



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 ЄКТС</i> |
| Семестровий контроль/ контрольні заходи | <i>залік</i> |
| Розклад занять | <i>https://schedule.kpi.ua/</i> |
| Мова викладання | <i>Українська</i> |
| Інформація про керівника курсу / викладачів | <i>Лектор: к.т.н., доцент, Тимощук О.Л., tymoshchuk.oksana@iit.kpi.ua Практичні/Лабораторні/ комп'ютерний практикум: к.т.н., доцент, Тимощук О.Л., tymoshchuk.oksana@iit.kpi.ua</i> |
| Розміщення курсу | <i>Платформа дистанційного навчання «Сікорський», Google classroom код курсу swsfm7d</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* -програм.

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; дозапису 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 рекурсії.

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 контекстно-вільного (КВ) та лексичного аналізаторів. Представлення знань. Методи виведення. Синтаксичний аналіз: моделі речень для КВ граматик; ієрархії Хомського та контекстно-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 контекстно-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 А.Черча. Редукція графів. Комбінаторна редукція.

Тема 11.2. Приклади мов функціонального програмування. Стили 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. Застосовуючі та відображаючі функціонали. Метод накопичування параметрів. Застосовуючі та відображаючі 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://efaidnbmnnnibpcajpcglclefindmkaj/ <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., Zelkowitz 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», *Googleclassroom*, код курсу *swsfm7d* *Обов'язковими для прочитання є Слайди лекцій, конспект лекцій та Методичні рекомендації з кредитного модуля «Мови і технології штучного інтелекту»*

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 o o358PROLOG 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 кредитного модуля

| Роз-діл | Тема | Найменування розділу, теми | Розподіл навчального часу | | | |
|---------|---|---|---------------------------|--------|--------------|-----|
| | | | всього | лекції | комп. практ. | СРС |
| 1. | Мови створення систем штучного інтелекту. | | 4 | 2 | | 2 |
| | 1.1. | Вступ до логічного програмування. | 1 | 1 | | |
| | 1.2. | Зв'язок формальної логіки та логічного програмування. | 3 | 1 | | 2 |

| Роз-діл | Тема | Найменування розділу, теми | Розподіл навчального часу | | | |
|---------|---|---|---------------------------|----------|--------------|----------|
| | | | всього | лекції | комп. практ. | СРС |
| 2. | Логічна парадигма програмування. Синтаксис та семантика Пролог-програм. | | 7 | 2 | 2 | 3 |
| | 2.1. | Основні поняття мови Prolog . | 0,5 | 0,5 | | |
| | 2.2. | Рекурсивні процедури. | 4 | 1 | 1 | 2 |
| | 2.3. | Декларативна та процедурна семантика мови Prolog . | 2,5 | 0,5 | 1 | 1 |
| 3. | Структура програми мовою Prolog . | | 3 | 2 | | 1 |
| | 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. | | 6 | 2 | 2 | 2 |
| | 4.1. | Позалогічні предикати мови Prolog | 5 | 1,5 | 2 | 1,5 |
| | 4.2. | Файли прямого доступу. | 1 | 0,5 | | 0,5 |
| 5. | Механізми створення повторів та рекурсій. | | 10 | 4 | 2 | 4 |
| | 5.1. | Основні методи створення повторів. | 5 | 2 | 1 | 2 |
| | 5.2. | Методи організації рекурсії. | 5 | 2 | 1 | 2 |
| 6. | Абстрактні типи даних мови Prolog . | | 14 | 6 | 4 | 4 |
| | 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 Prolog . | | 4 | 2 | | 2 |
| | 7.1. | Метаінтерпретатори в мові Prolog . | 2 | 1 | | 1 |
| | 7.2. | Експертні системи в мові Prolog . | 2 | 1 | | 1 |
| 8. | Динамічні бази даних (БД), експертні системи (ЕС), контекстно-вільний (КВ) аналіз засобами мови Prolog . | | 10 | 2 | 4 | 4 |
| | 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. | Алгоритми навчання в мові Prolog . | | 7 | 1 | | 6 |

| Роз-діл | Тема | Найменування розділу, теми | Розподіл навчального часу | | | |
|---------|--|--|---------------------------|----------|--------------|----------|
| | | | всього | лекції | комп. практ. | СРС |
| | 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. | Обробка природної мови засобами Prolog. | | 5 | 1 | | 4 |
| | 10.1 | Синтаксичний та семантичний аналіз. | 3 | 1 | | 2 |
| | 10.2 | Стохастичний підхід до аналізу мови. | 2 | | | 2 |
| | Контрольна робота 1 | | 7 | 1 | | 6 |
| 11. | Функціональна парадигма програмування. Мови функціонального програмування. | | 5 | 2 | | 3 |
| | 11.1 | Функція як природній формалізм мов функціонального програмування. | 3 | 1 | | 2 |
| | 11.2 | Приклади мов функціонального програмування. | 2 | 1 | | 1 |
| 12. | Основи функціонального програмування. | | 7 | 2 | 2 | 3 |
| | 12.1 | Типи рекурсій. | 4,5 | 1 | 2 | 1,5 |
| | 12.2 | Застосовуючі та відображаючі функціонали | 2,5 | 1 | | 1,5 |
| 13. | Lisp та імперативне програмування. | | 4 | 1 | | 3 |
| | 13.1 | Поняття замикання, функціональної абстракції, автофункції | 4 | 1 | | 3 |
| 14. | Елементи процедурного програмування в Lisp . | | 6 | 1 | 2 | 3 |
| | 14.1 | Зв'язок між процедурними та функціональними мовами програмування. | 6 | 1 | 2 | 3 |
| | Контрольна робота 2 | | 7 | 1 | | 6 |
| 15. | Стратегії пошуку в мові LISP . | | 3 | 1 | | 2 |
| | 15.1 | Стратегії неінформованого пошуку. Стратегії інформованого (евристичного) пошуку. | 3 | 1 | | 2 |

| Роз-діл | Тема | Найменування розділу, теми | Розподіл навчального часу | | | |
|-----------------------------|---|--|---------------------------|-----------|--------------|-----------|
| | | | всього | лекції | комп. практ. | СРС |
| 16. | Об'єктно-орієнтоване програмування засобами Common <i>LISP</i> Object System. | | 3 | 1 | | 2 |
| | 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 |
| Підготовка до заліку | | | 8 | 2 | | 6 |
| Усього в Y семестрі | | | 120 | 36 | 18 | 66 |

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></p> <p>РОЗДІЛ 1. Мови створення систем штучного інтелекту.</p> <p>Тема 1.1. Вступ до логічного програмування.</p> <p>Тема 1.2. Зв'язок формальної логіки та логічного програмування.</p> <p>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.</p> <p>Питання на СРС: Засвоєння пакетів систем Visual Prolog, SWI Prolog та Turbo-Prolog. Показати дію методу резолюції на заданій теорії.</p> |
| 2 | <p><u>Лекція 2.</u></p> <p>РОЗДІЛ 2. Логічна парадигма програмування. Синтаксис та семантика Prolog - програм.</p> <p>Тема 2.1. Основні поняття мови Prolog.</p> <p>Тема 2.2. Рекурсивні процедури.</p> <p>Тема 2.3. Декларативна та процедурна семантика мови Prolog.</p> <p>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 поняття рекурсії. Класичні приклади рекурсії. Лівобічна та правобічна рекурсії. Поняття семантичної моделі. Семантична модель мови Prolog в вигляді абстрактної машини. Декларативна та процедурна семантичні моделі мови Prolog.</p> <p>Питання на СРС: Засвоїти на практиці різницю між внутрішніми та зовнішніми цілями.</p> |

| | |
|---|---|
| | <p>Поекспериментувати з простими та складними запитами до фактів і правил. Зробити висновки щодо використання лівобічної рекурсії.</p> |
| 3 | <p><u>Лекція 3.</u> РОЗДІЛ 3. Структура програми мовою Prolog. Тема 3.1. Розділи програми. Тема 3.2. Стандартні предикати: арифметичні та управляючі. Синтаксис та призначення розділів програми. Опис доменів і предикатів. Використання зовнішніх та внутрішніх цілей. Діаграма успішного доведення цільового твердження. Предикати та твердження різних арностей. Арифметика мови <i>Prolog</i>: 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. Питання на СРС: Скласти приклади використання предиката not.</p> |
| 4 | <p><u>Лекція 4.</u> РОЗДІЛ 4. Застосування файлів на зовнішніх носіях. Тема 4.1. Позалогічні предикати мови Prolog. Тема 4.2. Файли прямого доступу. 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> РОЗДІЛ 5. Механізми створення повторів та рекурсії. 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. Питання на СРС: 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> 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. Питання на СРС: Розглянути "червоне" та "зелене" відсікання.</p> |
| 7 | <p><u>Лекція 7.</u> РОЗДІЛ 6. Абстрактні типи даних мови Prolog. Тема 6.1. Списки, черги, дерева, довідники. Операції над списками. 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>Питання на СРС: За допомогою операції конкатенації виконати: - конкатенацію двох та більше списків; - роз'єднання списків на два та більше.</p> |
| 8 | <p><u>Лекція 8.</u> Представлення множин двійковими деревами, двійкові довідники: додавання та вилучання елементів. Питання на СРС: Організація пошуку у двійкових довідниках та деревах. Методи представлення множин деревами.</p> |
| 9 | <p><u>Лекція 9.</u> 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.</p> |
| 10 | <p><u>Лекція 10.</u> РОЗДІЛ 7. Метаінтерпретатори, експертні системи в мові Prolog. Тема 7.1. Метаінтерпретатори в мові Prolog. Тема 7.2. Експертні системи в мові Prolog. Введення до метаінтерпретаторів. Оболонка до експертної системи на підставі правил. Семантичні мережі, фрейми та схеми 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. Питання на СРС: Гібридні системи: переваги та недоліки систем з потужними методами.</p> |
| 11 | <p><u>Лекція 11.</u> РОЗДІЛ 8. Динамічні бази даних (БД), контекстно-вільний (КВ) аналіз засобами мови Prolog. Тема 8.1. Системи управління базами даних (СУБД). Тема 8.2. Проектування контекстно-вільного (КВ) та лексичного аналізаторів. 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 граматик; ієрархії Хомського та контекстно-залежні граматики. Лексичний аналізатор речень. Доведення теорем засобами мови Prolog. Питання на СРС: Доведення теорем засобами Прологу.</p> |
| 12 | <p><u>Лекція 12.</u> РОЗДІЛ 9. Алгоритми навчання в мові Prolog. Тема 9.1. Машинне навчання, що визначається символічним представленням інформації. Тема 9.2. Нейромережеве машинне навчання. Тема 9.3. Машинне навчання на основі соціальних та емерджентних принципів. Пошук у просторі версій. Індуктивний алгоритм побудови дерева рішень. Теорія студіювання. Знання та навчання. Навчання без вчителя. Навчання з підкріпленням. Навчання перцептрона.</p> |

| | |
|----|---|
| | <p>Питання на СРС: 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>РОЗДІЛ 10. Обробка природної мови засобами Prolog.</p> <p>Тема 10.1. Синтаксичний та семантичний аналіз.</p> <p>Тема 10.2. Стохастичний підхід до аналізу мови.</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>Питання на СРС: Ймовірнісні мовні моделі. Підхід на основі дерева рішень. Граматичний аналіз та інші додатки стохастичного підходу.</p> |
| 13 | <p><u>Лекція 13.</u></p> <p>Контрольна робота №1</p> <p>РОЗДІЛ 11. Функціональна парадигма програмування. Мови функціонального програмування.</p> <p>Тема 11.1. Функція як природній формалізм мов функціонального програмування.</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>Тема 11.2. Приклади мов функціонального програмування. Стилi мов функціонального програмування. Конкретні реалізації: <i>SML, Hope, Miranda, Haskell, FP</i>, діалекти мови <i>LISP</i>.</p> <p>РОЗДІЛ 12. Основи функціонального програмування.</p> <p>Тема 12.1. Типи рекурсії. 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>Тема 12.2. Застосовуючи та відображаючи функціонали. Метод накопичування параметрів. Застосовуючи та відображаючи функціонали.</p> <p>РОЗДІЛ 13. LISP та імперативне програмування.</p> <p>Тема 13.1. Поняття замикання, функціональної абстракції, автофункції. Замикання. Функціональна абстракція. Автофункції. Відповідність між функціональними та імперативними програмами.</p> <p>Питання на СРС: Комбінації функцій відбору.</p> |
| 16 | <p><u>Лекція 16.</u></p> <p>РОЗДІЛ 14. Елементи процедурного програмування в LISP.</p> <p>Тема 14.1. Зв'язок між процедурними та функціональними мовами програмування.</p> <p>PROG-вирази у мові <i>LISP</i>. Побудова списків. Предикати, що написані за допомогою PROG-виразів. Інші конструкції процедурного програмування.</p> <p>Питання на СРС: Порівняльна характеристика програм на Ліспі, що написані засобами функціонального та процедурного програмування.</p> <p>Контрольна робота №2</p> |
| 17 | <p><u>Лекція 17.</u></p> |

| | |
|----|--|
| | <p>РОЗДІЛ 15. Стратегії пошуку в мові LISP.</p> <p>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.</p> <p>РОЗДІЛ 16. Об'єктно-орієнтоване програмування засобами Common LISP Object System.</p> <p>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.</p> <p>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.</p> |
| 18 | Залікова контрольна робота |

Роботи з комп'ютерного практикуму

| № КП | № розділу, теми | Зміст лабораторної роботи | К-сть годин |
|------|---|---|-------------|
| | | Дебати за темою | 2 |
| КП1 | Розділ 2, теми 2.1, 2.2. | Створення моделі відношень у заданій предметній області. Вступ до мови програмування Prolog. | 2 |
| КП2 | Розділ 4, тема 4.1; розділ 5, теми 5.1, 5.2. | Позалогічні предикати мови <i>Prolog</i> . Механізми реалізації повторів та фільтрації даних. | 2 |
| КП3 | Розділ 6, тема 6.1. | Списки, черги, дерева, довідники в Prolog | 4 |
| КП4 | Розділ 7, тема 7.1; розділ 8, тема 8.1. | Списки, черги, дерева, довідники в Python | 4 |
| КП5 | Розділ 11, теми 11.2, 11.3. | Засвоєння пакету <i>muLisp</i> . Програми з найпростішою структурою. | 2 |
| | | Програмування циклічних алгоритмів мовою <i>muLisp</i> засобами програмування: | 2 |
| | Розділ 12, тема 12.1; розділ 15, тема 15.1 | а) функціонального, б) процедурного. | |
| | | | 18 |

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, яку поділено на

дві контрольні роботи (12 годин), до залікової роботи (6 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. Поекспериментувати 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 силябусу.

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 кредитного модуля

Семестровий рейтинг студента з модуля (дисципліни) 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. Командні дебати | 20 |
| 3. Моделювання логічних задач | 20 |
| 4. Модульна контрольна робота | 10 |
| 5. Залікова робота | 10 |
| Всього | 100 |

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

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 б., КП2 – 10 б., КП3-КП5 – 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: КП1 – у період до 1-го календарного контролю; КП2-КП4 – у період до 2-го календарного контролю. КП5 – у період до виконання МКР.

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 дебатів та виконання завдання з моделювання логічних задач.

2.1. Дебати: «Штучний інтелект – великі можливості чи великі загрози?» - 20 балів

Розподіл на дві команди, кожна з яких відстоює «загрози» або «можливості».

Дебати включають два раунди: 1). Перший – презентація рішення. 2). Командні змагання «Питання-відповіді».

Оцінювання:

- Презентація – 4,5 бали;
- Змістовність вкладу за роллю – 5,5 балів;
- Коефіцієнт участі у командній роботі (тімлід оформлює у телеграм анонімне опитування за кожним членом команди та надає викладачу результати у вигляді скріншотів) – 4 балів;
- Участь у другому раунді – 6 балів (запитання до іншої команди – 3 бали; відповіді на запитання іншої команди – 3 бали).

2.2. Моделювання логічних задач (Моделювання у просторі станів, Продукційні системи) – 20 балів

- «Мавпа та банан», «Вовк, коза та капуста», «Задача Ейнштейна»
- Команда 2-3-4 студента, призначається один з підходів. Три парадигми – імперативна (програмна реалізація однією з мов C/C++, C#, Java), логічна (Prolog), мультипарадигменна (Python). Захист включає презентацію, демонстрацію коду та результатів.

3. Модульний контроль

Модульний контроль складається з однієї модульної контрольної роботи (МКР), що передбачено робочою навчальною програмою кредитного модуля. 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. Залікова контрольна робота

5.1. Обов'язкова для студентів, які навчаються за індивідуальним графіком асинхронно.

6. Штрафні заходи

6.1. Штрафні бали не нараховуються.

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 90 балів, який не атестований з обох календарних контролів – не більше 75 балів.

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

7. Заохочувальні бали

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

7.2. Доповіді на конференціях з теми відповідної дисципліни.

7.3. Призові місця у хакатонах за напрямом штучного інтелекту.

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

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 Тимошук О.Л.

Ухвалено кафедрою ММСА (протокол № 13 від 05.06.2024 р.)

Погоджено Методичною комісією НН ІПСА (протокол № 10 від 24.06.2024)

ДОДАТКИ

Додаток А

Зразок варіантів контрольних робіт.

Контрольна робота № 1. 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 хвост 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.