



**КАФЕДРА ЦИФРОВОЇ ЕКОНОМІКИ ТА  
БІЗНЕС-АНАЛІТИКИ**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА  
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**ЗАТВЕРДЖУЮ**

Декана

\_\_\_\_\_ доц. Стасишин А. В.  
(підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**КОНСПЕКТ ЛЕКЦІЙ  
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Експертні системи**

(назва навчальної дисципліни)

галузь знань: 05 «Соціальні та поведінкові науки»  
(шифр та найменування галузі знань)

спеціальність: 051 “Економіка”  
(код та найменування спеціальності)

спеціалізація: Інформаційні технології в бізнесі  
(найменування спеціалізації)

освітній ступінь: бакалавр  
(бакалавр/магістр)

**Розглянуто та ухвалено  
на засіданні кафедри цифрової економіки та  
бізнес-аналітики  
протокол № 1 від “28” серпня 2020 р.**

Зав. кафедри \_\_\_\_\_ Шевчук І.Б.  
(підпис)

**ЛЬВІВ 2020**

## Конспект лекції № 1

### Тема № 1. ВСТУП ДО ЕКСПЕРТНИХ СИСТЕМ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”.

**Мета лекції:** познайомити з поняттям штучного інтелекту; розглянути експертні системи як науки, що виділилась із штучного інтелекту; представити класифікацію експертних систем.

#### План лекції

1. Поняття штучного інтелекту.
2. Експертні системи як різovid штучного інтелекту.
3. Класифікація експертних систем.

**Опорні поняття:** штучний інтелект, експертні системи, подання знань, вирішення завдань, системи самонавчання, когнітивне моделювання, класична структура ЕС.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій /Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника.[Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации.[Електрон. ресурс]. – Режим доступу:<http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу:<http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

## ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

### 1.1. Поняття штучного інтелекту

Розвиток штучного інтелекту(ШІ) став можливим і розпочався тільки після створення ЕОМ – в 40х роках ХХст. Сам термін ШІ був запропонований в 1956 році в Стентфордському університеті на семінарі, який був присвячений розробці логічних, а не обчислювальних задач. Після визнання ШІ самостійною наукою, відбулося її розділення на 2-а основні напрями%

- Нейрокібернетика – основною ідеєю нейрокібернетики було те, що єдиний об'єкт. Який здатен мислити, - це людський мозок. Тому будь-який пристрій, що здатен мислити, повинен певним чином відтворювати структуру людського мозку. Звичайні компютери для реалізації цього принципу були не здатні, зважаючи на обмежені ресурси. Тому цей процес орієнтований на створення транспютерів – паралельних компютерів з великою кількістю процесорів і нейрокомпютерів, що моделюють структуру нервових структур мозку людини
- Кібернетика «чорного ящика» - не має значення, яка структура "мислячого" пристрою. Головне, щоб на задані вхідні дії він реагував так, як людський мозок. Цей напрям орієнтований на пошуки алгоритмів вирішення інтелектуальних задач на існуючих моделях комп'ютерів.

На сьогодні перелік досліджень та дисциплін із штучного інтелекту постійно збільшується. Коротко розглянемо кожен із напрямів розвитку

#### 1. Подання знань

Подання знань є найважливішою областю досліджень з штучного інтелекту. Це основа решти дисциплін. Знання мають форму описів об'єктів, взаємозв'язків і процедур. Наявність адекватних знань і здатність їх ефективно використовувати означає "уміння". Мозок людини добре пристосований до образної обробки інформації, але при виконанні обчислень стає безпорадним навіть в порівнянні з невеликим калькулятором.

Чи можуть комп'ютерні системи (КС) відтворити образну обробку, яка здійснюється людським мозком і якщо так, то яким чином? Створення загальної теорії або методу подання знань є стратегічною проблемою. Така теорія здатна відкрити можливість накопичення знань, які потрібні щодня для вирішення все нових і нових задач.

Проте для досягнення поставленої мети необхідно, перш за все, знайти спосіб вираження загальних закономірностей нашого світу, в цьому і полягає суть проблеми подання знань.

#### Вирішення задач

В загальному поданні, вирішення задач зводиться до пошуку шляху з деякої початкової точки в цільову або кінцеву. Людина робить це досить

ефективно за допомогою міркувань (дедуктивного логічного висновку), процедурного аналізу, аналогії і індукції. Комп'ютерні системи, принаймні в даний час, вирішують задачі тільки з використанням дедуктивного логічного висновку і процедурного аналізу. Характерним є те, що тип задачі визначає метод, найбільш відповідний для її вирішення.

Отже, задачі, які зводяться до процедурного аналізу, краще за все розв'язуються на комп'ютері. Облікові задачі, ведення рахунків, аналіз надходження готівки можуть служити прикладами процедурних задач, вирішуваних комп'ютером швидше і надійніше, ніж людиною.

[Продовжити перегляд](#)

## 1.2. Класифікація експертних систем

Клас ЕС сьогодні об'єднує декілька тисяч різних програмних комплексів, які можна класифікувати за різними критеріями. Розглянемо лише класифікацію за деякими глобальними критеріями [8].

### Класифікація за типом задачі, що вирішується

*Інтерпретація даних.* Під інтерпретацією розуміється визначення значення даних, результати якого повинні бути узгодженими і коректними. Звичайно передбачається багатоваріантний аналіз даних. Це одна з традиційних задач для ЕС.

*Діагностика.* Під діагностикою розуміється виявлення несправності деякої системи. Несправність – це відхилення від норми. Таке трактування дозволяє з єдиних теоретичних позицій розглядати і несправність устаткування в технічних системах, і захворювання живих організмів і всілякі природні аномалії. Важливою специфікою є необхідність розуміння функціональної структури системи діагностування.

*Моніторинг.* Основна задача моніторингу – безперервна інтерпретація даних в реальному масштабі часу і сигналізація про вихід тих чи інших параметрів за допустимі межі. Головні проблеми – пропуск тривожної ситуації і інверсна задача помилкового спрацьовування.

Складність цих проблем – в розмитості симптомів „тривожних” сигналів.

*Проектування.* Підготовка специфікацій на створення об'єктів з наперед визначеними властивостями. Під специфікацією розуміється весь набір необхідних документів: креслення, записка пояснення і т.д. Основні проблеми тут – отримання чіткого структурного опису знань про об'єкт. Для організації ефективного проектування необхідно формувати не тільки самі проектні рішення, але і мотиви їх ухвалення. Таким чином, в задачах проектування

тісно пов'язуються два основні процеси: процес виведення рішення і процес пояснення.

*Прогнозування.* Системи прогнозування логічно виводять вірогідні наслідки із заданих ситуацій. У системі прогнозування звичайно використовується параметрична динамічна модель, в якій значення параметрів підганяються під задану ситуацію. Наслідки, що виводяться з цієї моделі, складають основу для прогнозів з оцінками вірогідності.

*Планування.* Під плануванням розуміється знаходження планів дій, що відносяться до об'єктів, здатних виконувати деякі функції. У таких ЕС використовуються моделі поведінки реальних об'єктів з тим, щоб логічно вивести наслідки планованої діяльності.

*Навчання.* Системи навчання діагностують помилки при вивченні будь-якої дисципліни за допомогою ЕОМ і підказують правильні рішення. Вони акумулюють знання про гіпотетичного учня і його характерні помилки, потім в роботі здатні діагностувати слабкості в знаннях тих, кого навчають, і знаходити відповідні способи для їх ліквідації. Такі системи планують процес спілкування з учнем залежно від успіхів учня, з метою передачі знань.

[Продовжити перегляд](#)

### Контрольні питання

1. Поняття системи штучного інтелекту.
2. Нейрокібернетика та кібернетика “чорного ящика”.
3. Взаємозв'язок ЕС та СШ. Місце ЕС в структурі ШІ.
4. Вплив штучного інтелекту на ідеологію програмування.
5. Основні відмінності інтелектуальних програм від традиційних.
6. Поняття експертної системи та її прикладне значення.
7. Сутність експертного аналізу. Основні ознаки класу експертних програм.
8. Взаємозв'язок та відмінності ЕС, програмних засобів ШІ і традиційних програм.
9. Класична структура ЕС.
10. Класифікація систем штучного інтелекту.

Укладач: \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)

## Конспект лекції № 2

### Тема № 2. КОМПЕТЕНЦІЯ ЕКСПЕРТНИХ СИСТЕМ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** познайомити з поняттям штучного інтелекту; розглянути експертні системи як науки, що виділилась із штучного інтелекту; представити класифікацію експертних систем.

#### План лекції

1. Людська і штучна компетенція та їх порівняння.
2. Рівні реалізації експертних систем.

**Опорні поняття:** людська компетенція, штучна компетенція, переваги компетенцій, критерій вибору задач, типи експертів.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій /Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника.[Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации.[Електрон. ресурс]. – Режим доступу:<http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу:<http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

## ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

### 1. Людська і штучна компетенція та їх порівняння

Експерт володіє величезними знаннями, уміє швидко знайти головну інформацію для розпізнавання ситуацій. Завдяки навчанню й досвіду він може робити те, що звичайні люди виконувати не здатні. Експерт здійснює пошук у просторі альтернатив вибірково, зводячи до мінімуму зайву роботу, ухиляючись від малоуспішних зусиль, якнайшвидше відсікаючи безперспективні шляхи дослідження. На ранніх стадіях експерт досягає високої продуктивності, оскільки найефективніше використовує свій час.

Висококваліфіковані експерти дуже цінуються в усьому світі, одержують високу зарплату. Однак, людина-експерт може втратити свої навички, виявляти небажані суб'єктивні схильності, а при відсутності помічників і не мати достатньої інформаційної бази для прийняття рішень.

За цих умов, незважаючи на дорожнечу розробки комп'ютерної експертної системи (роки роботи інженерів і експертів), її початкова вартість компенсується потім якістю прийнятих рішень і надійністю в роботі, а при достатньому тиражуванні розробки – часто й доступною вартістю при експлуатації. Тож перед початком розробки необхідно врахувати всі переваги й недоліки ручної і машинної праці, тобто людської і штучної компетенції

Таблиця 2.1 – Переваги штучної компетенції над людською

Людська компетенція	Штучна компетенція
1. Нестійка, залежить від тренування	Стійка, не зменшується. Залежить тільки від стану техніки
2. Важко передається і тиражується	Передається як програмний продукт. Особливо ефективна як тиражування досвіду, що потрібний у декількох місцях одночасно
3. Важко документується	Документується як комп'ютерна програма
4. Іноді непередбачувана	Точно відповідає алгоритму
5. Дорога за вартістю	Може бути припустимою за вартістю
6. Неможливість роботи в умовах небезпечного середовища	Може використовуватися в умовах ядерних реакторів, хімічних виробництв і т. п.

### Критерії вибору задач, що реалізуються методами і засобами експертних систем

**Критерій** – це свого роду правило, за допомогою якого можна щонебудь оцінити.

До основних критеріїв вибору задач, що реалізуються методами і засобами експертних систем, можна віднести такі:

1. Дані і знання повинні бути надійні, достовірні, не змінюватись в часі, тобто бути стабільними в процесі вирішення задачі: не повинні

коректуватися і не повинні містити помилок і суперечностей.

2. Простір або область можливих рішень відносно невелика. Простір пошуку повинен бути невеликим, оскільки необхідно зосередитися на вузькій предметній області, для якої характерний невеликий об'єм знань, у тому числі і оснований на здоровому глузді.

3. В процесі вирішення задачі повинні використовуватися формальні міркування, а задача повинна бути не дуже проста і не дуже важка для експерта (з розрахунку приблизно 30 хв. для експерта).

4. Задача повинна бути поставлена чітко, тобто визначені цілі (або мета), які ставляться перед експертною системою в процесі консультації, необхідний набір евристик, які використовуються в процесі вирішення задачі людиною.

5. Повинен бути, принаймні, один експерт, що вміє чітко виражати свої думки: явно формулювати свої знання і пояснювати методи застосування цих знань для вирішення задачі. -

[Продовжити перегляд](#)

### Контрольні питання

1. Визначте переваги штучної компетенції над людською.
2. Визначте переваги людської компетенції перед штучною.
3. Визначте для вирішення яких типів задач рекомендується застосовувати ЕС.
4. Наведіть критерії доцільності розробки ЕС.
5. Наведіть критерії виправданості розробки ЕС.
6. Наведіть критерії можливості розробки ЕС.
7. Основні властивості експертної системи.
8. Вкажіть критерії оцінювання рівня реалізації експертної системи.
9. Рекомендації щодо розробки експертної системи.
10. Приведіть приклади експертних закономірностей, що застосовуються у різних галузях людської діяльності.

Укладач: \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)



## Конспект лекції № 3

### Тема № 3. ЗНАННЯ, МЕТОДИ ТА ЕТАПИ ЕКСПЕРТНИХ СИСТЕМ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** охарактеризувати студентам методи експертних систем; розглянути етапи проектування експертних систем; пояснити поняття знань в експертних системах.

#### План лекції

3. Поняття знань. Використання знань в експертних системах.
4. Характеристика методів експертних систем.
5. Етапи проектування експертних систем.

**Опорні поняття:** знання, дані, властивості даних, етапи проектування ЕС, ідентифікація проблеми, здобуття знань, структурування знань, процес агрегування та дезагрегування, види структурних ієрархій предметної області, формалізація проблеми як етап проектування ЕС, тестування як етап проектування ЕС.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

## ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

### 1. Поняття знань. Використання знань в експертних системах.

Головною відмінністю ЕС від інших програмних засобів є наявність бази знань, в якій зберігаються знання експертів про відповідну проблемну область. Знання зберігаються у вигляді сукупності записів деякою мовою подання знань, яка дозволяє легко змінювати та поповнювати базу знань в формі, яку розуміють спеціалісти. В традиційних же програмах знання жорстко „зашиті” в алгоритм і скорегувати їх може лише сам автор програми, тобто лише програміст (за умови, що він пам’ятає за яким принципом працює одна із множини створених ним програм).

До останнього часу саме різні мови подання знань (МПЗ) були центральною проблемою при розробці ЕС. Зараз існують десятки мов і моделей знань, серед яких найбільше поширення отримали продукції, семантичні сітки, фрейми, числення предикатів першого порядку і об’єктно-орієнтовані мови програмування. Вибір конкретної моделі визначається структурою знань в конкретній предметній області. Перед тим як обрати МПЗ і створити базу знань (БЗ) необхідно виявити цю структуру та виконати структурування знань.

Що ж таке знання? Загальне означення знань: «Перевірений практикою результат пізнання дійсності, правильне її відображення в мисленні людини». Однак таке означення не пояснює чим саме відрізняються програми, які основані на знаннях від традиційних прикладних програм.

Складність поняття знань полягає в множинності його матеріальних носіїв. За цією ознакою можна виділити п’ять основних форм знань:

Z1 - знання в пам’яті людини;

Z2 - матеріальні знання (підручники, довідники, статті і т. і.);

Z3 - поле знань (напівформалізований опис Z1 і Z2);

Z4 - знання мовою подання знань (формалізація Z3);

Z5 - база знань в комп’ютері (на машинних носіях інформації).

Наприклад, людина вирішує важкі і цікаві проблеми використовуючи не чіткі алгоритми, а виключно свій досвід в формі накопичених знань (Z1). Частину свого досвіду людство зберігає у вигляді книг чи заміток (Z2). Сукупність Z1 і Z2 створює знання про предметну область, які не пов’язані з машинною обробкою.

В ЕС використовують «екстракт» Z3 того, що вдалось отримати з Z1 і Z2. При цьому співвідношення емпіричного досвіду Z1 і його формалізованої теоретичної основи з книжок Z2, є різним для різних предметних областей.

Вважають, що чим більша вага  $Z1$  в процесі прийняття рішень, тим більш придатна ЕС для впровадження в дану область.

При створенні ЕС принциповим є етап розробки поля знань  $Z3$ . Поле знань – це деякий напівформалізований опис понять предметної області і зв'язків, що існують між ними. Наприклад у вигляді рисунка, таблиці, схеми, діаграми, мережі і т.д. В подальшому поле знань переписується деякою мовою подання знань, при цьому створюється модель знань  $Z4$ . Реалізація моделі за допомогою програмних засобів веде до виникнення п'ятої форми подання знань  $Z5$  – бази знань.

Важливо пам'ятати, що не у будь-якій предметній області доцільно виділяти знання. Знання важливі там, де присутні розмиті означення, зміни понять, залежність ситуацій від множини контекстів, де присутня велика доля невизначеної та нечіткої інформації, тобто в областях де переважають емпіричні знання, де накопичення фактів випереджає розвиток теорії (медицина, геологія, юриспруденція, фінанси тощо). Такі ж добре структуровані області знань як математика, фізика, теоретична механіка мають добре розвинутий формальний апарат для опису своїх закономірностей, що дозволяє успішно проводити машинне моделювання з використанням традиційного алгоритмічного програмування (без виділення рівнянень).

Таким чином, прийmemo в якості робочого, таке означення: **знання** – це основні закономірності предметної області, які дозволяють людині розв'язувати конкретні виробничі, наукові та інші задачі, тобто – факти, поняття, взаємозв'язки, оцінки, правила, евристики (або фактичні знання), а також стратегії прийняття рішень в цій області (стратегічні знання).

Чим же саме знання відрізняються від даних? Спробуємо виділити за аналогією п'ять форм даних:

D1 - результат спостереження об'єкта (наприклад, фіксація температури повітря на термометрі) або дані в пам'яті (наприклад, дата народження);

D2 - фіксація даних на матеріальному носії - таблиці, графіку, і т.і. (наприклад, таблиці Брадеса або дані про температуру протягом деякого інтервалу часу);

D3 - модель даних, деяка схема опису, яка пов'язує декілька об'єктів;

D4 - дані на мові опису даних;

D5 - база даних на машинному носії інформації.

Традиційно виділяють лише три рівні: зовнішній D1, логічний D3 та фізичний D5, які відповідають рівням знань  $Z1$ ,  $Z3$  і  $Z5$ .

Існує ще один аспект, який відображує відмінність даних і знань. Будь яке поняття, яке використовує людина, має два боки – екстенціонал та інтенціонал.

Екстенціонал – це визначення поняття через перелік понять нижчого рівня ієрархії або фактів, що відносяться до означення, тобто це набір конкретних фактів, що відповідають даному поняттю.

Інтенціонал – це визначення через поняття вищого рівня абстракції з вказуванням специфічних властивостей, тобто означення, або опис поняття через його властивості.

Наприклад, для поняття «поліклініка» екстенціоналом буде перелік типу: «поліклініка №2», «дитяча поліклініка», «обласна поліклініка» і т. і. Інтенціонал же в цьому випадку можна визначити так: «медична установа для надання амбулаторної допомоги за місцем проживання і роботи». Наприклад, для реляційної бази даних екстенціональними поданнями є конкретні факти про предметну область (рядок таблиці, або його стовець). Тобто екстенціональні відношення створюють просторову структуру такої бази даних, а інтенціональні - відповідають схемі бази даних, причому аргументами інтенціонального відношення виступають атрибути, домени яких використовувались для створення відповідного екстенціонального відношення.

Інтенціонал відокремлює знання від даних, які завжди задаються екстенціонально. Особливість інтенціоналу полягає в його точності, виразності і правильно виділених властивостях. Наприклад, коли давньогрецький вчений Платон визначив інтенціонал людини, як «тварина на двох ногах, яка не має пір'я», інший давньогрецький вчений Діоген спростував його означення, принісши обскубаного півня [13].

[Продовжити перегляд](#)

### **Контрольні питання**

1. Поняття знань.
1. Визначте та проаналізуйте форми знань та форми даних.
2. Порівняльна характеристика знань і даних.
3. Чи доцільно виділяти знання у будь-якій предметній області?
4. Поняття інтенціоналу та екстенціоналу.
5. Класифікація знань за поняттям глибини.
6. Основні положення методу «швидкого прототипування».
7. Процес та етапи проектування ЕС: загальна характеристика.
8. Ідентифікація проблеми як етап проектування ЕС.
9. Здобуття знань як етап проектування ЕС.
10. Структурування знань як етап проектування ЕС.
11. Процес агрегування та дезагрегування.

**Укладач:** \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)

## Конспект лекції № 4

### Тема № 4. РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** охарактеризувати студентам основні інструментальні засоби експертних систем; розглянути етапи розробки експертних систем.

#### План лекції

6. Інструментальні засоби експертних системах.
7. Етапи розробки експертних систем.

**Опорні поняття:** Засоби організації даних і знань, засоби підтримки, сервісні засоби, комплексні засоби, інтегровані засоби, демонстраційний прототип, дослідницький прототип, діючий прототип, промислова система, комерційна система,

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

## ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

### 1. Загальна характеристика інструментальних засобів

При розробленні експертних систем використовується спеціальний інструментарій, який дозволяє значно скоротити час розробки. Такий інструментарій включає як програмні, так і апаратні засоби. До апаратних засобів відносяться: ПЕОМ, інтелектуальні робочі станції, послідовні символні ЕОМ (ЛІСП і ПРОЛОГ машини), ЕОМ загального призначення та паралельні символні ЕОМ. Крім того, для розширення можливостей числових процесорів всіх типів випускаються спеціальні символні співпроцесори.

Загальну класифікацію програмного інструментарію можна показати так:

- 1) Процедурні мови, орієнтовані на обробку символної інформації (наприклад, ЛІСП і т.і.);
- 2) мови інженерії знань, тобто мови високого рівня, які орієнтовані на розробку ЕС (наприклад, PROLOG, OPS-5);
- 3) засоби автоматизації процесів конструювання, використання та модифікації ЕС (наприклад, RLL, HEARSAY-4 і т.і.);
- 4) пусті (базові) ЕС або „оболонки”, які не містять знань ні про яку ПО (наприклад, CODEX, EMYCIN, DECTOOLS і т.і.).

В наведеній вище класифікації інструментальні засоби (ІЗ) розташовані в порядку зменшення трудовитрат, необхідних при створенні за їх допомогою конкретних ЕС. При використанні інструментарію першого типу програміст вимушений самостійно програмувати всі компоненти ЕС на мові достатньо низького рівня. На другому рівні продуктивність різко зростає, однак за рахунок деякого падіння ефективності. ІЗ третього рівня дозволяють розробнику ЕС не розробляти всі або деякі компоненти ЕС, а обирати їх із заздалегідь сформованого набору. При використанні ІЗ четвертого рівня розробник повністю звільняється від роботи зі створення програм, оскільки він має в своєму розпорядженні пусту систему, яку необхідно наповнити знаннями з відповідної ПО [19].

Разом з тим, при використанні ІЗ третього та четвертого типів виникає і ряд проблем:

- реалізовані в них стратегії управління виведенням можуть не відповідати методам вирішення, які використовує експерт, наслідком чого можуть стати неефективні або взагалі неправильні рішення;
- мова подання знань може не підійти для даної ПО.

В наш час дуже широке розповсюдження отримали ІЗ, які називають оболонками, що налагоджуються. Вони дозволяють використовувати оболонку не в деякому «застиглому» вигляді, а генерувати її з деякої множини механізмів, які передбачені в ЕС (наприклад, ЕКО, ЕКСПЕРТИЗА, НЕКС). В останній час розмежують оболонки, які отримали назву "оболонок застосування" від тих оболонок, які призначені для здобуття знань. Крім того стали виділяти універсальні та спеціалізовані оболонки.

Таким чином, ми розглядатимемо інструментальні засоби побудови ЕС в якості систем програмування, які спрощують проектування ЕС. Відповідно, їх склад і структура визначаються особливостями вирішуваних експертними системами задач і технологій проектування ЕС. Необхідно відзначити, що вибір технології і інструментальних засобів реалізації ЕС – це ключове питання створення ЕС.

[Продовжити перегляд](#)

## **2. Стадії розробки експертних систем інструментарію**

Після завершення розробки першого прототипу, експерт і інженер зі знань мають нагоду оцінити, що саме буде включено в розробку остаточного варіанта системи. Для цього іноді необхідно виділити додаткові етапи (стадії існування) при переході ЕС від прототипу до промислового зразка:

- демонстраційний прототип,
- дослідницький прототип,
- діючий прототип,
- промислова система,
- комерційна система.

Найчастіше реалізується плавний перехід від демонстраційного прототипу до промислової системи.

### **Характеристики прототипів**

*Демонстраційний прототип:* система вирішує частину задачі, демонструючи життєздатність підходу (декілька десятків правил або понять).

*Дослідницький прототип:* система вирішує більшість задач, але нестійка в роботі і не повністю перевірена (декілька сотень правил або понять).

*Діючий прототип:* система надійно вирішує всі задачі на реальних прикладах, але для складної задачі вимагає багато часу і обсягів пам'яті.

*Промислова система:* система забезпечує високу якість рішень при мінімізації необхідного часу і пам'яті; переписується з використанням ефективніших засобів подання знань.

*Комерційна система*: промислова система, придатна до продажу, добре документована і забезпечена сервісом.

Таким чином, на третьому етапі роботи створюється "демонстраційний" прототип. Доцільно об'єднати його з дослідницьким, щоб відпрацювати більшу кількість правил. Інженер зі знань спільно з експертом заповнює БЗ системи значною кількістю правил.

Звичайно вважається, що для "демонстраційного прототипу" необхідний обсяг БЗ порядку сотні правил. При заповненні такої бази проводиться постійний контроль знань, що неперервно заносяться в систему, на несуперечність з уже існуючими знаннями за допомогою засобів, які є в робочій БЗ, або вручну. Отже, основне навантаження на третьому етапі несе інженер зі знань. Також на цьому етапі перевіряється успішність вибору експерта.

Про ступінь кваліфікації експерта можна судити за відсотком банальних висновків системи в загальному обсязі одержаних рішень.

В процесі роботи можуть виникнути складнощі з формалізацією одержаних знань відповідно до вибраної структури БЗ. Всі ці моменти фіксуються, але робота повинна продовжуватися з використанням вибраних засобів розробки ЕС. Тільки у разі виникнення конфлікту між можливостями вибраних засобів з фундаментальними властивостями знань, який неможливо ліквідувати без істотних порушень точності роботи ЕС, необхідно повернутися до вибору нового ІЗ.

[Продовжити перегляд](#)

## Контрольні питання

1. Поняття інструментальних засобів побудови експертних систем.
2. Апаратні та програмні інструментальні засоби.
3. Загальна класифікація програмного інструментарію.
4. Характеристика основних видів програмного інструментарію.
5. Інтегровані засоби програмного інструментарію.
6. Стадії розробки експертних систем.
7. Взаємозв'язок та послідовність етапів розробки експертних систем.
8. Процес та основні критерії оцінювання промислової ЕС.
9. Стадії розробки інструментальних засобів.
10. Універсальність інструментальних засобів для розробки ЕС.

**Укладач:** \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)



## Конспект лекції № 5

### Тема № 5. АНАЛІЗ ПРОЦЕСУ ЗДОБУТТЯ ЗНАНЬ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”.

**Мета лекції:** охарактеризувати поняття процесу здобуття знань; зробити огляд структуризації знань предметної області; розглянути методи здобуття знань.

#### План лекції

8. Поняття процесу здобуття знань.
9. Структуризація знань предметної області.
10. Методи здобуття знань.

**Опорні поняття:** Засоби організації даних і знань, засоби підтримки, сервісні засоби, комплексні засоби, інтегровані засоби, демонстраційний прототип, дослідницький прототип, діючий прототип, промислова система, комерційна система,

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

# ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

## 1. Поняття процесу здобуття знань

Наведемо означення терміна здобуття знань (knowledge acquisition),

"здобуття знань – це передача потенційного досвіду вирішення проблем, отриманого від деякого джерела знань, і перетворення його до вигляду, який надасть можливість використовувати його у програмі" [22].

Необхідно відзначити, що термін „здобуття знань” має узагальнений характер і зовсім нейтральний до способу передачі знань. Наприклад, передача може здійснюватися за допомогою спеціальної програми, що у процесі обробки великого масиву історій хвороби встановлює зв'язок між симптомами і захворюваннями.

Цей же термін застосовується і для позначення процесу взаємодії експерта зі спеціальною програмою, метою якого є:

- отримати будь-яким систематичним способом знання, якими володіє експерт, наприклад, пропонуючи експерту репрезентативні задачі і фіксуєчи пропоновані способи їх вирішення;
- зберегти отримані в такий спосіб знання в деякому проміжному вигляді;
- перетворити знання з проміжного подання у вигляд, придатний для практичного використання в програмі, наприклад у набір породжувальних правил. Перевага використання такої програми – зниження трудомісткості процесу, оскільки перенесення знань від експерта до системи здійснюється в один прийом.

Варто відзначити, що на практиці при виявленні знань у ході опитування експертів за робочий день вдається сформулювати від двох до п'яти "еквівалентів породжувальних правил". Причин такої низької продуктивності декілька:

- перш ніж приступити до опитування експертів, інженер зі знань, що не є фахівцем у даній предметній області, повинен витратити досить багато часу на ознайомлення з її специфікою і термінологією; тільки після цього процес опитування може стати продуктивним;
- експерти схильні думати про знайому їм область не стільки в термінах загальних принципів, скільки в термінах окремих типових об'єктів, подій і їхніх властивостей;
- для подання специфічних знань про предметну область потрібно підібрати придатну систему позначень і структурну оболонку, що само по собі є непростою задачею.

Як відомо, будь-яку складну задачу найкраще розбити на підзадачі, і саме так ми зробимо із задачею здобуття знань.

У роботі [22] пропонується виконати аналіз процесу здобуття знань у термінах моделі процесу проектування експертної системи (рис. 6.1):

1. *Ідентифікація.* Аналізується клас проблем, що передбачається вирішувати за допомогою проектованої системи, включаючи дані, якими потрібно оперувати, і критерії оцінювання якості рішень. Визначаються ресурси, доступні при розробці проекту, – джерела експертних знань; трудомісткість; обмеження за часом, вартістю та обчислювальними ресурсами.
2. *Концептуалізація.* Формулюються базові концепції і відношення між ними. Сюди ж входять і характеристика різних видів використовуваних даних, аналіз інформаційних потоків та на їх основі структур в предметній області, в термінах причинно-наслідкових зв'язків, відношень частина-ціле, постійне-тимчасове іт.п.
3. *Формалізація.* Починається спроба подати структуру простору станів і характер методів пошуку в ньому. Виконується оцінювання повноти і ступеня вірогідності (невизначеності) інформації й інших обмежень, що накладаються на логічну інтерпретацію даних, таких як залежність від часу, надійність і повнота різних джерел інформації.
4. *Реалізація.* Перетворення формалізованих знань у діючу програму, причому на перший план виходить специфікація методів організації керування процесом і уточнення деталей організації інформаційних потоків. Правила перетворюються у форму, придатну для виконання програмою в обраному режимі керування. Приймаються рішення про використовувані структури даних і розбиття програми на ряд більш-менш незалежних модулів.
5. *Тестування.* Перевірка роботи створеного варіанта системи на великому числі репрезентативних задач. У процесі тестування аналізуються можливі джерела помилок у поведженні системи. Найчастіше таким джерелом є наявний у системі набір правил. Виявляється, що в ньому не вистачає певних правил, одні – абсолютно коректні, а між деякими виявляються протиріччя.

Існує класифікація методів здобуття знань і це дозволяє інженерам зі знань залежно від конкретної задачі і ситуації обрати конкретний метод .

Комунікативні методи охоплюють всі види контактів з живим джерелом знань – експертом. Текстології – методи здобуття знань з документів і спеціальної літератури.

Розділення цих груп методів на верхньому рівні класифікації не означає їх антагоністичність, звичайно інженер зі знань їх комбінує.

Комунікативні методи в свою чергу також діляться на дві групи: активні і пасивні.

Пасивні – припускають провідну роль за експертом, а інженерзі знань лише фіксує (протоколює, записує на диктофон) міркування експерта під час його реальної роботи з прийняття рішень або записує те, що експерт вважає потрібним розповісти у вигляділекції.

При активних методах, навпаки, ініціатива повністю в руках інженера зі знань, який активно контактує з експертами різними способами.

Пасивні методи достатньо прості, але вимагають від інженера зі знань чітко аналізувати "потік свідомості" експерта і виявляти в ньому значущі фрагменти знань. Тут відсутній зворотний зв'язок, що ослаблює ефективність цих методів і визначає їх допоміжну роль.

Активні методи залежно від числа експертів, що передають свої знання, також діляться на дві групи. Якщо число експертів більше одного, то доцільно крім індивідуальних зустрічей застосовувати методи групових обговорень предметної області. Ці методи активізують мислення і дозволяють виявити нетривіальні аспекти знань. Проте індивідуальні методи і сьогодні є провідними, оскільки така делікатна процедура як отримання знань не передбачає зайвих свідків.

[Продовжити перегляд](#)

## **6.1 Контрольніпитання**

1. Поняття здобуття знань.
2. Аналіз процесу здобуттязнань.
3. Стадії здобуття знань.
4. Рівні аналізу знань.
5. Схема диференціації знань у системі KADS.
6. Онтологічний аналіз і здобуттязнань.
7. Класифікація та характеристика методів здобуття знань.
8. Концептуальна структура предметної області.
9. Функціональна структура предметної області.
- 10.Формалізація і програмна реалізація бази знань.

Укладач: \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПШБ, посада, науковий ступінь, вчене звання)

## Конспект лекції № 6

### Тема № 6. ПОДАННЯ ЗНАНЬ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** охарактеризувати поняття процесу здобуття знань; зробити огляд структуризації знань предметної області; розглянути методи здобуття знань.

#### План лекції

11. Поняття подання знань та її принципи.
12. Моделі подання знань.

**Опорні поняття:** подання знань, принципи подання знань, логічні моделі, продукційні моделі, семантичні мережі, фреймові моделі, модель дошки оголошень.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

# ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

## 1. Поняття подання знань

В галузі експертних систем подання знань означає не що інше, як систематизовану методику опису на машинному рівні того, що знає людина-експерт, яка спеціалізується в конкретній предметній області. Але помилково вважати, начебто подання знань зводиться до кодування в сенсі, аналогічному шифруванню. Якщо закодувати повідомлення, підставивши деяким регулярним образом замість одних символів інші, то отриманий результат не має нічого спільного з поданням змісту повідомлення в тому сенсі, як це припускається в теорії штучного інтелекту, навіть якщо отриманий код легко сприймається на машинному рівні і його можна зберігати в пам'яті комп'ютера.

Звернемо увагу хоча б на те, що в такому коді зберігається та лексична чи структурна неоднозначність, що притаманна природній людській мові. Наприклад, повідомлення "Відвідування тіточки може бути докучливим" буде настільки ж неоднозначним у закодованому виді, як і на "людській" мові. Переклад цього тексту в машинний код не позбавить нас від того, що це повідомлення можна трактувати і як твердження, що "набридає наносити візити тіточці", і як твердження, що "набридає, коли тіточка наносить візит".

Таким чином, один з парадоксів штучного інтелекту полягає в тому, що багато задач пошуку значеннєвого змісту, що легко вирішуються людиною, дуже важко реалізувати на ЕОМ і навпаки.

Будь-яке спілкування людини зі світом техніки припускає наявність деякого попереднього знання. Якщо, наприклад, хтось береться за пошук несправності в цифровій схемі, то можна припустити, що він володіє певними базовими знаннями з області електротехніки. Немає необхідності підкреслювати, що комп'ютер у чистому вигляді ніяких попередніх знань не має, а тому технічна експертність – набір якостей, що лежать в основі високого рівня роботи людей-фахівців при вирішенні проблем у певній вузькій області, – повинна включати і ці попередні знання.

І нарешті, подання припускає певну організованість знань. Подання знань повинне дозволити отримувати їх у потрібній ситуації за допомогою нескладного і більш-менш природного механізму. Простого переведення інформації (знань) у форму, придатну для збереження на машинних носіях, тут явно недостатньо. Для того щоб можна було досить швидко отримувати ті елементи знань, що найбільш придатні в конкретній ситуації, база знань повинна мати досить розвинуті засоби індексування і контекстної адресації. Тоді програма, що використовує знання, зможе керувати послідовністю застосування певних "елементів" знань навіть не маючи точної інформації про те, як вони зберігаються.

Звичайно, програмний код, виконуваний комп'ютером, повинен відповідати застосовуваній системі позначень, але це не можна вважати занадто серйозним обмеженням. Багато схем подання, на перший погляд, надзвичайно сильно відрізняються, проте виявляються насправді формально

еквівалентними, тобто все, що може бути виражене в одній системі подання, може бути виражене й в іншій

Перш ніж перейти до розгляду конкретних прикладів, давайте уточнимо термінологію, узявши за основу цитати з "класичних" робіт зі штучного інтелекту.

*Подання* (representation) у роботі вченого Уїнстона визначається як "безліч синтаксичних і семантичних угод, що уможлиблює опис предмета". У штучному інтелекті під "*предметом*" розуміється стан деякої проблемної області, наприклад об'єкти в цій області, їх властивості, відношення, які існують між об'єктами. *Опис* (description) "дозволяє використовувати угоди з подання для опису певних предметів". Синтаксис подання специфікує набір правил, що регламентують об'єднання символів для формування виразів мовою подання. Можна говорити про те, що вираз гарно чи погано сформований, тобто про те, наскільки він відповідає цим правилам. Зміст повинні мати лише добре сформовані вирази [28].

Загальноприйнятим в галузі штучного інтелекту є синтаксис у вигляді конструкції предикат-аргумент, що має форму

$$\langle \text{вираз} \rangle ::= \langle \text{предикат} \rangle (\langle \text{аргумент} \rangle, \dots, \langle \text{аргумент} \rangle)$$

У цій конструкції за  $k$ -значним предикатом повинні впливати  $k$  аргументів. Так, предикат „*at*” може бути двомісним відношенням, у якому перший аргумент виступає як ім'я деякого об'єкта, а в якості другого – його місцезнаходження (наприклад, кімната):

$$at(\text{робот}, \text{кімната } A)$$

Семантика подання специфікує, як повинен інтерпретуватися вираз, побудований відповідно до синтаксичних правил, тобто як з його форми можна отримати певний зміст. Специфікація звичайно виконується присвоєнням змісту окремим символам, а потім індукуванням присвоєння в більш складних виразах. Так, привласнюючи зміст символам „*at*”, робот, кімната  $A$ , ми можемо сказати, щовираз

$$at(\text{робот}, \text{кімната } A)$$

означає: робот знаходиться в кімнаті  $A$  (але не навпаки – кімната  $A$  знаходиться в роботі).

Процес вирішення проблеми, як правило, містить у собі поряд із поданням предметів навколишнього світу і судження про деякі дії. Як уже було відзначено деякі проблеми формулюються в термінах вихідного і цільового станів і множини операцій, які можна використовувати при спробах перетворити початковий стан у цільовий. Але тут залишається нез'ясованим питання про те, як можна подавати операції.

[Продовжити перегляд](#)

## Контрольні питання:

1. Поняття та принципи подання знань.
2. Логічні моделі подання знань: загальні положення.
3. Формальні мови.
4. Числення висловлювань.
5. Числення предикатів.
6. Логічне програмування. Мова PROLOG.
7. Принци презолуюцій.
8. Продукційні моделі подання знань: загальні положення.
9. Проектування продукційних експертних систем.
10. Характеристика структурних елементів ЕС продукційного типу.
11. Семантичні мережі як модель подання знань: загальні положення.
12. Класифікація семантичних мереж.
13. Використання семантичних мереж при розробці експертних систем.
14. Фреймові моделі подання знань: загальні положення.
15. Основи теорії фреймів.
16. Основні властивості фреймів.
17. Модель дошки оголошень: загальні положення.
18. Принципи організації систем з дошкою оголошень.
19. Система HEARSAY: загальна характеристика.
20. Використання джерел знань у HEARSAY-II.
21. Оболонка для створення систем з дошкою оголошень HEARSAY-III.

Укладач: \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)



## Конспект лекції № 7

### Тема № 7. СЕРЕДОВИЩЕ CLIPS. ЗАГАЛЬНА ХАРАКТЕРИСТИКА

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** охарактеризувати поняття процесу здобуття знань; зробити огляд структуризації знань предметної області; розглянути методи здобуття знань.

#### План лекції

1. Вступ до CLIPS. Історія створення CLIPS.
2. Робота з CLIPS. Синтаксис визначень.

**Опорні поняття:** ознайомитися з експертною оболонкою CLIPS, охарактеризувати основні переваги та недоліки, розглянути основи функціонування CLIPS.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.
- Експертні системи в міжнародних відносинах / Ю.О.Лунь, А.М. Козел, С.М. Ніколаєв. – Львів% Видавництво Львівської політехніки, 2011. – 196с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

## ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

### 1. Вступ до CLIPS. Історія створення CLIPS.

Спочатку абревіатура **CLIPS** була назвою мови — **C Language Integrated Production System** (тобто мова C, інтегрована із продукційними системами), зручної для розробки баз знань і макетів експертних систем.

Тепер **CLIPS** являє собою сучасний інструмент, призначений для створення експертних систем (expert system tool). **CLIPS** складається з інтерактивного середовища — експертної оболонки зі своїм способом подання знань, гнучкої й потужної мови й декількох допоміжних інструментів. Зараз, завдяки добрій волі своїх творців, **CLIPS** є абсолютно вільно розповсюджуваним програмним продуктом.

Зараз на ринку доступно не так багато експертних оболонок (інструментів, призначених для створення експертних систем). Незважаючи на те, що **CLIPS** поширюється безкоштовно, він досить успішно конкурує навіть із найвідомішими комерційними проектами. Кількість користувачів **CLIPS** росте рік у рік. Про це можна судити по активності відвідування сайтів, форумів і конференцій, присвячених **CLIPS**.

Поява мови **CLIPS** можна датувати 1984 р., місце народження **CLIPS** — космічний центр Джонсона NASA. Саме в цей час відділ штучного інтелекту (тепер Software Technology Branch) розробив множину прототипів експертних систем, що використовують сучасне програмне й технічне забезпечення. Однак, незважаючи на великий потенціал експертних систем, не багато хто із цих додатків дійшли до кінцевого споживача. Ця невдача обумовлювалася технологією створення експертних систем, який у той час оперували в NASA. Основні обмеження накладав мову **LISP**, використовувана як базова мова для розробки експертних систем.

**Як головні недоліки мови LISP можна виділити наступні три:**

- 1) недостатня адаптованість **LISP** до широкого кола стандартних комп'ютерів;
- 2) висока ціна технічного й програмного забезпечення, призначеного для роботи з **LISP**;
- 3) низька здатність інтеграції систем, написаних на **LISP**, із системами, написаними на інших мовах (виробництво вкладених додатків).

Співробітники відділу штучного інтелекту зрозуміли, що використання традиційних мов програмування, таких як C, усуне більшість виниклих проблем, і відділ почав пошуки виробників і постачальників інструментів для створення експертних систем, що оперують одним із традиційних мов програмування. Незважаючи на те, що число подібних інструментів було досить велике, ціна таких інструментів виявилася досить висока. Крім того, більшість із цих інструментів працювали на дуже невеликому числі платформ, а швидкість їхньої роботи залишала бажати кращого. Стало очевидно, що для одержання інструмента, що задовольняє всім вимогам NASA, необхідна розробка власного засобу для створення експертних систем.

**Прототип CLIPS був розроблений навесні 1985 р.**, деяким більш ніж за два місяці. Особлива увага була приділена створенню мови, сумісного з

мовами, що використовуються в NASA у той момент. Таким чином, синтаксис мови CLIPS був зроблений дуже схожим на синтаксис експертної оболонки ART, розробленою корпорацією Inference.

Незважаючи на те, що ART послужив прообразом, CLIPS розроблявся зовсім без допомоги Inference або доступу до вихідних кодів системи ART.

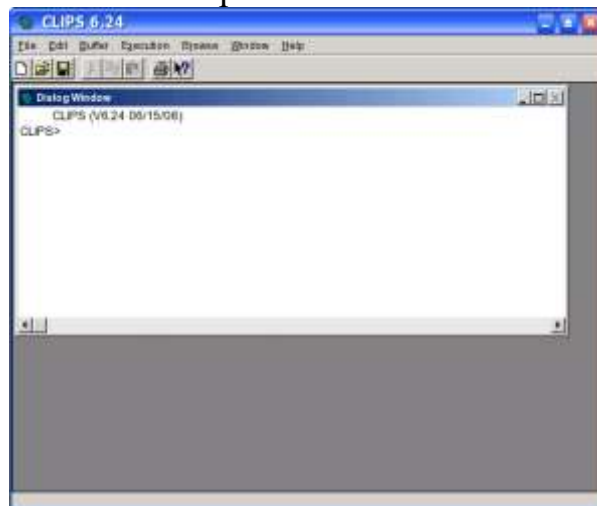
Завдяки тому, що CLIPS є вільно розповсюджуваним програмним продуктом з доступними вихідними кодами, останнім часом була випущена множина програм і бібліотек, що вдосконалять і доповнює можливості CLIPS. Деякі із цих продуктів є власністю їхніх компаній, що випустили, і призначені для внутрішнього використання або комерційного поширення, інші, як і сам CLIPS, поширюються вільно.

Як самі відомі приклади подібних проектів можна привести DLL/OCX-бібліотеку, що дозволяє використати механізм логічного висновку CLIPS у ваших додатках, FuzzyCLIPS, CLIPS++, CLIPS code generator.

[Продовжити перегляд](#)

## 2. Робота з CLIPS. Синтаксис визначень.

Для демонстрації прикладів, використовуваних у цій книзі, буде застосовуватися Windows-версія CLIPS 6.2. Незважаючи на повну сумісність із Apple Macintosh й UNIX-версіями, при роботі з даною книгою бажано використати саме Windows-версію середовища CLIPS. Зовнішній вигляд головного вікна CLIPS показаний на рис. 1.



**Рис. 1. Головне вікно CLIPS**

Windows-версія середовища CLIPS повністю сумісна з базовою специфікацією мови. Уведення команд здійснюється безпосередньо в головне вікно CLIPS. Однак у порівнянні з базовою Windows-версією надає множину додаткових візуальних інструментів (наприклад, менеджери фактів або правил), значно полегшуюче життя розробника експертних систем.

Експертні системи, створені за допомогою CLIPS, можуть бути запуснені трьома основними способами:

- уведенням відповідних команд і конструкторів мови безпосередньо в середовище CLIPS;

- використанням інтерактивного віконного інтерфейсу CLIPS (наприклад для версій Windows або Macintosh);
- за допомогою програм-оболонок, що реалізують свій інтерфейс спілкування з користувачем і механізмів, що використовують, знань і логічного висновку CLIPS.

Крім того, CLIPS при запуску дозволяє виконувати командні файли власного формату (ця можливість також доступна за допомогою команди batch). Для реалізації цієї можливості необхідно запустити CLIPS (у нашому випадку це файл CLIPSWin.exe) з одним із трьох наступних аргументів:

- -f<імені-файлу>
- -f2<імена-файли>
- -l<імені-файлу>

Текстовий файл, заданий за допомогою опції -f, повинен містити команди CLIPS.

### **Контрольні питання:**

1. Що таке CLIPS?
2. Основні переваги CLIPS.
3. Основні недоліки CLIPS.
4. Якими способами можуть бути запущені експертні системи, створені за допомогою CLIPS?
5. Для чого використовуються багатокрапки в CLIPS?
6. Для чого використовуються пробіли та символи табуляції в CLIPS?

**Укладач:** \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
 (підпис) (ПІБ, посада, науковий ступінь, вчене звання)

## Конспект лекції № 8

### Тема № 8. СЕРЕДОВИЩЕ CLIPS. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ

**Міжпредметні зв'язки:** Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

**Мета лекції:** ознайомитися з функціональними можливостями CLIPS.

#### План лекції

1. Основні елементи мови. Абстракції даних.
2. Подання знань. Об'єктно-орієнтовані можливості CLIPS.

**Опорні поняття:** ознайомитися з експертною оболонкою CLIPS, охарактеризувати основні переваги та недоліки, розглянути основи функціонування CLIPS.

#### Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій /Навчальний посібник. - К.: НАУ, 2012. – 132с.
- Експертні системи в міжнародних відносинах / Ю.О.Луць, А.М. Козел, С.М. Ніколаєв. – Львів: Видавництво Львівської політехніки, 2011. – 196с.

#### Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника.[Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации.[Електрон. ресурс]. – Режим доступу:<http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу:<http://www.inftech.webservis.ru/it/information/>

**Навчальне обладнання:** ТЗН, презентація тощо.

# ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

## 1. Основні елементи мови. Абстракції даних.

Синтаксис мови CLIPS можна розбити на три основних групи елементів, призначених для написання програм:

- примітивні типи даних;
- функції, що використовуються для обробки даних;
- конструктори, призначені для створення таких структур мови, як факти, правила, класи й т.д.

Розглянемо кожну із цих трьох груп більш докладно.

### Типи даних

CLIPS підтримує 8 *примітивних типів даних*: **float**, **integer**, **symbol**, **string**, **external-address**, **fact-address**, **instance-name**, **instance-address**.

Для *зберігання* чисельної інформації призначаються типи **float** й **integer**, для *символічної* - **symbol** й **string**.

Число в CLIPS може складатися тільки із символів цифр (0-9), десяткової крапки (.), знака (+ або -) і експонентного символу (e) з відповідним знаком, у випадку подання числа в експонентній формі. Нижче наведені приклади припустимих в CLIPS подань цілих і речовинних типів:

Приклад 1. *Подання* чисел в CLIPS

Цілі: **237 15 +12-32**

Речовинні: **237e3 15.09 +12.0 -32.3e-7**

Визначення цілого значення можна представити в такий спосіб:

Визначення 1. *Подання* цілого числа

**<ціле> ::= [+ | -]<цифра>+  
<цифра> ::= 0|1:2|3|4 | 5 | 6|7|8 | 9**

Речовинне значення має наступний синтаксис:

Визначення 2. *Подання* речовинного числа

**<речовинне> ::= <ціле>  
<експонента> | <ціле> . [експонента] |  
<беззнаковое-целое> [експонента] |  
<ціле> . <беззнаковое-целое> [експонента] <беззнаковое-целое> ::=  
<цифра>+<експонента>: := e|E  
<ціле>**

Якщо послідовність символів не відповідає наведеним вище визначенням цілого або речовинного числа, то дана послідовність сприймається CLIPS як значення типу **symbol**.

Значенням типу **symbol** може бути будь-яка послідовність символів, що починається з будь-якого не керуючого ASCII-символу. Значення типу **symbol** закінчується *обмежником*.

Обмежниками є будь-які невідображувані символи (наприклад, пробіл, символ табуляції або переходу на інший рядок), подвійні лапки, що відкривають або закриває кругла дужка, символи &, |, < й ~. Крапка з коми (;)

є символом початку коментарів і також може обмежувати значення типу `symbol`. Обмежувачі-символи-обмежники не можуть утримуватися в значенні `symbol`, за винятком `<`, що може бути першим символом значення. Значення типу `symbol` не може починатися із символу `?` або `$?`, але може містити ці символи. CLIPS є мовою, чутливим до регістра.

Значення типу, **string** являє собою рядок символів, укладену в подвійні лапки. Символ подвійних лапок також може бути включений у рядок. Для цього перед символом `"` необхідно поставити символ зворотної косої риси (`\`). Для включення в рядок символу зворотної косої риси необхідно використати два послідовних символи `\`. Приклади припустимих значень `string` наведені нижче:

[Продовжити перегляд](#)

### Контрольні питання:

1. Основні типи елементів, призначених для написання програмв синтаксисі мови CLIPS.
2. Що являє собою значення типу `string` в CLIPS.
3. Що таке факт в CLIPS.
4. Що таке функції в CLIPS.
5. Що таке вираз в CLIPS.
6. Що таке об'єкт в CLIPS.
7. Що таке модуль в CLIPS.
8. Які основні можливості ОПП ви знаєте?

Укладач: \_\_\_\_\_ Старух А.І., доцент, к.е.н.  
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)