

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

магістр


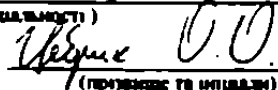

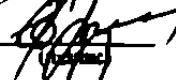

(освітній ступінь)

на тему: Методи та засоби побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину

Виконав: студент 6 курсу, групи СІм-61
спеціальності 123

"Комп'ютерна інженерія"

(шифр і назва спеціальності)

	 (підпис)	 (прізвище та ініціали)
Керівник	 (підпис)	<u>Луценко С.А.</u> (прізвище та ініціали)
Нормоконтроль	 (підпис)	<u>Пилип Е.В.</u> (прізвище та ініціали)
Рецензент	 (підпис)	<u>Скоробоган Н.А.</u> (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(назва набудовується після отримання диплома)

Факультет Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра Кафедра комп'ютерних систем та мереж
Освітній ступінь магістр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КС
Олександр П.М.
» 09 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Цибрик Олександр Олександрович
(прізвище, ім'я / по батькові)

1. Тема роботи Методи на засаді побудови спеціалізованого комп'ютерного системи для оптимізації якості бездротової зв'язі

Керівник роботи Душка Сергій Анатолійович Доктор технічних наук, професор КС
(прізвище, ім'я, по батькові, науковий ступінь, місце роботи)

Затверджені наказом по університету від «27» листопада 2019 року № 17854

2. Термін подання студентом роботи 26.12.2019р.

3. Вихідні дані до роботи Інформаційні системи, спеціалізовані комп'ютерні системи, моделі і методи побудови спеціалізованих комп'ютерних систем, моделі і методи аналізування інформаційних систем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз і проектування інформаційних систем.
2. Методи моделювання програмних систем.
3. Синтез автоматизованих систем для вирішення задачі оптимізаційної експертних систем.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Схема креслярської сторінки. Схема формування інформаційної моделі системи. Система IT вхід-вихід і обмінними даними.
2. Архітектура побудованої КС; таблиця моделі РК; Креслярська архітектура КСУ з описом лінійки даних; Креслярська архітектура КСУ з побудованими радіомодулями; Категоризація методів моделювання; Модель вхідного процесу моделювання; Інформаційна схема АСІЕ.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Організація роботи	Білик Н. Б.	<i>[Signature]</i>	<i>[Signature]</i>
Оцінює праці	Осипенко Г. М.	01.11.19	<i>[Signature]</i>
Безпека в НС	Стригун В. С., ст. вил. кадр ОК	<i>[Signature]</i>	<i>[Signature]</i>
Безпека	Лещенко В. М., доц.	<i>[Signature]</i>	<i>[Signature]</i>

7. Дата видачі завдання 30.09.2019

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Прізвище
1	Організація роботи	30.09.2019р.	Білик
2	Вибір теми	05.10.2019р.	Білик
3	Написання першого розділу	12.10.2019р.	Білик
4	Покращення і перевірка першого розділу	14.10.2019р.	Білик
5	Написання другого розділу	16.10.2019р.	Білик
6	Покращення другого розділу	18.10.2019р.	Білик
7	Написання третього розділу	20.10.2019р.	Білик
8	Покращення третього розділу	23.10.2019р.	Білик
9	Оформлення та безпека оформлення роботи	10.11.2019р.	Білик
10	Формат	12.11.2019р.	Білик
11	Оформлення і перевірка написаної роботи	16.11.2019р.	Білик
12	Оформлення та безпека оформлення роботи	19.11.2019р.	Білик
13	Оформлення та безпека оформлення роботи	15.12.2019р.	Білик
14	Підготовка до захисту	18.11.2019р.	Білик
15	Захист	26.12.2019р.	Білик

Студент

[Signature]
(підпис)Шарик О.О.
(підпис та ініціали)

Керівник роботи

[Signature]
(підпис)*[Signature]*
(підпис та ініціали)

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ І ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	13
1.1. Основні поняття інформаційних систем.....	13
1.2. Класифікація інформаційних систем.....	16
1.3. Теорія та методологія побудови систем.....	18
1.4. Комп'ютерні системи.....	26
1.5. Спеціалізовані комп'ютерні системи.....	29
1.6. Основи побудови спеціалізованих комп'ютерних систем.....	30
1.7. Глобальна модель, системні об'єкти, системні функції, функціональні об'єкти комп'ютерних систем.....	39
1.8. Класифікація архітектур та характеристики комп'ютерних систем.....	41
1.9. Висновки до розділу 1.....	43
РОЗДІЛ 2. МОДЕЛІ І МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНИХ СИСТЕМ..	44
2.1. Методи та засоби створення інформаційної системи	44
2.2. Аналіз якості та вимоги експлуатації бензину	47
2.3. Ідентифікаційна експертиза бензину	49
2.4. Застосування автоматизації при вирішенні задачі експертизи	50
2.5. Методи тестування програмних систем	52
2.6. Основні результати досліджень та методи інженерії тестування.....	56
2.7. Стратегії тестування, які базуються на аналізі ризику відмов	58
2.8. Моделі та методи вдосконалення процесу тестування.....	60
2.9. Висновки до розділу 2.....	61
РОЗДІЛ 3. СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ІДЕНТИФІКАЦІЙНОЇ ЕКСПЕРТИЗИ БЕНЗИНІВ.....	62
3.1. Структура автоматизованої системи ідентифікаційної експертизи бензинів.....	62
3.2. Структура та опис бази даних.....	65
3.3. Підсистема ідентифікації виробника палива.....	68

3.4. Архітектура модуля ідентифікаційної експертизи.....	70
3.5. Опис автоматизованої системи ідентифікаційної експертизи.....	72
3.6. Висновки до розділу 3.....	73
РОЗДІЛ 4. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....	74
4.1. Визначення стадій технологічного процесу та загальної тривалості проведення науково-дослідних робіт.....	74
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи.....	76
4.3. Розрахунок витрат на електроенергію.....	80
4.4. Розрахунок витрат на матеріали.....	80
4.5. Розрахунок суми амортизаційних відрахувань.....	81
4.6. Обчислення накладних витрат.....	82
4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт.....	83
4.8. Розрахунок ціни науково-дослідних робіт.....	84
4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень.....	84
РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	86
5.1. Організація охорони праці.....	86
5.2. Безпека в надзвичайних ситуаціях.....	88
5.2.1 Безпека життєдіяльності користувачів спеціалізованих комп'ютерних систем.....	88
5.2.2 Оцінка уразливості об'єкта, оснащеного спеціалізованою комп'ютерною системою від радіоактивного забруднення і проникаючої радіації.....	89
РОЗДІЛ 6. ЕКОЛОГІЯ.....	93
6.1. Добування електроенергії за рахунок спалювання мінерального палива. Забруднення довкілля при цьому та шляхи його зменшення.....	93

6.2. Робота з банками екологічної інформації.....	96
ВИСНОВКИ.....	99
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	101
Додаток А Тези конференцій.....	105
Додаток Б Код програми. Побудова нечіткої моделі.....	111
Додаток В Алгоритм нечіткого виведення.....	114

АНОТАЦІЯ

Методи та засоби побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину // Дипломна робота // Цебрик О.О. // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерних систем та мереж, група СІм-61 // Тернопіль, 2019 // с. – 117, рис. - 18, табл. – 16 , додат. – 3.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, СПЕЦІАЛІЗОВАНІ КОМП'ЮТЕРНІ СИСТЕМИ, ІДЕНТИФІКАЦІЙНА ЕКСПЕРТИЗА, НЕЧІТКА СИСТЕМА, НАФТОПРОДУКТИ, БЕНЗИН, C++, C#, XML, НЕЧІТКА МОДЕЛЬ.

Метою даної дипломної роботи є дослідження різних типів спеціалізованих комп'ютерних систем, а також в набутті вмінь та навичок практичного втілення математичного забезпечення в різних програмних середовищах для вирішення завдань.

Завдання дипломної роботи - дослідити різні спеціалізовані комп'ютерні системи для оцінювання якості бензину.

Предметом дослідження є методи та засоби побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину.

Дипломна робота складається з шести розділів.

Перший розділ містить огляд загальної теоретичної інформації, літературних джерел про інформаційні системи, спеціалізовані комп'ютерні системи.

В другому розділі здійснюється опис загальних підходів та методів досліджень побудови спеціалізованих комп'ютерних систем.

Третій розділ описує узагальнення результатів досліджень та досліджує різні типи математичного забезпечення комп'ютерних систем удосконалення інформаційної системи експертизи бензинів на основі нечіткої моделі даних.

В четвертому розділі проведено обґрунтування економічної ефективності, визначено стадії технологічного процесу та загальну тривалість проведення науково-дослідних робіт, обчислено витрати на оплату праці та відрахування на

соціальні заходи, проведено розрахунок витрат на електроенергію, на матеріали, суми амортизаційних відрахувань, обчислено накладні витрати, складено кошторис витрат та визначено собівартості науково-дослідних робіт розраховано ціни науково-дослідних робіт та визначено економічну ефективність і термін окупності капітальних вкладень.

П'ятий розділ присвячений питанням організація охорони праці та безпеки в надзвичайних ситуація.

В шостому розділі досліджено писання екології.

Об'єкт дослідження: процес контролю якості та ідентифікації зразків продукції.

ANNOTATION

Methods and means of construction of specialized computer systems for gasoline quality assessment // Diploma thesis // Tsebrik OO // Ivan Puliuyi Ternopil National Technical University, Faculty of Computer Systems and Networks, SIM-61 Group // Ternopil, 2019 // p. - 117, fig. - 18, tab. - 16, add. - 3.

Keywords: INFORMATION SYSTEM, SPECIALIZED COMPUTER SYSTEMS, IDENTIFICATION EXPERTISE, FUZZY SYSTEM, PETROLEUM PRODUCTS, GASOLINE, C ++, C #, XML, FUZZY MODEL.

The purpose of this thesis is to study the different types of specialized computer systems, as well as to acquire the skills and practical implementation of mathematical software in different software environments to solve problems.

The aim of the thesis is to explore various specialized computer systems for assessing the quality of gasoline.

The subject of the study is methods and tools for building specialized computer systems for gasoline quality assessment.

The thesis consists of six sections.

The first section provides an overview of general theoretical information, literature sources on information systems, specialized computer systems.

The second section describes the general approaches and research methods for building specialized computer systems.

The third section describes the synthesis of research findings and examines the different types of mathematical support for computer systems to improve the information system of gasoline expertise on the basis of a fuzzy data model.

In the fourth section, the economic efficiency substantiation was performed, the stages of the technological process and the total duration of the research works were determined, the costs of wages and deductions for social measures were calculated, the costs of electricity were calculated, materials, the amounts of depreciation, and the costs were calculated. cost estimate and cost of research, price of research work, cost effectiveness and payback period capital investments.

The fifth section focuses on the organization of occupational safety and health in emergencies.

Section 6 explores the writing of ecology.

Object of study: the process of quality control and identification of product samples.

ВСТУП

Актуальність теми. Третина бензину на українських автозаправних станціях є неякісною, про що свідчать дослідження Інституту споживчих експертиз. Заправляти авто неякісним бензином – це все одно, що постійно харчуватися у фаст-фудах. Для людини такий раціон може закінчитися виразкою та іншими недугами, а для машини – заміною двигуна. На українських дорогах все частіше їздять автомобілі відомих європейських автомобільних концернів, які випускають нові машини з урахуванням європейських вимог до якості бензину. Їздити на сучасних іномарках в Україні та заправлятися паливом, що відповідає лише старим стандартам, означає нищити мотор.

Питання визначення якості бензину виникали в споживачів та вивчалися аналітиками. Застосування спеціалізованих комп'ютерних систем для проведення робіт із збирання, накопичення, обробки, аналізу, прийняття рішень для визначення та з поліпшення якості засобами сучасних інформаційних технологій на території діяльності промислових підприємств незалежно від форм їх власності та організаційно-правової форми господарювання є пріоритетним напрямком діяльності. Дослідження в цій області сконцентровані на розробці та впровадженні спеціальних комп'ютерних програм, здатних імітувати, відтворювати ті області діяльності людини, які вимагають мислення, певної майстерності і накопиченого досвіду. До них відносяться задачі прийняття рішень, розпізнавання зразків палива та проведенні оцінювання якості різних марок бензину з застосуванням сучасних методів спеціалізованих комп'ютерних систем, які вже успішно застосовується в деяких областях техніки і життя суспільства – органічної хімії, пошуку корисних копалин, медичній діагностиці.

Дані системи повинні вирішувати задачі, що вимагають для свого рішення експертних знань у деякій конкретній області. У тій чи іншій формі експертні системи повинні мати ці знання. Такі системи використовують бази

даних (БД), або бази знань (БЗ). Система акумулює професійні знання керівників і фахівців, використовуючи їх для формування бази знань, яка містить набір взаємопов'язаних правил. При прийнятті рішень стає можливим аналіз наслідків різних рішень у вигляді питань «що буде, як-що...», не витрачаючи часу на трудомісткий процес програмування.

Мета і завдання дослідження. Метою дипломної роботи є дослідження методів та засобів побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину. Для досягнення мети в роботі необхідно розв'язати такі задачі:

- проаналізувати існуючі методи та інформаційні технології в галузі контролю якості продукції, у тому числі бензинів;
- розробити модель представлення даних для вирішення поставленої задачі;
- розробити методи налаштування та адаптації побудованої моделі;
- розробити автоматизовану систему ідентифікаційної експертизи (АСІЕ) на основі запропонованих моделей і методів;

Для досягнення поставленої мети необхідно було:

- проаналізувати параметри автомобільних бензинів та методи їх вимірювання;
- проаналізувати методичні та інструментальні похибки;
- розробити структури засобів ідентифікації автомобільних бензинів.

Метою моєї дипломної роботи є узагальнення знання в дослідженні різних типів математичного забезпечення спеціалізованих комп'ютерних систем оцінювання якості бензину, а також в набутті вмінь та навичок практичного втілення математичного забезпечення в різних програмних середовищах для вирішення певних прикладних завдань.

Завдання дипломної роботи - дослідити різні типи математичного забезпечення спеціалізованих комп'ютерних систем удосконалення інформаційної системи оцінювання якості бензину.

Об'єктом дослідження: є процес для оцінювання якості бензину з застосуванням спеціалізованих комп'ютерних систем.

Предметом дослідження: є методи та засоби побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину.

Методи дослідження. Для розв'язання поставлених задач використано теоретичні та експериментальні методи досліджень: аналізу даних, теорії розпізнавання образів з області обчислювального інтелекту, спеціалізовані комп'ютерні системи, теорії нечітких множин, об'єктно-орієнтованого програмування, теорії автоматичного регулювання, методах імітаційного моделювання на ЕОМ. Основні теоретичні результати перевірялись експериментальними методами.

Наукова новизна одержаних результатів. в основі отримання результатів дипломної роботи визначається тим, що розроблена й обґрунтована нечітка модель представлення та обробки даних для визначення якості бензину, що дозволило ефективно виконувати дослідження зразків продукції на основі процесів експертизи та накопичених знань експерта з застосуванням спеціалізованих комп'ютерних систем.

Практичне значення одержаних результатів. полягає в тому, що запропоновані в роботі моделі і методи дозволяють накопичувати і використовувати знання експерта для вирішення задач по визначенню якості бензину. Розроблено програмні засоби для проведення подальших досліджень з іншими предметними галузями і задачами розпізнавання образів в умовах невизначеності.

Публікації. Результати дослідження апробовано на VIII Міжнародної науково-технічної конференції молодих учених та студентів «Актуальні задачі сучасних технологій», Тернопіль, 2019.

VII науково-технічної конференції «Інформаційні моделі, системи та технології», Тернопіль, 2019.

Структура роботи. Дипломна робота складається із вступу, шести розділів та висновків, викладена на 117 сторінках друкованого тексту, містить 18 рисунків, 16 таблиць, 3 додатки та список використаних джерел.

РОЗДІЛ 1

АНАЛІЗ І ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1. Основні поняття інформаційних систем

Програмне забезпечення за період свого існування зазнало значних змін: від програм, які могли виконувати лише найпростіші логічні та операції з арифметичними даними до більш складних систем управління. Розглянемо основні напрями розвитку програмного забезпечення:

- виконання вирахувань;
- накопичення і обробка.

Спочатку створення комп'ютери мали велику вартість та використовувались силовими структурами переважно для виконання розрахунків пов'язаних з створенням ядерної зброї та ракетних технологій, а зараз вони використовуються у всіх сферах діяльності, а також приватними особами.

Сьогодні не можна навіть уявити функціонування підприємств та організацій без використання комп'ютерної техніки та використання різних систем та програм для виконання різних завдань.

Щоб вижити в конкурентній боротьбі сучасні компанії повинні вкладати свої грошові ресурси для розвитку у сфері новітніх інформаційних технологій, бо отримати перемогу може лише той, хто краще забезпечений та більш ефективно організований.

Інформаційні системи являються різноманітністю програм, але в залежності від предметної області вони відрізняються від програм і систем

своїми функціями, архітектурою та способом реалізації. Але існують властивості, які є загальними для всіх:

- вони використовуються для збору, зберігання та оброблення інформації через наявність ресурсів для відображення збереження та доступу до даних;

- так як системи мають використовувати користувачі, які не мають достатнього рівня інформаційної грамотності, інтерфейс для клієнта повинен бути легким у використанні, простими, зручними, з легко засвоюваним інтерфейсом, який надає користувачу доступ до всіх необхідних завдань та функцій, але має обмежувати використання будь яких дій непередбачених даним робочим місцем.

При розробці інформаційної системи приходиться вирішувати дві основні задачі:

- задачу розробки бази даних, які використовується для зберігання інформації;

- задачу розробки графічного інтерфейса клієнтської частини.

Інформаційна система являє собою прикладну програмну підсистему, яка орієнтується на збір, зберігання, пошук та обробку текстової та/або фактографічної інформації.

Найчастіше при використанні інформаційних систем застосовується діалог з користувачем.

В склад інформаційної системи в загальному входять типові програмні компоненти, а саме:

- діалоговий ввід-вивід;
- логіка діалога;
- прикладна логіка обробки даних;
- логіка керування даними;
- маніпулювання файлами та/або базами даних.

Корпоративна інформаційна система (КІС) включає в себе використання обчислювальної апаратної платформи на якій встановлено та налаштовано

стандартне програмне забезпечення, а також для виконання необхідних робіт встановлюється і спеціалізоване програмне забезпечення.

В сучасних умовах побудова на підприємстві корпоративної інформаційної системи є одним з інструментів для ведення успішного бізнесу.

На функціонування корпоративних інформаційних систем мають вплив багато факторів серед яких можна виділити три найважливіші:

- методики управління;
- загальні можливості та продуктивність комп'ютерних систем;
- підходи до технічної та програмної реалізації елементів інформаційної системи.

В процесі управління підприємством для підтримання конкурентного становища на ринку та підтримання рентабельності діяльності керівники компаній змушені шукати нові методи розвитку одним з яких є управління якістю. В наш час будь яка інформаційна система повинна відповідати новим тенденціям технологічного розвитку та бути «продвинутою» в технічному відношенні.

Для збільшення продуктивності інформаційної системи та забезпечення її функціональності використовуються різні засоби та методи такі як розвиток мережевих технологій та систем передавання даних, можливість пристосування комп'ютерної техніки до різноманітного устаткування та здійснюється постійний пошук нових, зручніших та універсальних методів програмно-технічної реалізації інформаційних систем одночасно з розвитком апаратної частини інформаційних частин.

В останні десятиліття на розвиток інформаційних систем здійснили вплив наступні чинники:

- новий підхід до програмування. Об'єктно-орієнтоване програмування майже витіснило модульне. Запровадження якого значно скоротило час на розробку складних інформаційних систем, спростило їх підтримку та розвиток;

- розвиток мережевих технологій. На зміну локальним інформаційним системам використовують клієнт-серверні та системи з багаторівневими архітектурами;

- розвиток мережі Інтернет. Дозволяє працювати по віддаленому доступі та можливість обслуговування покупців з застосуванням Інтернету.

В складі корпоративних інформаційних систем можна виділити такі складові:

- корпоративну мережу - комп'ютерну інфраструктуру, яка об'єднує мережеву, телекомунікаційну, програмну, інформаційну та організаційну інфраструктури.

Дана складова відображає системно-технічну, структурну сторону будь якої інформаційної системи, яка повністю визначає властивості системи, які відповідають за її успішну експлуатацію. Вимоги до інфраструктури стандартні, загальні, з відомими методами побудови та перевірені на практиці;

- взаємозв'язані функціональні підсистеми, які забезпечують вирішення задач та досягнення цілей організації. Вона повністю індивідуальна та залежить від специфіки роботи та поставлених задач і мети перед розробниками, що визначає функціональність інформаційної системи на прикладному рівні.

1.2. Класифікація інформаційних систем

Розглянемо класифікацію інформаційних систем для ознайомлення та визначення системи, яка найбільше підходить до розробки поставленої задачі.

1. Класифікація по масштабу:

- одиночні - використовуються на окремому комп'ютері від'єднаному від мережі. Створюють з використанням системи управління базами даних (СУБД).

- групові - орієнтовані на колективне використання на базі локальної мережі. При розробці використовують сервери баз даних (SQL-сервери)

- корпоративні - розраховані на крупні компанії, характерна система клієнт-сервер.

2. Класифікація по сфері застосування:

- системи обробки трансакцій – важливі вимоги до них висока продуктивність обробки трансакцій та гарантована доставка інформації при віддаленому доступі до бази даних (БД) по телекомунікації;

- система прийняття рішень – здійснюється збір та аналіз даних;

- інформаційно-довідкові системи – для пошуку інформації в мережі Інтернет;

- офісні інформаційні системи – автоматизація діловодства та документування.

3. Класифікація по способу організації системи на основі:

- архітектури файл-сервер;

- архітектури клієнт-сервер;

- багаторівневої архітектури;

- Інтернет/ інтранет-технологій.

Особливості варіантів побудови інформаційних додатків наведено в табл.

1.1 [1].

Таблиця 1.1

Типові функціональні компоненти інформаційної системи

Позначення	Найменування	Характеристика
PS	Presentation Services (засоби представлення)	Забезпечуються пристроями, які приймають ввід від користувача і відображають те, що повідомляє йому компонент логіки представлення PL, з використанням відповідної програмної підтримки
PL	Presentation Logic (логіка представлення)	Управляє взаємодією між користувачем та персональним комп'ютером. Обробляє дії користувачів при виборі команди в меню, натисканні кнопки або виборі елемента зі списку
BL	Business or Application	Список правил для прийняття рішень,

	Logic (прикладна логіка)	обчислення та операції, які повинен виконувати додаток
DL	Data Logic (логіка управління даними)	Операції з базою даних (SQL-оператори), які необхідно виконати для реалізації прикладної логіки управління даними)

Продовж. табл.1.1

Позначення	Найменування	Характеристика
DS	Data Services (операції з базою даних)	Дії СУБД, які викликаються для виконання логіки управління даними, такі як маніпулювання даними, визначення даних, фіксація чи повернення транзакцій. СУБД зазвичай компілює SQL-додатки
FS	File Services (файлові операції)	Дискові операції читання та запису даних для СУБД та інших компонентів. Зазвичай є функціями операційної системи (ОС).

Будь-яка інформаційна система має функціональні компоненти, які допомагають зрозуміти обмеження різноманітних архітектур інформаційних систем.

1.3. Теорія та методологія побудови систем

Поняття системи визначається, перш за все, виходячи з поняття "об'єкту пізнання". В.Д. Могилевський у книзі "Методологія систем" (1999р.) трактує ці поняття у взаємозв'язку наступним чином [2]:

"Об'єкт пізнання потрібно розглядати як систему, яка функціонує у середовищі взаємозв'язків своїх компонентів і взаємодіє з іншими системами". Цей підхід знайшов своє вираження у теорії систем та її прикладному аспекті - системному аналізі.

Теорія систем народилася як деякі узагальнення кібернетики шляхом поширення її ідей на складні організації компонентів систем. В той же час теорія систем не дає конкретних результатів для практичного застосування, в той час, як системний аналіз знайшов широке застосування в технічних задачах.

Такі атрибути, як складність, поліфункціональність, зв'язок з зовнішнім середовищем, прийняття рішень, самоорганізація і вдосконалення та управління складають зміст базисних понять теорії систем.

Класифікація систем визначається існуванням природних та штучних (технологічних та духовних) систем (рис. 1.1).



Рис.1.1. Загальна класифікація систем

В основі системи повинен бути деякий матеріальний продукт, який в сучасному розумінні включає в себе простір, матерію, енергію та інформацію. Для успішного функціонування таких надсистем виникає множина обслуговуючих їх вторинних та допоміжних класів систем. Таким чином, створюється ієрархія систем. Структура системи при цьому визначається сформульованою ціллю її функціонування. Надлишковість системи визначається умовою її надійності, тобто підсистеми, які не відповідають

функціям системи, відмирають та ліквідуються. Елементи системи повинні бути інформаційно, матеріально та енергетично пов'язані між собою для досягнення глобальної цілі системи. Системі притаманна властивість розвиватися, адаптуватися до нових умов шляхом створення нових зв'язків та компонентів.

Таким чином: система є особлива організація спеціалізованих елементів, об'єднаних в єдине ціле для рішення конкретної задачі.

Місце існування природних систем - біосфера, штучних систем - ноосфера, яка охоплена розумною людською діяльністю.

Важливим класом штучних систем є інформаційні системи, які складають основу вивчення даного предмету, який охоплює задачі проектування їх вузького класу, а саме інформаційних спеціалізованих комп'ютерних систем.

Методологія вивчення систем. Важливим кроком у побудові теорії систем є розробка аксіоматики, яка характеризує область і умови застосування теорії, а також вибір математичного апарату досліджень в галузі проектування та аналізу систем. Слід визначити термін - абстрактна теорія систем, яка відокремлює теорію від прикладних задач і формалізує математичний апарат, який відповідає методології побудови систем.

Абстрактна теорія систем нараховує 8 рівнів:

- символічний або лінгвістичний;
- теоретико-множинний;
- абстрактно-логічний;
- топологічний;
- логіко-математичний;
- теоретико-інформаційний;
- динамічний;
- евристичний.

Принциповими особливостями системи є:

- динамізм - змінність її станів у часі;
- інформаційність взаємодії її компонентів.

Вказані рівні та особливості систем складають фундаментальні основи теорії штучного інтелекту.

Можливість реалізації певних властивостей в системі базується на функціонуванні спеціалізованих елементів її архітектури. При цьому необхідна інформаційна взаємодія між елементами, які реалізуються за допомогою каналів зв'язків та сигналів, що визначає інформаційну властивість системи.

Аналіз інформаційних властивостей системи тісно пов'язаний з її складністю.

Найважливішою функцією системи є її стійкість. Порушення стійкості роботи системи пов'язане з поняттям рівноваги, яка визначається впливом її параметрів (рис. 1.2).

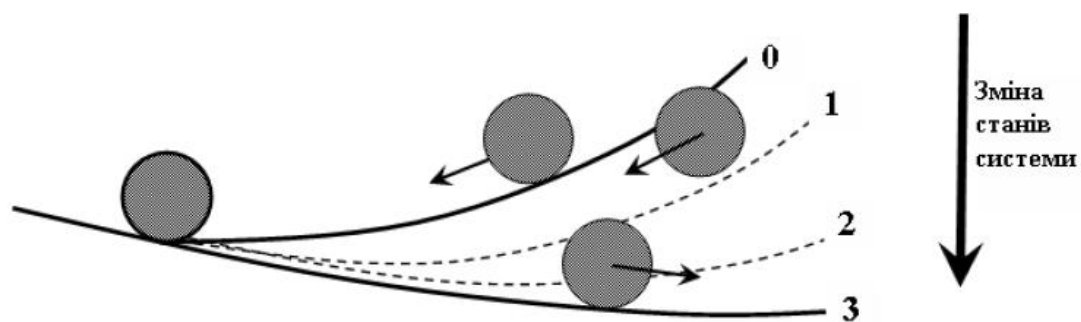


Рис.1.2. Вплив змін параметрів на стійкість

Втрата стійкості системи може відбуватися через зміну параметрів системи (біфоркація), через наявність непередбачених при створенні системи зовнішніх впливів, а також при порушенні інформаційних зв'язків в системі, коли змінюється її структура. В останньому випадку мова йде про структурну нестійкість.

Найбільш загальним представленням структури системи є оргграф взаємозв'язків між її компонентами, прикладом якої є схема формування математичної моделі системи (рис. 1.3).

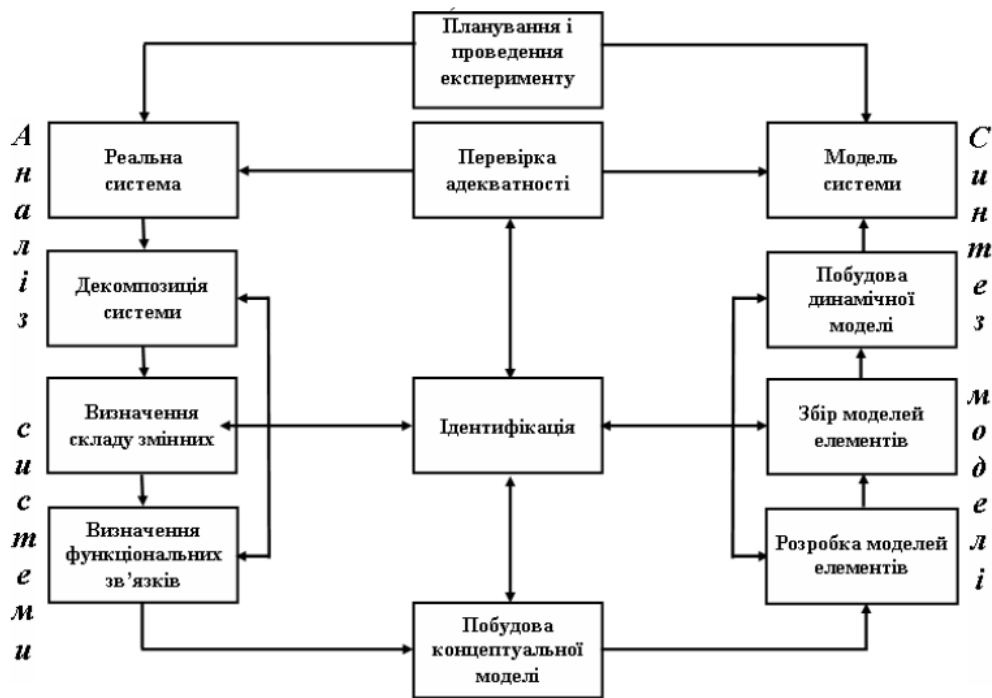


Рис.1.3. Схема формування математичної моделі системи

Важливим атрибутом системи є її модель, яка має наступне визначення. Моделлю називається спеціально синтезований для досліджень об'єкт, який характеризується необхідним ступенем подібності до досліджуваного об'єкта та адекватний цілям дослідження та характеристикам. При цьому при побудові моделі системи спочатку створюється мікро модель її окремих елементів, на основі яких будується макромодель або глобальна модель системи. В математиці моделлю називають деяку множину з заданим на ній набором відношень. В якості елементів множення виступають елементи системи, а відношеннями є зв'язки між ними.

Важливим класом моделей є концептуальні моделі. Вони описують перетворення інформації в системі і процеси її циркуляції по каналах зв'язку, тобто класифікуються як моделі руху даних у системі.

В теорії систем під рухом розуміють зміну станів, обумовлену зовнішніми та внутрішніми причинами. Концептуальні моделі є інструментом кількісного пізнання функцій системи, як множини з заданими на ній відношеннями.

Важливою характеристикою відображення моделі системи є її структура, під якою розуміють сукупність її функціональних елементів, об'єднаних зв'язками. При цьому в якості зв'язків можуть виступати матеріальні, енергетичні та інформаційні потоки.

Елементи є складовими частинами будь-якої S_0 . Кожен з елементів також може бути S_j , тобто

$$S_0 = F(S_1, S_2, \dots, S_j, \dots, S_k),$$

де k - число підсистем.

Елементи, які вступають та виходять із системи, відповідно називаються вхідними та вихідними.

Процес перетворення. В системах відбуваються процеси перетворення, у результаті яких змінюються стани системи, а вхідні елементи трансформуються у вихідні. В організованій системі цінність і корисність вхідних елементів збільшується, у протилежному випадку система деградує.

Вхідні елементи і ресурси. Вхідні елементи інформаційної системи - це сигнали та вхідні інформаційні потоки. Ресурсами при цьому виступають математичні засади алгоритми та програми формування, формалізації, перетворення, передавання, зберігання, архівації та використання даних.

Вихідні елементи. Вихідні елементи являють собою результати процесів перетворення в системі і розглядаються, як певна доцільність або прибуток від функціонування системи.

Ознаки системи та її компонентів. Системи, підсистеми та елементи можуть мати "кількісні" та "якісні" ознаки, які тісно пов'язані з теорією вимірювань.

Навколишнє середовище системи. Навколишнє середовище, з яким взаємодіє система, визначається певними границями по відношенню до ресурсів та споживачів системи.

На рис. 1.4 показана схема взаємодії системи з навколишнім середовищем.

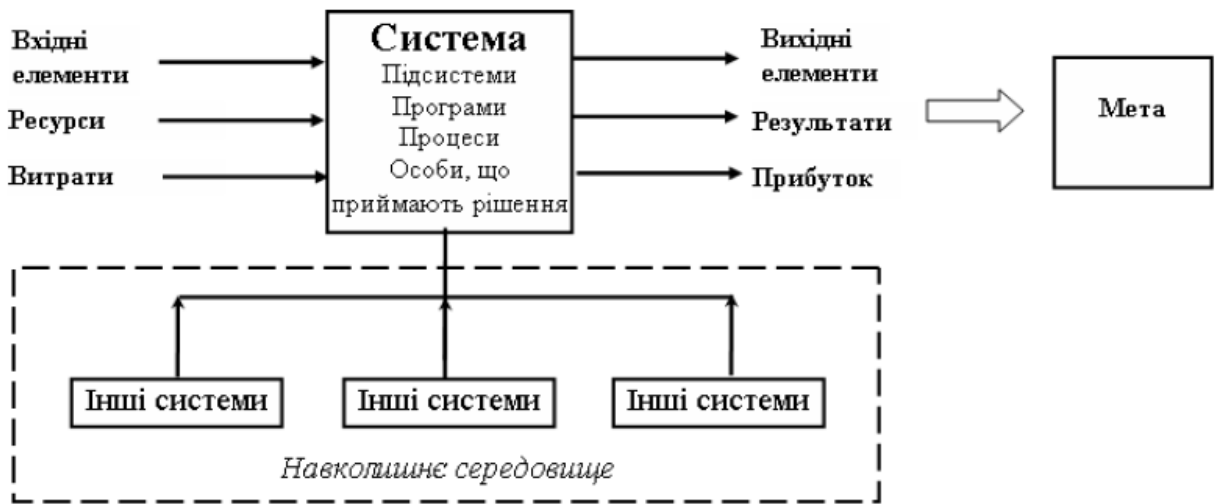


Рис.1.4. Система, її входи-виходи і навколишнє середовище

Задачі і цілі. При проектуванні систем першочергове значення має визначення завдань та цілей. Міра ефективності системи показує, наскільки досягаються цілі системи у процесі її функціонування.

Компоненти, завдання, програми. У ціленаправлених системах процес перетворення організується на основі використання функцій компонентів, завдань та програм.

Дії і рішення, які мають місце в системі, є прерогативою керуючих підсистем або елементів, які організують систему на досягнення поставлених цілей, а також контролюють та діагностують регламентні стани системи.

Поняття структури пов'язано з упорядкуванням відношень, які зв'язують елементи системи. Структура може бути простою або складною, однорівневою та ієрархічно-багаторівневою.

Оцінкою функціональних ресурсів структури системи є емерджентність, яка виражається відношенням числа зв'язків N_3 до числа елементів N_0 системи.

$$k_e = \frac{N_3}{N_0} \geq 2.$$

Стани та потоки. Існує різниця між станами і потоками в системах. Стан системи характеризується значеннями ознак системи у певний момент часу.

Переходи елементів системи з одного стану у інший викликають потоки, які визначають швидкість зміни ознак системи та її станів.

Поведінка системи. Поведінкою системи вважається зміна станів системи у часі.

При проектуванні систем особливої уваги заслуговують чотири важливі проблеми:

- визначення границь системи та границь навколишнього середовища, з якими взаємодіє система;
- встановлення чітких цілей системи;
- визначення структури програми та алгоритмів функціонування системи;
- опис управління системами.

Границі навколишнього середовища системи поширюються і охоплюють тільки ті компоненти, які контролюються і керуються системою. Проблема встановлення границь системи нерозривно пов'язана з визначенням цілей системи та вибором критеріїв ефективності. При цьому можлива множина суперечливих цілей і різних критеріїв оцінки ефективності системи.

Структура програми - це подання відношень всіх елементів у відповідності до функцій системи. Можливо визначити структуру програми як блок-схему залежностей між різними елементами або як можливі шляхи досягнення цілей системи.

Термін управління є загальним поняттям, яке включає всі дії компонентів системи, приймаючих рішення.

Рівні системи. Розрізняють наступні рівні систем:

- система високого рівня;
- система;
- підсистема.

Система високого рівня є керуючою по відношенню до системи, а підсистема може бути залежною, але автономною системою, яка має свою ціль.

У системах обов'язково повинні існувати додатні або від'ємні зворотні зв'язки, які забезпечують необхідну стійкість системи.

Організація є функціональною характеристикою системи, яка не тотожна складності її структури.

Математичні аспекти теорії систем. Дві системи можуть бути названі подібними на основі оцінки ступеня близькості їх математичних моделей.

Системи вважаються тотожними, якщо їх математичні структури моделей ізоморфні.

Рівні вивчення систем згідно концепції Боулдинга.

1. Емпіричний, коли знання про систему отримують тільки у результаті безпосереднього спостереження.

2. Формальний, на якому реалізуються проекти систем та інструкції їх функціонування.

3. Аванпроектний, коли знань достатньо для попередньої розробки проекту системи.

4. Теоретичний, на основі якого визначаються суттєві властивості системи.

5. Системно-теоретичний, коли визначені узагальнені характеристики систем.

Складність є продуктом об'єму та руху інформаційних потоків у системі.

Самоорганізація системи. Система, яка формує під дією зовнішніх впливів у результаті свого розвитку нову систему, яка відрізняється від початкової.

1.4. Комп'ютерні системи

Комп'ютерні системи можуть належати до двох класів: у залежності від способу опрацювання даних та інформаційних потоків:

- вбудовані КС;
- концентровані КС;
- розподілені КС пакетного опрацювання даних;
- розподілені КС реального часу.

Вбудовані КС, як правило, реалізуються на базі однієї обчислювальної платформи, мікропроцесора або мікроконтролера. При цьому всі обчислювальні процеси відбуваються в межах кристалу процесора та кристалів периферійного обладнання, яке за допомогою фізичних електронних чи оптоелектронних зв'язків взаємодіє з базовим процесором.

КС найчастіше використовуються в КСУ мобільних та автономних об'єктів управління. Приклад архітектури вбудованої КС показано на рис.1.5.

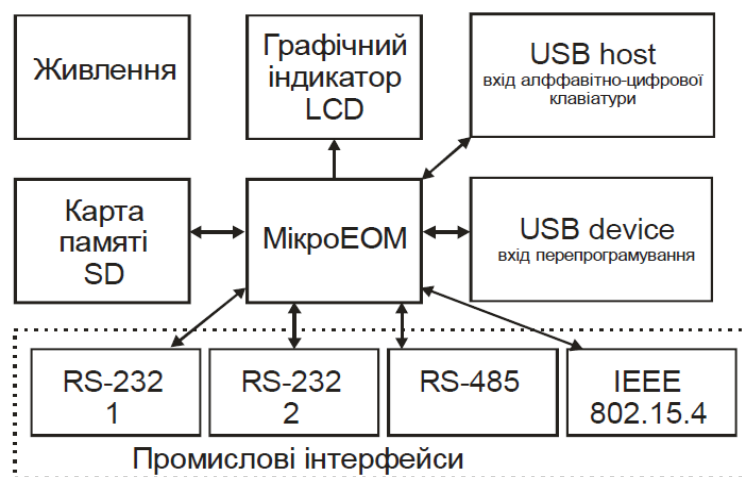


Рис.1.5. Архітектура вбудованої КС

Концентровані КС реалізуються на основі потужного процесора і багатьох терміналів, які на невеликій віддалі, наприклад, в межах однієї технологічної установки, машини, літака, судна або приміщення та офісу, взаємодіють через фізичні електромагнітні чи оптичні лінії зв'язку з центральним процесором.

Архітектура КС концентрованого опрацювання даних показана на рис. 1.6.

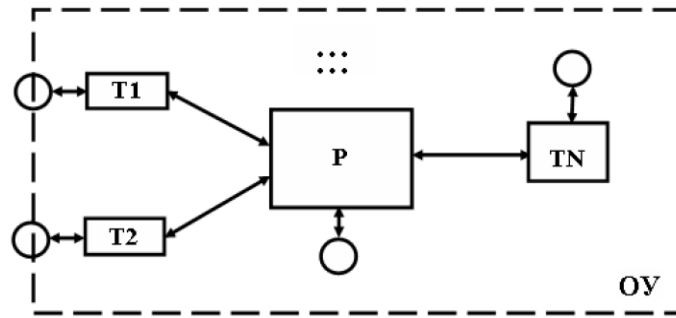


Рис.1.6. Архітектура концентрованої КС

Особливістю таких КС є використання засобів високошвидкісного паралельного та послідовного інтерфейсів, а також автономних сенсорів.

Розподілені КС (РКС) реалізуються на основі обчислювальних мереж та віддалених процесорів-сателітів, які обслуговуються та інформаційно взаємодіють з одним або багатьма системними серверами (СС). РКС можуть мати різні архітектури, які відображаються узагальнено моделлю на рис. 1.7.

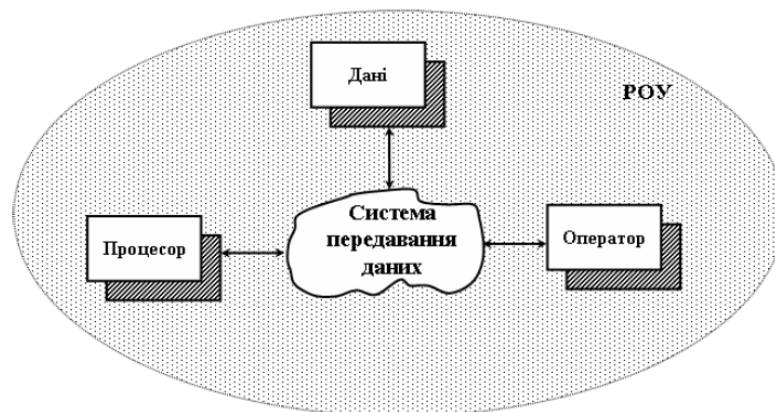



Рис.1.7. Узагальнена архітектура РКС (РОУ - розподілений об'єкт управління;  - ознака багаточисельності)

Розподілені КС реального часу можуть належати двом класам:

- універсальні РКС;
- спеціалізовані РКС або спеціалізовані КС (КСУ).

Універсальні РКС також можуть бути реалізовані на основі різних архітектур та топологій. Глобальна модель таких систем показана на рис.1.8.

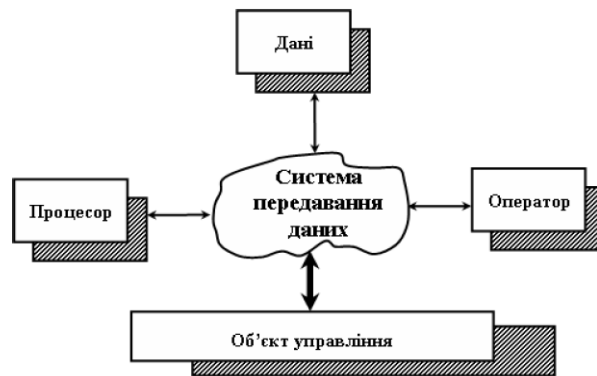


Рис.1.8. Глобальна модель РКС

З рис.1.8 видно, що в інформаційну структуру РКС та КСУ, крім процесорів, даних, системи передавання даних та операторів, обов'язково входить віддалений розподілений об'єкт управління.

Відмінність КСУ від універсальної РКС полягає в тому, що в КСУ особливі гіперхарактеристики. Наприклад, в залежності від умов експлуатації та необхідних характеристик обладнання КСУ (високі або низькі температури, вібрації, вибухонебезпечність, мінімальні габарити, особливо висока надійність, максимальна швидкодія, об'єм пам'яті та інші), а також унікальних особливостей об'єкта управління, КСУ максимально адаптується програмно-апаратними засобами до характеристик, необхідних алгоритмів та процесів опрацювання інформаційних потоків у реальному часі. При цьому особлива увага надається реалізації значного числа інформаційних моделей Д та ОУ, які оперативно використовуються для контролю відхилень станів ОУ від норми, виявлення передаварійних та попередження аварійних станів ОУ, а також обслуговуванню інформаційних потоків у квазістаціонарних станах ОУ.

1.5. Спеціалізовані комп'ютерні системи

Спеціалізовані комп'ютерні системи (КСУ) відрізняються від універсальних та інших класів комп'ютерних систем умовами максимізації або мінімізації системних характеристик їх компонентів:

Виходячи з ресурсних характеристик системних об'єктів глобальної моделі КСУ (див.рис.1.8), які описуються узагальненим функціоналом

$$\hat{A} \hat{m} = F(T, V, M, S),$$

де T - час використання ресурсу, V - швидкість виконання системних операцій (формування, передавання, цифрова обробка та зберігання даних), M - об'єм використовуваного ресурсу пам'яті, S - системні функції.

До характеристик кортежу функціоналу, на відміну від універсальних комп'ютерних систем, висуваються особливі мінімаксні вимоги наступного виду:

$$T = \min \vee \max;$$

$$V = \min \vee \max;$$

$$M = \min \vee \max;$$

$$S = \min \vee \max.$$

Вказані вимоги визначають базові параметри компонентів КСУ в залежності від класу, характеристик стаціонарності, екологічної безпеки, вибухонебезпечності, особливих умов експлуатації (великі глибини в морі, підземні об'єкти, космічні, стратегічні та ін. об'єкти).

Найчастіше одночасне досягнення суперечливих умов до системних характеристик компонентів КСУ є в принципі неможливим або недоцільним.

Тому задачі проектування та створення ефективних в архітектурному, функціональному та інформаційному аспектах є багатоваріантними з нечітко вираженими оптимумами, що потребує диференціації проектних рішень, стратегій проектування та побудови значного числа інформаційних моделей об'єктів управління, що обслуговують КСУ.

1.6. Основи побудови спеціалізованих комп'ютерних систем

Спеціалізовані комп'ютерні системи (СКС) належать до класу проблемно-орієнтованих комп'ютерних систем, вбудованих та розподілених

комп'ютерних систем (РКС) реального часу, а також суперфункціональних комп'ютерних систем (КС) спеціального призначення та відповідного програмно-апаратного виконання.

Проектні та експлуатаційні характеристики СКС визначаються не умовами універсальності та масового тиражування, а найчастіше особливими властивостями та характеристиками об'єктів управління, які вони обслуговують. СКС повинні володіти відповідними унікальними характеристиками алгоритмічної, структурної, ємнісної, апаратної складності, надійності та живучості в особливих умовах, часто, критичних чи жорстких обмеженнях на системні параметри часу, швидкодії, об'єму пам'яті та операційного інтелекту.

Основними інструментальними засобами сучасної інформатики є комп'ютерні системи різних класів і потужностей. За допомогою існуючих універсальних КС (УКС) можна розв'язувати багато задач наукового, виробничо-технічного та іншого характеру. Однак існують, як окремі задачі, так і цілі класи надзвичайно важливих задач, для розв'язання яких математичних та техніко-економічних якостей сучасних УКС вже не вистачає. Тут не варто доводити дієвість принципу спеціалізації інструментальних засобів взагалі, оскільки вся свідома технічна діяльність людства її підтверджує. Досить сказати, що цей принцип є ефективним і в галузі інформатики. Загальний аналіз причин створення і використання СКС показує, що ці причини можна розділити на три основні групи.

У першу групу об'єднані причини, які виникли внаслідок суперечностей між формальними математичними методами постановки і розв'язання задач, з одного боку, і загальними принципами організації та функціонування, а також технічними можливостями УКС, з другого боку. Саме математична сутність задач часто зумовлює необхідність створення СКС для їх розв'язання. Як приклад, можна навести нові нестандартні і неалгоритмічні методи, системи алгебраїчних, диференціальних та інтегральних рівнянь великої розмірності,

логічні та ймовірно-статистичні задачі, дії над матрицями і векторами, задачі в багатовимірних просторах.

До другої групи належать причини, які зумовлюються змістовною стороною задачі, що розв'язується СКС, та відображають специфіку відповідних предметних галузей. Для прикладу перелік таких галузей наведено нижче.

Третя група причин зумовлена особливими вимогами до якості реалізації КС, які звичайно полягають в екстремізації (тобто в максимальному наближенні до теоретичних границь) деяких їх характеристик, наприклад продуктивності (швидкодії), надійності (безвідмовності, живучості, відновлюваності, довговічності тощо), вартості та складності, точності, габаритів, маси та ін. Сюди ж належать вимоги, що визначають такі якості КС, як їх повна або часткова імплантація (конструктивне та функціональне суміщення) в інші системи, інформаційне поєднання з ними, пристосованість до умов експлуатації та кваліфікації обслуговуючого персоналу і т.д.

Слід мати на увазі, що реальні ситуації створення СКС найповніше характеризуються двома особливостями.

Перша особливість полягає в тому, що саме СКС є своєрідним засобом апробації нових методів автоматизації обробки інформації, які мають математичні корені. Наприклад, розпаралелювання та децентралізація обчислень, макрооперації та використання функціональних розширювачів, символна обробка та розв'язання задач у багатовимірних числових системах пройшли спочатку дуже ретельну перевірку в СКС і тільки після цього з'явилися в УКС.

Друга особливість пов'язана з тим, що реальні СКС є складними програмно-технічними комплексами, в яких необхідно задовільнити багато суперечливих вимог. Тому досягнення оптимальних функціональних якостей СКС може бути проблематичним, а тому доцільніше визначити ці якості як оптимізовані, тобто такі, що тією чи іншою мірою наближаються до оптимальних. Аналіз математичних методів оптимізації СКС показує, що вони

сприяють певною мірою виявленню недоліків таких систем і їхні «слабкі місця», простежувати взаємозв'язок характеристик системи, визначати загальний напрямок підвищення їх ефективності та оцінювати різні варіанти СКС. Однак ці методи, як правило, не дають конструктивних розв'язків і шляхів удосконалення СКС, не визначають змістовної сторони різних варіантів їх організації і реалізації. Генезис таких варіантів формальними математичними методами практично неможливий. Тому процес створення оптимізованих СКС має характер багатоступеневої ітераційної процедури, де в різних відношеннях комбінуються формальні та конкретно-змістовні методи, що відіграють аналітичну (оціночну) та синтетичну (генеративну) роль.

Отже, СКС - це КС для розв'язання багатьох відносно вузьких класів задач, оптимізовані в певній критеріальній сукупності.

Досі, протягом майже 50 років, принципово нові розробки КС базувались переважно на «проривних» успіхах мікроелектронної технології. Тут досить вказати на зв'язок поколінь КС та досягнутого мікроелектронікою рівня інтеграції. Однак виробництво електронних комплектуючих компонент для КС є дуже дорогим через потребу в надзвичайно великих капітальних вкладеннях у мікроелектронне виробництво. Тому протягом тих же 50-ти років виявилось, що створення комп'ютерної електронно-технологічної бази під силу далеко не всім країнам, навіть високорозвинутим. На сьогоднішній день ситуація в світі складається так, що виробництво основних електронних компонент для КС зосереджено переважно на 10-15 електронних фірмах, зокрема на таких, як Intel, TexasInstruments, Motorola. Ці фірми щороку вкладають величезні кошти в розвиток технологічної бази і, тим самим, ще більше закріплюють свої позиції лідерів. Чи означає це, що всі розробки в галузі КС зосереджено на цих фірмах? Ні, це зовсім не так. І тому є кілька причин.

По-перше, сам етап розробки електронних компонент сьогодні виділився в окрему галузь діяльності, яка має кілька напрямків. Перший з них передбачає створення систем автоматизованого проектування (САПР) електронних компонент. Тут також сформувалась група провідних фірм, зокрема

MentorGraphics, Synopsys, LSILogic, Cadence, DesignSystems, які створюють програмні продукти для різних етапів проектування електронних компонент. Разом з тим, цей напрямок значною мірою є відкритим для діяльності, особливо для тієї, що стосується вищого, системного, етапу проектування. Роботи по розвитку САПР цього етапу є на сьогодні найприоритетнішими, інформаційно місткими та стратегічно важливими. Другий напрямок передбачає створення комп'ютерних засобів підтримки проектування електронних компонент. Тут лідерство фірм-виробників робочих станцій Sun та HewlettPackard починає витіснятися виробниками робочих станцій на базі потужних персональних комп'ютерів з процесорами фірми Intel. Нарешті, останній, третій, напрямок передбачає безпосередньо розробку електронних компонент. Існує велика кількість фірм, розміщених у різних країнах, які в рамках угод займаються проектуванням електронних компонент для їх наступного виготовлення провідними мікроелектронними фірмами. Цей напрямок є найбільш відкритим для діяльності.

По-друге, самостійною галуззю діяльності є розробка і виготовлення УКС та СКС на базі електронних компонент провідних фірм-виробників. Тут поряд з кожною мікроелектронною фірмою сформувались групи системотехнічних компаній, які на базі продукції цієї фірми створюють відповідні УКС і СКС та програмні продукти. За кордоном ці фірми дістали назву «thirdparty». Більше того, в усіх розвинутих країнах здійснюється розробка і виробництво власних СКС різного призначення на базі імпортованих комплектуючих та на базі продукції «thirdparty». Наприклад, канадські, англійські, німецькі, французькі та інші фірми проводять такі розробки, використовуючи комплектуючі, що виробляються в знаменитій «кремнієвій долині» в США, а в самих США, крім власних, використовуються також комплектуючі з різних країн, зокрема, японські. Характерно, що розробку (добробку) СКС відповідно до потреб конкретного замовника здійснює значна кількість переважно невеликих фірм (15-50 працюючих). Так, щоб мати більший ринок збуту своєї продукції, провідні мікроелектронні фірми, крім випуску функціонально закінчених

виробів, роблять вироби з врахуванням побажань споживачів. Наприклад, розробивши потужне процесорне ядро (англійський термін «core»), фірма шляхом доповнення його необхідним обрамленням створює сімейство спеціалізованих комп'ютерів різноманітного призначення.

Україна в сучасних політичних і економічних умовах змушена розпочинати самостійну розробку та виробництво ряду СКС, оскільки існують предметні галузі та особливі вимоги до комп'ютерного обладнання і програмних засобів, що не дозволяють використовувати повністю або частково СКС закордонного виробництва. Серед предметних галузей є, наприклад, такі що безпосередньо пов'язані із забезпеченням суверенності і державності України: бортове та інше комп'ютеризоване обладнання військово-оборонного призначення: орієнтаційні та управлінські системи транспортного використання; засоби захисту інформації від несанкціонованого простору; обладнання комп'ютеризованих телекомунікаційних систем і мереж; засоби обробки сигналів і зображень відповідно до національних стандартів; банківська та податкова справа; енергетика (особливо ядерна) і засоби енергозбереження; геофізичні та гідрофізичні дослідження ресурсів України (політичні події змінили і відносну цінність таких ресурсів). Ряд інших галузей визначає рівень економічного розвитку держави: засоби сертифікації, промислової та сільськогосподарської продукції; засоби управління, обробки інформації та прийняття рішень в складних динамічних системах, що функціонують в реальному часі; розширене застосування нових інформаційних технологій (наприклад, штрихових кодів тощо); медична і промислова діагностична техніка; об'єктно-орієнтовані системи автоматизації проектування (наприклад, оптичних приладів і систем); аналіз і синтез мови; моделювання та реалізація функцій інших систем комп'ютерними засобами та ін.

На сьогодні в теорії СКС сформувалися два основні підходи до організації їх структур.

Перший ґрунтується на використанні стандартних універсальних процесорів, а врахування особливостей виконуваних задач реалізується, по суті, шляхом спеціалізації системного програмного забезпечення.

Другий підхід базується на використанні процесорів, орієнтованих на виконувані алгоритми (функції) апаратним способом. Такі процесори називають функціонально - або апаратно-орієнтованими (в англійській літературі - ASIC, ASIP).

Основна перевага процесорів універсальної архітектури - гнучкість. Запис програм виконання заданого набору алгоритмів у пам'ять програм дає можливість створити спеціалізований процесор із заданими функціями. Такі процесори підлягають перепрограмуванню заміною вмісту пам'яті програм. При створенні відповідних трансляторів програмне забезпечення СКС може бути написане мовами програмування високого рівня, що робить їх доступними для широкого кола користувачів. Крім того, дуже суттєвою перевагою цього підходу є можливість використання створених раніше програмних засобів. Разом з тим, існує ряд причин, через які використання стандартних процесорів універсальної архітектури для побудови СКС може бути недоцільним.

По-перше, це висока трудомісткість розробки, оскільки до складу СКС, крім самого універсального процесора, необхідно включити також засоби для реалізації інтерфейсних функцій, синхронізації, розширення пам'яті програм і пам'яті даних тощо. Сам процес розробки вимагає створення необхідного програмного забезпечення, наявності або створення технологічних програмних засобів для відпрацювання програм, а також програмно-апаратних засобів для відлагодження апаратної та програмної частин спеціалізованих процесорів.

По-друге, універсальна архітектура може бути дуже надлишковою у функціональному і структурному відношеннях для розв'язання однієї конкретної задачі (наприклад, для перетворення координат у системах орієнтації). Це може привести до надлишкової споживаної потужності, збільшення ступеня інтеграції, кількості виводів корпусів та розмірів кристалів.

По-третє, універсальний процесор може не задовольнити вимог щодо продуктивності СКС.

Забезпечити необхідну продуктивність можна шляхом побудови багатопроцесорних СКС. Однак їх використання може бути занадто дорогим. Тут мається на увазі вартість не стільки власне багатопроцесорної системи, скільки допоміжних засобів, що забезпечують використання такої системи.

Особливо багато проблем пов'язано з розробкою паралельних обчислювальних процесів, при реалізації яких всі процесори були б повністю завантажені.

В такому разі системне програмне забезпечення має бути здатним оперативно розв'язувати задачу оптимізаційного планування завантаження процесорів багатопроцесорної системи конкретної архітектури.

Такий підхід може вимагати великих витрат обладнання, що є небажаним (особливо при використанні СКС у бортових системах).

Зараз, коли досягнення мікроелектронної технології підтримується потужними САПР, другий підхід, що передбачає створення апаратно-орієнтованих на виконувани алгоритми процесорів, є реальною альтернативою універсальним стандартним процесорам.

По-перше, такий підхід забезпечує максимально можливу продуктивність при розв'язанні заданої задачі.

По-друге, він вимагає мінімальних витрат обладнання на побудову СКС для розв'язання заданої задачі за рахунок компромісу між програмними та апаратними засобами. Таке спільне проектування апаратної і програмної частин СКС дістало назву *hardware/softwareco-design*.

По-третє, сучасні мови опису апаратних засобів, зокрема VHDL, мають настільки високий рівень, що процес проектування апаратно-орієнтованого процесора не є набагато складнішим у порівнянні з розробкою спеціального програмного забезпечення для універсального процесора.

Якщо врахувати ще й можливості сучасних САПР щодо забезпечення повного відпрацювання моделей електронних компонент, включаючи їх роботу

в складі СКС, а також використання бібліотек раніше створених електронних компонент, то стає очевидним, що настав час, коли другий підхід починає витісняти перший.

З викладеного випливає, що розвиток СКС в Україні зумовлений, з одного боку, потребами в СКС, а з іншого - можливостями їх власної розробки і виробництва.

Світовий досвід створення СКС підтверджує життєздатність цього напрямку комп'ютерної науки і техніки в сучасних умовах, незважаючи на імпордне походження значної частини мікроелектронних комплектуючих виробів. Забезпечення необхідного кваліфікаційного рівня кадрового супроводу такого розвитку має бути однією з найважливіших задач національної вищої технічної школи.

Насичення функціональних можливостей традиційних інформаційних технологій опрацювання інформаційних потоків у базисі Радемахера (двійковій системі числення) та синтезу архітектур комп'ютерних систем на основі мереж Петрі потребує навчання і підготовки нового покоління фахівців по проектуванню СКС, які б володіли теоретичною базою, технологією і практикою застосування нелінійних інформаційних технологій опрацювання повідомлень у базисах Хаара, Крестенсона, Уолша та Галуа.

Перспективними знаннями у цій галузі є вміння формалізувати інформаційні потоки в СКС на основі «кольорових» мереж Петрі та матричних моделей руху даних.

Теоретичною основою таких моделей є теорія та інформаційна технологія організації та оцінки руху даних в складних розподілених КС, розроблена відомим американським системотехніком Дж.Мартіном.

Спеціалізована комп'ютерна система - це комп'ютерна система для розв'язання багатьох відносно вузьких класів задач, оптимізовані в певній критеріальній сукупності

1.7. Глобальна модель, системні об'єкти, системні функції, функціональні об'єкти комп'ютерних систем

Глобальна модель комп'ютерної системи вперше була запропонована Николайчуком Я.М. у вигляді взаємодії п'ятьох типів системних об'єктів:

Р - процесор,

Д - дані,

СПД - система передавання даних,

ОУ - об'єкт управління,

О - оператор.

Основні функції системних об'єктів:

Об'єкт управління (ОУ): вирішення технологічних задач; виконання завдань управління; формування повідомлень про свій стан.

Дані (Д): ідентифікація станів ОУ; нагромадження повідомлень; надання інформації.

Система передавання даних (СПД): організація руху даних; формування, передавання та виділення сигналів; комутація повідомлень.

Процесор (Р): обробка повідомлень; управління потоками даних; нагромадження інформації; організація діалогу; управління об'єктом.

Оператор (О): збір і формування повідомлень; їх обробка; діалог; прийняття рішень.

Кожен з наведених системних об'єктів може виконувати в середовищі КС чотири системні функції:

- формування даних ;
- передавання даних;
- цифрова обробка даних;
- приймання та зберігання даних.

Тобто, кожен з системних об'єктів (СО) може бути одним з функціональних об'єктів наступного типу:

- джерело інформації (ДІ);

- середовище передавання інформації (СПІ);
- середовище цифрової обробки інформації (СОІ);
- приймач інформації (ПІ).

Отже, системні об'єкти КС характеризуються дуальними (поліфункціональними) властивостями, що в значній мірі ускладнює методологію проектування та теоретичні основи оптимізації параметрів КС.

Очевидно, що для вивчення названих інтерфейсних взаємодій СО та використання їх при проектуванні КС необхідно описати взаємодію 25-ти їхніх пар.

В той же час, враховуючи, що теорія, методологія та техніка реалізації КС на основі стандартних технічних засобів, міжнародних протоколів та інтерфейсів достатньо повно подана у відповідних виданнях та інструкціях, при проектуванні низових рівнів проблемно-орієнтованих та спеціалізованих КС особливу увагу слід надавати вивченню системних характеристик об'єктів управління та їх взаємодії з іншими об'єктами КС.

Виходячи з класифікації п'яти системних об'єктів КС, можна побудувати таблицю пар їхньої взаємодії через інтерфейсні засоби комунікацій (табл.1.2).

Таблиця 1.2

Пари взаємодії системних об'єктів КС через інтерфейсні засоби комунікацій

	Р	Д	СПД	ОУ	О
Р	Р→Р	Р→Д	Р→СПД	Р→ОУ	Р→О
Д	Д→Р	Д→Д	Д→СПД	Д→ОУ	Д→О
СПД	СПД→Р	СПД→Д	СПД→СПД	СПД→ОУ	СПД→О
ОУ	ОУ→Р	ОУ→Д	ОУ→СПД	ОУ→ОУ	ОУ→О
О	О→Р	О→Д	О→СПД	О→ОУ	О→О

Дані взаємодії ОУ та СПД з іншими СО відображені в табл.1.2 відповідним фоном. Широка різноманітність реальних ОУ (наприклад, космічні апарати, літаки, підводні та наземні човни, атомні станції, енергетичні та нафтопромислові системи, транспортні засоби, інформаційні системи,

телекомунікації та інше) вимагає відповідної проблемної орієнтації та адаптації КС до характеристик ОУ при проектуванні та аналізі діючих КС.

Ця адаптація потребує по - новому осмислити формалізацію опису характеристик системних об'єктів.

1.8. Класифікація архітектур та характеристики комп'ютерних систем

В даний час існує широка різноманітність архітектур КСУ, до яких належать концентровані та розподілені системи опрацювання даних. До систем першого класу можна віднести монопольні архітектури, архітектури з розподіленим часом, архітектури з мультипрограмним та мультипроцесорним опрацюванням даних. Другий клас представлений значним числом однорівневих архітектур, в тому числі: магістральні, зіркові, кільцеві, систолічні. До класу багаторівневих розподілених архітектур КСУ слід віднести ієрархічні, багаторівневі-магістральні та зірково-магістральні архітектури.

Окремим класом безпроводних архітектур представлені безпроводні радіотехнічні інформаційні системи та комп'ютерні мережі наступного типу:

- безретрансляторні;
- з пасивними ретрансляторами;
- з активними ретрансляторами, в тому числі сотові мережі.

Комп'ютерні системи з оптичними каналами зв'язку охоплюються архітектурами на основі:

- дуплексних оптичних ретрансляторів;
- оптичних активних ретрансляторів;
- оптичних сканерів;
- волоконно-оптичних ліній зв'язку.

Значною оригінальністю архітектур характеризуються спеціалізовані комп'ютерні системи (КСУ), які часто можуть базуватися на об'єднанні окремих елементів різних типових архітектур.

До такого класу інформаційних систем, наприклад, належать:

- системи обліку витрат енергоносіїв з глибоким розпаралеленням потоків даних;
- комп'ютерні розподілені системи екологічного моніторингу;
- спеціалізовані охоронні системи;
- проблемно-орієнтовані корпоративні системи промислових та адміністративних організацій.

Така велика кількість архітектур інформаційних систем в значній мірі ускладнює вирішення задач оптимізації проектних рішень при побудові інформаційних систем, що потребує розробки відповідних моделей архітектур, які б дозволили шляхом формалізації структурних елементів різних мереж з єдиних позицій провести дослідження та порівняння їх системних характеристик.

Одним з перспективних підходів до вирішення такої задачі є використання теорії та технології побудови одномірних та багатомірних матричних моделей руху даних, що визначає актуальність таких досліджень.

На рис.1.9 подана класифікація архітектур КСУ з фізичними лініями зв'язку.

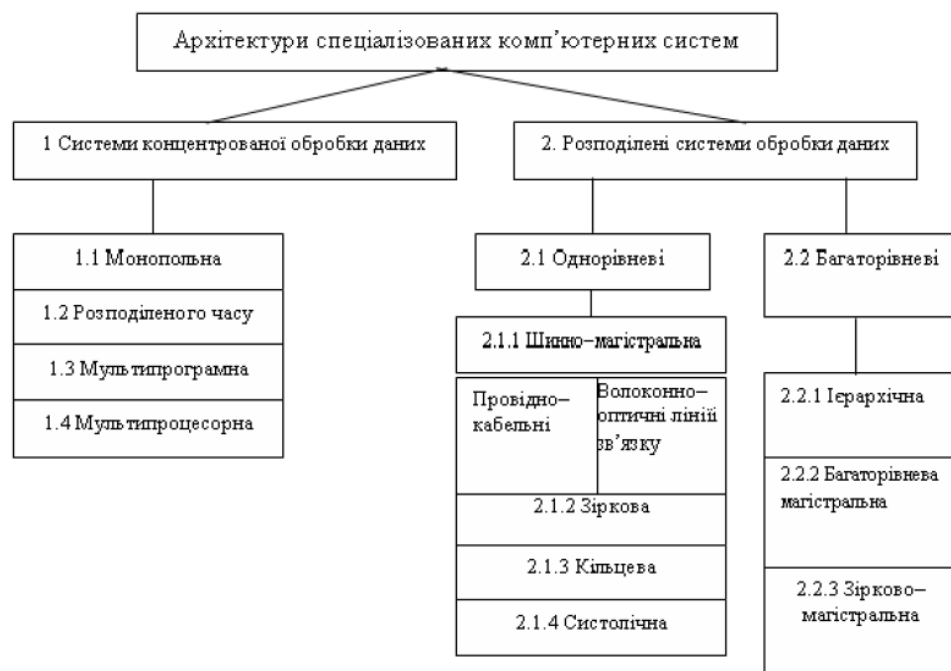


Рис.1.9. Класифікація архітектур КСУ з фізичними лініями зв'язку

На рис.1.10 подана класифікація архітектур КСУ з безпроводними лініями зв'язку.

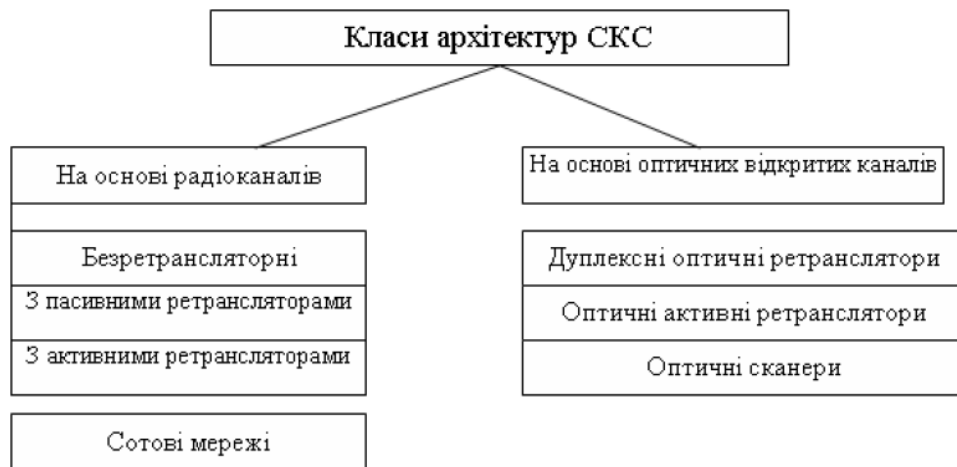


Рис.1.10. Класифікація архітектур КСУ з безпроводними радіоканалами

Подані класифікації архітектур КСУ дозволяють виконати формалізований вибір відповідного класу архітектури КСУ в залежності від їх проблемної орієнтації та необхідних системних характеристик.

1.9. Висновки до розділу 1

В розділі здійснено огляд загальних теоретичних інформацій, літературних джерел про інформаційні системи, спеціалізовані комп'ютерні системи.

Розглянуто основні поняття та класифікацію інформаційних систем опрацьовано теорію та методологію побудови систем.

Проведено аналіз комп'ютерні системи та вивчено спеціалізовані комп'ютерні системи. Описано основи побудови спеціалізованих комп'ютерних систем, глобальну модель, системні об'єкти, системні функції, функціональні об'єкти комп'ютерних систем.

На основі класифікації архітектур проведено порівняння ознайомлено з основними характеристиками комп'ютерних систем

РОЗДІЛ 2

МОДЕЛІ І МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНИХ СИСТЕМ

2.1. Методи та засоби створення інформаційної системи

У методологіях направлених на дані відмічаються частини програми фундаментованих даних. Програми створені методично (систематично) дивлячись на офіційну специфіку даних, що працює з ними. Опираючись на формальні специфікації, ми можемо створити засоби автоматичного програмування і досягнення вірності програм. Важлива увага зосереджена на різного роду абстракціях даних.

Метод об'єктно-орієнтованої розробки побудований на концепціях приховання інформації і загальних типів даних. Такий підхід переглядає всі ресурси (дані, модулі та системи), що виконують роль об'єктів. Кожен об'єкт ховає в собі деяку фігуру даних (або тип даних), описану набором правил, які знають, як жонглювати з цими даними. Використовуючи цю методологію, програміст може розробити свій власний академічний тип і відобразити проблемну зону у ці написані ним абстракції замінюючи традиційне зображення сфери, до якої є запитання, у замислі керуючої організації і організації відомостей мови реалізації. Схожий прийом представляється більш живим ніж методології, спрямовані на оперування (процес), через можливість розробляти у процесі проектування окремі типи академічних типів даних. На теперішній стадії програміст може зосередитись на проектні системи, не хвилюючись про деталі інформаційних об'єктів, які використовуються у системі.

Реалізація основних операцій методології:

- 1) найперше це знайти, в чому є проблема;
- 2) підготувати план кроків, що дають в сумі особливу стратегію, яка задовольняє вимогам системи;
- 3) втілити вищезазначену стратегію з простого плану до видимої форми;
 - а) визначити об'єкти і їхні значення;

- б) визначити, які можна зробити операції з певними об'єктами;
- в) створити інтерфейс;
- г) запустити всі створені операції.

Метод, що заснований на створенні концептуальних баз даних який пішов з класу методів, що орієнтовані на дані, закликаний надати розробнику вказівки у ході перетворень специфікацій у концептуальну схему бази даних. За мету в цьому підході взята модель з більш багатим, складнішим і повним значенням, щоб використати дану модель для спрощення роботи в проектуванні системи. В реальності це вид знань, паралелі яких проведені від реального світу до коду, що виконує ЕОМ. Сам процес розробки проекту розглядається як звичайна побудова простої моделі. Доступні для нас методи складання концептуальної моделі створені на зразках і прийомах звичайного узагальнення. Прогнозується, що розробник буде починати з найпростіших, натуральних класів об'єктів в подіях сфери де з'являється проблема. Всі наступні деталі існуючої системи будуть вводиться послідовними ітераціями через підкласи і їхню взаємодію.

Попередньо розглянуті методи містяться на модульному рівні. Метод, який використовує код і знаходиться на рівні кодингу, має назву: "Структурне програмування". Вищезазначений метод заснований на ідеї, що код простіше читати, обдумувати і писати.

Використовуючи прості базові структури, а точніше тільки три види базової структури (прямування, ітерацію і вибір), можливо представити абсолютно будь-яку складну систему. Структурне програмування об'єднує в собі певний список пов'язаних між собою проблем починаючи від методів програмування, захоплюючи проблему правильності написання і нотації і закінчуючи проблемою верифікації. Під час використання структурного програмування було виявлено, що вищезазначений спосіб не має сильного ефекту при створенні і розробці складних і великих систем. Щоб досягти високого рівня надійності і зменшення вартості буде доречно об'єднати засоби структурного програмування з архітектурними методами проектування, що

включає в себе проектування зверху – вниз, а також бібліотеки, що допомагають розвитку створення і розробки проекту.

При створенні інформаційної системи для зручності методи її створення прийнято ділити за ступенем автоматизації проектів, їх є три: головний – це оригінальний, також є “характерний” метод і автоматизований метод.

При використанні оригінального методу, або індивідуального, персонального, особистого і власноручного іншими словами, створюються особливі, приватні рішення, що є специфічними для будь-якого окремого об’єкту. Його переваги в тому, що в кінці ми отримуємо проект, що повністю відповідає всім окремим особливостям, якими б значними чи незначними вони не були. Якщо є переваги, відповідно є недоліки, потрібно згадати, що створення проекту таким методом є набагато довшим, порівняно з іншими методами і окремо треба згадати, що при такому методі є низька надійність, висока трудомісткість, збільшені терміни розробки системи, дуже поганий супровід і удосконалення інформаційної системи. Термін нормальної, прийнятної роботи і функціональності такого проектного рішення, зазвичай, не перевищує і року, тому потім потрібно удосконалювати рішення.

Формальними методами проектування прийнято ділити систему, що створюється на безліч різних частин (найчастіше це функції та алгоритми). Але для кожного методу потім потрібно створити певне проектне рішення, можливо навіть з різного роду модифікаціями, що буде використано при розробці інформаційної системи. Їх прийнято поділяти згідно з різним рівнем декомпозиції. Поділяють їх на:

- елементний – при використанні формальних рішень;
- компонентний – для використання разом з прикладеними програмами;
- об’єктний – при використанні характерних рішень ІС.

Засоби розробки типових ІС зобов’язані:

- комплексно об’єднувати в собі повністю весь процес розробки ІС;
- повинні бути об’єднані;
- мусять бути прості у вивченні, а також самому розробленні;

- у своїй групі повинні бути універсальними;
- повинен бути можливим процес розроблення в режимі інтерактивної взаємодії програміста з ЕОМ;
- важливою є спроможність дозволяти створювати адаптивні ІС;
- розробка мусить бути ефективною і економічною.

2.2. Аналіз якості та вимоги експлуатації бензину

Суттєва частина бензину на наших автозаправних станціях являється незадовільною з точки зору якості. Для споживачів певною мірою задоволення потреб прямо пропорційне рівню якості, яка визначається окремими, частковими показниками. Але головним показником для експлуатації бензину для споживачів є детонаційна стійкість бензину. Детонаційну стійкість ми можемо побачити в цифрах різних марок бензину, що показують октанове число, наприклад: А-92 або А-95. І тільки вищезазначена характеристика має користь, оскільки вона реально відображає роботу транспортного засобу, тобто характеристика оцінює рівень користі і шкоди палива для автомобіля. Якщо стандарт і реальне значення стійкості не відповідають один одному, то це призводить до ряду різних проблем, наприклад, від найвживанішої проблеми перевитрати пального, до шкоди навколишнього середовища, що прямопропорційно впливає на життєдіяльність оточуючих.

У великій кількості випадків під однією маркою бензину водію дають абсолютно іншу, яка кардинально відрізняється від заданих показників октанового числа.

Експерименти в області нафтопереробки проводять і по окремому показнику, і по повному дослідженню, щоб відповідало стандарту.

Ця експертиза містить в собі саму суть вивчення палива, зокрема бензину. До окремих об'єктів цієї експертизи відносять всі існуючі типи палива (бензину).

Експертизу проводять без залежності від місця, наприклад, для визначення виду речовини, або групи до якої вона належить.

При проведенні експертизи виявляють склад, групу і рід, до яких належить та чи інша перевіряюча речовина, при чому експертиза проводиться з відповідним рівнем якості до еталону і деяких інших послуг, якщо це необхідно.

Бензином вважається продукт, що є наслідком нафтопереробки, який використовується і вживається для тих чи інших механізмів. Взагалі існує не тільки автомобільне, але й авіаційне паливо. Вищезазначені палива застосовуються для двигунів внутрішнього згорання і для роботи мають необхідність примусово запалитись за допомогою іскри. Обидва ці палива мають спільні і навіть подібні характеристики, але вони використовуються в різних зонах.

Швидко і оперативно визначити за октановим числом бензин можна за допомогою портативних контрольних засобів. На сьогодні доступно багато різних засобів для контролю за якістю бензину, які проводять прямопропорційну залежність діелектричної проникності палива від октанового числа. Але результати такого контролю не підтримуються органами, що контролюють і слідкують за якістю палива, оскільки саме дослідження і експеримент проводиться за допомогою спрощених засобів вимірювання електричними моделями контролю. Така проблема вирішується двома способами: або дуже простими прийомами (ареометрами), або за допомогою використання складного устаткування.

Необхідні метрологічні характеристики з вищезазначених перші не можуть забезпечити, а другі мають потребу в постійних умовах і протяжних вимірюваннях. І тому розробка різних методів і засобів для перевірки автомобільного палива, щоб швидко оцінити його якість, є потрібною і назрілою темою. Ідентифікаційна експертиза проводиться під час виконання контролю якості даних взятих зразків продукції і направлена на те, щоб боротись з різними фальсифікаціями. Боротьба з такими фальсифікаціями є

актуальною, тому що така неякісна продукція приводить до збитків, при чому значних збитків, особливо страждає економічна складова і навколишнє середовище, адже вихлопні гази з неякісної продукції приносять дуже велику шкоду навколишньому середовищу. Окремо це відноситься до бензину, бо це найпоширеніше паливо і через різні особливості виробництва і транспортування набагато частіше піддається імітації.

Одною з важливих проблем, що найчастіше трапляється на очі експертам, заключається в тому, що в реальності ніщо не має абсолютної визначеності. Ми, як люди, здатні приймати вірні рішення, але вони можуть опиратись на часткову, не зовсім вірну, спотворену інформацію. Якщо узагальнювати, то, практично, найважливішою проблемою цілої науки є спроба втілити і використовувати міркування людини в сучасні комп'ютерні системи.

Хоч в сучасних базах даних є дуже велика кількість інформації, серед них є вагома частка неповної або часткової інформації, яку потрібно ще опрацювати. Особливо це стосується областей де яскраво виражається динаміка розвитку, сфера яких знаходиться у зоні ризику, в разі якщо інформація є неточною.

Отже ми мусимо враховувати фактор неповної інформації, що призводить до невизначеності, якщо ми хочемо вирішити проблему в реальності. Щоб описувати і працювати з даними в умовах неповної або невизначеної інформації найчастіше використовують теорію ймовірностей, але останнім часом досить часто використовують нечітку логіку.

2.3. Ідентифікаційна експертиза бензину

Завдання експертизи виникає від виробників бензину, що є серединою шляху на проміжку від виробників самого палива до станцій обслуговування, де заправляють автомобілі, це наслідок такої великої кількості підробок, тому необхідно шукати фальсифікати і боротись з ними. Потрібно знати виробника, що виробив бензин і його марку, ці дані знаходяться в лабораторіях на

дослідженнях нафтопродуктів, які забезпечують контроль над паливом і його якістю.

Автомобільний бензин отримується при переробці нафти і інших процесів, які мають суворий регламент фізикохімічних властивостей, але ми його отримуємо на виході після всіх переробок і маємо у вигляді звичайної суміші з різних компонентів.

Хімічний склад даного палива, може задовільняти нормативам, але він все рівно буде відрізнитись від іншого палива в залежності від того, яка сировина використовувалась і яка технологія застосовувалась при її обробці. Різний виробник бензину, робить бензин з своїми поправками, використовуючи інші компоненти, що допомагає відрізнити бензин одного виробника від іншого. Тому різна сировина, з точки зору хімічного складу, впливає на склад вже зробленого палива.

До лабораторії по контролю і перевірки якості палива не завжди потрапляють саме офіційні зразки продуктів, які мають паспорт, сертифікат якості і інші офіційні документи, в лабораторії часто звертаються неофіційні зразки, про яких іноді взагалі нічого не відомо.

Для палива, зразок якого був поданий офіційно, лабораторія перевіряє і вирішує задачу по контролю якості, яка заключається в тому, щоб порівняти задані характеристики з тими, що будуть отримані експериментально. Для зразків, які були подані неофіційно, лабораторія мусить визначити марку палива, його якість і обов'язково виробника, щоб заборонити продаж палива в разі виявлення фальсифікату.

2.4. Застосування автоматизації при вирішенні задачі експертизи

Людська діяльність з виходом на ринок персональних комп'ютерів, швидким розвитком комп'ютерних систем по управлінню базами даних, отримала великий поштовх і розвиток по автоматизації людської діяльності. З точки зору проблеми, що досліджується системи управління можуть

застосовуватись для збору і операцій з даними, а дані про кожний експеримент систематично додаються в базу даних. Сама автоматизація процесу в лабораторіях по перевірці і контролю якості палива і інших товарів має просто незаперечні переваги: це і значне скорочення часу на експертизу, і зведена до мінімуму кількість помилок, розробка сучасних мобільних лабораторій, яка дозволяє користуватись і проводити ідентифікаційну експертизу не тільки експертам, але й тим, хто не вважається екпертом в цій галузі, часто дослідження навіть проводять лаборанти за допомогою таких мобільних лабораторій.

У темі створення і розробці інтелектуальних комп'ютерних систем, що займаються автоматизацією процесів, що відносяться до розумової діяльності людей, проводиться дуже велика кількість різних досліджень. Під самою інтелектуальною системою мають на увазі системи, що навчаються самі, які можуть самостійно проводити логічні аналізи, синтези, проводити пошук по асоціаціям і т.п.

Виокремлюють два різних підходи при розробці інтелектуальних систем. Перший визначає створення програмно-апаратного комплексу, де методами штучного інтелекту реалізується певна частина функцій і сама інтелектуалізація системи не може бути основною метою. Наступний підхід має на увазі розробку інтелектуальної системи, щоб вирішувати певні проблеми, де інтелектуалізація має головну роль і мету.

Інтелектуальні системи мають свою класифікацію, вони поділяються на статичні і динамічні. Динамічні системи ми любимо називати "системами реального часу", вони також мають свої властивості і різницю в залежності від того в які обмеження швидкодії розробник ставить систему, а поділяються вони на: системи псевдо реальності, м'яку реальність часу і жорстку реальність. З вищезазначеного, перші з порівнянням від статичних отримують і обробляють дані, що доставляють їм ззовні і вирішують завдання швидше, ніж ми отримуємо якісь вагомні зміни оточуючого світу. М'які системи можуть допускати, щоб реакція на те, що відбувається була більше, ніж одна десята або

одна секунда. До жорстких систем прийнято ставити суворі умови, такі системи зобов'язуються реагувати швидше ніж пройде одна десята секунди.

При створенні різних інформаційних систем для опрацювання великих обсягів даних (біологічних, авіаційних, енергетичних і т.п.), з'являються кожного разу нові і складніші задачі з пошуку, опрацювання і автоматизації, їх не можна вирішити звичайними методами за допомогою штучного інтелекту або математичної статистики, або разом взятого штучного інтелекту і статистики. Щоб вирішити такі помилки застосовують нечітку логіку, еволюційні, генетичні алгоритми, штучні, нейронні і імунні мережі. Все вищеперераховане об'єднане в одну назву "м'які обчислення".

2.5. Методи тестування програмних систем

Тестування – обов'язкова частина програмної інженерії, є одним з варіантів покращення якості створених проектів, використовуючи цей процес, розробник усуває різні дефекти і баги, які могли залишитись не визначеними іншими методами перевірок.

Головною метою таких досліджень з інженерії тестування є розробка різних моделей і методів проведення тестів, щоб покращити обробку даних, створити окремі концепції теорії і практики в сфері тестування. Є багато напрямів дослідження в даній сфері, але головними вважаються:

- розбір і створення систематизованих методів тестування;
- розбір і обговорення вже існуючих методів надійності, створення нових алгоритмів і програм для їхнього представлення;
- створення на основі існуючих методів тестування нових корисних методів інженерії тестування;
- вирішення загальної структури процесу, який заплановує всі майбутні дії створення проекту, подальшої перевірки і оцінювання отриманого в результаті проекту, створення тесту систем опрацювання даних;

- розглядання сучасних методів і засобів покращення тестування і створення тесту для оцінки самого процесу тестування.

Висновки досліджень мають пряме відношення до інженерії тестування, завдяки цьому створили напрям програмної інженерії, що досліджує висновки досліджень, які були опубліковані (наприклад в звіті або методичках).

Якщо дивитись на неможливість поглибленого тестування складних програмних систем і також введення ліміту на час, вартість тесту, перед програмістами завжди з'являлась проблема правильної розстановки дій з тестування, щоб рівномірно розкинути ресурси і прийняти вірне, логічне рішення відносно початку і кінця тесту програмних систем.

В програмній інженерії, в загальному, як дисципліні створення програмних проєктів, головним вважається зосередження на покращенні якості і результативності програмних систем, використовуючи нові методи вимірювань і оцінки. Тобто, всі можливі типи розробки і користування програмним продуктом [3, 4].

В сучасності інженерний підхід має свої вимоги до тестів програмних систем, в яких основні є:

- перенесення початку тесту з кінця життєвого циклу програми на більш ранні. Це допомагає відійти від старої каскадної моделі до боку ітеративних моделей і створення процесу, що може себе керувати за допомогою тестування;
- вибір моделей і стратегій тесту, що задовольняють об'єктів і їхні рівні тестування, окрім того не підпадають під певні обмеження ресурсів у проєктах;
- шаблон процесу тестів, що також враховує вимоги програмної інженерії;
- пошук і створення певної кількості причин для припинення тесту.

В умовах дефіциту ресурсів для проєктів раптово велике значення стало мати керування ризиком, особливо ризиком того, що терміни написання програми зірвуться або вартість тестування не складеться в рамки, що має відношення не тільки процесів для управління тестів, але і знаходження

доцільних розмірів тестів і присвоєння кожній цілі своє місце для тесту враховуючи ризик відмови системи при виконанні програми.

Отже, проблема формування інженерного підходу до тестування - інженерії тестування, - важлива та актуальна складова програмної інженерії.

З початку 90-років проводилось багато досліджень в програмній інженерії і в результаті на програмну інженерію почали дивитись з іншої точки зору, тобто переосмислили методи і моделі в тестуванні, а також саму роль тестування. Звичайне, загальноприйняте уявлення про тести програмної системи для вирішення помилок, які допускаються на стадії розробки, кодингу і відлагодження, дало місце новому усвідомленню тестування у формі запланованого, цілеспрямованого тестування програмної системи на будь-якій стадії проекту.

Пізніше вищезазначений погляд на тестування має відбиток у У-моделі тесту, що впорядковує всі можливі дії, які пов'язані з цією моделлю. Така модель систематизує і впорядковує пару подій, які йдуть одна за одною: ліва гілка У-моделі, яка готова до тестування на протязі всіх випробувань і саме виконання тесту життєвого циклу програмних систем, як зазначено в регламенті ДСТУ і безліччю методів для окремих класів систем.

Дії підготування до тестів поділяються на певні етапи розробки програмних систем, які передують випробуванням. На кожному окремому кроці проводиться розбір відповідного цьому етапу розробки, щоб позначити цілі, об'єкти, сценарії і певні ресурси тестування, які відповідають заданому рівню випробування. Вищезазначений аналіз має дуже важливий і особливий побічний ефект. З одного боку поліпшується якість розглянутих напрацювань після виправлення некоректності, а з іншого боку – це розбір модельованих ситуацій, оптимізація кінцівки тестування.

З розвитком ітераційних моделей життєвого циклу виконання тестування зміщується на більш ранні етапи, бо ітераційні моделі можуть запуснути тест після готовності до тесту найпершої версії. Хоч і тестування наступних версій

включає в себе і попередні, це дає можливість провести ретельний тест цілої програми.

Стратегії тестів життєвого циклу ітераційних моделей розробляються з ідеєю умов примінення програмних систем. Метод побудови тестів відповідає правилам застосування програмних систем.

Метод побудови стратегії тестування, що відповідають зазначеним вимогам застосування програмних систем розвивався в сфері профілів програмних систем – опирався на базові принципи інженерії надійності. Профілем називається ціла множина додаткових сценаріїв роботи програмних систем, при чому появись може кожний з них з розрахованою ймовірністю появи. Під час процесу аналізу проблемних областей, множини поставлених завдань можуть будуватись з різними профілями, наприклад з профілем користувача або функціонального профілю і т.п.

Дивлячись на певні рамки часу для тестів – актуальною є проблема пошуку оптимального запуску програмної системи. Оптимальний час запуску залежний від трьох чинників, між якими потрібно знайти рівновагу, це надійність, ціна і час. Щоб вирішувати дану проблему було притінено цілу низку підходів, які засновані на окремих припущеннях, критеріях і правилах користування програмних систем, зокрема [5-7].

Найвідоміші підходи вирішення вищезазначених методів це дві крайності, удосконалення надійності програмної системи до максимуму і зниження витрат до мінімуму. Дана задача підходить до вбудованих систем, яку можна вирішити за допомогою використання динамічного програмування [8-11].

Маючи мету подбати про користувача, було розроблено підхід до оптимального поділу надійності програмних систем перед початком тестування, коли відомо технічні і цінові рамки [12-15].

Серед всіх вищезазначених підходах ніде не враховується серйозність того, що буде при відмові, розглянута тільки ймовірність цього. Також, з того часу різні напрямки тестів не знаходились в одній цілій темі.

Через це головні напрямки експериментів були спрямовані на формуванні ідеї тестування програмних систем, що в собі буде містити всі інші види діяльності здійснення тестів.

Окремо досліджуються створення певних моделей оцінки тестів для яких тестування підлягають структуризації на основі певних рівнів досконалості. Вищезазначені моделі призначаються і для аналізу всього процесу перевірки, і для пошуку нових шляхів поліпшення програмної системи.

2.6. Основні результати досліджень та методи інженерії тестування

При розробці програмних систем обов'язковою складовою є використання зрозумілих, адекватних методів, що забезпечує ефективність самого процесу. При експериментах з методами тестування прикладного забезпечення, процеси обробки даних були в рамках НДР [1 - 2]. Для проведення повного функціонального тестування прикладного програмного забезпечення були створені і підведені методи в системах обробки даних, наприклад метод "Розбиття на категорії".

Для кожного існуючого методу є певний критерій для набору вхідних тестів. Є важливим вибрати найбільш ефективний метод тесту з певними, незвичайними умовами для різного рівню тестування, це є складною проблемою, яка пов'язана з ризиком відмов програмних систем. Застосування методів рахується, як одна з задач, яку потрібно вирішити під час самого процесу побудови тестування, зробити аналіз і впорядкувати типові методи, які підходять до специфічних типів програмних систем.

Моделі тестів відрізняються різними підходами до розробки самих тестів. Традиційно їх розділяють тільки на дві частини – без доступу до відправного коду (чорний ящик) і з доступом до відправного коду (білий ящик) [16].

Але також є розширений список методів тестування, що заснована на різних підходах до початку проектування, список наведено в [17] і показано на рис. 2.1.

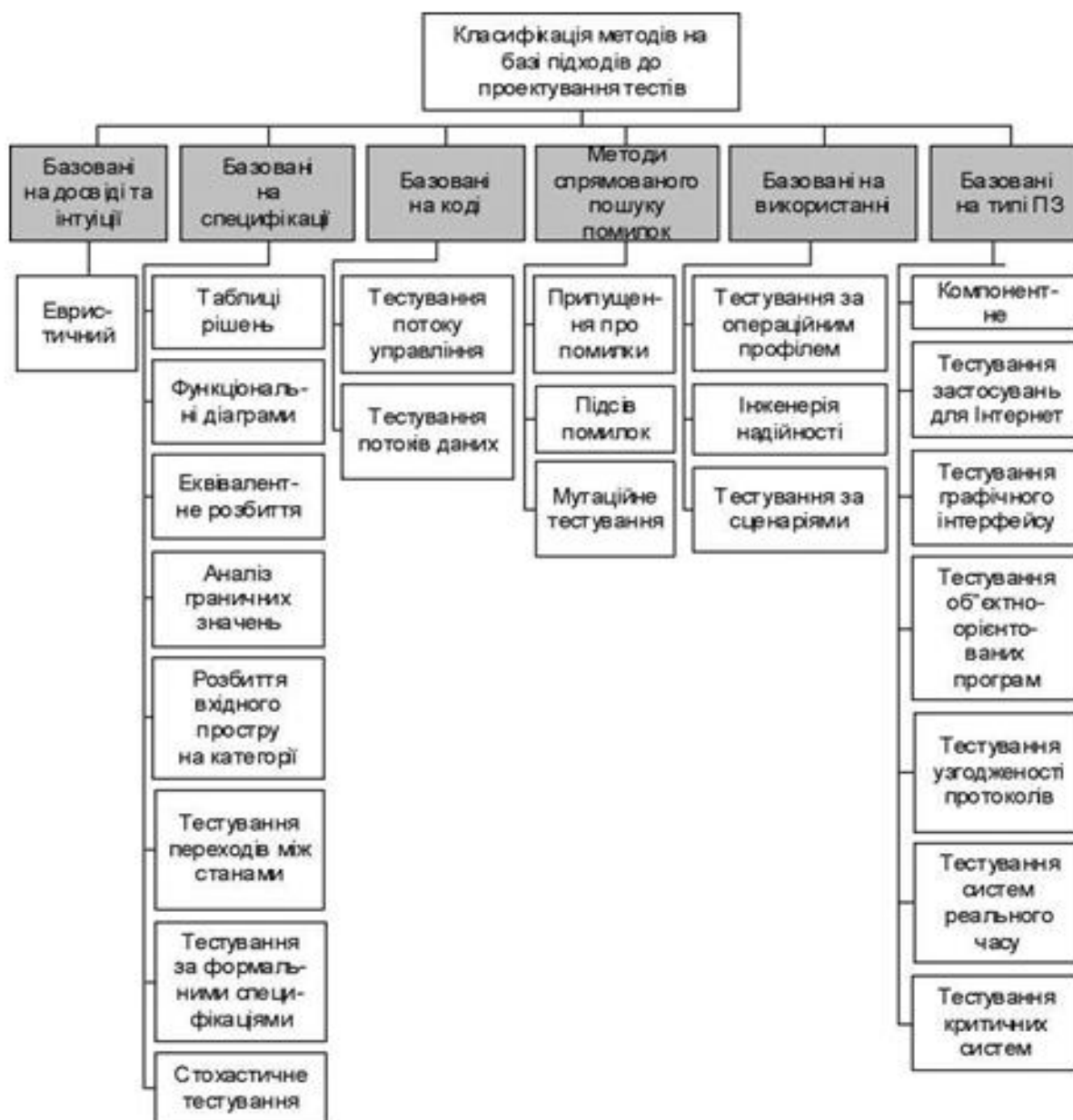


Рис. 2.1. Класифікація методів тестування

Метод використання ітераційних прототипів, як спосіб специфікації отримав право на життя з початку 90-х років, коли потужні САБЕ-інструменти (в таких інструментах автоматизовано специфічні інструменти і функції програмних систем) не були широко використовувані.

Метод ітераційних прототипів використовується і застосовується для запланованих тестувань при умові врахування моделі програмних систем, окрім цього, метод використовується для експериментів в об'ємах програмних систем.

2.7. Стратегії тестування, які базуються на аналізі ризику відмов

Практично найважливішою проблемою інженерії тестування є пошук потрібного розміру тесту і чисельного критерію припинення тестування. Аналізуючи вже відомі критерії, зрозуміло, що велика частка з них заснована на інженерії надійності.

В умовах дослідження тестувань, що були створені на основі модульного принципу, було прийнято оптимальний випуск систем, який заснований на ризиках відмов. Враховуючи профіль функціонування системи за допомогою сформованого методу оптимізації [18]. В центрі такого підходу лежить метод, що оцінює ризики відмов і модель, щоб визначити оптимальний час тесту.

Розробка профілю програмних систем це складова методу ризику і стратегії тестів, завдання якої у визначенні програмних компонентів, які вимагають довшого і детальнішого тестування [19, 20]. На рис. 2.2 показано головні етапи методу оцінки ризику відмов.

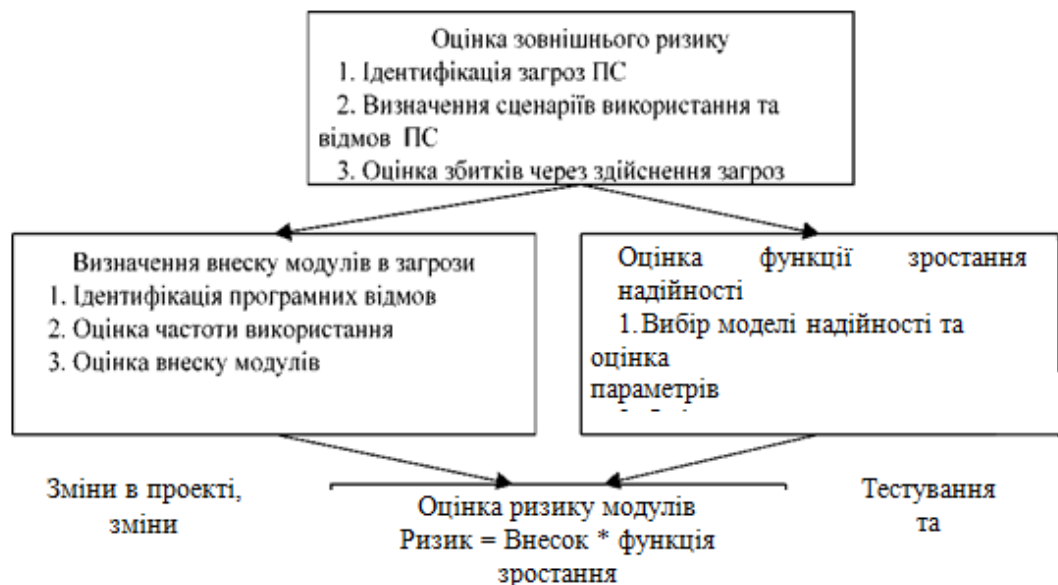


Рис. 2.2. Етапи методу оцінювання ризику відмов

Даний процес зображується у вигляді множини різних завдань з підготовки та оцінки результату тестів, які поділяються на 10 кроків процесу (рис. 2.3).

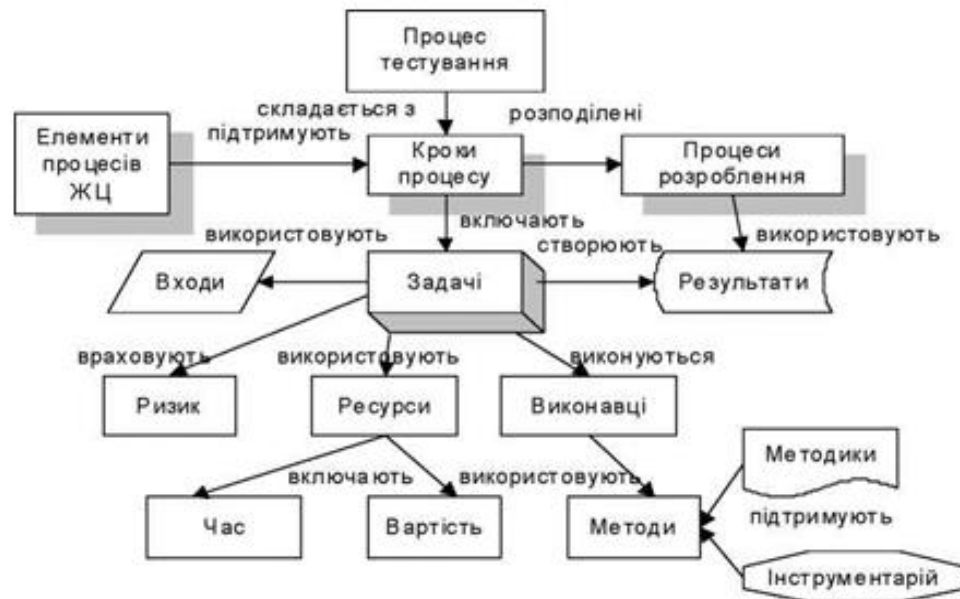


Рис. 2.3. Модель базового процесу тестування

Під час процесу підготовки створення програмних систем проводиться аналіз продуктів на всіх рівнях підготовки, без винятків, причиною проведення такого аналізу є необхідність визначити певні сценарії і ресурси тестування, які будуть відповідати певному рівню тестування. Висновки виконаних кроків повинні бути зафіксовані в створеному плані тестування.

Під час виконання процесів на кожному кроці при здійсненні фіксації висновків виконаних кроків потрібно порівнювати їх з результатами, які були очікувані на стадії планування програмних систем.

Висновки виконаних кроків аналізуються, щоб бути в курсі поточного стану програмної системи і вирішувати чи система була добре протестована на цьому рівні.

При кожному кроці створення проекту є затверджений набір завдань, які потрібно вирішити. При визначенні оптимального часу, щоб провести тестування використовують еталон припинення тестування. Оцінювання ризику відмов є частиною задачі, що відноситься до підготовки тестів на всіх рівнях.

При звичайному процесі роботи програмних систем, базовий процес розраховує: поділ завдання для всіх учасників проекту, вимоги до підготовки користувачів процесу, регламент документів, критерії початку, виконання і

завершення задачі. Щоб зберегти всю інформацію було розроблено шаблон, який поєднується з кожним кроком процесу.

2.8. Моделі та методи вдосконалення процесу тестування

Під час проведення досліджень інженерії тестування для виконання задачі базового процесу тестування було створено і розроблено відповідні методи для оцінки цього процесу. В кінці досліджень було застосовано існуючі моделі до моделей вітчизняних проектів, розроблено метод зрілості процесу тестування, а також метод експертної оцінки проекту.

Модель зрілості процесу тестування була розроблена і створена на основі моделі ТММ (п'ятирівнева модель). Кожен з рівнів має відношення до цілей, які заплановано досягти і вони відповідають зазначеному рівню зрілості. [21, 22]. Щоб перейти на верхній рівень, потрібно закінчити всі дії на нижньому рівні.

Кожний рівень, починаючи від першого (перший не рахується) написаний в своїй структурі. Структура характеризується рівнем зрілості, які описуються з діями тестування. Кожен рівень складається з:

- Прийомів засобів зрілості, які розрізняють ціль вдосконалення тестування;
- Під прийомів зрілості, що розрізняють межу, необхідні результати, щоб досягти поставлених завдань на кожному рівні;
- Дій і завдань для реалізації моделі до умови оцінювання.

Метод оцінювання зрілості виконуваних у проектах дій з тестування ПС ґрунтується на представленій моделі ТММ та полягає в анкетуванні провідних фахівців проектів та згортці результатів анкетування [23]. Використати оцінку можна, щоб визначити положення тестування і напрямку його поліпшення.

На даний момент дослідження з інженерії тестів стосуються створення нових методів тестів, а також поліпшення самого процесу тесту для систем

нової генерації – здебільшого для Web-сервісів. Вищеперераховані дослідження засновані на регламентах, створюваних у вказаних рамках досліджень.

2.9. Висновки до розділу 2

В даному розділі розглянуто моделі і методи тестування програмних систем. Проведено аналіз якості бензину та вимоги його експлуатації.

В даному розділі було проведено дослідження, що при розробці програмних систем обов'язковою складовою є використання зрозумілих, адекватних методів, що забезпечує ефективність самого процесу.

Визначено методи ідентифікаційної експертизу бензину та розглянуто способи застосування автоматизації при вирішенні задачі експертизи.

Розглянуто методи тестування програмних систем та основні результати досліджень. Під час процесу підготовки створення програмних систем проводиться аналіз продуктів на всіх рівнях підготовки, без винятків, причиною проведення такого аналізу є необхідність визначити певні сценарії і ресурси тестування, які будуть відповідати певному рівню тестування. Висновки виконаних кроків повинні бути зафіксовані в створеному плані тестування.

РОЗДІЛ 3

СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ІДЕНТИФІКАЦІЙНОЇ ЕКСПЕРТИЗИ БЕНЗИНІВ

3.1. Структура автоматизованої системи ідентифікаційної експертизи бензинів

Використовувати системи можуть працівники лабораторії оцінки якості палива і інших нафтопродуктів: експерти, приймальники.

Створена система призначена для роботи у лабораторії оцінки якості.

В даній системі вбудовано наступні функції:

- збирання інформації про зразки продукції, занесення їх в базу даних і створення вільного доступу до збереженої інформації;
- оцінка якості наданого зразка продукту на основі випробувань згідно з регламентом;
- достовірність взятого зразку продукції відповідно до поставленої задачі.

Розглянемо структуру створеної автоматизованої системи.

На вхід системи дані приходять ззовні, а саме з встановлених зовнішніх датчиків.

На виході при завершенні роботи система дає дані про взятий зразок продукту.

Проведення експертизи можливе не тільки в стаціонарній, але і в мобільній лабораторії.

В табл. 3.1 показано особливості атрибутів «Зразок».

Таблиця 3.1

Особливості значень атрибутів сутності «Зразок»

«Статус»	«Виробник»	«Власник»	«Постачальник»
Стандартний	Обов'язкове, взяте із списку відомих виробників	Фіксоване значення «лабораторія»	Будь-який
Скарга	Невідомий	Фіксоване значення «лабораторія»	АЗС
Експериментальний	Не встановлений	Нема обмежень	Нема обмежень
Контрольний	Невідомий	Фіксоване значення «лабораторія»	АЗС
Розвідувальний	Невідомий	Нема обмежень	АЗС

Конфігурація вигляду результату залежить отриманої вхідної інформації про зразок, практично характеризується таблицею «Зразок» (табл. 3.1). Структурна схема системи зображена на рис. 3.1.

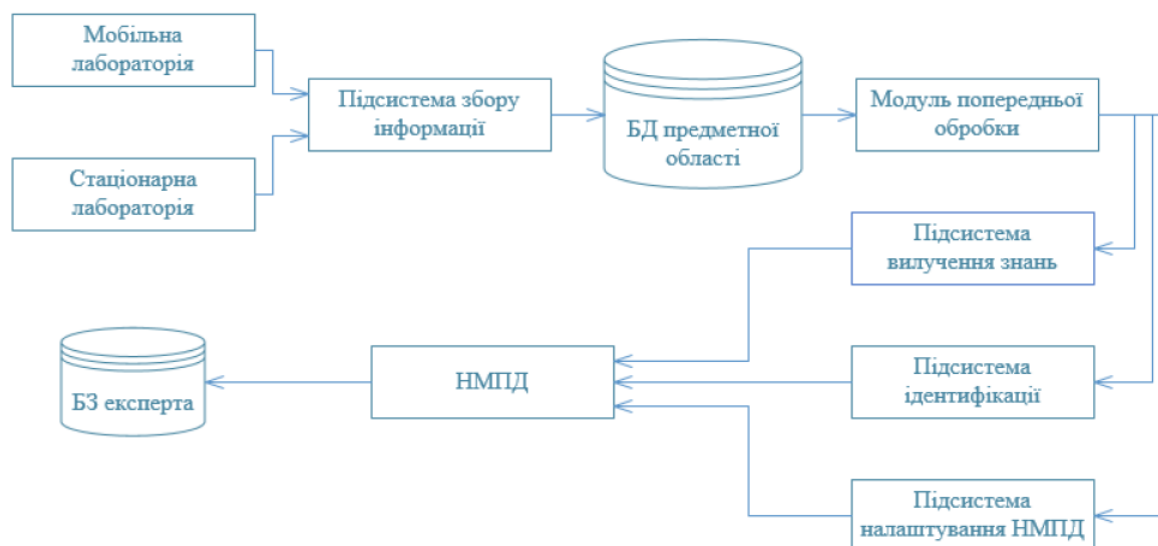


Рис. 3.1. Структурна схема АСІЕ

Нагромадженням знань в базу даних займаються три різних підсистеми: підсистема зібрання інформації, підсистема вилучення і підсистема

налагодження НМПД. У базі знань, по суті, нагромаджуються експертні знання, що знаходяться у вигляді нечіткої моделі, що лежить в основі системи.

Модуль попередньої обробки – стосується методів введення/виведення даних, вірності отримуваних даних і подальших методів обробки.

Модуль адміністрування БД – займається пошуком в базі, дані вивантажуються/завантажуються пакетно, створює звіти.

Підсистема, що займається ідентифікацією містить в собі основну частину технології нечіткої класифікації.

Система працює з зразком продукції і це її основна сутність. Прибори виконують експертизу зразку за допомогою фізико-хімічних методів.

В результаті експертизи зразка у висновку експерта, що видає система є тільки два можливих варіанти відповідей. Заявлені характеристики зразку спростовуються або підтверджуються.

Аналіз зразку не є окремим цілим процесом, він виконується в три етапи: контролювання зразку, експертиза і рішення про зразок.

В першому етапі контролю зразок підлягає порівнянню зразка до державних стандартів ДСТУ 4839:2007 або ДСТУ 4063-2001 після чого ставлять вердикт: стандартний - нестандартний.

Експертиза проводиться виконаннями експериментів, які зазначені умовами.

Всі отримані результати від експериментів збираються до купи і аналізуються, щоб прийняти рішення. Але іноді потрібно проводити додаткові дослідження, якщо є специфічні умови.

Визначення марки бензину і його виробника потрібне в таких випадках: щоб розвіяти сумніва «скаржника», розвіяти сумніви оптового клієнта і чи можна приймати продукт на нафтобазу, а також решта спірних питань. До кожного випадку є свої умови в поставленій задачі, щоб прийняти вірне рішення при розробці системи було передбачено специфічні форми введення.

Експертні знання, щоб вірно проаналізувати зразки знаходяться в базі знань, а набір зразків з датчиків, що зчитують зовнішню інформацію знаходяться у базі даних предметної області.

3.2. Структура та опис бази даних

На етапі створення системи було розглянуто різні моделі баз даних для автоматизованої програмної системи експертизи, вибір пав на реляційну модель бази даних.

Предметна область описується в базі даних.

Згідно з результатами аналізу, що були отримані в результаті експертизи лабораторії нафтопродуктів, була створена і представлена структура бази даних, зображення фізичної моделі буде представлено в рис. 3.2.

Для управління баз даних використовується SQLite, бо вона безкоштовна, її підтримують більшість середовищ розробки і вона популярна.

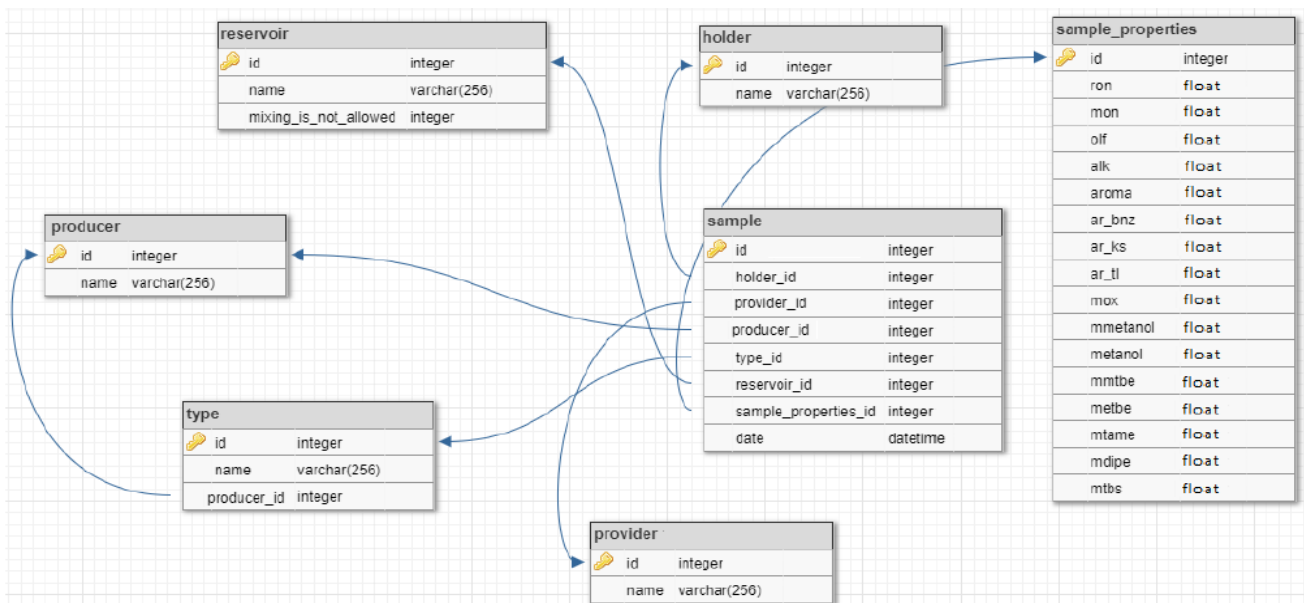


Рис. 3.2. Фізична модель бази даних предметної області

В табл. 3.2. перераховано всі назви таблиць баз даних.

Таблиця 3.2.

Імена таблиць бази даних

Назва таблиці в БД	Сутність
holder	Власник зразка палива
provider	Постачальник палива
producer	Виробник палива
type	Марка палива
reservoir	Резервуар, з якого отриманий зразок палива
sample	Зразок палива
sample_properties	Властивості зразка палива

Таблиця 3.3.

Опис таблиці holder

Стовбець	Тип	Значення
id	int	Ідентифікатор власника
name	varchar	Ім'я власника

Таблиця 3.4.

Опис таблиці provider

Стовбець	Тип	Значення
id	int	Ідентифікатор постачальника
name	varchar	Ім'я постачальника

Таблиця 3.5.

Опис таблиці producer

Стовбець	Тип	Значення
id	int	Ідентифікатор виробника
name	varchar	Ім'я виробника

Таблиця 3.6.

Опис таблиці type

Стовбець	Тип	Значення
id	int	Ідентифікатор типу
name	varchar	Назва типу
producer_id	int	Ідентифікатор виробника

Таблиця 3.7.

Опис таблиці reservoir

Стовбець	Тип	Значення
id	int	Ідентифікатор резервуару
name	varchar	Назва резервуару
mixing_is_not_allowed	int	флаг, дорівнює 1, якщо для резервуара не допускається змішування палива різних марок від різних виробників

Таблиця 3.8.

Опис таблиці sample

Стовбець	Тип	Значення
id	int	Ідентифікатор зразка
provider_id	int	Ідентифікатор постачальника
producer_id	int	Ідентифікатор виробника
type_id	int	Ідентифікатор типу
reservoir_id	int	Ідентифікатор резервуару
sample_properties_id	int	Ідентифікатор властивостей
date	datetime	Дата надходження на експертизу

Таблиця 3.9.

Описання таблиці `sample_properties`

Стовбець	Тип	Значення
id	int	Ідентифікатор запису
ron	float	Ок.число дослідн. методом
mon	float	Ок.число моторним методом
olf	float	Вміст олефінів
alk	float	Вміст алканів
aroma	float	Сумарний вміст ароматичних вуглеводів
ar_bnz	float	Вміст бензолу
ar_ks	float	Вміст ксилолів
ar_tl	float	Вміст толуолу
mox	float	Вміст кисню
mmetanol	float	Метанол
metanol	float	Етанол
mmtbe	float	Метил-трет-бутиловий ефір
metbe	float	Етил-трет-бутиловий ефір
mtame	float	ТАМ ефір
mdipe	float	ДІП ефір
mtbs	float	Трет-бутиловий спирт

Описання таблиць баз даних знаходиться в таблицях 3.3 – 3.9

3.3. Підсистема ідентифікації виробника палива

В цій підсистемі проводиться налаштування НМПД, обробка нових даних і перезапуск моделі після результатів обробки, окрім цього також визначення виробника на основі створеної моделі.

НМПД характеризується списком нечітких продукцій і множиною лінгвістичних змінних. Такі моделі завжди впорядковані, тому їх можна записувати в звичайному вигляді. Щоб описати нечіткий класифікатор вирішено взяти XML- формат.

XML-формат відомий тим, що має ієрархічну структуру і непогано вирішує питання зображення складних моделей даних в системах.

Для прикладу, нечіткий класифікатор створюється використовуючи наступну ієрархію: <Класифікатор> <Правила> <Лінгвістичні змінні>. XML-формат це уособлення зводу синтаксичних правил для зберігання упорядкованих даних і використовується системою організації обміну даних. Було запропоновано новий підтип XML-формату під назвою «Fuzzy Classifier System», формату fcs.

На рис. 3.3. приклад fcs-документа, що описує класифікатор для системи, складом з п'яти правил і шести ознак.

Кореневий елемент «fcs» обов'язково мусить містити в собі два вузла: «model» і «fuzzyrules».

Структура «model» зберігає в собі загальну інформацію про класифікатор: кількість класів (вузол «CN»), кількість атрибутів (вузол «AN»), назви образів (зображені в другорядних вузлах «class» вузла «classes»), назви атрибутів (зображені в другорядних вузлах «feature» вузла «features»), тип методу нечіткого виводу (вузли «Mftype», «AND», «OR»), обов'язково кількість в базі правил (вузол «RN»).

Структура «fuzzyrules» - описані всі правила «rule».

Належності описують додаткову інформацію: дату створення («date»), назву («id», «name»), версію («version») та інші важливі деталі.

Переваги використання XML-формату, щоб зберігати моделі даних:

- розвинена інтерпретованість висновків моделей користувачем;
- хороша взаємодія між окремими модулями створеної системи;
- побудувавши модель в цьому форматі, її можна використати в інших

системах.

```

1 <fcs version="1.0" id="test" date="10.09.2017">
2   <model>
3     <CN description="number of classes">5</CN>
4     <classes>
5       <class id="1">A92LN</class>
6       <class id="2">A95LN</class>
7       <class id="3">A95OR</class>
8       <class id="4">A80PR</class>
9       <class id="5">A76SH</class>
10    </classes>
11    <AN des="number of features">6</AN>
12    <features>
13      <feature id="1" type="integer" begin="0" end="100">OLF</feature>
14      <feature id="2" type="integer" begin="0" end="100">AROMA</feature>
15      <feature id="3" type="integer" begin="0" end="100">AR_BNZ</feature>
16      <feature id="4" type="integer" begin="0" end="100">HTBE</feature>
17      <feature id="5" type="integer" begin="0" end="100">RON</feature>
18      <feature id="6" type="integer" begin="0" end="100">NON</feature>
19    </features>
20    <RN des="number of rules">5</RN>
21    <AND>2</AND>
22    <OR>0</OR>
23    <MFtype>MFDiscrete</MFtype>
24  </model>
25  <fuzzyrules>
26    <rule consequent="1" weight="1" type="MFDiscrete">
27      <fs feature="OLF">
28        <size>63</size>
29        <x>12,89;13,20;13,32;13,45;13,57;13,69;13,82;13,94;14,06;14,19;14,31;14,43;14,56;14,68;14,80;
30 14,93;15,05;15,17;15,30;15,42;15,54;15,67;15,79;15,91;16,04;16,16;16,28;16,41;16,53;16,65;16,78;16,90;
31 17,02;17,15;17,27;17,39;17,52;17,64;17,76;17,89;18,01;18,13;18,26;18,38;18,50;18,63;18,75;18,87;19,00;
32 19,12;19,24;19,37;19,49;19,61;19,74;19,86;19,98;20,11;20,23;20,35;20,48;20,60;21,03;</x>
33      <y>0,00;0,40;0,40;0,43;0,48;0,50;0,48;0,50;0,50;0,52;0,52;0,48;0,50;0,45;0,43;0,40;0,43;0,43;0,50;
34 ;0,48;0,48;0,48;0,55;0,52;0,48;0,60;0,74;0,69;0,62;0,64;0,79;0,79;0,86;0,86;1,00;0,98;0,98;0,90;0,86;0,81;

```

Рис. 3.3. Приклад запису лінгвістичних змінних і правил нечіткої класифікації в форматі fcs

3. 4. Архітектура модуля ідентифікаційної експертизи

Цей модуль був створений для того, щоб реалізовувати моделі і методи в дипломній роботі для вирішення поставлених завдань розпізнавання образів.

Для написання бібліотеки була використана мова C ++.

Дуже важливою особливістю цієї бібліотеки те, що вона незалежна від предметної області.

Створення і розробка автоматизованих систем розпізнавання, ще одне призначення цієї бібліотеки.

На рис. 3.4. було зображено спрощену діаграму головних класів модуля.

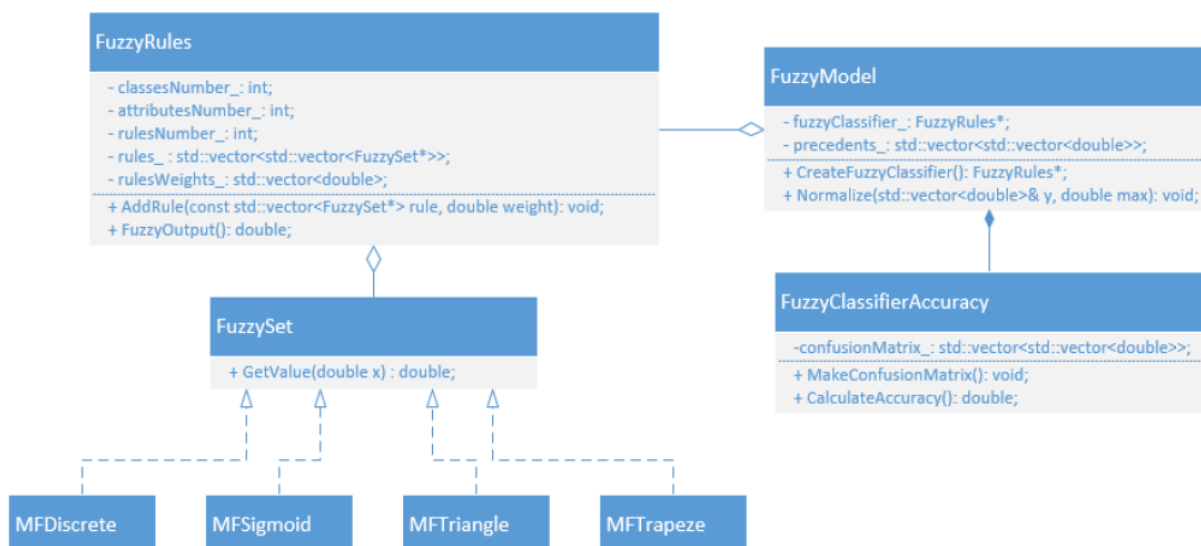


Рис. 3.4. Діаграма класів модуля ідентифікаційної експертизи

Для підтримування різних видів атрибутів був використаний `FuzzySet` (абстрактний). Клас `FuzzyModel` займається налаштуванням, повторним навчанням і приміненні нечіткого класифікатора, а клас `FuzzyClassifierAccuracy` займається контролем якості. Частина класу `FuzzyRules` займаються подаванням нечіткого класифікатора, кожна умова написана табличному вигляді, а лінгвістичні змінні через посилання на частини класу `FuzzySet`.

Клас `FuzzyModel` на основі множини прецедентів формує модель класифікатора. Клас `FuzzyRules` уособлює собою нечіткий класифікатор, що регулюється набором правил. Результат класифікації представляється у вигляді вектора впевненості α . Віртуальний клас `FuzzySet` для класів, що займаються описом функцій належності окремих типів є батьківським. Робота цього класу в тому, що він забезпечує, щоб робота класифікатора працювала незалежно з функціями будь-якого типу.

Класи, що реалізують функції приналежності `MFDiscrete` (сплайн першого порядку), `MFSigmoid` (сигмоїдальна), `MFTriangle` (трикутна), `MFTrapeze` (трапецієдальна). Завдання класу `FuzzyClassifierAccuracy`, оцінка якості створеного на основі моделі типу `FuzzyModel` класифікатора типу `FuzzyRules`. Оцінка проводиться в результаті порівняння вірного результату з тим результатом, що був отриманий за допомогою нечіткого класифікатора.

3.5. Опис автоматизованої системи ідентифікаційної експертизи

Автоматизована система ідентифікаційної експертизи – програмна система, в якій компоненти написані на мовах C++ (логіка програми) та C# (графічний інтерфейс). Щоб працювати з даними використовується SQLite.

Була розроблена спеціальна бібліотека класів для роботи з нечіткими моделями даних, вона називається “Бібліотекою для нечіткого розпізнавання” (Fuzzy Pattern Recognition Library).

Щоб зв’язувати компоненти в системі в програмній системі було використано представлення нечіткого класифікатора в xml-форматі.

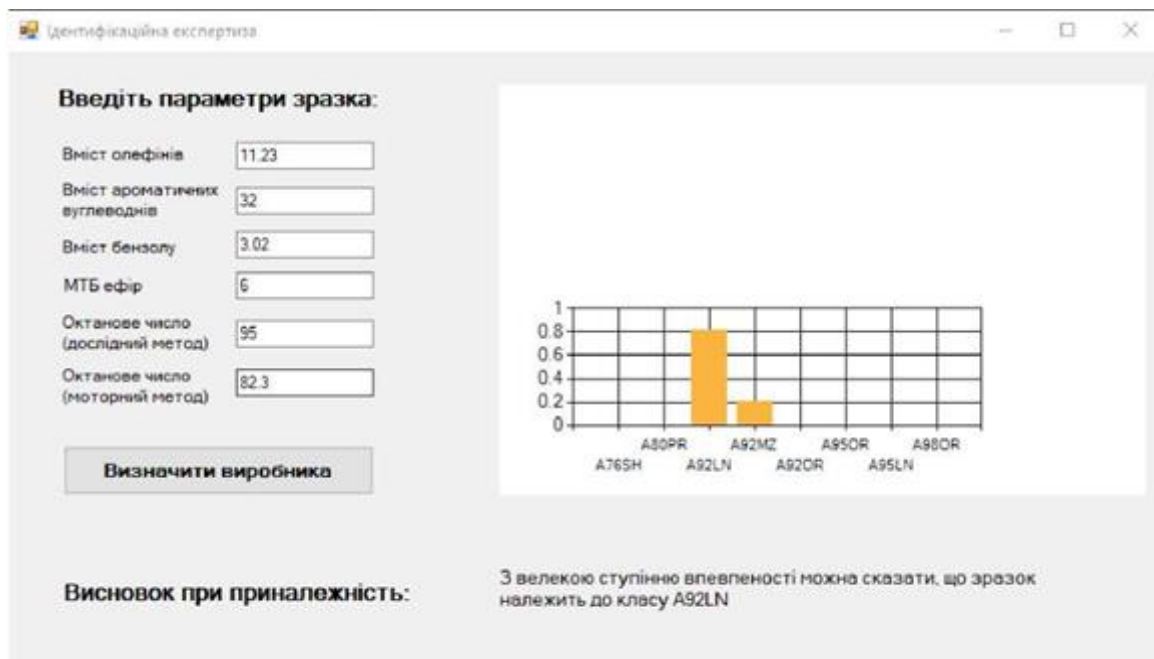


Рис. 3.5. Інтерфейс програми ідентифікаційної експертизи бензинів

Щоб представити нечітку модель даних була створена нечітка модель з методами її управління, а також розроблено комфортний і приємний візуальний інтерфейс користуючись мовою C# (рис. 3.5.).

Результат експертизи класифікатора подається у вигляді нечіткого інформаційного вектора, тому щоб оцінка була повною і достовірною, то результат подають у трьох формах: візуальній (за допомогою стовпчикової

діаграми), вербальній (лінгвістична інтерпретація значень вектора) і чисельній (нечіткий поріг).

3. 6. Висновки до розділу 3

Даний розділ присвячений побудові автоматизованої системи ідентифікаційної експертизи на прикладі бензинів. Побудована організаційна та функціональна структури для автоматизованих систем «м'якого» реального часу, призначеної для вирішення задачі ідентифікаційної експертизи продукції. Крім нафтопереробної галузі, система може бути використана, наприклад, в харчовій, хімічній та інших промисловостях.

Виконано аналіз інформаційних потоків контролю якості нафтопродуктів і побудована інформаційна модель предметної області. Це дозволило виділити структуру класів образів, яка використовується як вхідні дані задачі розпізнавання образів і побудувати реляційну базу даних інформаційної системи.

В основу методу ідентифікації зразка продукції, які використовується в інформаційній системі, покладена інформаційна технологія нечіткої класифікації. Розроблена бібліотека класів образів, яка реалізує всі математичні моделі і методи, запропоновані в дипломній роботі.

РОЗДІЛ 4

ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення науково-дослідних робіт

Економічне обґрунтування дипломної роботи магістра є суттю даного розділу, оскільки, дозволяє встановити доцільність проведення науково-дослідних робіт і економічно обґрунтувати доцільність застосування тих чи інших засобів.

Метою дипломної роботи магістра є дослідження методів та засобів побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину.

Як відомо, розробка надійної і ефективної системи вимагає значних затрат часу. Слід зауважити, що затрати часу залежать від кваліфікації розробника і його можливостей. Розробник повинен у достатній мірі володіти навиками програмування, вміти адекватно застосовувати математичний апарат, бути добре обізнаним з об'єктом дослідження.

Розробку даної системи можна поділити на такі етапи:

- 1) постановка задачі;
- 2) збір інформації по тематиці роботи наступне її опрацювання;
- 3) прийняття рішень щодо вибору оптимального шляху розв'язання поставленої задачі;
- 4) аналіз математичної моделі та методів побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину;
- 5) розробка алгоритму програми для оцінювання якості бензину;
- 6) налаштування середовища розробки і роботи вже готової програми;
- 7) написання програми;
- 8) написання і оформлення документації.

Для оцінки тривалості виконання окремих робіт використовують нормативи часу або попередній досвід.

До таких нормативів відносять тривалість написання операцій (команд), які в деяких підприємствах становлять: для одної операції - 0,5-1,6 год та 8 годин для п'яти операцій (тривалість зміни).

У разі їх відсутності звертаються до експертних оцінок по встановленню тривалості кожного етапу (стадії):

при трьох оцінках:

$$T_{ec} = (t_{min} + 4t_{н.й} + t_{max}) / 6, \quad (4.1)$$

при двох оцінках:

$$T_{ec} = (3t_{min} + 2t_{max}) / 5, \quad (4.2)$$

де T_{ec} – очікуване (середнє) значення тривалості виконання етапу (стадії); t_{min} , $t_{н.й}$, t_{max} – відповідно мінімальна, найбільш імовірна і максимальна оцінки тривалості виконання етапу (стадії).

Для визначення загальної тривалості проведення науково-дослідних робіт (розробки програмного продукту) доцільно дані витрат часу на виконання окремих стадій (етапів) звести у табл. 4.1.

Таблиця 4.1.

Основні етапи і час їх виконання у НДР

№ з/п	Етап	Середній час виконання етапу, год	
		інженер	керівник
1	2	3	4
1	постановка задачі	3	7

Продовж. табл.4.1

2	збір потрібної інформації і наступне її опрацювання	15	5
3	прийняття рішень щодо вибору оптимального шляху розв'язання поставленої задачі	3	2
4	аналіз математичної моделі та методів побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину	15	8
5	розробка алгоритму програми для оцінювання якості бензину	11	5
6	налаштування середовища розробки і роботи вже готової програми	3	1
7	написання програми	85	5
8	написання і оформлення документації	20	7
разом		155	40

Витрати часу наукового керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

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. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% – 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати – 21 дні/міс., що відповідає 168 год./міс. Розмір місячних окладів керівника та інженерів слід приймати згідно існуючих на даний час норм.

Основна заробітна плата розраховується за формулою:

$$Z_{осн} = T_c \times K_2, \quad (4.3)$$

де T_c – тарифна ставка, грн.; K_2 - кількість відпрацьованих годин.

Посадові оклади (тарифні ставки) за розрядами Єдиної тарифної сітки визначаються шляхом множення окладу (ставки) працівника 1 тарифного розряду на відповідний тарифний коефіцієнт. У разі коли посадовий оклад (тарифна ставка) визначені у гривнях з копійками, цифри до 0,5 відкидаються, від 0,5 і вище - заокруглюються до однієї гривні.

У 2019 році посадові оклади (тарифні ставки) розраховуються згідно з Законом України «Про Державний бюджет України на 2019 рік».

Мінімальна зарплата в 2019 р. складає 4173,00 грн., в погодинному розмірі 25,13 грн., прийmemo :

для інженера - 80,00 грн,

для керівника – 130,00 грн.

Тарифні ставки:

керівник проекту – 130,00 грн./год.,

інженер – 80,0 грн./год.

Основна заробітна плата становитиме:

$$Z_{осн} = T_{осн} \times K_{год} \quad (4.4)$$

Керівник проекту:

$$Z_{осн} = 130,00 \text{ грн.} \times 40 \text{ год.} = 5200,00 \text{ грн.}$$

Інженер:

$$Z_{осн} = 80,00 \text{ грн.} \times 155 \text{ год.} = 12400,00 \text{ грн.}$$

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

$$Z_{дод} = Z_{осн} \times K_{додл}, \quad (4.5)$$

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

Керівник проекту:

$$Z_{дод} = 5200,00 \text{ грн.} \times 0,15 = 780,00 \text{ грн.}$$

Інженер:

$$Z_{дод} = 12400 \text{ грн.} \times 0,10 = 1240,00 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{он}$) визначаються за формулою (4.6) і становлять:

$$B_{он} = Z_{осн} + Z_{дод} \quad (4.6)$$

Керівник проекту:

$$B_{он} = 5200,00 + 780,00 = 5980,00 \text{ грн.}$$

Інженер:

$$B_{он} = 12\,400,00 + 1240,00 = 13640,00 \text{ грн.}$$

Таким чином загальна сума становить 19620,00 грн.

Крім того, слід визначити відрахування на соціальні заходи:

- податок на доходи фізичних осіб: 18% 3531,60 грн.;
- військовий збір 1,5% 294,30 грн.;
- єдиний внесок 22% 4316,40 грн..

У сумі зазначені відрахування становлять 41,5%.

Отже, загальна сума відрахувань на соціальні заходи становитиме:

$$B_{C3} = \text{ФОП} \times 0,415 \quad (4.7)$$

$$B_{C3} = 19620,00 \text{ грн.} \times 0,415 = 8142,30 \text{ грн.},$$

де ФОП – фонд оплати праці, грн.

Проведені розрахунки витрат на оплату праці зведемо у наступну табл.

4.2.

Таблиця 4.2.

Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна платя, грн.	Нарах. на ФОП, грн.	Всього витрати на оплату праці, грн. 8=5+6+7
		Тарифна ставка, грн.	К-сть відпрацьов. год.	Фактично нарах. з/пл., грн.			
1	2	3	4	5	6	7	8
1.	Керівник проекту	130	40	5200,00	780,00	2481,70	8461,70
2.	Інженер	80	155	12400,00	1240,00	5660,60	19300,60
Разом				17600,00	2020,00	8142,30	27762,30

4.3. Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \times T \times S, \quad (4.8)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Згідно з постановою НКРЕКП України від 05.10.2018 р. № 1177 вартість електроенергії становить 243,71 коп./кВт.год.

Потужність комп'ютера – 380 Вт з підключеним маршрутизатором, кількість годин роботи обладнання згідно таблиці 4.1 – 250 годин.

$$Z_e = 0,380 \times 250 \times 2,4371 = 231,52 \text{ грн.}$$

4.4. Розрахунок витрат на матеріали

Результати розрахунку затрат на матеріали зводяться в таблицю 4.3.

Таблиця 4.3.

Визначення величини затрат на матеріал

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транспортно-заготівельні витрати, грн	Загальна сума витрат на матеріали, грн
Папір А4-80	пачка	1	100,00	100,00	-	100,00
Ватман	шт.	9	10,00	90,00	-	90,00

Продовж. табл.4.3

Заправка картриджа для лазерного принтера	шт.	1	90,00	90,00	-	90,00
Плата за користування Інтернетом	Грн.	1	170,00	170,00	-	170,00
Разом						450,00

4.5. Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

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:

$$A = \frac{B_6 \cdot H_A}{100} \quad (4.9)$$

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

B_6 – балансова вартість комп'ютера, на початок звітного періоду, грн.

H_a – норма амортизації, %.

$$A = \frac{22000,00 * 15\%}{100\%} = 3300,00 \text{ грн.}$$

4.6. Обчислення накладних витрат

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

Накладні витрати можуть становити 20% від суми основної та додаткової заробітної плати працівників:

$$H_e = V_{o.n.} \times 0,2, \quad (4.10)$$

$$H_e = 19620,00 \text{ грн.} \times 0,2 = 3924,00 \text{ грн.}$$

де H_e – накладні витрати, грн.,

$V_{o.n.}$ – суми основної та додаткової заробітної плати працівників, грн.

4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт

Собівартість (C_B) науково-дослідних робіт розрахуємо за формулою:

$$C_B = V_{o.n.} + V_{c.z.} + Z_{m.v.} + Z_e + T_v + A + H_e, \quad (4.11)$$

$$\begin{aligned} C_B &= 19620,00 + 8142,30 + 450,00 + 231,52 + 3\,300,00 + 3924,00 \\ &= 35667,82 \text{ грн.} \end{aligned}$$

Результати проведених вище розрахунків зведемо у табл. 4.4.

Таблиця 4.4.

Кошторис витрат на науково-дослідних робіт

Зміст витрат	Сума, грн.	В % до загальної суми
1	2	3
Витрати на оплату праці (основну і додаткову заробітну плату)	19620,00	55,01
Відрахування на соціальні заходи	8142,30	22,83
Матеріальні витрати	450,00	1,26
Витрати на електроенергію	231,52	0,65
Амортизаційні відрахування	3 300,00	9,25
Накладні витрати	3924,00	11,00
Собівартість	35667,82	100

4.8. Розрахунок ціни науково-дослідних робіт

Ціну науково-дослідних робіт можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \quad (4.12)$$

$P_{рен.}$ – рівень рентабельності, 30 %; K – кількість замовлень;

$B_{н.і.}$ – вартість носія інформації, грн.

Таким чином ціна рівна 46518,17 грн.

Визначимо величину прибутку:

$$\Pi = Ц - C_B \quad (4.13)$$

Згідно формули 4.13 отримаємо 10850,35 грн.

4.9. 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:

$$E_p = \Pi / C_v, \quad (4.14)$$

де Π – прибуток; C_v – собівартість.

$$E_p = 10850,35 / 35667,82 = 0,30$$

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

$$T_p = E_p \quad (4.15)$$

$$T_p = 1 / 0,30 = 3,33 \text{ р.}$$

Про доцільність розробки програми можна сказати при врахуванні наступних критеріїв:

Таблиця 4.5.

Техніко-економічні показники НДР

№ п/п	Показник	Значення
1	Собівартість, грн.	35667,82
2	Плановий прибуток, грн.	10850,35
3	Ціна, грн.	46518,17
4	Економічна ефективність	0,30
5	Термін окупності, рік	3,33

У результаті проведення розрахунків можна зробити висновок: розробка матиме оптимальну економічну ефективність 0,3 і термін окупності становитиме 3,33 року.

Варто зазначити, що дані розрахунки носять номінальний характер і основна їх мета оцінити приблизну вартість дослідження та створення даного продукту. Номінальний характер розрахунків зумовлений тим, що даний програмний продукт має дослідницьке призначення.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

В дипломній роботі здійснюється розробка системи експертизи бензину на основі спеціалізованих комп'ютерних систем, яка супроводжується виконанням ряду робіт з використанням ЕОМ. Тому необхідним є дотримання правил охорони праці, техніки безпеки та протипожежної безпеки при роботі з комп'ютерною технікою.

До діючих нормативних документів, що забезпечують охорону праці користувачів ЕОМ належать:

- Державні санітарні норми (ДСН) 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [24];
- НПАОП 0.00-7.15-18 “Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями” [25];
- ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [26];
- Закон України «Про охорону праці» [27].

Найповнішим документом з забезпечення охорони праці працівників ПК є "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин" ДСанПіН 3.3.2.007-98 [28].

З метою дотримання правил охорони праці робоче місце обладнане обладнанням для нейтралізації зарядів статичного поля та підвищення вологості повітря. Налаштована ефективна вентиляція і підтримується відносна вологість повітря на рівні 40-60% (також біля робочого місця є акваріум для створення комфортного і здорового мікроклімату).

На вікнах в приміщенні для комфортного регулювання освітлення навішено жалюзі. Головний потік світла прямує зліва, сонячні промені і

зайчики не заважають при розробці і дослідженні методів та засобів створення спеціалізованих комп'ютерних систем.

Згідно правил безпеки охорони праці робоче місце, де проводиться розробка методів та засобів спеціалізованих комп'ютерних систем для оцінювання якості бензину, розміщене так, щоб користувач сидів спиною до вікон. В будівлі штучне освітлення і працюють в більшості люмінесцентні лампи типу ЛБ. Також для більшого освітлення місця було встановлено додаткові світильники, які працюють при необхідності.

Щоб забезпечити комфортне і безпечне користування ПК і іншими приладами приміщення, робоче місце повинно мати площу більше 6 кв.м.

Кожного дня перед початком роботи проводиться вологе прибирання та очищення екрану ВДТ від пилу та інших забруднень.

При роботі з комп'ютером на користувача діє ряд небезпечних і шкідливих чинників: електромагнітні поля (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоне та іонізуюче випромінювання, шум і вібрація, статична електрика і інші. Для полегшення впливу від шкідливих чинників бажано робити перерви і не сильно напружувати очі. Для безпеки розробника методів та засобів спеціалізованих комп'ютерних систем потрібно виконувати роботи рівномірно згідно складності завдання.

Було проведено аналіз норм праці, шкідливих та небезпечних чинників, з якими стикається розробник комп'ютерних систем загалом, зокрема, з якими зіткнувся я, коли проводив дослідження про методи та засоби розробки спеціалізованих комп'ютерних систем, а також описано параметри і характеристики приміщення, заходи, які потрібно зробити для забезпечення належних умов роботи.

Таким чином розглянуто вимоги до охорони праці користувачів ЕОМ при дослідженні методів і розробці системи експертизи бензину на основі спеціалізованих комп'ютерних систем. Проведено дослідження безпечного користування комп'ютером при розробці методів та засобів спеціалізованих комп'ютерних систем.

5.2. Безпека в надзвичайних ситуаціях

5.2.1. Безпека життєдіяльності користувачів спеціалізованих комп'ютерних систем

При експлуатації спеціалізованої комп'ютерної системи для оцінювання якості бензину необхідно застосовувати регламентовані перерви, під час яких рекомендовано виконувати оздоровлюючі вправи.

- для творчої роботи з ПК (встановлення програм, редагування текстів та інше) - 15 хв. відпочинку після кожної години роботи;

- для введення інформації - 10 хв. відпочинку після кожної години роботи;

- для зчитування інформації (діалоговий режим роботи) - 15 хв. відпочинку через кожні 2 години роботи.

Без перерв не працювати більше, ніж 2 години.

Термін обідньої перерви складений правилами внутрішнього розпорядку підприємства.

Екран ВДТ має бути віддалений від очей користувача на 500-900 мм і мати противідблискове покриття. З метою запобігання шкоди дозволяється застосування приєкранних фільтрів, спеціальних екранів та іншого, що мають певний гігієнічний сертифікат.

Клавіатуру необхідно розташовувати на столі, без хитання, або окремо на відстані 100-300 мм від ближнього до робочого краю. Розташування клавіатури та її нахил (в межах 5°..15°) повинен відповідати вимогам користувача ПЕОМ.

Для зменшення несприятливого впливу під час використання пристрою типу «миша» слід надати досить велику вільну частину столу для комфортної роботи «миші» і легкого упору руки.

У робочому приміщенні (кімнаті), де встановлені комп'ютери, щодня потрібно виконувати вологе прибирання;

Приміщення, у якому знаходяться комп'ютери, потрібно провітрювати щогодини.

Після кожного часу роботи рекомендується робити десяти хвилинну перерву, яку зручно суміщати з провітрюванням. За будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин. Під час перерви не варто читати або дивитися телевізор. Перерва, яку Ви проводите за комп'ютером (наприклад, граючись або шукаючи матеріали в Інтернеті), просто не має сенсу;

Необхідно постійно слідкувати за станом екрану монітора: він має бути чистим, без плям та пилу. Крім того, обов'язково слідкуйте за чистотою окулярів – комп'ютерних чи звичайних;

Слідкуйте за поставою: ноги твердо стоять на підлозі чи на спеціальній підставці; стегна розташовані під прямим кутом до тулуба, а гомілки – під прямим кутом до стегон; сидіти потрібно прямо або злегка нахилившись вперед; пальці рук знаходяться на рівні зап'ястків або трохи нижче – у такому положенні вони найбільш рухливі; плечі мають бути розслаблені та вільно опущені, що сприяє розслабленню рук; відстань від очей до екрану монітора – не менше 55-60 см; центр екрану має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;

Щоб попередити „синдром сухого ока”, моргайте кожні 3-5 секунд;

5.2.2. Оцінка уразливості об'єкта, оснащеного спеціалізованою комп'ютерною системою від радіоактивного забруднення і проникаючої радіації

Оцінку уразливості варто розпочати із знаходження максимальних можливих значень рівня радіації і порції проникаючої радіації.

За певну стійкість об'єкту будемо приймати максимально можливу дозу радіації, що можуть винести люди під час праці.

Саму стійкість проти радіаційного поразення будемо оцінювати по такій схемі. Спершу знайти: граничні рівні радіації (Р/год.) на робочому місці, за яких можна працювати в звичному режимі або в режимі захисту проти

випромінювання; степінь захисту робочих; дозу випромінювання, що може одержати персонал; загублення сільськогосподарських тварин і пониження відсотку їхньої корисності; загублення рослин та відсотку їх врожаю; загублення та поразення лісних посаджень, що дає в результаті пониження самої діяльності лісогосподарських об'єктів; видержку роботи сільськогосподарських і лісогосподарських об'єктів.

Наступним кроком потрібно скласти висновки про максимальні рівні радіоактивного забруднення всього простору об'єкта і випромінювання радіації; степінь надавання захисту робітників, тварин і пристроїв, техніки, врожаю, кормів, води; можливість безперервної стійкої роботи об'єкта за умови, що сумарна доза опромінення працюючих не перевищуватиме допустимої дози; можливість виробництва запланованої, доброякісної продукції тваринництва, рослинництва і лісового господарства та заходи підвищення стійкості роботи об'єкта, підвищення рівня захисту працюючих, сільськогосподарських тварин і продукції тваринництва, рослин і врожаю, води і вододжерел.

Оцінка стійкості об'єкта до впливу хімічних і біологічних засобів. При оцінці стійкості об'єкта до впливу отруйних речовин (ОР) і сильнодіючих отруйних речовин (СДОР) необхідно визначити: тип ОР чи СДОР, межі осередку хімічного зараження і ураження, площу зони зараження; глибину поширення зараженого повітря; стійкість хімічних речовин на місцевості; час можливого перебування людей у засобах захисту органів дихання і шкіри; час можливого утримання сільськогосподарських тварин у захисних спорудах; кількість заражених людей, тварин; площі заражених рослин; зараження техніки; можливі втрати людей, тварин, загибель сільськогосподарських культур і лісових насаджень.

Проблема забезпечення природно-техногенної безпеки населення і територій була завжди актуальною, тому що існують тенденції до зростання втрат серед населення та заподіяння шкоди територіям, які виникають через природні явища, пов'язані з аваріями і різними катастрофами. Ризики

виникнення надзвичайних ситуацій природного і техногенного характеру з кожним роком постійно зростають.

Основні пріоритети у сфері цивільного захисту були зосереджені на забезпеченні безпеки життєдіяльності населення та оперативному реагуванні на надзвичайні ситуації та небезпечні події, підвищенні рівня захисту населення і територій від їх наслідків, охороні здоров'я, створенні системи поводження з відходами, забезпеченні високого рівня державного нагляду у сфері охорони праці та промислової безпеки тощо.

Негативний вплив явищ природного та техногенного походження призводить до виникнення надзвичайних ситуацій і небезпечних подій, загибелі людей, погіршення умов життя та діяльності населення країни та економічних збитків для населення та об'єктів господарювання.

Безпека життєдіяльності населення та численних господарських об'єктів у районах розвитку небезпечних природних і природно-техногенних процесів є однією з основних соціально-екологічних проблем сьогодення.

У підрозділі “Безпека в надзвичайних ситуаціях” було розглянуто правила поведінки працівника перед комп'ютером, щоб зменшити шкоду на здоров'я. Розробка і дослідження спеціалізованої комп'ютерної системи для оцінювання якості бензину є досить важливим, а відповідно і складним питанням, отже доводиться довго сидіти перед комп'ютером. Відповідно людина повинна дбати про те, щоб її робота була безпечною і дотримувалась всіх норм і правил охорони праці, техніки безпеки та протипожежної безпеки при роботі з комп'ютеризованою технікою.

Безпека в надзвичайних ситуаціях - це багатогранне поняття, під яким слід розуміти не лише забезпечення безпеки працівників у процесі виконання ними їх посадових обов'язків в нормальних умовах, воно охоплює різні заходи, серед яких доцільно виокремити профілактичні та превентивні процедури щодо професійних захворювань, організацію різних форм повноцінного відпочинку і харчування працівників під час робочої перерви, дотримання інструкцій запобігання і попередження надзвичайних ситуацій, поведінка в надзвичайних

ситуаціях і визначення впливу різних приладів, що оснащені спеціалізованими комп'ютерними системи на людину, забезпечення їх необхідним спецодягом та гігієнічними засобами і навіть надання соціальних пільг і гарантій. Правильний підхід до організації охорони праці на підприємстві, грамотне використання різних нематеріальних способів стимулювання працівників дають останнім необхідне почуття надійності, стабільності й зацікавленості керівництва у своїх співробітниках. Таким чином, завдяки налагодженій охороні праці знижується також плинність кадрів, що в свою чергу благотворно впливає на стабільність усього підприємства.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1. Добування електроенергії за рахунок спалювання мінерального палива. Забруднення довкілля при цьому та шляхи його зменшення

Спалювання мінерального палива супроводжується значними забрудненнями природного середовища. В результаті спалювання вуглеводневого палива в топках ТЕС, а також двигунах внутрішнього згоряння в атмосферу викидається вуглекислий газ, концентрація якого зростає приблизно на 0,25 % за рік. Це небезпечно, бо може викликати в майбутньому розігрівання атмосфери за рахунок парникового ефекту. З труб ТЕС і вихлопних труб автомобілів у атмосферу викидаються також окиси сірки й азоту, які є причиною виникнення кислотних дощів.

Атмосфера забруднюється також дрібними твердими частками золи, шлаку, неповністю згорілого палива (сажею). Для зменшення шкоди від цих забруднень вугілля очищають від сполук сірки перед його спалюванням у топках ТЕС, вловлюють з диму ТЕС окиси сірки й азоту за допомогою фільтрів, установок типу "циклон" тощо.

Для зменшення токсичності вихлопних газів автомобілів застосовують регулювання двигунів, впроваджують "екологічно чисті" марки пального, встановлюють на автомобілях спеціальні каталізатори, що допалюють чадний газ до вуглекислого тощо [29].

Оскільки разом з вугіллям у топку ТЕС потрапляє кількість пустих порід (сланців), що містять домішки природних радіоактивних елементів, частинки золи, що вилітають із труб ТЕС, є слаборадіоактивними. Отже, має місце радіоактивне зараження атмосфери й земної поверхні. Щоправда, воно не таке шкідливе, як радіоактивне забруднення від АЕС, тому що вугільні породи містять такі природні ізотопи (урану, торію тощо), які існували в біосфері мільйони років і до яких живий світ пристосувався - більшість рослин і тварин

не накопичують цих ізотопів у своєму організмі на відміну від штучних радіонуклідів, які викидають АЕС. Існуючі методи очищення газів від частинок золи дозволяють зменшити цей вид забруднення в 100-200 разів, таким чином зменшуючи радіоактивне забруднення від ТЕС до майже фонового рівня.

Після спалювання в топках ТЕС вугілля залишається багато твердих відходів (шлаку, золи). Ці відвали займають великі площі землі, забруднюють підземні й поверхневі води шкідливими речовинами.

Ще більші ділянки землі порушують величезні вугільні кар'єри. Так, шлакові відвали, терикони пустих порід і відпрацьовані кар'єри лише в Донбасі займають площу 50 тис. га, яка постійно збільшується. Зменшення шкоди від такого забруднення досягається утилізацією шлаків і пустих порід, з яких виготовляють будівельні матеріали, засипають ними яри, болота та кар'єри при їх рекультивації. Значний ефект дають економічні методи, зокрема, введення високої оплати за порушення земель, особливо родючих. Такі обмеження в більшості західних країн призвели до відмови використання кар'єрів у сільськогосподарських районах, оскільки плата за землю виявляється вищою, ніж та вигода, яку може дати відкритий метод розробки родовища [30].

Паливно-енергетичний цикл АЕС передбачає видобування уранової руди й вилучення з неї урану, переробку цієї сировини на ядерне паливо (збагачення урану), використання палива в ядерних реакторах, хімічну регенерацію відпрацьованого палива, обробку й поховання радіоактивних відходів. Усі ці операції супроводжуються небезпечним радіоактивним забрудненням природного середовища.

Радіація має дуже негативну особливість: усе, що контактує з радіоактивною речовиною саме стає радіоактивним, а, отже, небезпечним. Сказане цілком стосується й самих АЕС. Через 25-30 років експлуатації все їхнє обладнання, апаратура, місткості, приміщення, транспортні засоби тощо, стають настільки радіоактивними, що їх необхідно демонтувати й поховати на сотню років. Для поховання лише одного реактора потрібно близько 40 га землі [31].

Об'єкти теплоенергетики представляють собою джерела потенційного, до теперішнього часу практично кількісно не врахованого ризику для населення і навколишнього середовища. Комплексна оцінка процесів енергогенерації, енергоспоживання і пов'язаних з ними екологічних наслідків вкрай необхідна, оскільки енергетичні об'єкти паливно-енергетичного комплексу за ступенем впливу на навколишнє середовище належать до числа таких, що найбільш інтенсивно впливають на біосферу. Особливо актуальними стають питання регулювання відповідальності за збиток, у тому числі за екологічний збиток при створенні в нашій країні основ правової держави, при переході до ринкових відносин в економіці. Тут важливо знайти розумні економічні важелі, правильно співвідносити вигоди і втрати, доходи і витрати на компенсацію збитку.

Важливою задачею є розробка питань нормативного розмежування припустимих і неприпустимих впливів, оцінювання вартості екологічного збитку [32]. Спостерігаючи за щоденним накопиченням відходів, не можна не подивуватись з того, який потужний потік матеріалів усіх видів рухається лише в одному напрямку – від місця видобування ресурсів на смітник. Так само, як природні екосистеми залежать від кругообігу речовин, так стійке існування технологічного суспільства, зрештою, залежатиме від людської здатності і вміння рециклізувати практично всі види матеріалів. У зв'язку з цим найдоцільніше застосовувати не один метод, а розробляти комплексну програму ліквідації відходів.

Особливість екологічної шкоди полягає в тому, що в більшості випадків вона є непоправною або порівняно відновлювальною, оскільки відтворення компонентів природи пов'язано з довготривалим періодом. Екологічна шкода може бути відшкодована у такі способи: поновлення майна в природі (відтворення знищених природних ресурсів); відшкодування збитків, завданих природним компонентам (відновлення природних ресурсів); відшкодування збитків природокористувачу; компенсація витрат, спрямованих на оздоровлення навколишнього середовища та поліпшення його якості. В

більшості випадків відшкодування екологічної шкоди зводиться до відшкодування заподіяних збитків [33].

6.2. Робота з банками екологічної інформації

В епоху глобалізації багатство формується з потужної інформації або “хто володіє інформацією, той володіє світом”.

Сучасна глобальна екологічна криза, що поступово прогресує, потребує значної уваги та заходів з подолання. Вже на етапі зародження інформаційного суспільства, засади якого почали інтенсивно впроваджуватися у XXI ст., загострилися певні соціально-екологічні проблеми, ігнорування яких може призвести до невірних результатів. До таких найбільш явних і суттєвих проблем можна віднести наступні:

- 1) людина перебудовує сформовані природні системи, втручаючись у природній перебіг подій навколишнього середовища та організму самої себе;
- 2) виробництво принципово нових видів інформації обумовлених створенням невідомих природі матеріальних сутностей, які мають здатність до саморозвитку (нові речовини, біологічні види);
- 3) значні темпи науково-технічного прогресу призводять до неворотних змін у середовищі проживання людини;
- 4) відбувається протиставлення людини та природи, хижацьке споживання природних благ людиною.

Часто недостатня кількість інформації заважає повному розумінню причин і наслідків екологічної деградації, а соціальна байдужість стоїть на заваді вирішення цих проблем.

Формуючи екологічну свідомість громадян, в сучасних умовах необхідно використовувати всі переваги інформаційного суспільства, основою якого є виробництво та використання інформації, що призводить до структуризації суспільства. Сурди́ковська С.Т. стверджує, що “свідомість суспільства про необхідність охорони навколишнього середовища зростає значно швидше, ніж

обсяг інформації, що розкривається на цю тему, як державними, так і приватними підприємствами” [34]. Інформація за таких умов стає найважливішим і найдорожчим товаром.

Банки екологічної статистичної інформації – це вторинна накопичена інформація, певним чином упорядкована чи опрацьована. Зазвичай вона подається у вигляді збірників, щорічників.

Банком статистичної інформації про соціально-економічне становище держави є, по-суті, статистичний щорічник України, що представляє собою накопичену інформацію за ряд років. Його структура побудована так, щоб повністю відобразити всі грані соціально-економічних явищ і процесів.

Статистичний щорічник складається з таких розділів: національні рахунки, фінанси і кредит, ціни і тарифи, матеріально-енергетичні ресурси, промисловість, інвестиційна та будівельна промисловість, транспорт і зв'язок, торгівля і послуги, зовнішньоекономічна діяльність, структурні зміни в економіці, населення, зайнятість населення, доходи населення, освіта, наука та інформатика, культура і відпочинок, медичне обслуговування, правопорушення, природні ресурси та охорона навколишнього середовища, міжнародні зіставлення.

Для прикладу збірники “Довкілля України”, “Довкілля Житомирщини” є збірниками банку інформації про екологічне становище держави і її областей.

В основі забезпечення інформації дослідженнями лежать звичайні дані статистичної звітності, що знаходяться в наступних документах:

- Первісних документах статистичної звітності;
- Регіональних статистичних бюлетенях і статистичних щорічниках Держкомстату України;
- Матеріалах Національного банку України, Державного митного комітету України, Міждержавного статистичного комітету СНД, Українсько-Європейського центру з питань законодавства тощо.

Додатково ми можемо взяти за джерело матеріалів інформацію з спеціально організованих статистичних досліджень, різні аналітичні матеріали:

- Українських НДІ
 - Інститут економічного прогнозування НАН України;
 - Міжнародний центр перспективних досліджень;
 - Інститут економіки НАН України;
 - Інститут регіональних досліджень НАН України;
 - Інвестиційна компанія ДІКОМ та ін.
- Зарубіжних організацій
 - Міжнародний статистичний інститут;
 - Світовий банк;
 - Міжнародна організація праці;
 - Міжнародний банк реконструкції та розвитку;
 - Міжнародний валютний фонд;
 - Міжнародна фінансова корпорація;
 - Міжнародна асоціація розвитку;
 - Статистична комісія ООН;
 - Комісія з питань народонаселення Економічної та Соціальної ради ООН;
 - Організація економічного співробітництва і розвитку;
 - Конференція європейських статистиків та ін.

Інформацію про екологічні ситуації на окремих територіях чи об'єктах, про вплив антропогенної діяльності на стан довкілля та здоров'я людей можна отримати за даними екологічної експертизи, екологічний стан на окремих об'єктах описується в екологічних паспортах підприємства [35].

ВИСНОВКИ

Метою даної дипломної роботи було дослідження методів та засобів побудови спеціалізованих комп'ютерних систем для оцінювання якості бензину, а також в набутті вмінь та навичок практичного втілення математичного забезпечення в різних програмних середовищах для вирішення завдань.

Об'єктом дослідження є процес для оцінювання якості бензину з застосуванням спеціалізованих комп'ютерних систем.

В процесі виконання дипломної роботи було узагальнено знання в дослідженні різних типів математичного забезпечення спеціалізованих комп'ютерних систем оцінювання якості бензину.

Завданням дипломної роботи було дослідити різні типи математичного забезпечення спеціалізованих комп'ютерних систем удосконалення інформаційної системи оцінювання якості бензину.

Проведено аналіз спеціалізованих комп'ютерних систем для оцінювання якості бензину.

На основі нечіткої продукційної моделі розроблено узагальнений алгоритм побудови нечітких продукційних правил діагностування на основі зменшення кількості вхідних нечітких змінних.

При написанні дипломної роботи було проведено аналіз предмета дослідження і сутності науково-практичної задачі ідентифікаційної експертизи та доведено необхідність впровадження інформаційних технологій для її рішення, зокрема для задачі ідентифікаційної системи оцінювання якості бензину. Задача, яка була поставлена в роботі, представлена як задача розпізнавання образів. Запропоновано використання методів обчислювального інтелекту, а саме методу нечіткої класифікації, для її рішення.

Крім того висунуто і підтверджено гіпотезу про ефективність нечітких методів класифікації для вирішення задачі ідентифікаційної експертизи. Обґрунтовано ефективність опису класів образів нечіткими портретами, які

узагальнюють інформацію, отриману з навчальних вибірок. Сукупність таких нечітких портретів формує базу знань експерта.

Розроблено інформаційну технологію ідентифікаційної експертизи бензину. На основі моделей і методів, запропонованих в роботі, розроблено бібліотеку програмних компонентів, яка в подальшому може бути використана для досліджень з моделлю.

Проведено аналіз відомих моделей представлення знань і показано, що найбільш адекватною моделлю представлення знань є нечітка продукційна модель.

В основу процесу ідентифікаційної експертизи товару покладено процес аналізу показників якості. Такими є фізикохімічні показники, представлені в чисельному вигляді. Для обробки даних такого типу в області інформаційних технологій розвиваються методи аналізу даних.

Сучасний рівень розвитку інформаційних технологій і комп'ютерної техніки дозволяє за рахунок їх застосування в задачах ідентифікаційної експертизи удосконалити існуючі методи ідентифікації і розробляти нові, а також підвищити достовірність такої експертизи.

Важливою перевагою нечітких класифікаторів є інтерпретованість отриманих моделей. В основі нечіткого класифікатора лежать правила нечітких продукцій. Такі правила близькі людській інтуїції. Рішення, отримане методом нечіткої класифікації, моделює рішення, яке може бути отримано особою яка приймає рішення, в нашому випадку експертом.

Таким чином, застосування нечіткого підходу в задачах ідентифікаційної експертизи дозволяє моделювати когнітивні процеси експертизи.

В основу методу ідентифікації зразка продукції, які використовуються в інформаційній системі, покладена інформаційна технологія нечіткої класифікації. Розроблено бібліотека класів образів, яка реалізує всі математичні моделі і методи, запропоновані в дипломній роботі.

У дипломній роботі запропоновано нове вирішення актуальної науково-технічної задачі ідентифікаційної експертизи зразків товарів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Петров В.Н. Информационные системы: Учебник для вузов. – Питер.: «ЗАО Издательский дом «Питер»». – 2003. – 30-31с.
2. Николайчук Я.М., Возна Н.Я., Пітух І.Р. Проектування спеціалізованих комп'ютерних систем / Навчальний посібник / - Тернопіль: ТЗОВ "Тернограф". 2010. – 16-21с.
3. Лаврищева Е.М. Проблематика программной инженерии // Киев: Знання, 1991.19 с.
4. Лаврищева Е.М., Грищенко В.Н. Области знаний программной инженерии SWEBOOK и подход к обучению этой дисциплине // УСиМ. - 2005. - № 1.- 38 – 54 с.
5. Musa J.D., Ackerman A.F. Quantifying Software Validation: When to stop testing? // IEEE Software May 1989. - 19-27 p.
6. Kapur P., Garg R. Optimum Software release policies for Software reliability growth model under imperfect debugging // Rech. oper. - 1990. - 24, N 3. - 295 – 305 p.
7. Ohtera H., Yamada S. Optimum software release time considering an error-detection phenomenon during operation // IEEE Trans. Reliab. - 1990. - R 39. - N 5. - 596 – 599 p.
8. Мороз Г.Б. Пуассоновские модели роста надежности программного обеспечения и их приложение. Аналитический обзор // УСиМ. - 1996. - № 1 - 2. - 69 – 85 с.
9. Berman O., Ashrafi N., Optimization models for reliability of modular software systems // IEEE Trans. Software Reliability. - 1993. V. 19. - 1119 – 1123 p.
10. Cai, M., Lyu M., Wong K, Ko R. Component-based software engineering: Technologies, development frameworks, and quality assurance schemes// Proc. Asia-Pacific Software Engineering Conf., Dec. - 2000. - 372 – 379 p.
11. Fyffe D., Hines W., Lee N. System reliability allocation and a computational algorithm // IEEE Trans. Reliability. - 1968. - V. 17. - 64 – 69 p.

12. Tillman F. A., Hwang C. L., Kuo W. Determining component reliability and redundancy for optimum system reliability // IEEE Trans. Reliab. - 1977. V. 26. - 162 – 165 p.

13. Zahedi F. and Ashrafi N. Software reliability allocation based on structure, utility, price, and cost // IEEE Trans. Software Eng. -Apr., 1991. V. 17. - 345-355 p.

14. Yacoub S. M., Cukic B., and Ammar H. H. A component-based approach to reliability analysis of distributed systems // Proc. 18th IEEE Symp. Reliable Distributed Syst. - 1999. - 158 – 167 p.

15. Lyu M. R., Rangarajan S. et. Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development // IEEE Trans. on Reliab. V. 51. - 2002. JUNE - N 2. - 183 – 192 p.

16. Бабенко Л.П., Лаврищева Е.М. Основы программной инженерии. Учебник (укр. язык). - Киев: Знання, 2001. - 269 с.

17. Основы инженерии качества программных систем // Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, В.Ю. Суслов / Под ред. ИВ. Сергиенко. - Киев: Академперіодика, 2002. - 504 с.

18. Коротун Т.М. Тестирование информационных систем с учетом риска// Проблемы программирования. - Вып. 1-2 (спец. вып. по материалам конференции УкрПРОГ'2000) - 2000. - 385 – 392 с.

19. Коротун Т.М. Модель определения критерия завершения тестирования на основе анализа риска отказа и ее реализация // Проблемы программирования. - 2001. - № 1 - 2. - 63 – 68 с.

20. Мороз Г.Б., Коротун Т.М. Ризико-операційний підхід до вирішення проблеми оптимального випуску програмних систем // Проблеми програмування (Спец. вип. конференції УкрПРОГ-2006). - 2006. - № 2 - 3. - 231 – 236 с.

21. Лаврищева Е.М., Коротун Т.М. Построение процесса тестирования программных систем // Проблемы программирования. - 2002. - № 1 - 2. - 272 – 281 с.

22. Коротун Т.М. Моделі та методи інженерії тестування програмних систем в умовах обмежених ресурсів. Дис. канд. фіз.-мат. наук. Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, 2005. - Держ. обл. № 0405U003083.

23. Коротун Т.М. Совершенствование процесса тестирования программного обеспечения // Проблемы программирования. - 1998. - Вып. 3. - с. 59 – 64 с.

24. Гогіташвілі Г. Системи управління охороною праці: Навчальний посібник. – Львів: «Афіша», 2012. – 320 с.

25. Лесенко Г. Методика контролю стану умов та безпеки праці [Текст] // Охорона праці. – №9. – 2014. – 40 с.

26. Лисюк М. Обов'язки посадових осіб підприємства з охорони праці [Текст] // Справочник кадровика. К. – №2. – 2012. – 200 с.

27. Про охорону праці : Закон України від від 14 жовтня 1992 року № 2694-ХІІ в редакції від 1 січня 2015 року : [Електронний ресурс]. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/2694-15/page>

28. Москальова В. Основи охорони праці: Підручник. – Київ: ВД «Професіонал», 2011. – 672 с.

29. Бойчук Л Д., Соломенно Е.М., Бугай О.В. Екологія і охорона навколишнього середовища: Навч. посіб. — Суми: Університетська книга, 2003. — 284 с.

30. Дорогунцов СЛ., Коценко К.Ф., Аблова О.К. Екологія. — К.: КНЕУ, 2001. —162 с.

31. Білявський Г.О., Фурдуй Р.С., Костіков І.Ю.. Основи екології. – К.: “Либідь”, 2004.

32. Національна доповідь про стан навколишнього природного середовища в Україні. 1996. – К.: вид-во Раєвського, 1998.

33. Джигирей В.С.. Екологія та охорона навколишнього природного середовища. - К.: „ Знання”, 2002.

34. Вісник екологічної адвокатури. - 2006. - № 30. - С. 9-12; Кутузов В.И. Доступ к экологической информации: правовые аспекты [Текст] / В.И. Кутузов, А.А. Попов. - М., 2004. - 194 с.

35. Тарасова В.В. «Екологічна статистика» Київ: Центр учбової літератури, 2008. — 392 с.



УКРАЇНА
ТЕРНОПІЛЬ – 2019

УДК 004.9 **О. О.Цебрик**

Тернопільський національний технічний університет імені Івана Пулюя, Україна

МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ БЕНЗИНУ

О.О. Tsebryk

METHODS AND TOOLS FOR BUILDING SPECIALIZED COMPUTER SYSTEMS FOR GASOLINE QUALITY ASSESSMENT

Третина бензину на українських автозаправних станціях є неякісною, про що свідчать дослідження Інституту споживчих експертиз. Заправляти авто неякісним бензином – це все одно, що постійно харчуватися у фаст-фудах. Для людини такий раціон може закінчитися виразкою та іншими недугами, а для машини – заміною двигуна. На українських дорогах все частіше їздять автомобілі відомих європейських автомобільних концернів, які випускають нові машини з урахуванням європейських вимог до якості бензину. Їздити на сучасних іномарках в Україні та заправлятися паливом, що відповідає лише старим стандартам, означає нищити мотор.

Питання визначення якості бензину виникали в споживачів та вивчалися аналітиками. Застосування спеціалізованих комп'ютерних систем для проведення робіт із збирання, накопичення, обробки, аналізу, прийняття рішень для визначення та з поліпшення якості засобами сучасних інформаційних технологій на території діяльності промислових підприємств незалежно від форм їх власності та організаційно-правової форми господарювання є пріоритетним напрямком діяльності. Дослідження в цій області сконцентровані на розробці та впровадженні спеціальних комп'ютерних програм, здатних імітувати, відтворювати ті області діяльності людини, які вимагають мислення, певної майстерності і накопиченого досвіду. До них відносяться задачі прийняття рішень, розпізнавання зразків палива та проведенні оцінювання якості різних марок бензину з застосуванням сучасних методів спеціалізованих комп'ютерних систем, які вже успішно застосовується в деяких областях техніки і життя суспільства – органічної хімії, пошуку корисних копалин, медичній діагностиці.

Дані системи повинні вирішувати задачі, що вимагають для свого рішення експертних знань у деякій конкретній області. У тій чи іншій формі експертні системи повинні мати ці знання. Такі системи використовують бази даних (БД), або бази знань. Система акумулює професійні знання керівників і фахівців, використовуючи їх для формування бази знань, яка містить набір взаємопов'язаних правил. При прийнятті рішень стає можливим аналіз наслідків різних рішень у вигляді питань «що буде, як-що...», не витрачаючи часу на трудомісткий процес програмування.

Мета проведених досліджень полягала у вивченні особливостей ідентифікації математичної моделі прийняття рішень в експертних системах управління на основі нечіткої логіки. Проведено аналіз відомих моделей представлення знань і показано, що найбільш адекватною моделлю представлення знань є нечітка продукційна модель.

На основі нечіткої продукційної моделі розроблено узагальнений алгоритм побудови нечітких продукційних правил діагностування на основі зменшення кількості вхідних нечітких змінних. Було проведено аналіз предмета дослідження і сутності науково-практичної задачі ідентифікаційної експертизи та доведено необхідність впровадження інформаційних технологій для її рішення, зокрема для задачі ідентифікаційної експертизи бензинів. Задача, яка була поставлена в роботі, представлена як задача розпізнавання образів. Запропоновано використання методів обчислювального інтелекту, а саме методу нечіткої класифікації, для її рішення.

КОД ПРОГРАМИ

Побудова нечіткої моделі

```

int CModelProcessing::fs_Frequency(TDoubleArray x, TDoubleArray xy[2], double
beta, double alfa)
{
    xy[0].clear(); xy[1].clear();
    //xy[0] fuzzzset X. It's length is not equal to sample length
    //xy[1] Mju
    //step of sliding
    double step; //sliding window width
    double wind;
    //iterators for main data
    TDoubleArray::iterator the_x;
    TDoubleArray::iterator the_x2;
    std::sort(x.begin(), x.end());
    double min = x.front(), max = x.back();
    if (max - min < 0.01)
    {
        xy[0].push_back(min);
        xy[1].push_back(1);
        return 1;
    }
    double xN = x.size();
    //-----PARAMS -----
    //for step of slidig method
    int fStep = 0;
    switch (fStep)
    {
    case 0:
    {
        //my method by coefs
        double h_middle = (max - min) / (xN - 1);
        step = beta*h_middle;
        wind = (step) / alfa;
        break;
    }
    };
    //-----SLIDING -----
    the_x = x.begin();
    double xyN = floor((max - min) / step); //number of steps
    double Point_first = (*the_x) + ((max - min) - xyN*step) / 2; //left edge of
function
    double Point, beg, end;
    xy[0].push_back(Point_first - (wind / 2));
    xy[1].push_back(0);
    int i_step = 0; // number of step
    int k, maxK = 0; //counter for points in window on step i_step
    do {
        k = 0; //the number of points in the wind
        Point = Point_first + step*i_step;
        beg = Point - wind / 2;
        end = Point + wind * 2;
        while (the_x != x.end() && (*the_x) < beg)
        {
            ++the_x;

```



```

    }
    the_x2 = the_x;
    while (the_x2 != x.end() && (*the_x2) <= end)
    {
        k++; if (k>maxK)maxK = k;
        ++the_x2;
    }
    i_step++;
    xy[0].push_back(Point);
    xy[1].push_back(k);
} while (the_x != x.end() && i_step <= xyN);
xy[0].push_back(Point_first + step*i_step + (wind / 2));
xy[1].push_back(0);
//check edges
if (xy[0].at(0)<0 && xy[1].at(0) >= 0)
{
    xy[0].at(0) = 0; xy[1].at(0) = xy[1].at(1);
    //IT is left edge with 1
};
fs_Normalize(xy[1], maxK);
return 1;
};

```

```

void CModelProcessing::fs_Normalize(TDoubleArray &y, int max)
{
    TDoubleArray::iterator the_y;
    the_y = y.begin();
    while (the_y != y.end())
    {
        (*the_y) /= max;
        ++the_y;
    };
};

```

```

CFuzzyMFSigmoid *CModelProcessing::fs_CreateSigmoid(CFuzzyMFDiscrete *FS, double param)
{
    CFuzzyMFSigmoid *FSs = new(CFuzzyMFSigmoid);
    TDoubleArray xy[2];
    xy[0] = ((CFuzzyMFDiscrete *)FS)->xy[0];
    xy[1] = ((CFuzzyMFDiscrete *)FS)->xy[1];
    int xyN = xy[0].size();
    if (xyN>0)
    {
        double porog = param;
        double TopX_left_inx, TopY_left = -1;
        double TopX_right_inx, TopY_right = -1;
        for (int i = 0; i<xyN; i++)
        {
            double znach = xy[1].at(i);
            if (znach>1 - porog)
            {
                if (TopY_left == -1)
                {
                    TopY_left = znach;
                    TopX_left_inx = i;
                    TopY_right = znach;
                    TopX_right_inx = i;
                }
                else {
                    TopY_right = znach;
                }
            }
        }
    }
};

```

```

        TopX_right_inx = i;
    }
}
}
double midLeftSide = fs_getMiddleforSigmoid(xy, 0, TopX_left_inx, -1);
double midRightSide = fs_getMiddleforSigmoid(xy, TopX_right_inx, xyN - 1, 1);
double g1, g2, b1, b2;
g2 = midLeftSide;
b2 = midRightSide;
double midLeftSide_q = xy[0].at(0) + (xy[0].at(TopX_left_inx) - xy[0].at(0)) /
2;
double midRightSide_q = xy[0].back() - (xy[0].back() - midRightSide) / 2;
g1 = -log(0.01 / 0.99) / (xy[0].at(TopX_left_inx) - midLeftSide_q);
b1 = -log(0.01 / 0.99) / (xy[0].at(TopX_right_inx) - midRightSide_q);
FSs->SetCoefs(g1, g2, b1, b2);
FSs->SetMinMax(xy[0].front(), xy[0].back());
} //end if (xyN>0)
return FSs;
};

double CModelProcessing::fs_getMiddleforSigmoid(TDoubleArray xy[2], int beg_p, int
end_p, int side)
{
    double mid;
    double x_half1 = -1;
    double x_half2 = -1;
    double y1, y2;
    y2 = xy[1].at(beg_p);
    for (int i = beg_p + 1; i <= end_p; i++)
    {
        y1 = y2;
        y2 = xy[1].at(i);
        if (y1 == 0.5 && y2 == 0.5)
        {
            x_half1 = i - 1;
            x_half2 = i;
        }
        else if (((side == -1) && (y1 <= 0.5) && (y2 >= 0.5)) || ((side == 1) &&
(y1 >= 0.5) && (y2 <= 0.5)))
        {
            double x_t = (xy[0].at(i) - xy[0].at(i - 1))*(0.5 - y1) / (y2 - y1) +
xy[0].at(i - 1);
            if (x_half1 == -1)
            {
                x_half1 = x_t;
                x_half2 = x_t;
            }
            else x_half2 = x_t;
        }
    }
    if (x_half1 != -1)
        mid = x_half1 + (x_half2 - x_half1) / 2;
    else
        mid = xy[0].at(0);
    return mid;
}

```

Алгоритм нечіткого виведення

```

int CModelProcessing::ts_GenerateSamples(int flag)
{
    int Fold;
    if (flag>1) { Fold = flag; flag = 2; };
    P_all_ind.clear();
    // int CountofSamples; //=1;//for multy samples generation
    TTwoDimDoubleArray::iterator the_P;
    TIntArray P_ind;
    TIntArray::iterator the_P_ind;
    TIntStringMap::iterator the_map;
    int classNum;
    switch (flag)
    {
    case 1: //every 3rd element in class for testing
    {TIntIntMap classElementCounter;
    the_map = NC.begin();
    while (the_map != NC.end())
    {
        classElementCounter[(the_map).first] = 0;
        the_map++;
    };
    the_P = P_all.begin();
    while (the_P != P_all.end())
    {
        classNum = (*the_P).back();
        if (++classElementCounter[classNum] == 3)
        {
            P_ind.push_back(0); classElementCounter[classNum] = 0;
        }
        else {
            P_ind.push_back(1);
        }
        ++the_P;
    };
    P_all_ind.push_back(P_ind);
    };
    break;
    case 2:
    int C = NC.size();
    int N = P_all.size();
    for (int i = 0; i<N; i++) P_ind.push_back(1); //iáó-àpùèà
    for (int i = 0; i<Fold; i++)
    {
        TIntArray T = P_ind;
        P_all_ind.push_back(T);
    }
    TIntIntMap classElementCounter;
    the_map = NC.begin();
    while (the_map != NC.end())
    {
        classElementCounter[(the_map).first] = 0;
        the_map++;
    };
    for (int p = 0; p<N; p++)
    {
        classNum = P_all.at(p).back();
        P_all_ind.at(classElementCounter[classNum]).at(p) = 0; //òâñòíâúâ
        if ((++classElementCounter[classNum]) == Fold)

```

```

        {
            classElementCounter[classNum] = 0;
        }
    } //end for p
    break;
};
return 1;
};

int CModelProcessing::ts_InferenceBySample(int sample_index, int flag)
{
    if (FCAccuracy != NULL) free(FCAccuracy);
    FCAccuracy = new CClassifierAccuracy(NC);
    TDoubleArray data;
    TIntDoubleMap alfa;
    int classNum, classNum_rigth;
    int CountElements = 0;
    TTwoDimDoubleArray::iterator the_x_line;
    the_x_line = P_all.begin();
    int num = 0;
    cerr << "INFERENCE ";
    while (the_x_line != P_all.end())
    {
        if (GetSampleRow(num, data, classNum_rigth, flag, sample_index))
        {
            FClassifier->Inference(data, alfa, classNum);
            FCAccuracy->AddElement(classNum_rigth, alfa, classNum);
            CountElements++;
        };
        the_x_line++; num++;
    }; //end while The_x_line
    FCAccuracy->MakeConfusionMatrix();
};

CFuzzyRules *CModelProcessing::CreateBaseClassifier(int sample_index)
{
    free(FClassifier);
    FClassifier = new CFuzzyRules(NC.size(), NA.size());
    TDoubleArray data;
    TDoubleArray MF[2];
    CFuzzySet* FS;
    TRule_line FSarray; //array of fuzzy sets pointer
    TIntStringMap::iterator the_class;
    TIntStringMap::iterator the_attr;
    int CL; AnsiString CL2;
    the_class = NC.begin();
    while (the_class != NC.end())
    {
        CL = (*the_class).first;
        CL2 = (*the_class).second;
        the_attr = NA.begin();
        FSarray.clear();
        while (the_attr != NA.end())
        {
            GetSampleCow((*the_class).first, (*the_attr).first, data, sample_index);
            if (p_USE_SLIDING_PARAM)
            {
                fs_Frequency(data, MF, p_SLIDING_BETA, p_SLIDING_ALFA);
            }
            else {

```

```

        fs_Frequency(data, MF);
    };
    FS = new CFuzzyMFDiscrete();
    ((CFuzzyMFDiscrete *)FS)->Initial(MF);
    //MEDIAN FILTRATION
    if (p_MEDIANA)
        ((CFuzzyMFDiscrete *)FS)->MedianFiltration(2, 0);
    if (p_SIGMOID)
    {
        CFuzzyMFSigmoid *FS_new = fs_CreateSigmoid((CFuzzyMFDiscrete *)FS);
        free(FS);
        FS = FS_new;
    };
    FSarray.push_back(FS);
    AnsiString info = (*the_class).second + " " + (*the_attr).second;
    ++the_attr;
}
FClassifier->AddRule(FSarray, (*the_class).first);
++the_class;
}
return FClassifier;
};

int CModelProcessing::GetSampleCow(int ClassNum, int AttributeNum, TDoubleArray &x, int
sample_index)
{
    x.clear();
    //for index variant
    bool fIndx = false;//1- if we use index array P_all_ind
    TIntArray::iterator the_indx;
    if (sample_index>0 && !P_all_ind.empty())
    {
        fIndx = true;
        the_indx = (P_all_ind.at(sample_index - 1)).begin();
    };
    double d;
    vector <double> line;
    iterlv12 = P_all.begin();
    while (iterlv12 != P_all.end()) {
        line = *iterlv12;
        if (line.back() == ClassNum)
        {
            if ((fIndx && (*the_indx) == 1) || !fIndx)
            {
                d = line.at(AttributeNum - 1);
                x.push_back(d);
            };
        };
        ++iterlv12;
        if (fIndx) { ++the_indx; };
    };
};

int CModelProcessing::GetSampleRow(int num, TDoubleArray &x, int &classNum_right, int
flag, int sample_index)
{
    x.clear();
    //for index variant
    bool fIndx = false;//1- if we use index array P_all_ind
    TIntArray::iterator the_indx;

```

```

if (sample_index>0 && !P_all_ind.empty())
{
    fIndx = true;
    the_indx = (P_all_ind.at(sample_index - 1)).begin();
};
if ((fIndx && (*(the_indx + num)) == flag) || fIndx == false)
{
    TDoubleArray::iterator the_line = (*(P_all.begin() + num)).begin();
    while (the_line != (*(P_all.begin() + num)).end() - 1)
    {
        x.push_back(*the_line);
        the_line++;
    }
    classNum_right = (*the_line);
    return 1;
}
else { return 0; };
};

int CModelProcessing::getIndexClass(int classnum)
{
    vector <double> line;
    int count = 0;
    iterlv12 = P_all.begin();
    while (iterlv12 != P_all.end()) {
        line = *iterlv12;
        if (((int)(*iterlv12).back()) == classnum)
        {
            count++;
        }
        ++iterlv12;
    }
    return count;
}

```