Міністерство освіти і науки України

Дніпропетровський національний університет імені Олеся Гончара

Факультет прикладної математики

Кафедра комп’ютерних технологій

**ДИПЛОМНА РОБОТА**

**за ОКР бакалавра**

|  |  |
| --- | --- |
| Тема: | Розробка програмного забезпечення для лінгвістичного розбору текстів |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Виконавець**: | студ. групи ПК-10-1  Павленко Кристина Геннадіївна  «….»………………….  ………………..(підпис) |
| **Допускається до захисту** |  |  | |
| Завідувач кафедри ПКТ  проф., д.ф. –м.н. Гук Н.А.  «….»…………………. | | **Керівник**: | доц., к.т.н.  Золотько К.Є.  «….»………………….  .…………….. (підпис) |
| …………………………(підпис) | |  |  |
|  | | **Рецензент:** | доц., к.ф.-м.н. , ПМЗ  Кузнєцов К.А.  «….»………………….  …………………….(підпис) |

**Керівник з розділу**

**«Охорона праці»**: ст.викл., ТБЖ

Козєєв І.С.

«….»………………….

.…………….. (підпис)

Дніпропетровськ

2014

**РЕФЕРАТ**

Дипломна робота:\_\_с.,\_\_\_рис.,\_\_\_\_табл.,\_\_\_\_джерел,\_\_\_\_додатки.

**Об'єктом дослідження є** ……………………...............................................................................................

**Мета роботи -** ……………………………………………………………

**Методика (метод) дослідження**  ……………………………………….

**Одержані висновки та їх новизна**..........................................................

**Результати досліджень можуть бути застосовані при ................................ ................................**

**Перелік ключових слів**: 5**–**10 слів або словосполучень із тексту роботи, які характеризують її зміст, у називному відмінку (ВЕЛИКИМИ БУКВАМИ, ЧЕРЕЗ КОМУ).

**ANNOTATION**

**ЗМІСТ**

ВСТУП 5

Постановка задачі 5

РОЗДІЛ1. Інформаційний огляд 8

1.1 Експертні системі та їх характеристика 8

1.2 Огляд існуючих експертних систем та сфери їх застосування 13

1.3 Переваги та недоліки експертніх систем 13

РОЗДІЛ2. ОГЛЯД(АНАЛІЗ?) ПРЕДМЕТНОЇ ОБЛАСТІ 14

[2.1.1. Присудок (The Predicate)](#_Toc359209120)

[2.1.2. Підмет (The Subject). Загальна характеристика](#_Toc359209121)

2.[1.3. Узгодження присудка з підметом](#_Toc359209121)

РОЗДІЛ3. ТЕОРЕТИЧНА ЧАСТИНА 19

3.1 Теорія множин 19

3.2 Теорія графів 21

3.3 Лінгвістична змінна 26

РОЗДІЛ4. ПРАКТИЧНА РЕАЛІЗАЦІЯ 35

4.1 Вибір програмного інструменту 35

4.1.1 Середовище розробки 35

4.1.2 Мова програмування 35

4.2 Опис програми 41

РОЗДІЛ5. РЕЗУЛЬТАТИ ТА АНАЛІЗ ВИКОНАНОЇ РОБОТИ 35

РОЗДІЛ6. ОХОРОНА ПРАЦІ 35

[6.1. Аналіз виробничих факторів](#_Toc359209133)

[6.1.1. Шкідливі та небезпечні фактори](#_Toc359209133)

[6.1.2. Розташування робочого місця програміста](#_Toc359209133)

[6.1.3. Параметри мікроклімату та шуму](#_Toc359209133)

[6.1.4. Виділення озону при роботі принтерів](#_Toc359209133)

[6.2. Заходи до поліпшення умов праці.](#_Toc359209134)

[6.2.1. Електронебезпека на робочому місці](#_Toc359209133)

[6.2.2. Організаційна безпека](#_Toc359209133)

6.3. Розрахунок та проектування інженерно-технічного заходу захисту від шкідливого виробничого фактору

[6.4. Висновки](#_Toc359209136)

[ВИСНОВКИ](#_Toc359209138)

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 65

ДОДАТОК 68

**ВСТУП**

У житті сучасного суспільства важливу роль відіграють автоматизовані інформаційні технології . З плином часу їх значення безперервно зростає. Але розвиток інформаційних технологій відбувається дуже нерівномірно : якщо сучасний рівень обчислювальної техніки та засобів зв'язку вражає   
уяву , то в області смислової обробки інформації успіхи значно скромніші.

Ці успіхи залежать , насамперед, від досягнень у вивченні процесів людського мислення , процесів мовного спілкування між людьми і від уміння моделювати ці процеси на ЕОМ.

Коли мова йде про створення перспективних інформаційних технологій , то проблеми автоматичної обробки текстової інформації , представленої на природних мовах , виступають на передній план . Це визначається тим , що мислення людини тісно пов'язане з його мовою. Більш того , природна мова є інструментом мислення. Він є також універсальним засобом спілкування між людьми - засобом сприйняття , накопичення , зберігання , обробки і передачі інформації. Проблемами використання природної мови в системах автоматичної обробки інформації займається наука комп'ютерна лінгвістика . Ця наука виникла порівняно недавно - на порозі п'ятдесятих і шістдесятих років минулого сторіччя. За минулі півстоліття в області комп'ютерної лінгвістики були отримані значні наукові та практичні результати : були створені системи машинного перекладу текстів з одних природних мов на інші , системи автоматизованого пошуку інформації в текстах , системи автоматичного аналізу та синтезу усного мовлення та багато інших. Дана робота присвячена побудові оптимального комп'ютерного інтерфейсу засобами комп'ютерної лінгвістики при проведенні лінгвістичних досліджень.

# **ПОСТАНОВКА ЗАДАЧІ**

Задачею дипломної роботи є створення програмного забезпечення для розбору речення в англійської мови.

Основними вимогами до проекту є:

* знаходження підмета то присудка;
* простий інтерфейс програми;

В системі повинен діяти постійний автоматичний контроль над операціями, що виконуються, та даними, з якими вони працюють.

Система повинна мати зручний зовнішній інтерфейс, який орієнтований на користувача. Потрібно забезпечити коректну роботу системи.

**РОЗДІЛ 1.ІНФОРМАЦІЙНИЙ ОГЛЯД**

**1.1 Експертні системі та їх характеристика**

Експертна система — це методологія адаптації [алгоритму](http://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) успішних рішень одної сфери науково-практичної діяльності в іншу. З поширенням комп’ютерних технологій це тотожна інтелектуальна [комп’ютерна програма](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%E2%80%99%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), що містить знання та аналітичні здібності одного або кількох експертів щодо деякої галузі застосування і здатна робити логічні висновки на основі цих знань, тим самим забезпечуючи вирішення специфічних завдань ([консультування](http://uk.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BD%D1%81%D1%83%D0%BB%D1%8C%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F&action=edit&redlink=1), [навчання](http://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F), [діагностика](http://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D0%BD%D0%BE%D1%81%D1%82%D0%B8%D0%BA%D0%B0), [тестування](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), [проектування](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) тощо) без присутності [експерта](http://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82) (спеціаліста в конкретній проблемній галузі).   
 Також визначається як система, яка використовує [базу знань](http://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D1%8C) для вирішення завдань у деякій предметній галузі. Цей клас програмного забезпечення спочатку розроблявся дослідниками [штучного інтелекту](http://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%82) в [1960-ті](http://uk.wikipedia.org/wiki/1960-%D1%82%D1%96) та [1970-ті](http://uk.wikipedia.org/wiki/1970-%D1%82%D1%96) та здобув комерційне застосування, починаючи з [1980-их](http://uk.wikipedia.org/wiki/1980-%D1%82%D1%96). Часто термін система, заснована на знаннях використовується в якості синоніма експертної системи, однак можливості [експертних](http://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82) систем ширші за можливості систем, заснованих на.

Однак узгодженого визначення експертних систем не існує. Натомість автори дають визначення залежно від застосування, структури таких систем. Ранні визначення експертних систем припускали застосування виведення нових знань на основі правил.

Схожі дії виконує [програма-майстер](http://uk.wikipedia.org/w/index.php?title=%D0%9C%D0%B0%D0%B9%D1%81%D1%82%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0)&action=edit&redlink=1) (wizard) . Як правило, майстри застосовують в [системних програмах](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) для інтерактивного спілкування з користувачем (наприклад, при [інсталяції ПЗ](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D0%B0%D0%BB%D1%8F%D1%86%D1%96%D1%8F_%D0%9F%D0%97)). Головна відмінність майстрів від ЕС — відсутність [бази знань](http://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D1%8C); всі дії жорстко запрограмовані. Це просто набір форм для заповнення користувачем.

Інші подібні програми — [пошукові](http://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA%D0%BE%D0%B2%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) або [довідкові](http://uk.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D0%B2%D1%96%D0%B4%D0%BA%D0%BE%D0%B2%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) системи. За запитом користувача вони надають найвідповідніші ([релевантні](http://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C)) розділи бази статей, альтернативність вибору котрих визначає суб’єкт формування запиту.

Тож ми бачимо обмеження методології експертних систем за наявності корисних якостей у коректних межах застосування.

Характеристики ЕС

Експертна система відрізняється від інших прикладних програм наявністю таких ознак:

* Моделює механізм [мислення](http://uk.wikipedia.org/wiki/%D0%9C%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F) людини при застосуванні для розв'язання задач в цій предметній області. Це істотно відрізняє експертні системи від систем математичного моделювання або комп'ютерної анімації. Однак, ЕС не повинні повністю відтворювати психологічну модель фахівця в цій області, а повинні лише відтворювати за допомогою комп'ютера деякі методики розв'язання проблем, що використовуються експертом.
* Система, окрім виконання обчислювальних операцій, формує певні висновки, базуючись на тих знаннях, якими вона володіє. Знання в системі, зазвичай, описані деякою спеціалізованою мовою і зберігаються окремо від програмного коду, що формує висновки. Компонент збереження знань прийнято називати [базою знань](http://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D1%8C).
* Під час розв'язання задач основну роль відіграють евристичні і наближені методи, що, на відміну від алгоритмічних, не завжди гарантують успіх. Евристика, в принципі, є правилом впливу, що в машинному вигляді відображає деяке знання, набуте людиною разом із накопичуванням практичного досвіду розв'язання аналогічних проблем. Такі методи є наближеними в тому сенсі, що, по-перше, вони не потребують вичерпної вихідної інформації, а, по-друге, існує певний ступінь впевненості в тому, що запропонований розв'язок є правильним.

Експертні системи відрізняються і від інших видів програм із галузі штучного інтелекту.

* Експертні системи застосовуються для предметів реального світу, операції з якими зазвичай вимагають великого досвіду, накопиченого людиною. Експертні системи мають яскраво виражену практичну направленість для застосування в науковій або комерційній сфері.
* Однією з основних характеристик експертної системи є її швидкодія, тобто швидкість отримання результату та його достовірність. Дослідницькі програми штучного інтелекту можуть бути і не дуже швидкими, натомість, експертна система повинна за прийнятний час знайти розв'язок, що був би не гіршим за розв'язок, що може запропонувати фахівець в цій предметній області.
* Експертна система повинна мати можливість пояснити, чому запропоновано саме цей розв'язок і довести його обґрунтованість. Користувач повинен отримати всю інформацію, необхідну йому для того, аби переконатись в обґрунтованості запропонованого розв'язку.

Етапи розробки

* Етап ідентифікації проблем - визначаються завдання, які підлягають вирішенню, виявляються цілі розробки, визначаються експерти і типи користувачів.
* Етап витягання знань - проводиться змістовний аналіз проблемної області, виявляються поняття і їх взаємозв'язки, визначаються методи розв'язання задач.
* Етап структуризації знань - обираються ІС і визначаються способи подання всіх видів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність цілям системи зафіксованих понять, методів рішень, засобів представлення й маніпулювання знаннями.
* Етап формалізації - здійснюється наповнення експертом бази знань. У зв'язку з тим, що основою ЕС є знання, даний етап є найбільш важливим і найбільш трудомістким етапом розробки ЕС. Процес придбання знань поділяють на вилучення знань з експерта, організацію знань, що забезпечує ефективну роботу системи, і представлення знань у вигляді, зрозумілому ЕС. Процес придбання знань здійснюється інженером зі знань на основі аналізу діяльності експерта з вирішення реальних завдань.
* Реалізація ЕС - відбувається створення одного або декількох прототипів ЕС котрі вирішують поставлені задачі.
* Етап [тестування](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) - проводиться оцінка обраного способу представлення знань в ЕС в цілому.

**1.2 Огляд існуючих експертних систем та сфери їх застосування**

Можна навести такі відомі експертні системи, як приклад існуючих:

* [CLIPS](http://uk.wikipedia.org/wiki/CLIPS) — [мова програмування](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), використовується для створення експертних систем
* [Dendral](http://uk.wikipedia.org/wiki/Dendral) — аналіз даних [мас-спектрометрії](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81-%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D1%96%D1%8F)
* [Dipmeter Advisor](http://uk.wikipedia.org/w/index.php?title=Dipmeter_Advisor&action=edit&redlink=1) — аналіз даних, отриманих під час пошуку [нафти](http://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%84%D1%82%D0%B0)
* [Jess](http://uk.wikipedia.org/w/index.php?title=%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_Jess&action=edit&redlink=1) — від [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Java Expert System Shell, оболонка експертних систем на [Java](http://uk.wikipedia.org/wiki/Java). Рушій CLIPS реалізований на мові програмування Java, використовується для створення експертних систем
* [MQL 4](http://uk.wikipedia.org/w/index.php?title=MQL_4&action=edit&redlink=1) — MetaQuotes Language 4, спеціалізована мова програмування для опису фінансової стратегії
* [Mycin](http://uk.wikipedia.org/wiki/Mycin) — діагностика інфекційних хвороб крові та рекомендація антибіотиків
* [Prolog](http://uk.wikipedia.org/wiki/Prolog) — мова програмування, використовується для створення експертних систем
* [R1 / XCON](http://uk.wikipedia.org/wiki/R1_/_XCON)(експертна система) — обробка замовлень
* [SHINE Real-time Expert System](http://uk.wikipedia.org/w/index.php?title=SHINE_Expert_System&action=edit&redlink=1) — від [англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Spacecraft Health INference Engine, рушій для отримання даних про стан і безпеку космічного корабля
* [STD Wizard](http://uk.wikipedia.org/w/index.php?title=STD_Wizard&action=edit&redlink=1) — експертна система для рекомендації та вибору медичних аналізів (діагностики)

Класифікація ЕС:

* Класифікація ЕС за зв'язком з реальним часом
* Класифікація ЕС за типом проблемного середовища
* Класифікація за типом ЕОМ
* Класифікація за ступенем інтеграції з іншими програмами
* Класифікація ЕС за завданням, що вирішується

Сфера застосування та перспектива розвитку:

Експертні системи досить давно використовуються у діагностиці, зокрема у медичній та автомобільній.

Експертна система стану двигуна [SUN SMP 4000](http://uk.wikipedia.org/w/index.php?title=SUN_SMP_4000&action=edit&redlink=1)

Також експертні системи можна використовувати в [прогнозуванні](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), [плануванні](http://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), [контролі](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C), [управлінні](http://uk.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F) та [навчанні](http://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F).

Наприклад, експертні системи вже застосовуються в банківській справі в таких напрямках:

* програмах аналізу інвестиційних проектів;
* програмах аналізу стану валютного, грошового та фондового ринку;
* програмах аналізу кредитоспроможності чи фінансового стану підприємств і банків.

Процес створення експертних систем значно змінився за останні роки. Завдяки появі спеціальних інструментальних засобів побудови експертних систем значно скоротились терміни та зменшилась трудомісткість їх розробки. Інструментальні засоби, що використовуються при створенні експертних систем, можна розбити на три класи:

* мови програмування, орієнтовані на створення експертних систем ([Лісп](http://uk.wikipedia.org/wiki/Lisp), [Пролог](http://uk.wikipedia.org/wiki/Prolog), [Smalltalk](http://uk.wikipedia.org/wiki/Smalltalk), [FRL](http://uk.wikipedia.org/w/index.php?title=FRL&action=edit&redlink=1), [InterLisp](http://uk.wikipedia.org/w/index.php?title=InterLisp&action=edit&redlink=1) та такі загальновживані, як: [Сі](http://uk.wikipedia.org/wiki/C), [Асемблер](http://uk.wikipedia.org/wiki/Assembler), [Паскаль](http://uk.wikipedia.org/wiki/Pascal),[Фортран](http://uk.wikipedia.org/wiki/FORTRAN), [Бейсик](http://uk.wikipedia.org/wiki/Basic));
* середовища програмування ([Delphi](http://uk.wikipedia.org/wiki/CodeGear_Delphi), [Java](http://uk.wikipedia.org/wiki/Java));
* порожні експертні системи (наприклад, оболонка [EXSYS Professional 5.0 for Windows](http://uk.wikipedia.org/w/index.php?title=EXSYS&action=edit&redlink=1)).

На американському і західноєвропейських ринках систем штучного інтелекту організаціям, які бажають створити експертну систему, фірми-розробники пропонують сотні інструментальних засобів для їх побудови. Нараховуються тисячі розроблених вузькоспеціалізованих експертних систем. Це свідчить про те, що експертні системи складають дуже вагому частину програмних засобів.

**1.3 Переваги та недоліки експертних систем**

Експертні системи відзначаються певними перевагами над людьми-експертами при використанні. Зокрема, експертна система:

* переважає можливості людини при вирішенні надзвичайно громіздких проблем;
* не має упереджених думок, тоді як експерт може користуватися побічними знаннями і легко піддається впливу зовнішніх факторів;
* не робить поспішних висновків, нехтуючи певними етапами знайдення рішення;
* забезпечує діалоговий режим роботи;
* дозволяє роботу з інформацією, що містить символьні змінні;
* забезпечує коректну роботу з інформацією, яка містить помилки, за рахунок використання ймовірнісних методів досліджень;
* дозволяє проводити одночасну обробку альтернативних версій;
* за вимогою пояснює хід кроків реалізації програми;
* забезпечує можливість обгрунтування рішення та відтворення шляху його прийняття.

Але навіть найкращі з існуючих експертних систем мають певні обмеження у порівнянні з людиною-[експертом](http://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82), які зводяться до таких:

* Більшість експертних систем не цілком придатні для широкого використання. Якщо [користувач](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87) не має деякого досвіду роботи з цими системами, у нього можуть виникнути серйозні труднощі. Багато експертних систем доступні лише тим експертам, які створювали їх бази знань. Тому потрібно паралельно розробляти відповідний користувацький інтерфейс, який би забезпечив кінцевому користувачу властивий йому режим роботи;
* "Навички" системи не завжди "зростають" після сеансу експертизи, навіть коли проявляються нові знання;
* Все ще залишається проблемою приведення знань, отриманих від експерта, до вигляду, який забезпечував би їх ефективне використання;
* Експертні системи, як правило, не можуть набувати якісно нових знань, не передбачених під час розробки, і тим більше не володіють здоровим глуздом. Людина-експерт при розв'язанні задач звичайно звертається до своєї інтуїції або здорового глузду, якщо відсутні формальні методи рішення або аналоги розв'язування даної проблеми.

**РОЗДІЛ 2.ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ**

**2.1 Загальний аналіз структури речення в англійській мові**

**2.1.1 Присудок (The Predicate)**

В англійській мові є три типи присудка :

А) простий дієслівний ( the simple verbal predicate) ;

Б) складений іменний ( the compound nominal predicate);

В) складений дієслівний ( the compound verbal predicate) .

Простий дієслівний присудок (the Simple Verbal Predicate) виражається смисловим дієсловом в особовій формі, яка може бути простою або складною формою будь-якого часу, стану або способу.

I'm going to the library this morning. Shall I change your book for you?

Stop! Wait for me! I'm coming too.

Ann is not at home.— And where is she?

На відміну від української мови простий дієслівний присудок, виражений дієсловом to be, ніколи не опускається (так само як і дієслово-зв'язка to be):

Is Pete at home? — No, he is at school.

Петро вдома? — Hi, він у школі.

Присудок, виражений дієсловом to have або to have got, на українську мову звичайно перекладається дієсловом є (був, було):

I have (have got) a nice bag.

У мене є гарненька сумочка.

Особливим різновидом простого дієслівного присудка є присудок, виражений зворотом there is (there are; there has been; there have been; there was, there were; there will be) — є, буде, був і т. п.

Слово there тут не є прислівником і самостійного значення не має. У реченні з присудком there is може вживатись, і прислівник there (там).

There were many people there. (Там було багато народу.)

Речення з присудком there is вживаються для вираження наявності (або відсутності) якогось предмета (явища) або особи, виражених підметом, якщо їх не виділяє той, хто говорить, з ряду їм подібних предметів (осіб), або якщо цей предмет чи особа ще невідомі співрозмовникові. Відповідно зчислювані іменники після присудка there is вживаються в однині з неозначеним артиклем, а у множині — з неозначеними займенниками some, many, few, a few і їх еквівалентами (a lot of, plenty of і т. п.), а також з числівниками або із заперечним займенником nо.

Незчислювані іменники після звороту there is вживаються без артикля або з неозначеними займенниками some, any, little, a little, much і їх еквівалентами (a great deal of і т. п.), а також із заперечним займенником по.

У стверджувальних реченнях з присудком there is підмет ставиться після присудка.

В українській мові речення з there is відповідають реченням, які починаються звичайно обставиною часу або місця і з присудком, вираженим дієсловами бути, перебувати, існувати, стояти, лежати і т. п. (Ці дієслова в українській мові часто пропускаються.)

There is a good park in our town. (У нашому місті є хороший парк.)

There are many beautiful flowers in our park.

(У нашому парку (росте) багато красивих квітів.)

До складу присудка з there можуть входити також і модальні дієслова.

There can be no doubt about it.

There is to be a festival in our town next summer.

Особливим різновидом присудка є Фразовий присудок (the Group Verb Predicate) складається з дієслова з ослабленим значенням в особовій формі (найчастіше з дієслів to have і to take) та іменника. Це сполучення утворює єдине смислове і граматичне ціле. Воно часто буває рівнозначне простому дієслівному присудку.

I have dinner. = I dine.

Фразовий присудок звичайно означає коротку (однократну) дію. Найбільш уживаними сполученнями цього типу є: to have dinner (breakfast, supper); to have a lesson (lessons); to have a-talk; to have a look; to have a cold; to have a good time; to have a rest; to have a bite; to have a drink; to have a smoke; to have a quarrel; to have (take) a walk; to take a bath; to take a seat; to take a tram (a bus і т. n.); to take care; to pay a visit; to pay attention; to catch cold.

Let's have a swim.

Did you have a talk with him?

Складений іменний присудок (the Compound Nominal Predicate) складається з дієслова-зв'язки to be в особовій формі і предикатива (іменної частини), вираженого звичайно іменником або прикметником. Предикатив означає якусь ознаку підмета, а дієслово-зв'язка з'єднує підмет і предикатив і виражає особу, число, час і стан.

John is my friend.

You are very kind, thank you.

На відміну від української мови дієслово-зв'язку to be в англійській мові не пропускають. Наприклад:

Our town is very beautiful.

Крім дієслова to be, як дієслово-зв'язка можуть вживатися також такі дієслова:

дієслова to seem, to look, to appear, із значенням здаватися, виглядати і дієслово to feel — почувати:

You look tired. (Ти виглядаєш втомленим.)

дієслова to get, to become, to turn із значенням ставати:

It's getting dark. (Стає темно. Темніє.)

дієслова to remain, to keep, to continue із значенням залишатися, продовжувати:

Keep quiet, please.

Предикатив (іменна частина присудка) може бути виражений такими словами і групами слів:

іменником у загальному (рідко — присвійному) відмінку без прийменника або з прийменником:

This is my sister Ann. She is a student.

прикметником або дієприкметником:

Snow is white.

займенником-іменником у загальному або називному відмінку або присвійним займенником в абсолютній формі; особовий займенник І в функції предикатива звичайно вживається в об'єктному відмінку:

It's me.

кількісним або порядковим числівником:

We were only two.

I was the first to come.

інфінітивом або інфінітивним зворотом:

Our plan was to start at once.

герундієм:

My hobby is collecting postage stamps.

прислівником:

The lesson is over. Our time is up.

комплексом з інфінітивом або герундієм:

My father is against my entering the college this autumn.

Складений дієслівний присудок поділяється на два види :

the Modal Verbal Predicate

the Aspect Verbal Predicate

Модальний дієслівний присудок (the Modal Verbal Predicate) складається з модального дієслова (can, may, must, should, need, ought, be, have, have got) в особовій формі і основи інфінітива або інфінітива з часткою to (після дієслів ought, be і have) смислового дієслова.

Другою частиною модального присудка може також бути сполучення інфінітива дієслова-зв'язки to be і предикатива.

Модальний присудок виражає відношення особи до дії або стану (можливість, необхідність, імовірність і т. п.)

Can you swim?

Не ought to be more polite.

Фазовий дієслівний присудок – це присудок який складається з певних дієслів які позначають початок, продовження, закінчення дії, та інфінітива або герундія.

Ось, для прикладу, декілька фазових дієслів : to start, to begin (починати), to go on (продовжувати), to finish, to stop, to cease (закінчувати, припиняти).

His friend began to study foreigne languages.

She started crying

**2.1.2 Підмет (The Subject). Загальна характеристика**

Підмет може бути виражений такими частинами мови:

іменником у загальному відмінку:

What was the homework for today?

Займенником – іменником у загальному або називному відмінку або присвійним займенником в абсолютній формі:

We are pupils of the seventh form at school No. 14.

It's raining hard.

інфінітивом або інфінітивним зворотом; у цьому випадку підмет звичайно ставиться після дієслова-присудка, а перед останнім ставиться формальний підмет it:

То answer this question is not so easy.

(Найчастіше: It's not so easy to answer this question.)

герундієм або герундіальним зворотом: у цьому випадку підмет іноді стоїть після дієслова, а перед присудком ставиться формальний підмет it:

Speaking and reading English is the best way to learn English.

кількісним або порядковим числівником:

How many nails do you need? Four will do.

підрядним реченням:

It's a great pity that he is absent.

Підмет називається складним, якщо він виражений синтаксичними комплексами з інфінітивом, дієприкметником або герундієм. Складні підмети, виражені комплексом з герундієм або інфінітивним for-комплексом, звичайно вводяться формальним підметом it і ставляться після присудка. Складний підмет, виражений комплексом «Називний відмінок з інфінітивом», складається з двох частин, між якими стоїть присудок.

Не is expected to come today (складний підмет, виражений комплексом «Називний відмінок з інфінітивом»).

Або:

Your trying to do that alone won't be of any use (складний підмет, виражений комплексом з герундієм).

Займенник it як підмет вживається в двох основних значеннях:

в значенні особового займенника 3-ї особи однини або вказівного займенника:

Where is the dining-room? — It's over there.

без будь-якого значення, як формальний підмет

В українській мові часто вживаються речення і звороти без підмета (безособові і неозначено-особові). В англійській мові такі речення не вживаються і їм відповідають речення з формальним підметом it або з іншими типами підмета. Наприклад:

Морозить. - It's freezing.

У простому реченні може бути два або більше підметів, з'єднаних сполучниками and — і, та, either...or— або ... або, neither ... not — ні ... ні, not only ... but also — не тільки... але й, as well ... as — як ... і або безсполучниковим способом. Такі підмети називаються однорідними.

Neither you nor he is right.

**2.1.3 Узгодження присудка з підметом**

Присудок в англійській мові узгоджується з підметом в особі та числі настільки, наскільки особа виражена у формах дієслова-присудка.

What are you doing? — I'm preparing for my lessons.

Jane, weren't we supposed to review lesson nine?

My brother Nick plays tennis well, my sisters play basket-ball.

Якщо підмет виражений збірними іменниками people—люди, police — поліція, cattle— худоба та ін., то дієслово-присудок стоїть у множині.

There were few people at the booking-office.

Our police are people's police.

Якщо підметом є Іменник news—новини, або назви наук arithmetics — арифметика, mathematics — математика, physics — фізика та ін., дієслово-присудок стоїть в однині.

Is there any news about him?

Mathematics is his strong point.

Після деяких іменників форма числа дієслова-присудка в англійській мові відрізняється від відповідної української в зв'язку з особливостями утворення множини .

His clothes are quite clean. (його одяг зовсім чистий.)

У реченні з однорідними підметами, з'єднаними сполучниками and, both... and або безсполучниковим способом, присудок стоїть у множині.

John and Tom are brothers.

Zinc, copper and lead are coloured metals.

Якщо два однорідні підмети з'єднані сполучниками either ... or, neither ... nor, not only ... but also, то дієслово-зв'язка узгоджується з другим з них.

Either you or your friend is wrong.

**РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА**

**3.1 Теорія множин**

Теорія множин — розділ [математики](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), в якому вивчаються загальні властивості [множин](http://uk.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) (переважно [нескінченних](http://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D0%BA%D1%96%D0%BD%D1%87%D0%B5%D0%BD%D0%BD%D0%B0_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0)). Виділення теорії множин в самостійний розділ математики відбулося на рубежі XIX і XX століть. Теорія множин зробила дуже великий вплив на розвиток сучасної математики — вона є фундаментом ряду нових розділів математики, дозволила по-новому поглянути на класичні розділи математики і глибше зрозуміти сам предмет математики.

Сучасні дослідження теорії множин була започатковані [Георгом Кантор](http://uk.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D1%80%D0%B3_%D0%9A%D0%B0%D0%BD%D1%82%D0%BE%D1%80) і [Ріхардом Дедекіндом](http://uk.wikipedia.org/wiki/%D0%A0%D1%96%D1%85%D0%B0%D1%80%D0%B4_%D0%94%D0%B5%D0%B4%D0%B5%D0%BA%D1%96%D0%BD%D0%B4) в 1870-х роках. Після відкриття [парадоксів](http://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%BE%D0%BA%D1%81%D0%B8_%D1%82%D0%B5%D0%BE%D1%80%D1%96%D1%97_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD) наївної теорії множин, на початку XX століття були запропоновані численні системи аксіом, серед яких найвідомішою є [система Цермело-Френкеля](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD_%D0%A6%D0%B5%D1%80%D0%BC%D0%B5%D0%BB%D0%BE-%D0%A4%D1%80%D0%B5%D0%BD%D0%BA%D0%B5%D0%BB%D1%8F), з [аксіомою вибору](http://uk.wikipedia.org/wiki/%D0%90%D0%BA%D1%81%D1%96%D0%BE%D0%BC%D0%B0_%D0%B2%D0%B8%D0%B1%D0%BE%D1%80%D1%83).

Основні поняття

В основі теорії множин лежать первинні поняття: [множина](http://uk.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) та елемент множини. Елемент множини перебуває щодо множини у відношенні бути елементом множини (позначається як x \in A[[4]](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD#cite_note-4) — «x є елемент множини A»). Серед похідних понять найважливішими є наступні:

* [порожня множина](http://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D0%BE%D0%B6%D0%BD%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) — множина, яка не містить елементів, позначається зазвичай \varnothing;
* [підмножина](http://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) і [надмножина](http://uk.wikipedia.org/w/index.php?title=%D0%9D%D0%B0%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0&action=edit&redlink=1) — множина, яка складається тільки з елементів іншої множини, та множина, до якої належать усі елементи іншої множини, відповідно;
* [сімейство множин](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%96%D0%BC%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD&action=edit&redlink=1);
* [простір (універсум)](http://uk.wikipedia.org/wiki/%D0%A3%D0%BD%D1%96%D0%B2%D0%B5%D1%80%D1%81%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) — множина, що є надмножиною всіх множин;
* [конституента](http://uk.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D0%B5%D0%BD%D1%82%D0%B0&action=edit&redlink=1).

Над множинами визначені наступні [операції](http://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97_%D0%BD%D0%B0%D0%B4_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8):

* [об'єднання (або сума)](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%B4%D0%BD%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD) (позначається як \ A \cup B);
* [перетин (або добуток)](http://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%82%D0%B8%D0%BD_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD) (позначається як \ A \cap B);
* [різниця](http://uk.wikipedia.org/wiki/%D0%A0%D1%96%D0%B7%D0%BD%D0%B8%D1%86%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD) (позначається як \ A \setminus B, рідше \ A - B);
* [симетрична різниця](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0_%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D1%86%D1%8F) (позначається як A\,\triangle\,B, рідше A\,\dot{-}\,B).
* [доповнення](http://uk.wikipedia.org/wiki/%D0%94%D0%BE%D0%BF%D0%BE%D0%B2%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD) (позначається як \ \setminus A, або -A\,);

Для множин визначені наступні [бінарні відношення](http://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%BD%D0%B0%D1%80%D0%BD%D0%B5_%D0%B2%D1%96%D0%B4%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%BD%D1%8F):

* [відношення рівності](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D1%96%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD&action=edit&redlink=1) (позначається як A = B\,);

[відношення включення](http://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) (позначається як A \subset B, або A \subseteq B).

**3.1 Теорія множин**

Теорія графів — розділ [математики](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), що вивчає властивості [графів](http://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). Наочно граф можна уявити як геометричну конфігурацію, яка складається з точок ([вершини](http://uk.wikipedia.org/w/index.php?title=%D0%92%D0%B5%D1%80%D1%88%D0%B8%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D0%B0&action=edit&redlink=1)) сполучених лініями ([ребрами](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%B5%D0%B1%D1%80%D0%BE_%D0%B3%D1%80%D0%B0%D1%84%D0%B0&action=edit&redlink=1)). У строгому визначенні графом називається така пара множин G = (V, E), де V є підмножина будь-якої зліченної множини, а E - підмножина V × V.

Визначення [графу](http://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) є настільки загальним, що цим терміном можна описувати безліч подій та об'єктів повсякденного життя. Високий рівень абстракції та узагальнення дозволяє використовувати типові [алгоритми](http://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) теорії графів для вирішення зовнішньо несхожих задач у [транспортних](http://uk.wikipedia.org/w/index.php?title=%D0%9C%D0%B5%D1%80%D0%B5%D0%B6%D0%B0_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B0&action=edit&redlink=1) і [комп'ютерних мережах](http://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%B5%D0%B6%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0), [будівельному](http://uk.wikipedia.org/wiki/%D0%91%D1%83%D0%B4%D1%96%D0%B2%D0%BD%D0%B8%D1%86%D1%82%D0%B2%D0%BE) [проектуванні](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), молекулярному [моделюванні](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) тощо.

Теорія графів знаходить застосування, наприклад, в геоінформаційних системах (ГІС). Існуючі або запроектовані будинки, споруди, квартали і т. п. розглядаються як вершини, а з'єднують їхні дороги, інженерні мережі, лінії електропередачі і т. п. - як ребра. Застосування різних обчислень, вироблених на такому графі, дозволяє, наприклад, знайти найкоротший об'їзний шлях або найближчий продуктовий магазин, спланувати оптимальний маршрут.

Теорія графів містить велику кількість невирішених проблем і поки не доведених гіпотез.

Формальне означення графа:

Нехай X = { x1,…,xn} – деяка скінченна множина (множина вершин), M2 – множина всіх невпорядкованих пар елементів з X,

M2 = { (xi,xj ): xi ∈ X, xj ∈ X, i ≠ j}

1. Граф G(X,W) – пара множин X,W ⊂ M2. Множина X – це множина вершин, множина W – це множина ребер. Якщо (xi,xj ) ∈ W, то ми говоримо, що ребро (xi,xj ) сполучає вершину xi з вершиною xj; інша термінологія – ребро (xi,xj ) і вершини xi та xj інцидентні.

2. Граф G(X,W) називається повним, якщо W = M2.

Якщо множина X містить n вершин, то, очевидно, число ребер повного графа дорівнює Cn2. Повний граф з n вершинами позначається Kn.

3. Граф G(X,W) називається порожнім, якщо W = ∅.

4. Вершини xi та xj графа G(X,W) інцидентні, якщо (xi,xj ) ∈ W.

5. Степенем d( xi ) вершини xi графа G(X,W) називається число вершин xj ,які інцидентні вершині xi (число відрізків, які виходять з вершини xi ).

6. Якщо d( xi ) =1, то вершина xi називається кінцевою вершиною графа G(X,W). Якщо d( xi ) = 0, то вершина xi називається ізольованою.

**3.1 Лінгвістична змінна**

Лінгвістична змінна - в теорії нечітких множин, змінна, яка може приймати значення фраз з природної або штучної мови. Наприклад, лінгвістична змінна «швидкість» може мати значення «висока», «середня», «дуже низька» та інші. Фрази, значення яких бере змінна, в свою чергу є іменами нечітких змінних і описуються нечітким безліччю.

Математичне визначення

Лінгвістичної змінної називається п'ятірка { x , T ( x ) , X , G , M } , де   
x - ім'я змінної; T ( x ) - деякий безліч значень лінгвістичної змінної x , кожне з яких є нечіткою змінною на безлічі X; G є синтаксичне правило для утворення імен нових значень x ; M є семантична процедура, що дозволяє перетворити нове ім'я , утворене процедурою G , в нечітку змінну ( задати вид функції приналежності ) , асоціює ім'я з його значенням , поняттям .

T ( x ) також називають базовим терм -множиною , оскільки воно задає мінімальну кількість значень , на підставі яких за допомогою правил G і M можна сформувати інші допустимі значення лінгвістичної змінної.   
Безліч T ( x ) і нові утворені за допомогою G і M значення лінгвістичної змінної утворюють розширене терм- безліч .

**РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ**

# **4.1 Вибір програмного інструменту**

**4.1.1 Середовище розробки**

Microsoft Visual Studio - лінійка продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як у рідному, так і в керованому кодах для всіх платформ, підтримуваних Microsoft Windows (Windows Mobile, Windows CE, . NET Framework, .NET Compact Framework і Microsoft Silverlight).

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і як відладчик машинного рівня. Решта інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

## Visual Studio 2010 була випущена 12.04.2010 року разом з. NET Framework 4.0. Visual Studio включає підтримку мов C # 4.0 і Visual Basic, NET 10.0, а також мови F#, який був відсутній у попередніх версіях.

**4.1.2. Мова програмування С#**

Cі Шарп (англ. C Sharp) - елегантна, сторого типізована об'єктно-орієнтована мова програмування загального призначення, призначена для розробки різноманітних безпечних і потужних додатків, які виконуються в середовищі. NET Framework. Цю мову було створено у 2001 році. На мові C# можна розробляти звичайні клієнтські програми Windows, веб-служби XML, розподілені компоненти, додатки типу "сервер-клієнт", додатки баз даних та інші. У Visual C# 2008 мається розширений редактор коду, конструктори зі зручним для користувача інтерфейсом, вбудований відладчик і багато інших засобів для спрощення розробки додатків на мові C # версії 3.0.

Синтаксис C # дуже виразний, але простий у вивченні. Усі, хто знайомий з мовами C, C + + або Java з легкістю дізнаються синтаксис з фігурними дужками, характерний для мови C #. Розробники, які знають будь-який з цих мов, як правило, зможуть домогтися ефективної роботи з мовою C # за короткий термін. Синтаксис C # робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вирази і прямий доступ до пам'яті, чого немає в Java. С# підтримує універсальні методи і типи, забезпечуючи вищий рівень безпеки і продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власну поведінку ітерації, яке може легко використовуватися в клієнтському коді. У C# 4.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласної конструкцією мови.

Як об'єктно-орієнтована мова, C# підтримує поняття інкапсуляції, успадкування та поліморфізму. Всі змінні і методи, включаючи метод Main - точку входу програми - інкапсулюються в визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які мають пріоритет над віртуальні методи в батьківському класі, необхідно ключове слово override, щоб виключити випадкове повторне визначення. У мові C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, реалізовує інтерфейси, але не підтримує спадкування.

На додаток до основних описаним об'єктно-орієнтованим принципам, мова C# спрощує розробку компонентів програмного забезпечення завдяки кільком інноваційним конструкціям мови, до числа яких входять наступні:

* Інкапсульовані підписи методів, звані делегатами, які підтримують суворо-типізовані повідомлення про події.
* Властивості, що виступають в ролі методів доступу для закритих змінних-членів.
* Атрибути з декларативними метаданими про типи під час виконання.
* Вбудовані коментарі XML-документації.
* LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів в різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, в мові C# можна використовувати процес, який називається "Interop". Процес Interop дозволяє програмам на C# виконувати практично будь-які дії, які може виконувати вихідний додаток на C++. Мова C# підтримує навіть покажчики і поняття "небезпечного" коду для тих випадків, коли прямий доступ до пам'яті має вкрай важливе значення.

Процес побудови C# в порівнянні з C і C++ простий і є більш гнучким, ніж в Java. Немає окремих файлів заголовка, а методи і типи не потрібно оголошувати в певному порядку. У вихідному файлі C# може бути визначено будь-яке число класів, структур, інтерфейсів і подій.

**РОЗДІЛ 5. РЕЗУЛЬТАТИ ТА АНАЛІЗ ВИКОНАНОЇ РОБОТИ**

**РОЗДІЛ 6. ОХОРОНА ПРАЦІ**

##### Охорона праці являє собою систему законодавчих актів, соціально-економічних, організаційних, технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, що забезпечують безпеку, збереження здоров'я і працездатності людини в процесі праці. Закон України «Про охорону праці» визначає основні положення з охорони праці та регулює взаємовідносини між працівниками та адміністрацією.

##### Робота фахівця в сучасному автоматизованому і механізованому виробництві являється процесом взаємодії людини і машини при якій відбувається мобілізація психологічних і фізичних функцій людини. Тому важливим є дотримання працівником режиму праці та відпочинку.

## **6.1Аналіз виробничих факторів**

### **6.1.1Шкідливі та небезпечні фактори**

На підприємствах і в організаціях, що використовують відеодисплейні термінали, особливо – де використовується праця програмістів, технологічний процес та використовуване обладнання є потенційними джерелами шкідливих і небезпечних виробничих факторів, що можуть несприятливо впливати на стан здоров’я працівників, а також на їхніх нащадків.

При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, виявлення та аналіз небезпечних та шкідливих факторів виробництва користувача ПК та особливо програмістів є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

При роботі з використанням персонального комп‘ютера відзначають наступні шкідливі та небезпечні фактори, які можуть загрожувати здоров’ю робітників на робочих місцях:

* несприятливі кліматичні (мікрокліматичні) умови у повітрі робочої зони;
* напружені зорові роботи;
* інтелектуальні навантаження;
* монотонність праці;
* нервово-емоційна напруженість праці;
* невідповідність ергономічних показників робочого місця діючим вимогам;
* фізична важкість виконуваної роботи (статичні навантаження на кістково-м’язовий апарат людини);
* шуми;
* електромагнітні випромінювання;
* ризики ураження електричним струмом;
* ризики виникнення пожеж;
* недостатня, або надмірна освітленість робочого місця;
* ризик виникнення надзвичайних ситуацій природного або штучного характеру на об’єкті або території.

### **6.1.2Розташування робочого місця програміста**

Робоче приміщення є кімнатою (Рис. 6.1), що містить 2 робочих місця програмістів, розташоване на 7-му поверсі 10-поверхової будівлі.

Загальна площа приміщення складає 16,8 м2, а загальний об‘єм – 46,2 м3. Висота стелі  – 2,75 м.

Площа приміщення, що припадає на одне робоче місце, становить 8,4 м2, а об’єм – 23,1 м3.

Робочі місця розташовані біля вікна. Відстань між столами 50 см. Це менше від рекомендованої величини.

Робочі столи виготовлені з ДСП та деревини і мають розміри 1,8 м \* 0,9 м, що відповідає встановленим рекомендованим величинам. 

Рис.6.1. План робочого приміщення

Сидіння робочих місць відповідають ергономічним вимогам для найбільш зручного положення тіла при роботі з ПК (Рис. 6.2).

Відстань між користувачем та ВДТ складає приблизно 700 мм. Висота робочої поверхні столу становить 750 мм.

На столах встановлено освітлювальні прилади з регульованим положенням.

Відеодисплейний термінал на кожному з робочих місць розташований на відстані 500 мм від краю поверхні стола. Окрім терміналу, кожне робоче місце містить власне комп’ютер, клавіатуру та маніпулятор типу миші.

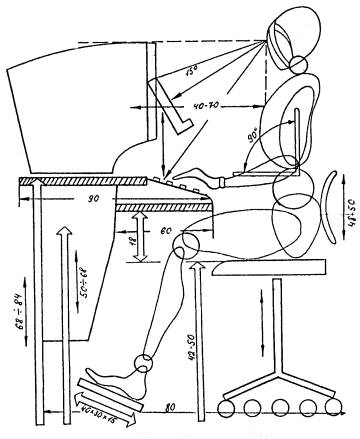
На одному з робочих столів встановлено лазерний принтер. У приміщенні також маються: журнальний столик, тумба біля стіни, вбудована шафа. 

Рис.6.2. Положення тіла при роботі з ПК

### **6.1.3 Параметри мікроклімату та шуму**

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт “Санітарні норми мікроклімату виробничих приміщень” ДСН 3.3.6.042-99.

Для постійних робочих місць, якими є робочі місця операторів ПК, встановлені оптимальні параметри мікроклімату, а при неможливості їх дотримання використовують допустимі параметри. Робота оператора ПК за енерговитратами відноситься до категорії легких робіт. В таблиця 4.1 наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу.

Таблиця 6.1

Параметри мікроклімату для приміщень з ПК

|  |  |  |
| --- | --- | --- |
| Період року | Параметр мікроклімату | Величина |
| Холодний | Температура повітря в приміщенні | 22...24°С |
| Відносна вологість | 40... 60% |
| Швидкість руху повітря | до 0,1 м/с |
| Теплий | Температура повітря в приміщенні | 23...25 °С |
| Відносна вологість | 40...60% |
| Швидкість руху повітря | 0,1...0,2 м/с |

Виміряні за допомогою приладів температура та вологість у робочому приміщені відповідають вказаним у таблиці 6.1 для теплого періоду року.

Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціювання повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у таблиці 4.2.

Розташовані у приміщенні 2 ПК і принтер являються джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ являється гранично допустима густина потоку енергії Іг.д, (вт/м2), яка встановлюється в залежності від площі опромінюваної поверхні тіла людини (Sопр).

Таблиця 4.2

Норми подачі свіжого повітря в приміщення з ПК

|  |  |
| --- | --- |
| Характеристика приміщення | Об'ємна витрата свіжого повітря, що подається в приміщення, мЗ на одну людину в годину |
| Об'єм до 20м3 на людину | Не менше 30 |
| 20... 40 м3 на людину | Не менше 20 |
| Більше 40 м3на людину | Може біти використана природна вентиляція |

Нормовані рівні складають:

Іг.д =35 Вт/м2 при Sопр > 50%;

Іг.д =70 Вт/м2 при Sопр ~ 25-50%;

Іг.д =100 Вт/м2 при Sопр < 25%.

Рівень шуму у приміщені не перевищує 50 дБ і відповідає нормам.

### **6.1.4 Виділення озону при роботі принтерів**

Озон внесений до списку речовин, максимальні значення концентрації яких на робочих місцях обмежені і строго визначені.

Вміст озону не повинен перевищувати концентрацій, вказаних у переліку «Гранично допустимих концентрацій забруднюючих речовин в атмосферному повітрі населених місць» № 3086-84 від 27.08.84 р. Для озону концентрація не повинна перевищувати 0,1 мг/м3 (ГОСТ 12.1.005-88).

Для запобігання виділення принтером озону слід застосовувати озонові фільтри. Необхідно взяти до уваги, що озоновий фільтр перестає діяти, якщо принтер завантажений мало або простоює без роботи. Тому слід міняти його через проміжки часу, рекомендовані виробником.

## **6.2 Заходи щодо поліпшення умов праці**

### **6.2.1 Електробезпека на робочому місці**

Встановлюються значення гранично допустимих рівнів напруги і струмів ГОСТ 12.1.038-82. Розглянуте приміщення відноситься до класу приміщень без підвищеної небезпеки поразки електричним струмом. В даному приміщенні відсутні ознаки підвищеної чи особливої небезпеки (вологості, провідникового пилу, струмоведучих основ (металевих, земляних), підвищеної температури.

Застосовувана техніка (ПК і принтер) відносяться до електроустановок напругою до 1000 В, підключення яких здійснюється від мережі однофазного змінного струму напругою 220 В, частотою 50 Гц. струм спрацьовування 25 А.

Усунути небезпеку поразки струмом при замиканні на корпус можна за допомогою установки в приміщенні нульового захисного провідника. Призначення нульового захисного провідника – створення для струму короткого замикання ланцюга з малим опором, щоб цей струм був достатнім для швидкого відключення ушкодженої установки від мережі.

Крім того, на робочому місці оператора ПЕОМ необхідно забезпечити захист від статичної електрики.

В приміщеннях розрядні струми статичної електрики найчастіше виникають при дотику обслуговуючого персоналу до кожного з елементів ПЕОМ.

Такі розряди можуть привести до виходу з ладу ПЕОМ. Вони впливають на працюючих, погіршують умови праці.

Для зниження величини виникаючих розрядів статичної електрики, покриття технологічних підлог варто виконувати з одношарового антистатичного лінолеуму марки АСН. До загальних заходів захисту від статичної електрики можна віднести зволоження повітря (до 75%), іонізацію повітря. Дані захисні міри регламентує ГОСТ 12.4.124-83.

### **6.2.2 Організаційна безпека**

Для запобігання перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для розминання м’язів очей.

Для розслаблення зорових м‘язів слід виконати наступні вправи:

* заплющити очі і масажувати їх коловими рухами пальців одну-дві хвилини;
* трьома пальцями кожної руки легко натиснути на верхні повіки, потім через одну-дві секунди зняти пальці з повік. Повторити п’ять разів;
* сісти вільно, не напружуючись, на низьке крісло, і виконувати колові рухи очима за годинниковою стрілкою та проти. Повторити по п’ять разів і після кожного покліпати.

## **6.3 Розрахунок та проектування інженерно-технічного заходу захисту від шкідливого (небезпечного) виробничого фактору (освітленість приміщення)**

Нормованим параметром природного освітлення є коефіцієнт природного освітлення (КПО).

КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота оператора ПК відноситься до робіт високої точності, для яких при використанні бокового освітлення КПО=2%.

Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт.

За даними вимірювань рівень природної освітленості поверхні, де розташований ПК, складає 200 лк при освітленості тієї ж поверхні відкритим небосхилом в 20000 лк, тобто КПО = 1,2%, що не відповідає нормам.

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи, які в порівнянні з лампами розжарювання мають ряд істотних переваг:

* за спектральним складом світла вони близькі до природного світла;
* мають підвищену світлову віддачу (у 2–5 разів вищу, ніж у ламп розжарювання);
* мають триваліший термін служби (до 10 тис. годин).

В приміщенні використовуються світильники серії Л 201. Кожен світильник комплектується двома лампами. Тобто необхідно використовувати 2 світильника із 4 працюючими лампами в них. На момент атестації робочого місця оператора працювало дві лампи, тому рівень штучного освітлення не задовольняв санітарним нормам.

Для покращення умов праці рекомендується збільшити рівень загальної освітленості приміщення шляхом встановлення двох додаткових ламп.

## **6.4 Висновки**

Було проведено дослідження робочого приміщення програміста на дотримання норм охорони праці. Виявлені недоліки в освітленні робочого місця та дані рекомендації щодо їх усунення. Запропоновані заходи щодо поліпшення умов та підвищення безпеки праці програміста у приміщенні.