

---

# КОМП'ЮТЕРНІЕ НАУКИ

---

УДК 004.4; 004.8

## СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ – ОСНОВА СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЇХ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

*ЗАЯЦЬ В.М., ЗАЯЦЬ М.М.*

На основі системного підходу (від простого до складного) робиться класифікація сучасних систем штучного інтелекту (СШІ), аналізуються способи подання даних і знань в СШІ, відзначаються особливості кожної з них та вказуються напрями доцільного застосування та подальшого розвитку. Проводиться порівняльний аналіз програмних засобів та мов програмування для їх реалізації.

**Ключові слова:** програмне забезпечення, системи штучного інтелекту, декларативні мови програмування, семантичні сітки, фрейми, продукційні системи, Лісп, Пролог, універсальні мови моделювання, логіко-лінгвістичні моделі.

**Key words:** software, artificial intelligence, declarative programming language, frames, Lisp, Prolog, universal modeling language

### 1. Вступ

З часу появи робіт зі штучному інтелекту їх розвиток пішов у двох напрямках.

Моделювання діяльності мозку, його психофізіологічних властивостей з метою відтворення на ЕОМ або за допомогою інших технічних засобів інтелекту чи штучного розуму – так званий біонічний напрям.

Імітація розумової поведінки людини з використанням засобів комп'ютерної техніки. Це основний прагматичний напрям розвитку робіт зі штучного інтелекту. За останні десятиліття досягнуто особливих успіхів. У цьому напрямку психофізіологічна діяльність мозку залишається в стороні. Тут комп'ютер розглядається як інструмент, на якому можуть бути написані добрі чи погані програми, що тією чи іншою мірою імітують розумову діяльність людини.

Поняття “штучний інтелект” в цьому напрямі розуміється метафорично, а дослідники зайняті розробленням програм, які дозволяють засобам комп'ютерної техніки відтворювати процеси, що є результатом розумової діяльності людини. Нажаль, на обчислювальну техніку нерідко поширюють ідею антропоморфізму. Можна часто зустріти вирази “машина прийняла рішення”, “машина розпізнає ситуацію”, “машина грає в шахи”.

Трапляються цілком невіглаські уявлення про “машини, що думають” в буквальному розумінні цих слів.

Насправді за всією цією “розумовою” діяльністю комп'ютера стоїть людина, яка для нього склала програму-інструкцію, реалізація якої і забезпечує імітацію розумової діяльності людини, але не більше.

Задачі, які розв'язуються в штучному інтелекті, є основою сучасних інформаційних технологій. Вчорашні інформаційні технології – це передовсім технології паперові, оскільки їх реалізація потребує колосальної кількості паперу. Традиційне використання комп'ютерної техніки в цьому плані становища не змінює. Старі інформаційні технології використовують комп'ютерну техніку через ряд посередників: фізиків, математиків, програмістів, спеціалістів з предметних областей. Знання про цьому в кожній предметній області фіксуються на паперових носіях.

У сучасних інформаційних технологіях спочатку, за літературними джерелами вивчають проблему, на її основі формують модель, створюють алгоритми її опрацювання на комп'ютері, відтак розв'язують задачу та аналізують її результати. При цьому забезпечується можливість отримання нових знань. Такі технології дозволяють можливість не лише ґрунтовно вивчити проблему, але й сприяють удосконаленню технічних і програмних засобів

У традиційних комп'ютерних інформаційних технологіях домінують моделі, які мають синтаксичний, описовий характер, а не суто процедурний, інструктивний характер. Для прикладу, математичні моделі у вигляді системи диференціальних рівнянь можуть описувати процеси в об'єктах найрізноманітнішої природи. Тобто для них характерний універсалізм. Якщо пред'явити користувачеві самі моделі (без коментарів), то він не зможе визначити, який об'єкт, які процеси описуються, а може зробити найзагальніші висновки про те, з якими класами об'єктів зіставимі ці моделі. Коментарі про конкретні предметні знання залишаються поза ЕОМ.

Становлення нової інформаційної технології зумовлено тим, що в теорії штучного інтелекту (а раніше в теорії управління, запропонованій академіком Д.А. Поспеловим) були розроблені логіко-

лінгвістичні моделі [2]. Вони дають змогу формалізувати конкретні змістовні знання про об'єкти і процеси, що в них відбуваються, і ввести в комп'ютер, поряд з математичними, логіко-лінгвістичні. До них належать семантичні сітки, фрейми, продукційні системи, які часто об'єднують як програмно-апаратні засоби систем штучного інтелекту (ПЗ СШ).

Мета даної роботи – на основі системного підходу від простого до складного провести класифікацію сучасних систем штучного інтелекту, дати їх порівняльну характеристику, проаналізувати способи подання даних і знань та вказати напрями подальшого їх вдосконалення.

## 2. Особливості сучасних систем штучного інтелекту

Системи штучного інтелекту, які функціонують за принципами нової інформаційної технології, поділяють на:

- Інтелектуальні інформаційно-пошукові системи (системи типу питання-відповідь), які забезпечують в процесі діалогу взаємодію користувача-непрограміста з базами даних і знань на професійній мові користувача, близькій до природної.
- Розрахунково-логічні системи, які дають змогу користувачам, що не є професійними програмістами і фахівцями в прикладній математиці, розв'язувати в діалоговому режимі професійні задачі, використовуючи складний математичний апарат і відповідне програмне забезпечення.
- Експертні системи, що дозволяють комп'ютеризувати області, в яких знання можна подати в експертній описовій формі, але використання математичних моделей, характерних для точних наук, ускладнене або й взагалі неможливе.

Саме завдяки експертним системам штучний інтелект набув стратегічного значення в розвитку науки і управління, проектування і екології, космосу і фізики Землі, освіти і матеріалознавства. Особливого значення набувають так звані гібридні експертні системи – об'єднання традиційних експертних систем з розрахунково-логічними, тобто в гібридних експертних системах логіко-лінгвістичні моделі використовуються поряд з математичними.

Зазначимо ще одну особливість нової інформаційної технології – адаптацію і гнучкість програмних систем відносно задач з тієї чи іншої предметної області. У всіх цих випадках не можна априорі сказати, за допомогою якої моделі, алгоритму і системи розв'язується задача чи розпізнається ситуація.

Так, коли встановлюють медичний діагноз, невідомо заздалегідь, яке правило буде використано з наявної бази знань. Це характерно для всіх інтелектуальних пакетів прикладних задач.

Отже, всі системи штучного інтелекту орієнтовані на знання, а тому подальший прогрес систем штучного інтелекту і комп'ютерних інформаційних технологій визначається розвитком трьох основних теоретичних проблем:

- представлення знань – центральна проблема штучного інтелекту;
- комп'ютерна лінгвістика – забезпечує процес природно-мовного спілкування з ЕОМ та процес автоматизованого перекладу текстів з іноземних мов;
- комп'ютерна логіка, яка має особливе значення для розвитку експертних систем. Мета її – моделювання людського мислення і перетворення програмування з мистецтва в науку.

Штучний інтелект можна визначити як комплексну науково-технічну проблему, для розв'язання якої дослідження ведуться у чотирьох напрямках:

- а) моделювання на комп'ютері окремих функцій творчих процесів: ігрові задачі, автоматичне доведення теорем, автоматичний синтез програм, аналіз музичних творів, розпізнавання та ідентифікація складних динамічних об'єктів;
- б) зовнішня інтелектуалізація комп'ютерів: фундаментальні дослідження із створення діалогового інтерфейсу (зовнішня в тому розумінні, що інтелектуальні системи на початковому етапі створюють на ЕОМ існуючої архітектури, враховуючи комп'ютери неймановської структури);
- в) внутрішня інтелектуалізація комп'ютерів – це створення комп'ютерів нової архітектури, основаної на принципах штучного інтелекту і призначеної для створення інтелектуальних систем;
- г) цілеспрямована поведінка роботів, що здатні автономно вирішувати завдання, які поставила людина.

Перш ніж аналізувати проблеми штучного інтелекту, розглянемо, як науково-технічний прогрес привів до становлення нових інформаційних технологій.

В основу науково-технічного прогресу покладено зростання знань, яке відбувається швидше, ніж за експонентою.

Якщо  $x(t)$  – маса знань, то, вважаючи, що знання зростають за законом експоненти, маємо

$$x(t) = px(t),$$

де  $p$  – постійна величина. Звідси

$$x(t) = x(0)e^{pt}.$$

У наукознавстві вживають такий термін, як період подвоєння знань  $T$ . Безсумнівно,  $P$  і  $T$  є функціями часу і повинні бути уточнені. Величина  $T$  постійно падає, а  $P$  – зростає.

Дані про зміну  $T$  і  $P$ , як видно з таблиці, показують, що з 50-х років минулого століття спостерігається інформаційний вибух, який виявляється у збільшенні самого показника експоненти.

Дані про період подвоєння знань та коефіцієнт пропорційності

Рік	1900	1950	1970	1981	1990	2000
$T$ (роки)	50	10	5	2.5	1.2	0.6
$P$ (рік-1)	0.014	0.07	0.14	0.28	0.55	1.15

Зростання знань приводить до ускладнення народного господарства. Для ефективного управління господарськими механізмами виникає потреба в комп'ютеризації народного господарства та науки, яка має в наші дні таке саме значення, як індустріалізація країни в 20 – 30-ті роки ХХ ст.

Комп'ютеризація означає масове використання обчислювальної техніки, а тому професійних програмістів не вистачає, щоб забезпечити масове використання ЕОМ у народному господарстві. Традиційні комп'ютерні технології характеризуються наявністю посередників між ЕОМ і користувачами, до яких належать аналітики, програмісти, постановники задач, оператори ЕОМ.

Крок до позбавлення від посередників – це введення термінального доступу до ЕОМ, що значно зменшує кількість посередників у ланці ЕОМ - користувач. Для ефективного розв'язання своїх задач користувач повинен здійснити ще один крок – забезпечити прямий доступ до комп'ютера.

Інший шлях забезпечення прямого доступу до комп'ютера полягає у створенні засобів програмного забезпечення, які реалізуються всередині комп'ютерів, тобто через створення інтелектуального інтерфейсу ЕОМ. Його можна реалізувати на основі логіко-лінгвістичних моделей та інших методів й ідей, розроблених в області штучного інтелекту. Досягнення в області штучного інтелекту настільки значні, що якщо раніше йшлося про ін-

дустрію обробки даних, то тепер йдеться про індустрію інтелектуальних систем.

З використанням обчислювальної техніки пов'язані такі поняття, як програми та дані: на початкових етапах програміст розробляв програму, а потім вводив у неї дані. Відтак сталися значні зміни – дані були відділені від програми. З'явилися дані у вигляді різних структур: реляційні (табличні), мережі, ієрархічні та системи управління базами даних (СУБД), банки даних та знань.

Для відділення даних від програм використовувалися засоби опису даних, які містяться в мовах програмування. У таких мовах, як Фортран та Алгол розвивались порівняно нескладні засоби опису структур даних. Складніші засоби містять мови Паскаль, Сі, Сі++. Паралельно розвивалися способи запису даних на зовнішніх носіях інформації. Тут фундаментальним стало поняття файлу, що має ім'я і містить всі необхідні структуровані записи даних про різні об'єкти, з якими працює система.

Отже, файл можна розглядати як інформаційну модель об'єкта.

У процесі подання даних на зовнішніх носіях можна виділити такі етапи:

- способи формування записів даних у файлах, ведення файлів; організацію доступу до них повністю визначають користувачі;
- керування файлами та доступ до них забезпечується операційною системою;
- створення баз даних і систем управління ними з метою їх оптимальної обробки в інтересах усього підприємства.

З появою нових інформаційних технологій з'явилися і нові способи подання даних і програм у США.

Таким чином, можна зробити висновок, що нові інформаційні технології є основою сучасних США, розвиток яких своєю чергою є могутнім стимулом для створення як нового програмного забезпечення, так і появи нових мов програмування та засобів для їх ефективно технічної реалізації.

### 3. Способи подання даних та знань в США

Сьогодні можна говорити про новий етап представлення даних у пам'яті комп'ютера – створення інформаційно-розрахункових мереж, а на їх основі – розподілених баз колективного користувача.

З появою СШ з'явилися і нові поняття, такі як база знань. Безумовно, що дані й структура бази даних певною мірою відображають знання про предметну область і її структуру. Проте специфічними ознаками, що відрізняють знання від даних, пов'язаних з їх поданням у комп'ютері, можна вважати такі чотири ознаки [2, 22]:

- внутрішня інтерпретованість;
- структурованість;
- зв'язність;
- активність.

Якщо звернутися до структур даних, то деякі із ознак, властиві знанням, будуть справедливими і для даних. Так, інтерпретованість явно простежується в реляційній базі даних, де імена стовпців є атрибутами відношень, імена яких вказані в рядках.

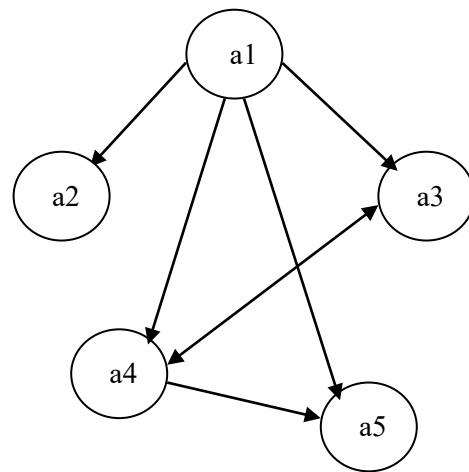
Другу ознаку – структурованість можна подати декомпозицією складного об'єкта на простіші і встановленням зв'язків між простими. Це зв'язки типу відношень клас-підклас, вид-підвид, список-підсписок. Ці зв'язки можна спостерігати в сіткових та ієрархічних моделях даних. Ці самі відношення при бажанні можна задати і в табличних базах даних.

Що стосується третьої ознаки – зв'язність, то практично неможливо знайти аналогів у згадуваних базах даних, оскільки наші знання зв'язані не лише в розумінні структури. Вони відображають закономірності у фактах, процесах, явищах та причинно-наслідкові відношення між ними.

Стосовно четвертої ознаки (активність) склалася така ситуація, що при використанні комп'ютерів нові знання породжуються програмами, а дані пасивно зберігаються в пам'яті комп'ютера. Людині властива пізнавальна активність. І це принципово відрізняє знання від даних. Таким чином, протиріччя у знаннях стимулюють здобуття нових знань. Таким же стимулом до знань є неповнота. Загалом, не завжди вдається розмежувати дані і знання. Однак вони існують і привели до появи спеціальних формалізмів у вигляді моделей подання знань в комп'ютері, які відображають всі чотири ознаки знань. Сьогодні є, принаймні, чотири види моделей і відповідних їм мов представлення знань: мови (моделі) семантичних мереж; системи фреймів; логічні мови (моделі) та продукційні системи.

Розглянемо моделі семантичних мереж. У семантичній сітці вершинам поставлені у відповідність

поняття (об'єкти, події, процеси, сигнали), а дугам – відношення на множині понять. З цієї причини мову семантичних мереж ще називають реляційною. Тут досить виразно проступає така ознака знань як зв'язність. У семантичній сітці можна відобразити знання, які виражені на природній мові. Візьмемо речення: “Мисливець сів на пліт, переїхав на другий берег, поклав у корзину качку і взяв корзину з качкою.” Можна виділити п'ять об'єктів: мисливець (a1), пліт (a2), другий берег (a3), корзина (a4), качка (a5). Ці об'єкти зв'язані відношеннями: сів на (Г1), переїхав на (Г2), поклав у (Г3) і взяв (Г4). Сітку, що відповідає цьому реченню, показано на рисунку.



Семантична мережа подання речень природною мовою

Згідно з логікою реального світу ми можемо доповнити вихідний текст відношеннями, які явно не містяться в тексті. Доповнений текст матиме вигляд: Мисливець сів на пліт і на плоті переїхав на другий берег. На другому березі була качка. Мисливець поклав качку в корзину. Мисливець взяв корзину з качкою.

Треба вказати на одну важливу обставину: в індоевропейських мовах, що підтвердили експериментальні дослідження, є не більше двохсот відношень, що не зводяться одне до одного. Комбінації цих відношень дозволяють виразити інші відношення, які фіксуються в текстах. Ця обставина дозволяє сподіватися, що будь-яку предметну область можна описати за допомогою скінченної множини відношень і здійснити автоматичну по-

будову семантичних мереж безпосередньо з тексту.

Частковим випадком семантичних мереж є сценарії або однорідні сітки. Це сітки, в яких об'єкти зв'язані єдиними відношеннями строгого або не строгого порядків з різною семантикою.

Для прикладу, якщо об'єктами-поняттями будуть роботи різного виду (чи окремі операції деякої роботи), а єдиними відношеннями строгого порядку – відношення слідування, то ми отримаємо сітковий графік комплексу робіт. Очевидно, сценарії є зручним засобом складання планів.

Семантична сітка загального виду дає змогу зрозуміти відмінності між поняттями бази даних і бази знань. Предметна область – це множина допустимих станів її компонент. Множина, представлена через загальні поняття і відношення між ними, утворює базу знань у вигляді інтенціональної семантичної сітки. З іншого боку, залежно від ситуації, компоненти предметної області мають конкретне значення, властивості, характеристики. Всі ці конкретні значення будуть відображені в так званій екстенціональній семантичній сітці або бази даних природної структури.

Ще один важливий аспект семантичних мереж – це представлення в них елементів загального виду за допомогою логічних міркувань та використання прикладних програм. Ці елементи можна реалізувати використовуючи так звані віртуальні відношення.

Але слід зазначити, що семантичні сітки, незважаючи на їх різноманіття, мають ряд недоліків. Надто довільна структура і різні типи вершин та зв'язків потребують потужного програмного забезпечення для їх обробки. Це зумовило появу різних модифікацій семантичних мереж: синтагматичні ланки, сценарії, фрейми тощо. Коротко зупинимося на фреймових моделях.

Термін фрейм (рамка) асоціюється з природним уявленням людини про об'єкт як щось довершене, обрамлене. Формально під фреймом розуміють структуру такого вигляду:

$$f[\langle v_1, g_1 \rangle, \langle v_2, g_2 \rangle, \dots, \langle v_k, g_k \rangle],$$

де  $f$  – ім'я фрейма; пара  $\langle v_i, g_i \rangle$  – це  $i$  є слот, де  $v_i$  – ім'я слота, а  $g_i$  – його значення.

Часто фрейми, свою чергою, поділяють на дві групи: фрейми-описи та рольові фрейми. Приклади фреймів:

1) описовий фрейм:

[ $\langle$ фрукти $\rangle$ ,  $\langle$ виноград, болгарський 20 кг $\rangle$ ,  
 $\langle$ вишня, володимирська 200 кг $\rangle$ ];

2) рольовий фрейм:

[ $\langle$ перевезти $\rangle$ ,  $\langle$ що, прокат 300 кг $\rangle$ ,  
 $\langle$ куди, Москву 200 кг $\rangle$ ,  $\langle$ звідки, Череповець $\rangle$ ,  
 $\langle$ чим, залізничним транспортом $\rangle$ ].

Якщо в наведених прикладах і в загальному виразі забрати всі значення слотів, то отримаємо фрейм, який названо прототипом фрейму: фреймом-інтенціоналом. Фрейми з конкретними значеннями слотів називаються фреймами-прикладями, фреймами-екземплярами.

Фрейми властивість вкладеності, тобто значення слоту може виступати ціла система імен слотів глибшого рівня. Така структура запису фреймів дає змогу успішно оперувати з гігантськими масивами даних, реалізовувати і ефективно управляти базами даних та створювати нові бази знань. Очевидно, для ефективного використання таких вкладених даних в одну структур слід використовувати мови програмування та відповідне програмне забезпечення, що оптимально пристосоване для цих цілей.

#### 4. Характеристика програмних засобів та мов програмування для реалізації СШ

Для розроблення та реалізації сучасних СШ розроблено відповідні технічні та програмні засоби, які можуть бути реалізовані за допомогою спеціальних мов програмування, які пристосовані для цих цілей. Для створення СШ найбільш вдалими є декларативні мови програмування, які задають не послідовність виконання дій для досягнення поставленої мети, а описують об'єкт дослідження і взаємозв'язки між його складовими. В результаті цього маємо логічну або функціональну модель об'єкта досліджень. Використовуючи вбудовані механізми опрацювання даних і програм в таких мовах, можемо отримувати нові знання про об'єкт дослідження, оперативно аналізувати його динаміку і управляти його поведінкою.

В основу сучасних мов логічного програмування покладено мову Пролог [7-13], що з'явилася в 1970

році водночас з такими поширеними мовами як Паскаль і Сі. Орієнтація мов логічного програмування – це нетрадиційне застосування комп'ютерної техніки: розуміння природної мови, створення баз знань, інтелектуальних систем прийняття рішень [4, 22-25], систем розпізнавання об'єктів, експертних систем [1, 3, 5, 6], систем перекладу текстів, інформаційно-аналітичних систем та інших задач, які прийнято відносити до проблем штучного інтелекту [2, 5, 22, 33]. Принципова відміна мови Пролог від традиційних мов у тому, що програма на мові логіки описує не процедуру розв'язання задачі, а створює логічну модель предметної області – деякі факти відносно властивостей даної предметної області і відношення між цими властивостями, а також правила вислуду нових властивостей і відношень на основі заданих. Отже, Пролог є описовою мовою, а не процедурною. Основні поняття мови досвідченими програмістами сприймаються без особливих затруднень, а от практичне втілення цих знань в корисні і ефективні програми викликає певні ускладнення.

В основу сучасних мов функціонального програмування покладено мову Лісп, яка запропонована Дж. Маккарті [32] в 1960 р. і орієнтована на розв'язання задач нечислового характеру. Англійська назва мови Lisp є скороченням виразу опрацювання списків і підкреслює область її застосування. У вигляді списків зручно відображати алгебраїчні вирази, множини, графи, правила виводу, граматику природних мов та інші складні об'єкти. Списки є однією з найбільш гнучких форм представлення інформації в пам'яті комп'ютера.

Після появи Ліспу [13-19] рядом авторів розроблено інші алгоритмічні мови, орієнтовані на розв'язання задач штучного інтелекту, серед яких слід відзначити Пленер, Снобол, Рефал, Пролог [5]. Тим не менше мови функціонального програмування успішно розвиваються і майже за сорокарічну історію існування з'явилося ряд діалектів функціональної мови: Common Lisp [14], MacLisp, R-Lisp [21], InterLisp, Standart Lisp, Reduce [22] та ряд інших.

Відзначимо, що відміни між ними не носять принципового характеру і зводяться, в основному, до несуттєвої різниці у формі запису програми та набору вбудованих функцій. Тому програміст, який освоїв одну із мов функціонального програмування, без особливих зусиль в стані освоїти будь-яку іншу, володіючи основними методами та алгоритмами побудови функціональних програм.

Незважаючи на очевидні переваги декларативних мов перед процедурними, довгий час ці мови роз-

вивалися і застосовувалися у вузькому колі спеціалістів з проблем штучного інтелекту. Лише протягом останніх двох десятиліть декларативні мови програмування викликали зацікавлення широкого кола спеціалістів з інформатики. Зростання популярності цих мов зумовлено виходом теорії штучного інтелекту в сферу прикладного програмування.

На сьогоднішній день з'явилися престижні національні проекти створення ЕОМ нових поколінь, в яких інтелектуальний інтерфейс, орієнтований на користувача-непрофесіонала в інформатиці, займає центральне місце.

Пролог був прийнятий як базова мова в японському проекті ЕОМ п'ятого покоління, який орієнтований на дослідження методів логічного програмування і штучного інтелекту.

Теоретичною основою Прологу є символічна логіка, яка дає механізм обчислення предикатів. Цій мові притаманний ряд властивостей, які відсутні в традиційних мовах програмування. До них відносяться: механізм логічного доведення з пошуком і поверненням, вбудований механізм співставлення зі зразком, проста і виразна структура даних з можливістю її зміни. Пролог відрізняє однакову структуру даних і програм. Оскільки вони не відрізняються, то їх можна змінювати під час роботи Пролог-системи. В Пролозі, як і у функціональних мовах, відсутні вказівники, оператори присвоєння та переходу. Природним методом програмування тут є рекурсія. Ряд важливих рис теорії програмування впливає з декларативності мов, що дозволяє розуміти програму не досліджуючи динаміки її виконання. Однак істинна потужність цих мов ґрунтується не на одній якійсь властивості, а в сукупному і грамотному поєднанні всіх. Мови декларативного характеру орієнтовані не на розробку розв'язків, а на систематичний і формалізований опис задачі з тим, щоб розв'язок впливав із складеного опису. В сучасному програмуванні позначився перехід від пошуку розв'язку до визначення того, який розв'язок потрібний. Якщо мови логічного програмування ідеально пристосовані для реалізації процедури логічного висновку, то функціональні мови [14-21] дозволяють будь-яку синтаксично правильно побудовану програму на довільній мові програмування записати у вигляді функції, тим самим оптимальним чином побудувати інтерпретатор досліджуваної мови програмування.

## 5. Системний підхід до освоєння та застосування декларативних мов програмування

Традиційні (алгоритмічні) мови програмування порівняно з декларативними (описовими) є доволі об'ємні і громіздкі, оскільки не дають змоги:

- максимально використовувати можливості сучасної комп'ютерної техніки для забезпечення ефективності програмних засобів;
- ясно і наочно відображати алгоритми програми, щоб забезпечити легкість перевірки та модифікації останніх.

Строго функціональні та мови логічного програмування, що відносяться до декларативних мов програмування [7-21], є доволі простими, а тому забезпечують достатньо високий ступінь виразності програм, порівняно з традиційними мовами (Algol, Fortran, Pascal, C, Delphi, C++). Ряд функціональних програм можуть ефективно працювати на сучасних комп'ютерах, тим не менше не так ефективно, як відповідні програми з оператором присвоєння. Це пов'язано зі структурою архітектури сучасних комп'ютерів. Окрім того, вибір дещо іншої структури представлення даних, ніж це прийнято в інструментальному Ліспі, забезпечує як більшу ясність представлення програм, так і підвищення їх ефективності з використанням сучасних комп'ютерів старої архітектури.

З одного боку, сучасні мови програмування повинні ефективно використовувати сучасні машини, а з іншого – ясно виражати програмні алгоритми, щоб полегшити перевірку останніх. Строго функціональна мова, будучи простою за своєю структурою, демонструє вищий ступінь виразності порівняно з традиційними мовами, де існує оператор присвоєння. Це зв'язано, великою мірою, зі способом вибору структур даних. Дещо інший їх вибір можна забезпечити підвищенням ефективності функціональних програм і на сучасних комп'ютерах. Одна з фундаментальних властивостей мови програмування, що дає можливість ясно описати обчислення на цій мові, – простота семантики мови. Велика перевага функціональної мови, як і логічної, в тому, що тут є кілька основних понять, кожне з яких має просту семантику. Зокрема, семантика функціональної мови розуміється в термінах значень, які мають вирази, але аж ніяк не в термінах дій і послідовності їх використання. Але з практичного погляду було б справедливіше зробити висновок, що строго функціональна мова є надзвичайно елементарною і її розширення значно б підвищили ефективність і ясність деяких класів вираховань. Очевидно, необхідно розрізнити суто синтаксичні розширення і розширення, які вима-

гають зміни семантики мови. Зміна семантики вимагає обережності, оскільки це ускладнює розуміння (ясність) вже відлагоджених функціональних програм.

Будь-яку систему позначень для опису алгоритмів і структури даних можна назвати мовою програмування, хоча, як правило, вимагається ще й реалізація мови на комп'ютері.

Сьогодні розроблено сотні різних мов програмування. Ще в 1969р. Саммет [32] наводить список з 120 мов, які доволі широко застосовуються. Ця приголомшлива кількість мов програмування протирічить тому, що більшість програмістів у своїй практиці використовують декілька мов програмування, а значна їх частина – одну або дві мови. Виникає питання про доцільність освоєння різних мов програмування, якщо навряд чи буде можливість їх реалізації. Тим не менше, якщо не обмежуватися поверховим ознайомленням з мовою, а мати глибокі уявлення про поняття, які лежать в основі конструювання програм на даній мові, то поза всяким сумнівом можна переконатися у доцільності освоєння різних мов програмування, виходячи з таких міркувань:

1. Внаслідок вивчення різних мов програмування покращується розуміння конкретної мови, її понять та основних методів і прийомів, що в ній використовуються. Типовим прикладом може бути рекурсія. При правильному її використанні можна отримати елегантну ефективну програму, а застосування її до простого алгоритму могло б привести до астрономічного збільшення часових затрат. З іншого боку, недоступність використання рекурсії в таких мовах, як Фортран, Кобол та розуміння основних принципів і методів реалізації рекурсії може внести ясність в певні обмеження мови, які, на перший погляд, є надуманими.
2. Значення мов програмування розширює запас корисних програмістських конструкцій і сприяє розвитку мислення. Працюючи з структурами даних однієї мови, виробляють і відповідну структуру мислення. Вивчаючи конструкції інших мов та методи їх реалізації, розширюють програмістський тезаурус.
3. Знання великої кількості мов програмування дає змогу обґрунтовано вибирати певну мову для розв'язання конкретної задачі.
4. Освоєння нової мови програмування, як і природної мови людського спілкування, завжди легше, якщо відомі декілька мов.
5. Знання принципів побудови різноманітних мов програмування полегшує розробку нової мови.

Конструювання сучасних мов програмування сьогодні далеке від досконалості. Кожна з відомих мов має свої недоліки та переваги.

Більшість сучасних мов програмування є універсальними, оскільки дають змогу записати будь-який алгоритм цією мовою, якщо не накладати обмежень на час виконання програми та місткість пам'яті, алгоритмічну складність тощо. Якщо хто-небудь запропонує нову мову програмування, то вона, очевидно, буде універсальною, якщо ігнорувати обмеження на пам'ять або час. Порівнюючи різні мови програмування, слід виходити не з кількісного співвідношення того, що вони дозволяють зробити, а з якісних відмінностей, що визначають елегантність (короткість і наочність), легкість (прозорість) та ефективність (швидкодія та технічні засоби) програмування на них. Це порівняння слід здійснювати в контексті конкретної сфери застосування.

Очевидно для детального розуміння особливостей, синтаксису, семантики, діапазону прикладних застосувань декларативних мов програмування (Lisp [14-21, 24, 28], Prolog [7-13, 28], UML [31-33]), які покликані обслуговувати задачі штучного інтелекту [2, 4-6, 8, 10-12, 22, 33], при їх вивченні слід застосовувати системний підхід. Суть підходу зводиться до викладу матеріалу від простого до складного з наведенням численних прикладних задач, які в змозі розв'язати на даній мові програмування. Поряд з цим слід чітко уявляти обмеження, які накладені на використовувану мову програмування та сферу її прикладних застосувань.

При системному підході [30] забезпечується можливість постійного порівняння різних описових мов програмування [27], постійна зміна і оновлення прикладних задач та освоєння нових можливостей логічних предикатів у Пролозі чи функціональних описів в Універсальному Ліспі.

Авторам видається, що при викладанні декларативних мов програмування такий системний підхід можна забезпечити шляхом викладання лекційного, практичного матеріалу чи постановки лабораторних робіт у вигляді аплікацій, виконаних на мові програмування UML [31] чи іншій об'єктно-орієнтованій мові моделювання [5, 6, 12, 22-29], яка забезпечує можливість як постійного поновлення підготовленого матеріалу, так і постійного порівняння з іншими підходами, їх оптимізації і подальшого вдосконалення.

Можна дійти висновку, що досягти прогресу для отримання і освоєння нового інформаційного матеріалу можна лише у процесі постійного удосконалення та оновлення набутих знань, вмінь та на-

вичок. Коли такого матеріалу накопичується значний обсяг, то без системного його впорядкування від простого до складного, класифікації за алгоритмічною складністю, функціональними можливостями та простотою подання і зберігання даних подальший розвиток неможливий.

## 6. Висновки

При побудові СШ доцільно застосовувати системний підхід, рухаючись від простого і очевидного до складного і малоімовірного на перший погляд. При цьому слід враховувати особливості як самої системи, так і програмних засобів, які розробляються або використовуються.

Видається доцільним, поряд з процедурними, застосування декларативних мов програмування, зокрема, логічного та функціонального, які дозволяють будувати логічну модель об'єкта досліджень та застосовувати ефективні механізми її аналізу й експлуатації.

Для ефективного подання даних і програм найбільш продуктивними є логіко-лінгвістичні моделі [2, 22, 32] (семантичні мережі; системи фреймів; логічні моделі; продукційні системи; функціональне подання [28]; реляційні бази і банки даних та знань).

Суттєвий прогрес в галузі створення СШ неможливий без побудови принципово нових комп'ютерів, які відходять від неймановської структури, а пов'язані з нейронною структурою [34], що наближена до природних потреб користувача.

**Література:** 1. Вендров А. М. CASE-технологии: современные методы и средства проектирования информационных систем / А. М. Вендров. М.: Финансы и статистика, 1998. 175 с. 2. Поспелов Г. С. Искусственный интеллект – основа новой информационной технологии / Г. С. Поспелов. К.: Наукова думка, 1995. 360 с. 3. Заяць В. М. Методи, алгоритми та програмні засоби для моделювання і аналізу динаміки складних об'єктів на основі дискретних моделей: Монографія / В. М. Заяць. Львів: Новий світ – 2000, 2009. 400 с. 4. Катренко А. В. Теорія прийняття рішень: Підручник / А. В. Катренко, В. В. Пасічник, В. П. Пасько. К.: Видавнича група ВНУ, 2009. 448 с. 5. Робертс Ф. С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. М.: Наука, 1986. 386 с. 6. Томашевський В. М. Моделювання систем: Підручник / В. М. Томашевський. К.: Видавнича група ВНУ, 2004. 352 с. 7. Клоксин У., Меллиш К. Программирование на языке Пролог. М.: Мир, 1987. 336 с. 8. Хоггер К. Введение в логическое программирование. М.: Мир. 1988. 348 с. 9. Братко И. Программирование на языке Пролог. 10. Макаллистер Дж. Искусственный интеллект и Пролог на микро-ЭВМ, М.: Машиностроение, 1990. 240 с. 11. Ин Ц., Соломон Д. Использование Тур-



бо-Пролога: Пер. с англ. М.: Мир, 1993. 608 с. **12.** Бакаев А.А., Гриценко В.И., Экспертные системы и логическое программирование. К.: Вища школа, 1995. 246 с. **13.** Заяць В.М. Логічне програмування: Ч. 1: Конспект лекцій з дисципліни «Логічне програмування» для студентів базового напрямку 6.08.04 «Програмне забезпечення автоматизованих систем». Львів: Видавництво Національного університету «Львівська політехніка», 2002. 48 с. **14.** Хендерсен П. Функциональное программирование. Применение и реализация. М.: Мир, 1983. 349 с. **15.** Бадаев Ю.И. Теория функционального программирования. Мови CommonLisp та AutoLisp. Київ, 1999. 150 с. **16.** Заяць В.М. Конспект лекцій з курсу функціональне програмування. Львів, 1999. 55 с. **17.** Маурер У. Введение в программирование на языке Лисп. М.: Мир, 1976. 104 с. **18.** Хювенен Э., Слептян Й. Мир Лиспа. Т.1. Введение в язык Лисп и функциональное программирование: Пер. с финск. М.: Мир, 1990. 447 с. **19.** Хювенен Э., Слептян Й. Мир Лиспа. Т.2 Методы и системы программирования. Пер. с финск. М.: Мир, 1990. 447 с. **20.** Еднерал В.Ф., Крюков А.П., Родионов А.Я. Язык аналитических вычислений REDUCE. М.: Из-во МГУ, 1984. 176 с. **21.** Лавров С.С., Силигадзе Г.С. Автоматическая обработка данных. Язык Лисп и его реализация. М.: Наука, 1978. 176 с. **22.** Уинстон П. Искусственный интеллект. М.: Мир, 1980. 513 с. **23.** Грин Д., Кнут Д. Математические методы анализа алгоритмов. М.: Мир, 1987. 120 с. **24.** Фостер Дж. Обработка списков.. М.: Мир, 1974. 280 с. **25.** Вирт Н. Алгоритмы + структуры данных = программы: Пер. с англ. М.: Мир, 1985. 213 с. **27.** Бердж В. Методы рекурсивного программирования. М.: Машиностроение, 1983. 248 с. **28.** Заяць В.М., Заяць М.М. Логічне і функціональне програмування. Навч. посібник. Рівне: 216. 400 с. **29.** Erich Gamma, Richard Helm, Ralih Johnson, John Vlissides. Wzorce projektowe, 2012. **30.** Заяць В.М., Маєвський Яцек. Системний підхід до освоєння та викладання мов логічного і функціонального програмування //21-й Міжнародний молодіжний форум «Радіоелектроніка і молодь в XXI столітті». Харків, ХНУРЕ., 2017. С. 4-6. **31.** Stanislaw Wrycha i inni. UML 2.1. Cwiczenia. ISBN: 978-83-246-0612-2012. **32.** Maccarthy J. Recursive functions of symbolic expressions and their computation by machine // Comm. ACM.: 1960. Vol.3. P. 184-195. **33.** Nilsson N.J. Principles of Artificial Intelligence. Tioga. Springer-Verlag, 1960. 164 с. **34.** Бондаренко М. Ф., Хаханов В. И., Гузь О. А., Шабанов-Кушнарченко Ю. П. Инфраструктура мозкоподобных обчислювальних процесів. Харків: ХНУРЕ, 2010. 169 с.

Надійшла до редколегії 03.05.2017

**Рецензент:** д-р фіз.-мат. наук, проф. Яворський І.М.

**Заяць Василь Михайлович**, д-р техн. наук, професор Технологічно-Природничого Університету (м. Бидгощ, Польща). Наукові інтереси: числові методи в прикладних застосуваннях, математичне і комп'ютерне моделювання динамічних систем, розпізнавання об'єктів зі складною динамічною природою, електрорадіотехніка та інформатика. Захоплення і хобі: туризм, велоспорт, театр; щоденні піші прогулянки 7 км. Адреса: Україна, 79070, Львів, просп. Червона Калина, 39, кв. 4. Тел.: дом. (+38) (032) 222-43-48; роб.: (+48) 52-340-53-30; моб.: (+38)(032) 0678362473.

**Заяць Марія Михайлівна**, ст. викладач НУ «Львівська політехніка» (м. Львів, Україна). Наукові інтереси: інформаційні технології, інженерія знань, математичне і комп'ютерне моделювання складних об'єктів, логічне програмування. Захоплення і хобі: туризм, театр, якісне приготування вишуканих страв. Адреса: Україна, 79070, Львів, просп. Червона Калина, 39, кв. 4. Тел.: дом. (+38) (032) 222-43-48; роб.: (+38) (032) 258-25-38 моб.: (+38)(032) 0989327526.

**Zaiatc Vasyl M.**, DSc., Professor, Professor University of Science and Technology (m. Bydgoszcz, Poland). Research interests include numerical methods in applied applications, mathematical and computer modeling of dynamic systems, recognition of objects of complex dynamic nature, electro - radiotehnika and informatics. Hobbies: hiking, biking, theater; daily walks 7 km. Address: Ukraine, 79070, Lviv, avenue Chervona Kalyna, 39, flat 4. Tel.: Home (38) (032) 222-43-48; Job: (+48) 52-340-53-30; Mob.: (+38) (032) 0678362473.

**Zaiats Mary M.**, senior teacher National University "Lviv polytechnic" (m. Lviv, Ukraine). Research interests include information technology, engineering knowledge, mathematical and computer modeling of complex objects, logic programming. Interests and hobbies: hiking, theater, high-quality gourmet food. Address: Ukraine, 79070, Lviv, avenue Chervona Kalyna, 39, flat 4. Tel.: Home. (38) (032) 222-43-48; Job.: (+38) (032) 258-25-38 Mob.: (+38) (032) 0989327526.