



# МОВИ І ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки 124 Системний аналіз</i>
Освітня програма	<i>Системи і методи штучного інтелекту Системний аналіз і управління</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>очна(денна)/дистанційна/змішана</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити ЕКТС</i>
Семестровий контроль/ контрольні заходи	<i>залік</i>
Розклад занять	<i><a href="https://schedule.kpi.ua/">https://schedule.kpi.ua/</a></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор:завідувач кафедри ММСА, к.т.н., доцент,Тимощук О.Л., <a href="mailto:tymoshchuk.oksana@ill.kpi.ua">tymoshchuk.oksana@ill.kpi.ua</a> Практичні/Лабораторні/ комп'ютерний практикум: к.т.н., доцент,Тимощук О.Л., <a href="mailto:tymoshchuk.oksana@ill.kpi.ua">tymoshchuk.oksana@ill.kpi.ua</a></i>
Розміщення курсу	<i>Платформа дистанційного навчання «Сікорський», Google classroom</i>

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Основні завдання кредитного модуля полягають в опануванні теоретичного та практичного базису щодо мов та технологій штучного інтелекту. **Метою викладання дисципліни** є формування знань, вмінь і навичок, необхідних для осмисленого вибору з подальшим використанням засобів функціонального та логічного програмування. **Основні задачі дисципліни** — ознайомити студентів з теоретичним та технологічним базисом проектування, визначенням та реалізацією як мов програмування, так і засобів завдання та дослідження поведінки програм, реалізованих цими мовами на прикладі мов LISP та Prolog.

**Згідно з вимогами освітньо-професійної програми після засвоєння дисципліни студенти мають продемонструвати такі результати навчання:**

**знання:** концепції та методи математичної логіки; теорію формальних мов та граматики; теорію математичних основ лямбда-числення; функціональний підхід до програмування; зв'язок формальної логіки та логічного програмування; синтаксис та семантику мов програмування Prolog та LISP; загальні правила роботи із сучасним стандартним програмним забезпеченням,

**уміння:** вибрати та обґрунтувати вибір підходу до програмування: функціонального або логічного; описати логічну модель заданої предметної області засобами мов програмування, що вивчаються; використовувати логічну та функціональну парадигму для побудови моделей неформалізованих задач; спроектувати систему управління базою даних; коректно інтерпретувати одержані результати,

**досвід:** спроектувати структуру програми в залежності від обраної парадигми; застосовувати існуючі та сучасні мови функціонального та логічного програмування у відповідних напрямках штучного інтелекту. Вивчення дисципліни дає можливість одержати фундаментальні знання стосовно сутності, об'єктивних закономірностей, принципів, методів та технологій щодо штучного інтелекту. Результатом вивчення дисципліни є формування у майбутніх фахівців належних компетентностей щодо мов та технологій штучного інтелекту з можливістю подальшого ефективного використання у проектній діяльності в ІТ індустрії.

### **Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

В структурно-логічній схемі навчання дисципліна передбачає попередню підготовку з курсу «Дискретний аналіз» (розділи: Алгебра висловлювань, Алгебра предикатів, Числення висловлювань, Числення предикатів, Формальні мови та граматики, Елементи теорії автоматичного доведення теорем), «Теорія програмування» (розділи: Класифікація мов та підходів до програмування, Формальні мови та граматики, Функціональна парадигма програмування, Логічна парадигма програмування), «Побудова та аналіз алгоритмів» (розділи: Рекурсивні функції та алгоритми, Алгоритми штучного інтелекту, Динамічні структури даних), «Об'єктно-орієнтоване програмування» (розділи: Інкапсуляція, Об'єкти і класи, Успадкування, Індивідуальність). Дисципліна забезпечує наступні дисципліни навчального плану: «Методи та системи штучного інтелекту», «Бази даних», «Експертні системи» (як базова мова при написанні робіт з комп'ютерного практикуму), а також всі спеціальні дисципліни, що потребують комп'ютерного моделювання.

## **2. Зміст навчальної дисципліни**

### **РОЗДІЛ 1. Мови створення систем штучного інтелекту.**

**Тема 1.1. Вступ до логічного програмування.** Вступ: предмет дисципліни, її обсяг, зміст, роль та зв'язок з іншими дисциплінами навчального плану. Огляд літератури по курсу. Розвиток мов штучного інтелекту: ретроспектива та перспектива. Сучасні парадигми програмування для рішення задач штучного інтелекту.

**Тема 1.2. Зв'язок формальної логіки та логічного програмування.** Застосування елементів алгебри логіки в логічному програмуванні. Фрази Хорна. Метод резолюцій.

### **РОЗДІЛ 2. Логічна парадигма програмування. Синтаксис та семантика *Prolog* -програми.**

**Тема 2.1. Основні поняття мови *Prolog*.** Визначення основних понять: елементи мови, алфавіт мови, фрази, терми, факти, атоми, змінні, константи. Запити до фактів: з константами, змінними; прості, складені. Квантифікація змінної в запиті. Синтаксична конструкція: правило. Квантифікація змінної в правилі. Загальна схема доведення цільового твердження.

**Тема 2.2. Рекурсивні процедури.** Визначення поняття рекурсії. Класичні приклади рекурсії. Лівобічна та правобічна рекурсії.

**Тема 2.3. Декларативна та процедурна семантика мови *Prolog*.** Поняття семантичної моделі. Семантична модель мови *Prolog* в вигляді абстрактної машини. Декларативна та процедурна семантичні моделі мови *Prolog*.

### **РОЗДІЛ 3. Структура програми мовою *Prolog*.**

**Тема 3.1. Розділи програми.** Синтаксис та призначення розділів програми. Опис доменів і предикатів. Використання зовнішніх та внутрішніх цілей. Діаграма успішного доведення цільового твердження.

**Тема 3.2. Стандартні предикати: арифметичні та управляючі.** Предикати та твердження різних арностей. Арифметика мови *Prolog*: інфіксні, префіксні та постфіксні операції; адитивні, мультиплікативні, порівняння, відношення; інтерпретація предиката **is**. Предикати управління: коректне припинення виконання: **exit**; аналіз змінної на конкретизацію: **bound, free**; заперечення: предикат **not**; управління бектрекінгом: предикати **fail, cut**.

### **РОЗДІЛ 4. Застосування файлів на зовнішніх носіях.**

**Тема 4.1. Позалогічні предикати мови *Prolog*.** Опис файлового домену. Предикати для роботи з файлами. Послідовності дій під час запису в файл; читанні з файлу; модифікації існуючого файлу; допису в кінець існуючого файлу.

**Тема 4.2. Файли прямого доступу.** Засоби проектування програм. Програми обробки файлів. Послідовність дій щодо роботи з файлами прямого доступу.

### **РОЗДІЛ 5. Механізми створення повторів та рекурсій.**

**Тема 5.1. Основні методи створення повторів.** Предикат **fail**. Метод відкоту після неуспіху. Предикат **cut**. Типи відсікання. Метод відсікання та відкоту. Метод повтору, визначений користувачем. Засоби управління перебором.

**Тема 5.2. Методи організації рекурсії.** Проста рекурсія. Складна рекурсія. Використання рекурсії для генерації ряду чисел. Проектування рекурсивного правила для знаходження суми ряду (користуючись **cut** та без нього). Засоби створення процедури обчислення факторіалу.

### **РОЗДІЛ 6. Абстрактні типи даних мови *Prolog*.**

**Тема 6.1. Списки, черги, дерева, довідники. Операції над списками.** Структури даних мови *Prolog*. Поняття "список". Атрибути списку. Графічне представлення списків. Метод розділу списку на хвіст і голову. Операції над списками: пошук елемента в списку, конкатенація списків, поділ списків на задану кількість списків, додавання елементів до списку, сортування списку, представлення множин двійковими деревами, двійкові довідники: додавання та вилучання елементів.

**Тема 6.2. Типи даних: рядки. Вбудовані предикати роботи з рядками.** Поняття "рядки". Доменний тип **string**. Використання ASCII- кодів для запису символів. Вбудовані предикати роботи з рядками. Об'єднання рядків. Роз'єднання рядків. Пошук символу або підрядка.

### **РОЗДІЛ 7. Метаінтерпретатори, експертні системи в мові *Prolog*.**

**Тема 7.1. Метаінтерпретатори в мові *Prolog*.** Введення до мета інтерпретаторів. Оболонка до експертної системи на підставі правил. Семантичні мережі, фрейми та схеми в мові *Prolog*.

**Тема 7.2. Експертні системи в мові *Prolog*.** Структура експертних систем та їх проектування. Експертні системи, що базуються на правилах та рішення задач на основі цілі. Евристика та управління в експертній системі. Судження на підставі моделей: приклад NASA. Гібридні системи: переваги та недоліки систем з потужними методами.

## **РОЗДІЛ 8. Динамічні бази даних (БД), контекстне-вільний (КВ) аналіз засобами мови *Prolog*.**

**Тема 8.1. Системи управління базами даних (СУБД).** Статичні бази даних: складні області, однорівневі, дворівневі складені домени. Конструкції альтернативних доменів. Отримання структурованої інформації з статичної БД. Реляційні БД. Предикати роботи з БД. Проектування СУБД.

**Тема 8.2. Проектування контекстне-вільного (КВ) та лексичного аналізаторів.** Представлення знань. Методи виведення. Синтаксичний аналіз: моделі речень для КВ граматик; ієрархії Хомського та контекстне-залежні граматики. Лексичний аналізатор речень. Доведення теорем засобами мови *Prolog*.

## **РОЗДІЛ 9. Алгоритми навчання в мові *Prolog*.**

**Тема 9.1. Машинне навчання, що визначається символьним представленням інформації.** Пошук у просторі версій. Індуктивний алгоритм побудови дерева рішень. Теорія студіювання. Знання та навчання. Навчання без вчителя. Навчання з підкріпленням. Навчання перцептрона.

**Тема 9.2. Нейромережеве машинне навчання.** Навчання перцептрона. Навчання за методом зворотнього поширення. Конкурентне навчання. Синхронне навчання Хебба. Мережі асоціативної пам'яті.

**Тема 9.3. Машинне навчання на основі соціальних та емерджентних принципів.** Генетичні алгоритми. Системи класифікації та генетичне програмування. Штучне життя та емерджентне навчання.

## **РОЗДІЛ 10. Обробка природної мови засобами *Prolog*.**

**Тема 10.1. Синтаксичний та семантичний аналіз.** Специфікації та синтаксичний аналіз з використанням контекстне-вільних граматик. Об'єднання знань про синтаксис та семантику. Сумісне використання синтаксичних та семантичних конструкцій в системі визначень DCG.

**Тема 10.2. Стохастичний підхід до аналізу мови.** Ймовірнісні мовні моделі. Підхід на основі дерева рішень. Граматичний аналіз та інші додатки стохастичного підходу.

## **РОЗДІЛ 11. Функціональна парадигма програмування. Мови функціонального програмування.**

**Тема 11.1. Функція як природній формалізм мов функціонального програмування.** Понятійний апарат. Чисті функції, функціональність. Математичні основи: лямбда-числення Черча. Редукція графів. Комбінаторна редукція.

**Тема 11.2. Приклади мов функціонального програмування.** Стилі мов функціонального програмування. Конкретні реалізації: *SML, Hope, Miranda, Haskell, FP*, діалекти мови *LISP*.

## **РОЗДІЛ 12. Основи функціонального програмування.**

**Тема 12.1. Типи рекурсій.** Рекурсивні структури даних. Рекурсії нульового, першого та вищих порядків. Рекурсія за значенням. Рекурсія за аргументами. Паралельна рекурсія. Взаємна рекурсія. Функціонали.

**Тема 12.2. Застосовуючи та відображаючи функціонали.** Метод накопичування параметрів. Застосовуючи та відображаючи функціонали.

## **РОЗДІЛ 13. *LISP* та імперативне програмування.**

**Тема 13.1. Поняття замикання, функціональної абстракції, автофункції.** Замикання. Функціональна абстракція. Автофункції. Відповідність між функціональними та імперативними програмами.

## **РОЗДІЛ 14. Елементи процедурного програмування в *LISP*.**

**Тема 14.1. Зв'язок між процедурними та функціональними мовами програмування.** PROG-вирази у мові *LISP*. Побудова списків. Предикати, що написані за допомогою PROG-виразів. Інші конструкції процедурного програмування.

#### **РОЗДІЛ 15. Стратегії пошуку в мові *LISP*.**

**Тема 15.1. Стратегії неінформованого пошуку. Стратегії інформованого (евристичного) пошуку.** Застосування засобів абстрагування для реалізації загальних алгоритмів пошуку в ширину, в глибину, «жадібний» алгоритм. Управління пошуком у просторі станів засобами мови *LISP*.

#### **РОЗДІЛ 16. Об'єктно-орієнтоване програмування засобами Common *LISP Object System*.**

**Тема 16.1. Визначення класів, екземплярів, родових функцій та методів в CLOS. Реалізація успадкування в CLOS.** Підтримка системою CLOS концепції об'єктно-орієнтованого програмування: інкапсуляції, поліморфізму, успадкування.

**Тема 16.2. Приклад моделювання діагностичної системи виробничого процесу.** Моделювання термостату. Створення моделі виробничого процесу в термінах взаємодії компонентів. Опис взаємодії компонентів в термінах передачі повідомлень між об'єктами.

### **3. Навчальні матеріали та ресурси**

#### **Базова**

1. Логічне і функціональне програмування : навч. посіб. [для студентів ВНЗ України] / В. М. Заяць, М. М. Заяць ; Нац. ун-т "Львів. політехніка". — Кам'янець-Подільський (Хмельниц. обл.) ; Львів : Гордукова І. Є., 2016. — 398 с. : іл., табл., портр.
2. Зубенко В. В. Програмування: навчальний посібник (гриф МОН України) / В. В. Зубенко, Л. Л. Омельчук. — К. : ВПЦ «Київський університет», 2011. — 623 с.
3. Нікітченко М. С. Теоретичні основи програмування: навчальний посібник / М.С Нікітченко — Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010. — 121с.
4. Bratko, Ivan. Prolog programming for artificial intelligence. Addison-Wesley Publishing Company, Inc.1986 chrome-extension://efaidnbmnnnibpcajpcgiclfefindmkaj/ <https://silp.iiita.ac.in/wp-content/uploads/PROLOG.pdf>
5. Bratko, I. (1982) Knowledge-based problem-solving in AL3. In: Machine Intelligence 10 (J. E. Hayes, D. Michie, Y. H. Pao, eds.).
6. Математичні основи логічного програмування: навчальний посібник / В.І.Месюра, Н.В.Лисак, О.І.Суприган. – Вінниця: ВНТУ, 2013. – 94 с.
7. Pratt T.W., Zelkovitz M.V. Programming languages, design and implementation (4th ed.). Prentice Hall, 2000
8. Функційне програмування : Навч. посіб. для студ. вищих навч. закл., що навч. за спец. "Програм. забезп. автоматиз. систем" / В. М. Заяць; Нац. ун-т "Львів. політехніка". - Л. : Бескид Біт, 2003. - 159 с.
9. Gabbrielli, Maurizio (2010). *Programming languages principles and paradigms*. London, New York: Springer,. ISBN 9781848829145.
10. Robert W. Sebesta: *Concepts of Programming Languages*, 9th ed., Addison Wesley 2009.
11. Мови та технології штучного інтелекту. Частина 1. "Алгоритми штучного інтелекту на мові PROLOG": Методичні рекомендації до виконання робіт з комп'ютерного практикуму / Укладач: О.Л.Тимошук — К.:НН ІПСА КПІ ім. Ігоря Сікорського, 2022. – 25с.

*Методичні матеріали викладені в Електронному Кампусі та на Платформі дистанційного навчання «Сікорський», Google Classroom. Обов'язковими для прочитання є Слайди лекцій,*

## Додаткова література

1. Ellis Horwood. Buchanan, B. G. and Shortliffe, E. H. (1984, eds.) Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.
2. Addison-Wesley. Duda, R., Gaschnig, J. and Hart, P. (1979) Model design in the Prospector consultant system for mineral exploration. In: Expert Systems in the Microelectronic Age (D. Michie, ed.). Edinburgh University Press.
3. Hammond, P. (1984) Micro-PROLOG for Expert Systems. In: MigroPROLOG: Programming in Logic (K. L. Clark, F. G. McCabe, eds.). Prentice-Hall.
4. Michie, D. (1979, ed.) Expert Systems in the Microelectronic Age. Edinburgh University Press. 357 о о358PROLOG PROGRAMMING FOR ARTIFICIAL INTELLIGENCE
5. Quinlan, J. R. (1983) Inferno: a cautious approach to uncertain reasoning. Zhe Computer Journal 26: 255-270.
6. Reiter, J. (1980) ALIX: An Expert System Using Plausible Inference. Oxford: Intelligent Terminals Ltd.
7. Shortliffe, E. (1976) Computer-based Medical Consultations: MYCIN.
8. Elsevier. Weiss, S. M. and Kulikowski, C. A. (1984) A Practical Guide to Designing Expert Systems.
9. Chapman and Hall. Winston, P. H. (1984) Artificial Intelligence (second edition). AddisonWeslev

## Навчальний контент

### 4. Методика опанування навчальної дисципліни (освітнього компонента)

Викладання дисципліни проводиться згідно існуючої методики організації навчального процесу у вищих учбових закладах з використанням практичних занять, комп'ютерного практикуму, а також самостійної роботи студентів. Методика вивчення дисципліни «Мови і технології штучного інтелекту» передбачає синтетичне поєднання теоретичних, практичних знань та знань набутих під час самостійної роботи. Методика викладання дисципліни передбачає як надання теорії логічного та функціонального програмування, так і наведення значної кількості належних прикладів. Практичний матеріал, що наданий на лекції, є основою при написанні робіт з комп'ютерного практикуму. Обов'язковою умовою є опрацювання сучасної літератури. Теоретичні та практичні знання набуті на лекції повинні бути наочно закріплені на заняттях з комп'ютерного практикуму (КП) у формах, зазначених у рейтинговій системі. В процесі викладання дисципліни для активізації навчального процесу рекомендується використовувати проблемні лекції, практичні заняття-дискусії, тренінги, ролові ігри, відкриті обговорення, а також кейс-методи і презентації.

## Структура освітнього компонента

Роз-діл	Тема	Найменування розділу, теми	Розподіл навчального часу			
			всього	лекції	комп. практикум	СРС

Роз-діл	Тема	Найменування розділу, теми	Розподіл навчального часу			
			всього	лекції	комп. практикум	СРС
1.	Мови створення систем штучного інтелекту.		<b>4</b>	<b>2</b>		<b>2</b>
	1.1.	Вступ до логічного програмування.	1	1		
	1.2.	Зв'язок формальної логіки та логічного програмування.	3	1		2
2.	Логічна парадигма програмування. Синтаксис та семантика Пролог-програм.		<b>7</b>	<b>2</b>	<b>2</b>	<b>3</b>
	2.1.	Основні поняття мови <b>Prolog</b> .	0,5	0,5		
	2.2.	Рекурсивні процедури.	4	1	1	2
	2.3.	Декларативна та процедурна семантика мови <b>Prolog</b> .	2,5	0,5	1	1
3.	Структура програми мовою <b>Prolog</b> .		<b>3</b>	<b>2</b>		<b>1</b>
	3.1.	Розділи програми.	0,5	0,5		
	3.2.	Стандартні предикати: арифметичні та управляючі предикати.	2,5	1,5		1
4.	Застосування файлів на зовнішніх носіях.		<b>6</b>	<b>2</b>	<b>2</b>	<b>2</b>
	4.1.	Позалогічні предикати мови <b>Prolog</b>	5	1,5	2	1,5
	4.2.	Файли прямого доступу.	1	0,5		0,5
5.	Механізми створення повторів та рекурсій.		<b>10</b>	<b>4</b>	<b>2</b>	<b>4</b>
	5.1.	Основні методи створення повторів.	5	2	1	2
	5.2.	Методи організації рекурсії.	5	2	1	2
6.	Абстрактні типи даних мови <b>Prolog</b> .		<b>14</b>	<b>6</b>	<b>4</b>	<b>4</b>
	6.1.	Списки, черги, дерева, довідники. Операції над списками	11	4	4	3
	6.2.	Типи даних: рядки. Вбудовані предикати роботи з рядками	3	2		1
7.	Метаінтерпретатори, експертні системи в мові <b>Prolog</b> .		<b>4</b>	<b>2</b>		<b>2</b>
	7.1.	Метаінтерпретатори в мові <b>Prolog</b> .	2	1		1
	7.2.	Експертні системи в мові <b>Prolog</b> .	2	1		1
8.	Динамічні бази даних (БД), експертні системи (ЕС), контекстно-вільний (КВ) аналіз засобами мови <b>Prolog</b> .		<b>10</b>	<b>2</b>	<b>4</b>	<b>4</b>

Розділ	Тема	Найменування розділу, теми	Розподіл навчального часу			
			всього	лекції	комп. практикум	СРС
	8.1.	Системи управління базами даних (СУБД).	7,5	1,5	4	2
	8.2.	Проектування контекстно-вільного (КВ) та лексичного аналізаторів.	2,5	0,5		2
9.	<b>Алгоритми навчання в мові <i>Prolog</i>.</b>		<b>7</b>	<b>1</b>		<b>6</b>
	9.1.	Машинне навчання, що визначається символьним представленням інформації.	3	1		2
	9.2.	Нейромережеве машинне навчання.	2			2
	9.3.	Машинне навчання на основі соціальних та емерджентних принципів.	2			2
10.	<b>Обробка природної мови засобами <i>Prolog</i>.</b>		<b>5</b>	<b>1</b>		<b>4</b>
	10.1	Синтаксичний та семантичний аналіз.	3	1		2
	10.2	Стохастичний підхід до аналізу мови.	2			2
	<b>Контрольна робота 1</b>		<b>7</b>	<b>1</b>		<b>6</b>
11.	Функціональна парадигма програмування. Мови функціонального програмування.		<b>5</b>	<b>2</b>		<b>3</b>
	11.1	Функція як природний формалізм мов функціонального програмування.	3	1		2
	11.2	Приклади мов функціонального програмування.	2	1		1
12.	Основи функціонального програмування.		<b>7</b>	<b>2</b>	<b>2</b>	<b>3</b>
	12.1	Типи рекурсій.	4,5	1	2	1,5
	12.2	Застосовуючі та відображаючі функціонали	2,5	1		1,5
13.	<b><i>Lisp</i> та імперативне програмування.</b>		<b>4</b>	<b>1</b>		<b>3</b>
	13.1	Поняття замикання, функціональної абстракції, автофункції	4	1		3
14.	<b>Елементи процедурного програмування в <i>Lisp</i>.</b>		<b>6</b>	<b>1</b>	<b>2</b>	<b>3</b>
	14.1	Зв'язок між процедурними та функціональними мовами програмування.	6	1	2	3

Роз-діл	Тема	Найменування розділу, теми	Розподіл навчального часу			
			всього	лекції	комп. практикум	СРС
	<b>Контрольна робота 2</b>		<b>7</b>	<b>1</b>		<b>6</b>
15.	Стратегії пошуку в мові <i>LISP</i> .		<b>3</b>	<b>1</b>		<b>2</b>
	15.1	Стратегії неінформованого пошуку. Стратегії інформованого (евристичного) пошуку.	3	1		2
16.	Об'єктно-орієнтоване програмування засобами Common <i>LISP</i> Object System.		<b>3</b>	<b>1</b>		<b>2</b>
	16.1	Визначення класів, екземплярів, родових функцій та методів в CLOS. Реалізація успадкування в CLOS.	1,5	0,5		1
	16.2	Приклад моделювання діагностичної системи виробничого процесу.	1,5	0,5		1
	<b>Підготовка до заліку</b>		<b>8</b>	<b>2</b>		<b>6</b>
Усього в Y семестрі			<b>120</b>	<b>36</b>	<b>18</b>	<b>66</b>

### Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
1	<p><u>Лекція 1.</u>  <b>РОЗДІЛ 1. Мови створення систем штучного інтелекту.</b>  <b>Тема 1.1. Вступ до логічного програмування.</b>  <b>Тема 1.2. Зв'язок формальної логіки та логічного програмування.</b>  Вступ: предмет дисципліни, її обсяг, зміст, роль та зв'язок з іншими дисциплінами навчального плану. Огляд літератури по курсу. Розвиток мов штучного інтелекту: ретроспектива та перспектива. Сучасні парадигми програмування для рішення задач штучного інтелекту. Застосування елементів алгебри логіки в логічному програмуванні. Фрази Хорна. Метод резолюцій.  <b>Питання на СРС:</b> Засвоєння пакетів систем Visual Prolog, SWI Prolog та Turbo-Prolog. Показати дію методу резолюції на заданій теорії.</p>
2	<p><u>Лекція 2.</u>  <b>РОЗДІЛ 2. Логічна парадигма програмування. Синтаксис та семантика Prolog - програм.</b>  <b>Тема 2.1. Основні поняття мови Prolog.</b>  <b>Тема 2.2. Рекурсивні процедури.</b>  <b>Тема 2.3. Декларативна та процедурна семантика мови Prolog.</b>  Визначення основних понять: елементи мови, алфавіт мови, фрази, терми, факти, атоми, змінні, константи. Запити до фактів: з константами, змінними; прості, складені. Квантифікація змінної в запиті. Синтаксична конструкція: правило. Квантифікація змінної в правилі. Загальна схема доведення цільового твердження. Визначення</p>

	<p>поняття рекурсії. Класичні приклади рекурсії. Лівобічна та правобічна рекурсії. Поняття семантичної моделі. Семантична модель мови <i>Prolog</i> в вигляді абстрактної машини. Декларативна та процедурна семантичні моделі мови <i>Prolog</i>.</p> <p><b>Питання на СРС:</b> Засвоїти на практиці різницю між внутрішніми та зовнішніми цілями. Поекспериментувати з простими та складними запитами до фактів і правил. Зробити висновки щодо використання лівобічної рекурсії.</p>
3	<p><u>Лекція 3.</u></p> <p><b>РОЗДІЛ 3. Структура програми мовою <i>Prolog</i>.</b></p> <p><b>Тема 3.1. Розділи програми.</b></p> <p><b>Тема 3.2. Стандартні предикати: арифметичні та управляючі.</b></p> <p>Синтаксис та призначення розділів програми. Опис доменів і предикатів. Використання зовнішніх та внутрішніх цілей. Діаграма успішного доведення цільового твердження.</p> <p>Предикати та твердження різних арностей. Арифметика мови <i>Prolog</i>: інфіксні, префіксні та постфіксні операції; адитивні, мультиплікативні, порівняння, відношення; інтерпретація предиката <b>is</b>. Предикати управління: коректне припинення виконання: <b>exit</b>; аналіз змінної на конкретизацію: <b>bound</b>, <b>free</b>; заперечення: предикат <b>not</b>; управління бектрекінгом: предикати <b>fail</b>, <b>cut</b>.</p> <p><b>Питання на СРС:</b> Скласти приклади використання предиката <b>not</b>.</p>
4	<p><u>Лекція 4.</u></p> <p><b>РОЗДІЛ 4. Застосування файлів на зовнішніх носіях.</b></p> <p><b>Тема 4.1. Позалогічні предикати мови <i>Prolog</i>.</b></p> <p><b>Тема 4.2. Файли прямого доступу.</b></p> <p>Опис файлового домену. Предикати для роботи з файлами. Послідовності дій під час запису в файл; читанні з файлу; модифікації існуючого файлу; дозапису в кінець існуючого файлу. Засоби проектування програм. Програми обробки файлів. Послідовність дій щодо роботи з файлами прямого доступу.</p>
5	<p><u>Лекція 5.</u></p> <p><b>РОЗДІЛ 5. Механізми створення повторів та рекурсії.</b></p> <p><b>Тема 5.1. Основні методи створення повторів.</b></p> <p>Предикат <b>fail</b>. Метод відкоту після неуспіху. Предикат <b>cut</b>. Типи відсікання. Метод відсікання та відкоту. Метод повтору, визначений користувачем. Засоби управління перебором.</p> <p><b>Питання на СРС:</b> Написати правила, які використовують відсікання, для: 1) обчислення максимуму; 2) процедури перевірки належності списку, яка дає одне рішення; 3) додавання елемента до списку без дублювання; 4) класифікації об'єктів.</p>
6	<p><u>Лекція 6.</u></p> <p><b>Тема 5.2. Методи організації рекурсії.</b></p> <p>Проста рекурсія. Складна рекурсія. Використання рекурсії для генерації ряду чисел. Проектування рекурсивного правила для знаходження суми ряду (користуючись <b>cut</b> та без нього). Засоби створення процедури обчислення факторіалу.</p> <p><b>Питання на СРС:</b> Розглянути "червоне" та "зелене" відсікання.</p>
7	<p><u>Лекція 7.</u></p> <p><b>РОЗДІЛ 6. Абстрактні типи даних мови <i>Prolog</i>.</b></p> <p><b>Тема 6.1. Списки, черги, дерева, довідники. Операції над списками.</b></p>

	<p>Структури даних мови <i>Prolog</i>. Поняття “список”. Атрибути списку. Графічне представлення списків. Метод розділу списку на хвіст і голову. Операції над списками: пошук елемента в списку, конкатенація списків, поділ списків на задану кількість списків, додавання елементів до списку, сортування списку.</p> <p><b>Питання на СРС:</b> За допомогою операції конкатенації виконати: - конкатенацію двох та більше списків; - роз’єднання списків на два та більше.</p>
8	<p><u>Лекція 8.</u></p> <p>Представлення множин двійковими деревами, двійкові довідники: додавання та вилучання елементів.</p> <p><b>Питання на СРС:</b> Організація пошуку у двійкових довідниках та деревах. Методи представлення множин деревами.</p>
9	<p><u>Лекція 9.</u></p> <p><b>Тема 6.2. Типи даних: рядки. Вбудовані предикати роботи з рядками.</b></p> <p>Поняття “рядки”. Доменний тип string. Використання ASCII- кодів для запису символів. Вбудовані предикати роботи з рядками. Об’єднання рядків. Роз’єднання рядків. Пошук символу або підрядка.</p>
10	<p><u>Лекція 10.</u></p> <p><b>РОЗДІЛ 7. Метаінтерпретатори, експертні системи в мові <i>Prolog</i>.</b></p> <p><b>Тема 7.1. Метаінтерпретатори в мові <i>Prolog</i>.</b></p> <p><b>Тема 7.2. Експертні системи в мові <i>Prolog</i>.</b></p> <p>Введення до метаінтерпретаторів. Оболонка до експертної системи на підставі правил. Семантичні мережі, фрейми та схеми в мові <i>Prolog</i>. Структура експертних систем та їх проектування. Експертні системи, що базуються на правилах та рішення задач на основі цілі. Евристика та управління в експертній системі. Судження на підставі моделей: приклад NASA.</p> <p><b>Питання на СРС:</b> Гібридні системи: переваги та недоліки систем з потужними методами.</p>
11	<p><u>Лекція 11.</u></p> <p><b>РОЗДІЛ 8. Динамічні бази даних (БД), контекстно-вільний (КВ) аналіз засобами мови <i>Prolog</i>.</b></p> <p><b>Тема 8.1. Системи управління базами даних (СУБД).</b></p> <p><b>Тема 8.2. Проектування контекстно-вільного (КВ) та лексичного аналізаторів.</b></p> <p>Статичні бази даних: складні області, однорівневі, дворівневі складені домени. Конструкції альтернативних доменів. Отримання структурованої інформації з статичної БД. Реляційні БД. Предикати роботи з БД. Проектування СУБД. Представлення знань. Методи виведення. Синтаксичний аналіз: моделі речень для КВ граматики; ієрархії Хомського та контекстно-залежні граматики. Лексичний аналізатор речень. Доведення теорем засобами мови <i>Prolog</i>.</p> <p><b>Питання на СРС:</b> Доведення теорем засобами Прологу.</p>
12	<p><u>Лекція 12.</u></p> <p><b>РОЗДІЛ 9. Алгоритми навчання в мові <i>Prolog</i>.</b></p> <p><b>Тема 9.1. Машинне навчання, що визначається символьним представленням інформації.</b></p> <p><b>Тема 9.2. Нейромережеве машинне навчання.</b></p> <p><b>Тема 9.3. Машинне навчання на основі соціальних та емерджентних принципів.</b></p>

	<p>Пошук у просторі версій. Індуктивний алгоритм побудови дерева рішень. Теорія студювання. Знання та навчання. Навчання без вчителя. Навчання з підкріпленням. Навчання перцептрона.</p> <p><b>Питання на СРС:</b> Навчання перцептрона. Навчання за методом зворотнього поширення. Конкурентне навчання. Синхронне навчання Хебба. Мережі асоціативної пам'яті. Генетичні алгоритми. Системи класифікації та генетичне програмування. Штучне життя та емерджентне навчання.</p> <p><b>РОЗДІЛ 10. Обробка природної мови засобами Prolog.</b></p> <p><b>Тема 10.1. Синтаксичний та семантичний аналіз.</b></p> <p><b>Тема 10.2. Стохастичний підхід до аналізу мови.</b></p> <p>Специфікації та синтаксичний аналіз з використанням контекстно-вільних граматики. Об'єднання знань про синтаксис та семантику. Сумісне використання синтаксичних та семантичних конструкцій в системі визначень DCG.</p> <p><b>Питання на СРС:</b> Ймовірнісні мовні моделі. Підхід на основі дерева рішень. Граматичний аналіз та інші додатки стохастичного підходу.</p>
13	<p><u>Лекція 13.</u></p> <p><b>Контрольна робота №1</b></p> <p><b>РОЗДІЛ 11. Функціональна парадигма програмування. Мови функціонального програмування.</b></p> <p><b>Тема 11.1. Функція як природній формалізм мов функціонального програмування.</b></p> <p>Понятійний апарат. Чисті функції, функціональність. Математичні основи: лямбда-числення А.Черча. Редукція графів. Комбінаторна редукція.</p>
14	<p><u>Лекція 14.</u></p> <p><b>Тема 11.2. Приклади мов функціонального програмування. Стил мов функціонального програмування.</b> Конкретні реалізації: <i>SML, Hope, Miranda, Haskell, FP</i>, діалекти мови <i>LISP</i>.</p> <p><b>РОЗДІЛ 12. Основи функціонального програмування.</b></p> <p><b>Тема 12.1. Типи рекурсії.</b> Рекурсивні структури даних. Рекурсії нульового, першого та вищих порядків. Рекурсія за значенням. Рекурсія за аргументами. Паралельна рекурсія. Взаємна рекурсія. Функціонали.</p>
15	<p><u>Лекція 15.</u></p> <p><b>Тема 12.2. Застосовуючи та відображаючи функціонали.</b> Метод накопичування параметрів. Застосовуючи та відображаючи функціонали.</p> <p><b>РОЗДІЛ 13. LISP та імперативне програмування.</b></p> <p><b>Тема 13.1. Поняття замикання, функціональної абстракції, автофункції.</b> Замикання. Функціональна абстракція. Автофункції. Відповідність між функціональними та імперативними програмами.</p> <p><b>Питання на СРС:</b> Комбінації функцій відбору.</p>
16	<p><u>Лекція 16.</u></p> <p><b>РОЗДІЛ 14. Елементи процедурного програмування в LISP.</b></p> <p><b>Тема 14.1. Зв'язок між процедурними та функціональними мовами програмування.</b></p> <p>PROG-вирази у мові <i>LISP</i>. Побудова списків. Предикати, що написані за допомогою PROG-виразів. Інші конструкції процедурного програмування.</p> <p><b>Питання на СРС:</b> Порівняльна характеристика програм на Ліспі, що написані засобами</p>

	функціонального та процедурного програмування. <b>Контрольна робота №2</b>
17	<u>Лекція 17.</u> <b>РОЗДІЛ 15. Стратегії пошуку в мові LISP.</b> <b>Тема 15.1. Стратегії неінформованого пошуку. Стратегії інформованого (евристичного) пошуку.</b> Застосування засобів абстрагування для реалізації загальних алгоритмів пошуку в ширину, в глибину, «жадібний» алгоритм. Управління пошуком у просторі станів засобами мови LISP.
18	<u>Лекція 18.</u> <b>РОЗДІЛ 16. Об'єктно-орієнтоване програмування засобами Common LISP Object System.</b> <b>Тема 16.1. Визначення класів, екземплярів, родових функцій та методів в CLOS.</b> Реалізація успадкування в CLOS. Підтримка системою CLOS концепції об'єктно-орієнтованого програмування: інкапсуляції, поліморфізму, успадкування. <b>Тема 16.2. Приклад моделювання діагностичної системи виробничого процесу.</b> Моделювання термостату. Створення моделі виробничого процесу в термінах взаємодії компонентів. Опис взаємодії компонентів в термінах передачі повідомлень між об'єктами.

### Роботи з комп'ютерного практикуму

№ КП	№ розділу, теми	Зміст лабораторної роботи	К-сть годин
1.	ОК, які належать до пререквізитів	Моделювання логічних задач імперативними мовами програмування з використанням ШІ. Частина 1.	2
2. КП1	Розділ 2, теми 2.1, 2.2.	Створення моделі відношень у заданій предметній області. Вступ до мови програмування Пролог.	2
3. КП2	Розділ 4, тема 4.1;	Позалогічні предикати мови <i>Prolog</i> . Механізми	2
	розділ 5, теми 5.1, 5.2.	реалізації повторів та фільтрації даних.	
4. КП3	Розділ 6, тема 6.1.	Списки, черги, дерева, довідники в	4
		Prolog	
5. КП4	Розділ 7, тема 7.1;	Списки, черги, дерева, довідники в	2
	розділ 8, тема 8.1.	Python	
6. КП5	Розділ 11, теми 11.2, 11.3. Розділ 12, тема 12.1; розділ 15, тема 15.1	Функціональне програмування. Програми з найпростішою структурою. Списки, черги, дерева, довідники. Програмування циклічних алгоритмів обраною мовою ФП засобами програмування:	4

№ КП	№ розділу, теми	Зміст лабораторної роботи	К-сть годин
1.	ОК, які належать до пререквізитів	Моделювання логічних задач імперативними мовами програмування з використанням ШІ. Частина 1.	2
		а) функціонального, б) процедурного.	
7.	Розділи 1-16	Моделювання логічних задач. Частина 2.	2

18 годин

### 5. Самостійна робота здобувача вищої освіти

*Види самостійної роботи включають підготовку до дебатів за заданою темою, завдань з комп'ютерного практикуму з зазначених тем та проведення тестування створених програмних продуктів (24 години), підготовку до модульної контрольної роботи (підсумкової) (12 годин), а також опрацювання тем, які не увійшли до лекцій (24 години).*

№ з/п	Назва теми, що виноситься на самостійне опрацювання	К-ть годин СРС
1	Засвоєння пакетів систем Visual Prolog, SWI Prolog та Turbo-Prolog. Показати дію методу резолюції на заданій теорії.	2
2	Засвоїти на практиці різницю між внутрішніми та зовнішніми цілями. Проекспериментувати з простими та складними запитами до фактів і правил. Зробити висновки щодо використання лівобічної рекурсії.	2
3	Скласти приклади використання предиката not.	1
4	Написати правила, які використовують відсікання, для: 1) обчислення максимуму; 2) процедури перевірки належності списку, яка дає одне рішення; 3) додавання елемента до списку без дублювання; 4) класифікації об'єктів.	2
5	Розглянути "червоне" та "зелене" відсікання.	2
6	За допомогою операції конкатенації виконати: - конкатенацію двох та більше списків; - роз'єднання списків на два та більше.	2
7	Організація пошуку у двійкових довідниках та деревах. Методи представлення множин деревами.	2
8	Гібридні системи: переваги та недоліки систем з потужними методами.	2
9	Доведення теорем засобами Прологу.	2
10	Навчання перцептрона. Навчання за методом зворотнього поширення. Конкурентне навчання. Синхронне навчання Хебба. Мережі асоціативної пам'яті. Генетичні алгоритми. Системи класифікації та генетичне програмування. Штучне життя та емерджентне навчання.	2
11	Ймовірнісні мовні моделі. Підхід на основі дерева рішень. Граматичний аналіз та інші додатки стохастичного підходу.	2
12	Комбінації функцій відбору.	1
13	Порівняльна характеристика програм на Ліспі, що написані засобами функціонального та процедурного програмування.	2

		24

## Політика та контроль

### 6. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які ставляться перед студентом:

- щодо політики академічної доброчесності: при захисті робіт з комп'ютерного практикуму забороняється використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті;
- щодо політики академічної доброчесності: при написанні модульної контрольної роботи забороняється використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті;
- на заняттях бажано відключати телефони;
- за несвоєчасну подачу звіту з роботи комп'ютерного практикуму штрафні бали не нараховуються;
- діє система заохочувальних балів.

### 7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

**Поточний контроль:** експрес-опитування, опитування за темою заняття, МКР, дебати тощо

**Календарний контроль:** провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

**Семестровий контроль:** залік

**Умови допуску до семестрового контролю:** мінімально позитивна оцінка за індивідуальне завдання / зарахування усіх лабораторних робіт / семестровий рейтинг більше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

### Рейтингова система оцінювання освітнього компоненту

Семестровий рейтинг студента з освітнього компонента (ОК) складається з вагових балів за кожний вид контролю, що наведено в табл. 1

#### СИСТЕМА РЕЙТИНГОВИХ БАЛІВ

Таблиця 1

Вид контролю	Бал
1. Роботи з комп'ютерного практикуму (РКП) 5 робіт	40
2. Моделювання логічних задач	40
2.1. Частина 1 – з використанням штучного інтелекту (20 балів)	
2.2. Частина 2 – з використанням декларативних мов програмування (20 балів)	
4. Виконання завдань з переліку СРС	10

## КРИТЕРІЇ ОЦІНЮВАННЯ

### 1. Робота з комп'ютерного практикуму

Виконання та захист РКП складається з двох частин, кожна з яких оцінюється окремо.

Загальна кількість балів за кожну РКП становить 0÷10 балів. Роботи з кожного з п'яти комп'ютерних практикумів оцінюються наступним чином:

КП 1 — 5 балів

КП 2 — 15 балів (захист 10 балів, звіт 5 балів)

КП 3 (3.1) — 10 балів

КП 4 (3.2) — 5 балів

КП 5 (3.3) — 5 балів

Робота вважається виконаною та захищеною, якщо студент отримав не менше ніж по 60% з кожного з п. 1.1 та 1.2.

1.2. Захист теоретичного матеріалу з теми РКП дорівнює:

КП1 – 5 балів,

КП2 – 15 балів,

КП3-КП5 – 10 балів, 5 балів, 5 балів. Захист у протоколі за окремими вимогами в залежності від мови програмування.

Вимоги до захисту КП 3: на захист розписати покрокове виконання запиту з константами в аргументах предиката з власної роботи. КП 4: на захист у протоколі провести аналіз вбудованих функцій та структурованих типів мови Python (на C). КП 5: на захист у протоколі проаналізувати особливості обраної функціональної мови (LISP, Haskell, F# тощо) та як працюють рекурсії у функціональних мовах. КП3, КП4, КП5 - письмовий автономний захист, що завантажується у Classroom на платформі Сікорський.

Термін виконання та захист РКП визначається згідно з графіком навчального процесу: Моделювання логічних задач – Частина 1, КП1, КП2 – у період до 1-го календарного контролю; КП3, КП4, Моделювання логічних задач – Частина 2 – у період до 2-го календарного контролю. КП5, Звіт з виконання СРС – у період до виконання МКР – заліковий тиждень.

### 2. Самостійна робота

Робоча навчальна програма дисципліни передбачає такий вид роботи, як самостійна робота студентів (СРС). СРС є обов'язковим видом роботи та включає підготовку до КП, МКР та виконання завдання з моделювання логічних задач.

#### **2.1. Моделювання логічних задач. (Моделювання у просторі станів, Продукційні системи).**

**Частина 1. Використання технологій штучного інтелекту для написання програмного коду – 20 балів.**

- «Мавпа та банан», «Вовк, коза та капуста», «Задача Ейнштейна», «Пригоди в дивних печерах», «Задача про ханойські вежі», «Задача про N королев», «Задача планування».

Команда 2-3-4 студента, призначається один з підходів. Імперативна (програмна реалізація однією з мов C/C++, C#, Java), мультипарадигменна (Python). Захист включає презентацію, демонстрацію коду та результатів.

Розподіл на команди, кожна з яких презентує реалізацію **обраної логічної задачі**. Аналіз результатів.

## **2.2. Моделювання логічних задач (Моделювання у просторі станів, Продукційні системи).**

### **Частина 2. Реалізація на декларативній мови програмування. – 20 балів**

- Захист включає презентацію, демонстрацію коду та результатів.

### **2.3. Виконання роботи зі списку СРС на 10 балів.**

### **3. Модульний контроль**

Модульний контроль складається з однієї модульної (підсумкової) контрольної роботи (МКР), що передбачено робочою навчальною програмою ОК. Примірники варіантів контрольної роботи наводяться у Додатку до робочої навчальної програми з кредитного модуля.

3.1. МКР включає два теоретичних та одне практичне питання. МКР оцінюється в межах 0÷10 балів. Допуском до МКР є виконання та захист 1÷5 робіт з комп'ютерного практикуму.

### **4. Відвідування лекцій**

4.1. Обов'язкове - у разі синхронного дистанційного навчання.

4.2. Не обов'язкове - у разі асинхронного дистанційного навчання при наявності оформлення індивідуального графіку навчання.

### **5. Календарний контроль**

5.1. У разі синхронного навчання за несвоєчасне виконання та захист РКП та інших завдань студент, який не атестований без поважних причин з одного календарного контролю може отримати не більше 85 балів, який не атестований з обох календарних контролів – не більше 75 балів.

5.2. Деякі види робіт можуть оцінюватися як 0 балів у разі низької якості виконання.

### **6. Заохочувальні бали**

6.1. Участь в модернізації робіт з комп'ютерного практикуму.

6.2. Доповіді на конференціях з теми відповідної дисципліни.

6.3. Призові місця у хакатонах та конкурсах стартапів за напрямом штучного інтелекту.

6.4. За результатами неформальної освіти у разі часткового співпадіння тем програми дисципліни та програми курсу неформальної освіти.

Примітка: для студентів, які навчаються за індивідуальним графіком асинхронно всі види робіт мають бути завантажені до [classroom.google.com](https://classroom.google.com).

## **Додаткова інформація з дисципліни (освітнього компонента)**

- *Можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за відповідною тематикою;*
- *Участь у науково-практичній конференції «Системні науки та інформатика» за темою дипломної роботи за погодженням з науковим керівником;*
- *Участь у конкурсі стартапів Sikorsky Challenge з власним стартапом.*
- *Відео-записи лекцій доступні для перегляду на платформі дистанційного навчання «Сікорський».*
- *Перелік питань, які виносяться на семестровий контроль, наведено в додатку А.*

## Робочу програму навчальної дисципліни (силабус):

Складено завідувачем кафедри математичних методів системного аналізу к.т.н., доцентом Тимощук О.Л.

Ухвалено кафедрою ММСА (протокол № 3 від 08.10.2025 р.)

Погоджено Методичною комісією НН ІПСА (протокол № 2 від 09.10.2025 р.)

## ДОДАТКИ

### Додаток А

#### Зразок варіантів контрольних робіт.

Модульна (підсумкова) контрольна робота. Мова програмування Пролог.

##### Варіант 1.

1. Універсальна та екзистенціальна квантифікація змінної в правилі.
2. Арифметика Турбо-Прологу (приклад, типи чисельних доменів).
3. Програма пошуку в статистичній БД.

##### Варіант 2.

1. Декларативна семантика Прологу (визначення, приклади).
2. Розділ предикатів.
3. Програма запису в файл (послідовність кроків, вбудовані предикати).

##### Варіант 3.

1. Рекурсивні процедури.
2. Складені домени.
3. Процедура сортування елементів списку.

##### Варіант 4.

1. Процедурна семантика Прологу (визначення, приклади).
2. Розділ clauses.
3. Процедура влучення в список.

##### Варіант 5.

1. Визначення списків. Метод ділення на хвіст і голову.
2. Розділ domains.
3. Перетворення рядків у список символів.

Контрольна робота № 2. Мова програмування Lisp.

##### Варіант 1.

1. Реалізація функціональної абстракції.
2. Паралельна рекурсія.
3. Програма обчислення довжини списку.

##### Варіант 2.

1. Призначення автофункції.
2. Визначення списків, атомів, чисел.
3. Програма обчислення копії списку.

##### Варіант 3.

1. Призначення замикання.
2. Взаємна рекурсія.
3. Програма визначення належності елемента списку.

##### Варіант 4.

1. Рекурсії по аргументах і по значеннях.
2. S-вирази: призначення та синтаксис.

3. Програма обчислення суми ряду.

**Варіант 5.**

1. Метод накопичуючих параметрів.
2. Lambda-функції.
3. Програма обчислення добутку ряду.