n.}$ – вартість носія інформації, грн.;

ПДВ – ставка податку на додану вартість, (20 %).

$$Ц = 2185,30 \cdot (1 + 0,3) \cdot (1 + 0,2) = 3409,1 \text{ грн.}$$

4.10. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{\Pi}{C_v} \quad (4.14)$$

де, Π – прибуток;

C_v – собівартість.

$$\Pi = \text{Ц} - \text{С} \quad (4.15)$$

$$E_p = (3409,1 - 2185,3) / 2185,3 = 0,56$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p} \quad (4.16)$$

В нашому випадку $T_p = 1/0,56 = 1,79$, що є нормальним, оскільки допустимим вважається термін окупності до 5 років.

Всі дані внесемо в зведену таблицю 4.5 техніко-економічних показників

Таблиця 4.5

Техніко-економічні показники розробки сайту

№п/п	Показник	Значення
1.	Собівартість, грн.	2185,30 грн.
2.	Плановий прибуток, грн.	1223,21 грн.
3.	Ціна, грн.	3409,1 грн.
4.	Економічна ефективність	0,56
5.	Термін окупності, рік	1,79

4.11. Висновки до четвертого розділу

Проведено обчислення основних техніко економічних показників економічної ефективності. Загальна вартість пропонованих робіт по розробці програмного продукту становить 3409,1 грн. Оскільки ефективність становить 0,56, що є високим показником то проводити дані роботи варто і вкладені кошти окупляться за 1,79 року.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

Дотримання правил охорони праці є невід'ємною частиною в дослідженні методів організації API для інформаційної системи. В загальному, поняття охорона праці в інформаційній комп'ютерній системі являє собою дотримання всіх вимог і нормативів, що присутні в законодавчих актах про охорону праці. Закони цієї області спрямовані на якісну і безпечну експлуатацію робочих приладів і приміщень, дотримання санітарно-гігієнічних умов праці і захист від інших небезпечних чинників на підприємстві. Ці засоби є одним з складників для коректної роботи методів і засобів оцінювання якості інформаційної системи та забезпечення її якості роботи в цілому.

В основних законодавчих актах про охорону праці приділяється велика увага поліпшенню умов праці в усіх галузях господарства, впровадженню сучасних засобів техніки безпеки і забезпеченню санітарно - гігієнічних умов, що запобігають виробничому травматизму і професійним захворюванням.

Основні задачі охорони праці при використанні комп'ютерних і інформаційних технологій:

- 1) аналіз впливу факторів виробничого середовища на здоров'я і працездатність користувачів персональних комп'ютерів (ПК);
- 2) вдосконалювання методів оцінки працездатності і стану здоров'я користувачів ПК;
- 3) розробка і впровадження організаційно-технічних, гігієнічних і соціально-економічних заходів щодо раціоналізації виробничого середовища;

- 4) розробка і впровадження профілактичних і оздоровчих заходів, що дозволяють зберігати здоров'я людини і підвищувати її працездатність;
- 5) вдосконалення методик навчання користувачів ПК питанням охорони праці.

Охорона життя і здоров'я людини є пріоритетним напрямком соціальної політики держави. В Україні прийнято закон прямої дії «Про охорону праці», який регламентує захист конституційного права працівників на безпечні умови праці. Законодавство України про охорону праці складається із загальних законів України та спеціальних законодавчих актів. Загальними законами України, що визначають основні положення з охорони праці є Конституція України, Закон України «Про охорону праці», Кодекс законів про працю (КЗпП), Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності»

Крім загальних законів України, правові відносини у сфері охорони праці регулюються спеціальними законодавчими актами, указами і розпорядженнями Президента України, рішеннями уряду України, нормативними актами міністерств та інших центральних органів державної виконавчої влади. Сьогодні проводиться значна робота з питань вдосконалення законодавчої бази з питань охорони праці, адаптації українського законодавства з охорони праці до вимог Європейського Союзу.

Спеціальними законодавчими актами є державні нормативні акти з охорони праці (ДНАОП), що являють собою правила, стандарти, норми, положення, інструкції та інші документи, яким надано чинність правових норм, обов'язкових для виконання.

Вимогам нормативних актів з охорони праці мають відповідати:

- умови праці на кожному робочому місці;
- безпека технологічних процесів, машин, механізмів, обладнання й інших засобів виробництва;

- стан засобів колективного та індивідуального захисту;
- санітарно-побутові умови.

Для забезпечення безпеки ПК, периферійні пристрої та устаткування для обслуговування, ремонту і налагодження, електропроводи і кабелі за виконанням і ступенем захисту повинні відповідати класу зони за ПУЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ЕОМ, периферійних пристроїв і устаткування для обслуговування, ремонту та налагодження виконується як окрема групова трьохпровідна мережа, шляхом прокладки фазного, нульового робочого і нульового захисного провідників. Нульовий захисний провідник використовується для занулення (заземлення) електронних пристроїв. Використання нульового робочого провідника як нульового захисного провідника забороняється. Площа перетину нульового робочого і нульового захисного провідника в груповій трьохпровідній мережі повинна бути не менше площі перетину фазного провідника.

У приміщенні, де одночасно експлуатується чи обслуговується понад п'ять персональних ПК, на видному і доступному місці встановлюється аварійний резервний вимикач, що може цілком відключити електричне живлення приміщення крім освітлення. Штепсельні з'єднання та електророзетки, крім контактів фазного і нульового робочого провідників, повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх повинна бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазного і нульового робочого провідників.

Особливо неприпустимим є підключення ПК, периферійних пристроїв і устаткування для обслуговування, ремонту та налагодження до звичайної двопровідної електромережі.

При обробці факторів пожежної безпеки при розробці мережі, необхідно врахувати, що будинки і ті їхні частини, в яких розміщуються ПК,

повинні мати не нижче II ступеня вогнестійкості. Приміщення для обслуговування, ремонту і налагодження ЕОМ повинні відноситися за пожежо вибухонебезпекою до категорії В, а за класом приміщення - до Па.

Приміщення з ЕОМ повинні бути обладнані системою автоматичної пожежної сигналізації із димовими пожежними сповіщувачами з розрахунку 2 шт. на кожні 20 м² площі приміщення. В інших помешканнях допускається встановлювати теплові пожежні сповіщувачі.

Приміщення з ЕОМ мають бути оснащені переносними вуглекислотними вогнегасниками з розрахунку 1 на 50 м², але не менше 2 на приміщення. Підходи до засобів пожежогасіння повинні бути вільними.

Влаштованість робочих місць повинна забезпечувати належні умови освітлення приміщення і робочого місця, оптимальні параметри мікроклімату, ергономічних характеристик основних елементів робочого місця також враховувати наявність шуму і вібрації, м'якого рентгенівського випромінювання, електромагнітного випромінювання, ультрафіолетового та інфрачервоного випромінювання, електростатичного поля між екраном і оператором, відсутність пилюки, озону, оксидів азоту та аероіонізації.

Отже, в даному підрозділі розглянуто основні законодавчі акти і нормативи при використанні інформаційних технологій, приведено основні стандарти і правила щодо влаштування робочих місць і приміщень що використовують комп'ютерну техніку.

5.2. Безпека в надзвичайних ситуаціях

Метою дипломної роботи є дослідити методи організації АРІ для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника опії об'єктів. У дипломній роботі проаналізовано якісні характеристики роботи інформаційної системи. Окремо розглянута робота програмної частини системи і робота користувача з інтерфейсом за

показниками характеристик. При роботі інформаційної системи окрім багатьох технічних характеристик важливо враховувати і захищеність користувачів. Тому безпека в надзвичайних ситуаціях відіграє велику роль в інформаційних системах. Одним з шкідливих факторів, з яким може зіткнутися працівник системи це іонізуюче випромінювання.

Іонізуюче випромінювання — це будь-яке випромінювання, яке прямо або опосередковано викликає іонізацію навколишнього середовища (утворення позитивно та негативно заряджених іонів).

Іонізуюче випромінювання існує протягом всього періоду існування Землі, воно розповсюджується в космічному просторі. Природними джерелами іонізуючих випромінювань є космічні промені, а також радіоактивні речовини, які знаходяться в земній корі.

Штучними джерелами іонізуючих випромінювань є ядерні реактори, прискорювачі заряджених частинок, рентгенівські установки, штучні радіоактивні ізотопи, прилади засобів зв'язку високої напруги тощо. Як природні, так і штучні іонізуючі випромінювання можуть бути електромагнітними (фотонними або квантовими) і корпускулярними.

У результаті дії іонізуючого випромінювання на організм людини в тканинах можуть виникати складні фізичні, хімічні та біологічні процеси. При цьому порушується нормальне протікання біохімічних реакцій та обмін речовин в організмі.

В залежності від поглинутої дози випромінювання та індивідуальних особливостей організму викликані зміни можуть носити зворотний або незворотний характер. При незначних дозах опромінення уражені тканини відновлюються. Тривалий вплив доз, які перевищують гранично допустимі межі, може викликати незворотні зміни в окремих органах або в усьому організмі й виразитися в хронічній формі променевої хвороби. Віддаленими наслідками променевого ураження можуть бути променеві катаракти, злоякісні пухлини.

При вивченні дії на організм людини іонізуючого випромінювання були виявлені такі особливості:

- висока руйнівна ефективність поглинутої енергії іонізуючого випромінювання, навіть дуже мала його кількість може спричинити глибокі біологічні зміни в організмі;
- присутність прихованого періоду негативних змін в організмі, він може бути досить довгим при опроміненнях у малих дозах;
- малі дози можуть підсумовуватися чи накопичуватися;
- випромінювання може впливати не тільки на даний живий організм, а й на його нащадків (генетичний ефект);
- різні органи живого організму мають певну чутливість до опромінення. Найбільш чутливими є: кришталик ока, червоний кістковий мозок, щитовидна залоза, внутрішні (особливо кровотворні) органи, молочні залози, статеві органи;
- різні організми мають істотні відмінні особливості реакції на дози опромінення;
- ефект опромінення залежить від частоти впливу іонізуючого випромінювання. Одноразове опромінення у великій дозі спричиняє більш важкі наслідки, ніж розподілене у часі.

Очі людини уражаються при дозах 2...5 Гр (200...500 рад). Встановлено, що професійне опромінення із сумарною дозою 0,5...2 Гр (50...200 рад), отримане протягом 10-20 років, призводить до помутніння кришталика.

Основними документами, якими регламентується радіаційна безпека в Україні, є: Норми радіаційної безпеки України (НРБУ-97) та Основні санітарні правила України (ОСПУ).

У НРБУ-97 виділяють три категорії осіб щодо ризику іонізуючого опромінення:

- категорія А - персонал, який безпосередньо працює з радіо активними речовинами;
- категорія Б - персонал, що безпосередньо не працює із радіо активними речовинами, але за умови розміщення їх на робочих місцях або місцях проживання може потрапити під дію опромінення;
- категорія В - все населення країни.

Для осіб категорій А і Б НРБУ-97 встановлюють ліміти ефективної й еквівалентної доз за календарний рік. Обмеження опромінення категорії В (населення) здійснюється введенням лімітів річної ефективної та еквівалентної доз для критичних груп осіб категорії Б. Остання означає, що значення річної дози опромінення осіб, що входять до критичної групи, не повинно перевищувати ліміту дози, встановленого для категорії В.

Таблиця 5.1

Ліміти доз сумарного внутрішнього і зовнішнього опромінення

Ліміти доз, мЗв·рік ⁻¹	Категорія опромінюваних осіб		
	А	Б	В
$ЛД_E$ (ліміт ефективної дози)	20	2	1
Ліміти еквівалентної дози:			
$ЛД_{lens}$ (для кришталика ока)	150	15	15
$ЛД_{skin}$ (для шкіри)	500	50	50
$ЛД_{extrim}$ (для кистей та стоп)	500	50	-

Для захисту від зовнішнього опромінювання, яке має місце при роботі із закритими джерелами випромінювання, основні зусилля необхідно направити на попередження переопромінення персоналу шляхом:

- збільшення відстані між джерелом випромінювання і людиною (захист відстанню);
- скорочення тривалості роботи в зоні випромінювання (захист часом);
- екранування джерела випромінювання (захист екранами).

Під закритими джерелами радіаційного випромінювання розуміють такі, які виключають можливість потрапляння радіоактивних речовин в навколишнє середовище.

Під внутрішнім опроміненням розуміють вплив на організм людини випромінювань радіоактивних речовин, що потрапляють всередину організму. На дверях приміщень, у яких проводиться робота з відкритими джерелами радіоактивного випромінювання, повинен знаходитися знак радіаційної небезпеки - на жовтому фоні три червоних пелюстки.

Висновки.

У цьому підрозділі було розглянуто питання наявності іонізуючого випромінювання як негативної характеристики інформаційної системи та захисту від нього. Наведено основні законодавчі документи які регламентують цивільну безпеку в Україні. Описано вплив випромінювання на організм людини та норми радіаційної безпеки. Наведено основні методи захисту від зовнішнього випромінювань. Захист від внутрішнього опромінення вимагає виключення безпосереднього контакту з радіоактивними речовинами у відкритому вигляді та попередження потрапляння їх у повітря робочого простору.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1. Актуальність охорони навколишнього середовища

Охорона довкілля та раціональне використання природних ресурсів є невід'ємною частиною процесу суспільного розвитку української держави, адже природні ресурси є основою життєдіяльності населення та економіки держави, тому забезпечення їх збереження, відтворення та невиснажливого використання є однією з основних передумов сталого соціально-економічного розвитку країни.

Сучасне екологічне становище України не може розглядатись без минулого нашої країни, без історії природокористування, без врахування важливої моделі: людина – виробництво – природа. Зміни, які відбуваються внаслідок людської діяльності, негативно впливають на довкілля, тому в сучасному світі надзвичайно важливого значення набула справа охорони навколишнього природного середовища.

Як свідчить досвід, проводити ефективну політику невиснажливого розвитку в державі досить важко, навіть за умов процвітаючої економіки. Тим складнішою виглядає ця проблема в Україні – молодій державі, яка переживає успадковану кризу і змушена одночасно вирішувати безліч проблем: політичних, економічних, соціальних, екологічних.

6.2. Статистичне оцінювання техногенних впливів

Процес незворотного перетворення людиною частин біосфери на техногенні об'єкти і території дістав назву техногенезу, а частина біосфери, штучно перетворена в результаті життєдіяльності людини і заповнена її

продуктами, називається технічною оболонкою біосфери(техносферою).

Основне завдання комплексної оцінки техногенного впливу:

- вивчення техногенних чинників забруднення довкілля;
- класифікація джерел забруднень;
- визначення джерел походження забруднень;
- всі проблемами людства, які породжують забруднення біосфери.

Техногенні чинники забруднення довкілля об'єднують у такі групи:

- атмосферні: хімічне, фізичне, механічне і теплове забруднення;
- водні: океани і моря, забруднення поверхневих і підземних вод;
- ґрунтові: хімічне, ерозійне забруднення, ущільнення, засолення, заболочення;
- геологічні: негативні екзогенні процеси - зсуви, підтоплення, обвали, абразії берегів;
- біотичні: деградації екосистем, збіднення біорізноманіття, мутації, зникнення лісів і пасовищ, біогенна акумуляція шкідливих речовин;
- комплексні.

Вплив техносфери на стан атмосфери:

- найбільший вплив на стан атмосфери чинять теплоенергетика, металургійна промисловість, підприємства хімічної та будівельної індустрії, автотранспорт, що викидають у повітря пил, важкі метали, вуглеводні, оксиди карбону, бенз(а)пірен та інші речовини;
- найбільший вплив на хімічний склад атмосферного повітря чинить спалювання кам'яного вугілля;
- найпотужнішим негативним техногенним чинником є енергетика.

Підприємства чорної металургії утворюються пил та оксид сірки, хімічна і

нафтохімічна промисловість продукують майже у два рази менше викидів при значно більшій різноманітності забруднюючих речовин;

- серед усіх видів транспорту автомобільний посідає перше місце за кількістю і різноманітністю забруднюючих речовин, а також за кількістю незворотних змін ландшафтів та інших негативних впливів на довкілля.

Будь-який вплив людини на природні екосистеми призводить до їх змін, які викликають позитивні чи негативні наслідки для економіки і для всього суспільства. При вирішенні сучасних екологічних проблем великого значення набуває комплексна оцінка регіональних екологічних проблем, яка базується на глибокому вивченні та врахуванні всіх природних і соціально-економічних умов і факторів регіонів. Суть такої оцінки полягає в дослідженні просторової структури історично складених природно-територіальних комплексів та проведенні на цій основі розділу території країни (районування) на природні зони (області), округи та райони.

Методичні підходи. Існують різні підходи до екологічної оцінки територій:

- економічна оцінка – відображає наслідки для економічної діяльності суспільства (економічне зростання, ефективне використання ресурсів та ін.);

- соціальна оцінка – виявляє наслідки для соціального життя суспільства (рівень життя, культурне середовище, екологічна освіта);

- природно-ресурсна оцінка – виявляє наслідки для природних екосистем (зниження біорізноманіття, антропогенний вплив на живі організми).

Моделі, що використовуються при комплексних оцінках. Для отримання екологічних оцінок застосовуються різні моделі:

- блокові моделі системи «населення – господарство – природні системи» розробляються у вигляді загальної (базової) і часткової(територіальної);

- матричні моделі будуються у вигляді прямих і зворотних матриць;
- картографічні моделі створюються для визначення територіальних зв'язків між впливами і наслідками;
- статистичні моделі відображають кількісні залежності між впливами – змінами – наслідками.

6.3. Статистичний аналіз екологічності виробництва

Загальна класифікація еколого-економічних показників з метою оцінки й аналізу екологічності виробництва (ЕВ) у промисловості може бути подана за такими ознаками: за змістом, за рівнем визначення, за часовим інтервалом, за об'єктом оцінки, за характером використання:

1. За змістом еколого-економічних показників:

- натуральні – показники екологічності (екобезпечності) технологічних процесів, техніки (включаючи природоохоронне устаткування), виробничо-господарської діяльності в цілому та її окремих складових. Наприклад, ступінь очищення промислових викидів, комплексність використання сировини (ресурсна ефективність виробництва), обсяги викидів (скидів) шкідливих речовин у навколишнє середовище;
- натурально-вартісні – еколого-економічний збиток у розрахунку на одиницю товарної продукції в натуральному вираженні, збиткоємність маси викиду (скиду), екологічний результат у розрахунку на одну гривню капітальних вкладень;
- вартісні – розмір економічного збитку в розрахунку на одиницю продукції у вартісному вираженні, повні екологічні витрати виробництва, екологічні платежі за забруднення довкілля;
- локальні – показники вимірюють окремий параметр ЕЕРВ і можуть бути основою формування інтегральних показників, а також

використовуватися для аналізу впливу екологічних чинників (показників) на узагальнюючі результати виробничо-господарської діяльності;

- узагальнюючі – показники є головною, підсумковою і регулюючою оцінкою еколого-економічної ефективності технологічних процесів, забезпеченості підприємства основними природоохоронними фондами, рівня впливу виробництва на навколишнє природне середовище.

2. За часовим інтервалом: ретроспективні, поточні, фактичні, оперативні, прогнозні, планові.

3. За об'єктом оцінки:

- виробництво в цілому, окремі етапи відтворювальних процесів (виробничо-технологічні, переробні, організаційні, природоохоронні, ресурсозбережні);

- виробництво конкретних видів продукції (послуг);

- види (складові) виробничо-господарської діяльності підприємств (виробнича, інвестиційна).

4. За характером використання:

- регулюючі (дієві) – це показники, що безпосередньо застосовуються в процесі регулювання (управління) екологічності виробництва і якості навколишнього середовища, а також стану екосистем у процесі використання;

- індикаторні – показники, за допомогою яких може здійснюватися узагальнююча характеристика ЕВ у процесі аналізу;

- допоміжні показники забезпечують розрахунок комплексних, узагальнюючих еколого-економічних показників; можуть відігравати допоміжну роль при прийнятті складних, управлінських рішень.

Еколого-економічний рівень промислового виробництва може бути представлений у вигляді системи показників, які формують основні складові екологічної діяльності підприємства у взаємозв'язку з виробничо-господарською діяльністю. Система показників еколого-економічного рівня

виробництва враховує основні аспекти природоохоронної діяльності підприємства (організаційно-технічний рівень, управління, фінансування і т.д.), а також ступінь впливу виробництва на навколишнє середовище у взаємозв'язку з кінцевими результатами.

Показники організаційно-технічного рівня екологічної діяльності підприємств (ЕД):

- екологічність, екобезпечність технологічного процесу;
- озброєність і забезпеченість природоохоронними фондами;
- організаційний рівень ЕД;
- рівень управління ЕД.

Виділення показників організаційно-технічного рівня екологічної діяльності підприємств у межах оцінки ЕВ в принципі дозволяє говорити про організаційно-еколого-економічний рівень господарювання.

Показники рівня фінансової забезпеченості і платоспроможності ЕД:

- фінансування ЕД підприємством і державою;
- платоспроможність платного природокористування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хомів Б. А. Застосування вагових коефіцієнтів при оцінюванні opinii текстової інформації у web-документах / Б.А. Хомів, С.А. Лупенко / Матеріали XV наукової конференції «Природничі науки та інформаційні технології», Тернопіль, 14 – 15 грудня 2011 р. – Тернопіль: ТНТУ, 2011 – С. 95.
2. R. Quirk. A Comprehensive Grammar of the English Language, 2nd Revised edition : / Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, Jan Svartvik. – Longman, New York, 1989. – 1779 p.
3. J. Carbonell, Subjective Understanding: Computer Models of Belief Systems. PhD thesis, Carbonell Jaime Guillermo, 1979. – 300 p.
4. Y. Wilks and J. Bien, "Beliefs, points of view and multiple environments," in Proceedings of the international NATO symposium on artificial and human intelligence, USA, New York, NY: Elsevier North-Holland, Inc., 1984. – pp. 147-171.
5. Хомів Б. А. Формування інтегральної оцінки об'єкта у задачах OPINIONMINING / Б.А. Хомів, С.А. Лупенко / Матеріали XIX міжнародної конференції «Проблеми прийняття рішень в умовах невизначеності», Мукачєво, 23 – 27 квітня 2012 р. – Київ: Київський національний університет імені Тараса Шевченка, 2012 – С. 224-225.
6. Моделирование емоційного забарвлення текстової інформації : (Матеріали 14 міжнародної конференції «Системний аналіз та інформаційні технології», 24 квітня 2012 р.) [Електронний ресурс] / Б.А. Хомів, С.А. Лупенко // Інститут прикладного системного аналізу Національного технічного університету України «Київського політехнічного інституту». – С. 209-210. – Режим доступу. : <http://sait.kpi/books/sait2012.ebook.pdf>

7. Хомів Б. А. Програмна реалізація методу обчислення вагових коефіцієнтів та інтегральної оцінки об'єктів в текстових даних / Б.А. Хомів, С.А. Лупенко / Матеріали II-ї науково-технічної конференції «Інформаційні моделі, системи та технології», Тернопіль, 25 квітня 2012 р. – Тернопіль: ТНТУ, 2012 – С. 42.

8. Хомів Б. А. Математичне моделювання опінії та застосування лінгвістичних змінних в задачах opinion mining /Б.А. Хомів, С.А. Лупенко, О. А. Пастух / Матеріали I-ї міжнародної наукової конференції «Інформація, комунікація, суспільство», Львів, 25-28 квітня 2012 р. – Львів: НУ ЛП, 2012 – С. 210-211.

9. Хомів Б. А. Компаративний аналіз математичних моделей, методів та засобів оцінювання опінії в текстових даних інтернет-ресурсів / Б. А. Хомів, С. А. Лупенко, А. С. Сверстюк // Вісник Хмельницького національного університету. – 2011. – № 6. – С. 7-16.

10. Оптимизация программ на PHP // PHP по-русски– Режим доступа:<http://php.spb.ru/php/speed.html>. – Дата останнього доступу 12.11.2013.

11. Bing L. Web Data Mining. Exploring Hyperlinks, Contents and Usage Data, Second Edition : / Liu Bing. – Springer, 2011. – 622 p.

12. A. Kay. Object oriented programming, 2nd Revised edition – Longman, New York, 1992. – 36 p.

13. Тестирование кода на PHP // Фрилансер – Удаленная работа, советы фрилансерам, работа на дому – Режим доступа: <http://www.freelancer.ua/unit-тестирование-php/2006/11/02/>. – Дата останнього доступу 10.11.2013.

14. J.Bollen; H.Maoand X.-J.Zeng «Twitter mood predicts the stock market». – 2010. – 8 p.

15. Измерение скорости работы скрипта // Портал для программистов: здесь есть все для программирования – Режим доступа: <http://devoid.com.ua/php/php-beginners/izmerenie-skorosti-raboti-scripta-v-php.html>.

16. Визначення витрат на створення ПЗ автоматизованих систем / П. Андон, В. Суслов, Т. Коротун, Г. Коваль, О. Слабоспицька – Проблеми програмування. – 1998. - №3. – С. 23 – 34.

17. Конверс Т. PHP 5 и MySQL. Библия пользователя / Т. Конверс, Д. Парк, К. Морган – Изд-во «Вильямс» – 2009 – 1216 с.

18. Закон України «Про цивільну оборону» [Електронний ресурс]. – Режим доступа: URL: <http://zakon0.rada.gov.ua/laws/show/2974-12> – Назва з екрану

19. Оцінка умов праці [Електронний ресурс]. – Режим доступа: URL: <http://studentbooks.com.ua/content/view/1332/76/1/9/> – Назва з екрану

20. Тарасова В.В. Екологічна статистика / В.В Тарасова – К.: «Центр учбової літератури», 2008. – 247с.

ВИСНОВКИ

В дипломній роботі узагальнено та вирішено задачу створення API для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об'єктів. В результаті дослідження методів організації API запропоновано використання протоколу SOAP, що дало змогу розширити функціональні можливості системи. При цьому отримані такі результати:

1. Проаналізовано предметну область галузі Opinion Mining.
2. Здійснено аналіз існуючих систем для видобування та оцінювання opinii.
3. Обґрунтовано використання вагових коефіцієнтів при обчисленні інтегрального показника opinii об'єктів, що дозволило підвищити точність аналізу вхідних даних.
4. Проведено аналіз методів організації API та основних технологій які використовувалися для розробки інформаційної системи.
5. Обґрунтовано використання мови програмування PHP, що дозволить досягнути гнучкості в розробці та підвищити швидкодію.
6. Використано розширювану мову розмітки, що спростила процес передачі інформації та редактор Adobe Dreamweaver CS 5.5, що оптимізувало процес розробки інформаційної системи.
7. Описано властивості та способи конфігурування HTTP серверу Apache.
8. Розроблено алгоритми ідентифікації користувачів, обчислення вагових коефіцієнтів та повний алгоритм інформаційної системи.
9. Написано код програми, а також проведено тестування на швидкодію, стабільність та правильність виконання обчислень.
10. Запропоновано шляхи оптимізації програми для підвищення швидкодії виконання PHP скрипта.

11. Обґрунтовано економічну ефективність розробки та розраховано вартість програмного продукту.

12. Проаналізовано законодавчі акти щодо охорони праці в приміщеннях, де експлуатується інформаційна система і досліджено вплив іонізуючого випромінювання на організм людини і способи боротьби з ним.

13. Досліджено статистичні аналізи екологічності виробництва та оцінювання техногенних впливів, а також обґрунтовано актуальність охорони навколишнього середовища.

АНОТАЦІЯ

Тема дипломної роботи: “Дослідження методів організації API для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об’єктів”. // Дипломна робота // Забурко Микола Петрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2013 // с. – 100 , рис. – 12 , табл. – 14, аркушів А1 – 9 , додат. – 4 , бібліогр. – 20.

Ключові слова: OPINION MINING, ОПІНІЯ, ВАГОВИЙ КОЕФІЦІЄНТ, API, ІНФОРМАЦІЙНА СИСТЕМА, SOFTWARE

Метою роботи є обґрунтування методу організації та створення API для інформаційної системи обчислення вагових коефіцієнтів та інтегрального показника opinii об’єктів.

Обґрунтовано використання в роботі вагових коефіцієнтів при обчисленні інтегрального показника opinii об’єктів, що дозволило підвищити точність аналізу вхідних даних.

Розроблена інформаційна система дає змогу оцінити opinii за кількістю позитивних, негативних та нейтральних коментарів по відношенню до компонент об’єктів, й в майбутньому покращити характеристики об’єктів.

ANOTATION

Subject : "Investigation API methods for information systems computing weight coefficients and integral index for objects opinion". / / Diploma work / / Zaburko Mykola / / Ternopil State Technical University named after Ivan Puluj , department of computer information systems and software engineering , group CIm-61 / / Ternopil , 2013 / / p. - 100, fig. - 12 , tab. -14, -9 A1 sheets , add. - 4 ref. - 20.

Keywords : OPINION MINING, OPINIYA , WEIGHT COEFFICIENTS, API, INFORMATION SYSTEM , SOFTWARE.

The aim is to reasoning the organization method of creating an information system API for computing weight coefficients and integral index for objects opinion.

Proved of use in the application the weight coefficients when calculating the integral index objects opinion, thus improving the accuracy of the input data.

The developed information system makes it possible to evaluate opinion by the number of positive, negative and neutral comments related to the component objects, and in the future to improve the properties of objects.