КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ
Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015
Кафедра програмної інженерії та кібербезпеки

ЗАТВЕРДЖЕНО
вченою радою
(прот. 10 п. 10 від "21" червня 2018 р.)
Ректор
_____ А.А. Мазаракі

ФУНКЦІОНАЛЬНЕ ТА ЛОГІЧНЕ ПРОГРАМУВАННЯ /
FUNCTIONAL AND LOGICAL PROGRAMMING

ПРОГРАМА ТА РОБОЧА ПРОГРАМА / SYLLABUS

освітній ступінь	**магістр /** master
галузь знань	**12 Інформаційні технології /** Information Technology
спеціальність	**121 Інженерія програмного забезпечення /** Software Engineering
спеціалізація	**Інженерія програмного забезпечення /** Software Engineering

Київ 2018

Розповсюдження і тиражування без офіційного дозволу КНТЕУ заборонено

Автор Т.В. Савченко, канд. техн. наук, доцент

Програму та робочу програму розглянуто і затверджено на засіданні кафедри програмної інженерії та інформаційних систем 15 травня 2018 р., протокол №26.

Рецензенти: Н.О. Котенко, канд. пед. наук, ст.викл.
Р.А. Серебряков, канд. техн. наук, доцент,
начальник відділу програмного забезпечення
ТОВ «АЛАРЕД ГРУП»

Навчальна програма

ФУНКЦІОНАЛЬНЕ ТА ЛОГІЧНЕ ПРОГРАМУВАННЯ /
FUNCTIONAL AND LOGICAL PROGRAMMING

ПРОГРАМА ТА РОБОЧА ПРОГРАМА / SYLLABUS

освітній ступінь	**магістр /** master
галузь знань	**12 Інформаційні технології /** Information Technology
спеціальність	**121 Інженерія програмного забезпечення /** Software Engineering
спеціалізація	**Інженерія програмного забезпечення /** Software Engineering

ВСТУП

Програма та робоча програма дисципліни «Функціональне та логічне програмування» призначені для студентів освітнього ступеня «магістр» спеціальності 121 «Інженерія програмного забезпечення».

«Функціональне та логічне програмування» належить до обов'язкових дисциплін, які формують основні фахові компетентності циклу професійної підготовки магістра з інженерії програмного забезпечення.

Дисципліна *«Функціональне та логічне програмування»* передбачає вивчення двох основних розділів: 1. Функціональне програмування. 2. Логічне програмування.

Функціональне та *логічне програмування* (ФЛП) – два напрямки програмування за своїми принципами і методологією цілком відмінні від традиційного Фон-Нейманівського процедурного програмування, що базується на роботі з пам'яттю. Функціональне та логічне програмування орієнтоване на нетрадиційне використання обчислювальної техніки: розуміння природної мови, бази знань, експертні системи та інші задачі. Знання методів і засобів функціонального та логічного програмування є обов'язковим елементом культури сучасного програміста.

Вивчення дисципліни дозволить студентам оволодіти методами функціонального та логічного програмування, теоретичними знаннями щодо принципів функціонального підходу при розробці програм; основними підходами і засобами логічного програмування з використанням різноманітних структур даних у програмуванні та їх застосуванні при побудові ефективних алгоритмів; вмінням користуватись сучасним програмним забезпеченням.

Програма та робоча програма дисципліни «Функціональне та логічне програмування» складаються з таких розділів:

1. Мета, завдання та результати вивчення (компетентності) дисципліни, її місце в освітньому процесі.
2. Зміст дисципліни.
3. Структура дисципліни та розподіл годин за темами (тематичний план).
4. Тематика та зміст лекційних, лабораторних занять та самостійної роботи студентів.
5. Список рекомендованих джерел.

1. МЕТА, ЗАВДАННЯ ТА РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ (КОМПЕТЕНТНОСТІ), ЇЇ МІСЦЕ В ОСВІТНЬОМУ ПРОЦЕСІ

Мета вивчення дисципліни «Функціональне та логічне програмування» полягає у набутті студентами знань, необхідних для формування теоретичної бази знань та практичних навичок використання методів та засобів функціонального та логічного програмування для розв'язання складних і неформалізованих задач, що зустрічаються в реальних економічних, організаційних і виробничих системах.

Завданням вивчення дисципліни «Функціональне та логічне програмування» є:

1) набуття: знань з теоретичних основ в галузі побудови, розробки та експлуатації програмних продуктів, що використовують засоби логічного та функціонального програмування;
2) ознайомлення: з сучасним станом та перспективами розвитку функціонального та логічного програмування;
3) оволодіння: практичними навичками створення програмних продуктів засобами логічного та функціонального програмування, застосування відповідних методів, моделей і алгоритмів розв'язання інтелектуальних задач.

Предметом дисципліни є сукупність теоретичних і практичних проблем, які пов’язані з проектуванням, розробкою та практичним використанням програмних продуктів, що використовують засоби логічного та функціонального програмування.

Компетентності зазначеної дисципліни – отримати певну систему знань, на основі якої фахівець зможе розв'язувати задачі професійної діяльності в галузі експлуатації, проектування та розробки інтелектуальних систем, застосовувати на практиці методи, моделі і алгоритми автоматичного отримання управлінських рішень в сучасних інформаційно-обчислювальних комплексах.

Місце в освітньому процесі: відповідно до структури навчального плану, знання і практичні навички, отримані під час вивчення дисципліни «Функціональне та логічне програмування» базуються на знаннях з наступних дисциплін: «Об'єктно-орієнтоване програмування», «Операційні системи», «Експертні системи», «Алгоритми та структури даних», «Бази даних» та «Штучний інтелект» і є основою для підготовки випускних кваліфікаційних робіт за спеціальністю та в подальшій практичній роботі за фахом.

2. ЗМІСТ ДИСЦИПЛІНИ

РОЗДІЛ 1. ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ

Тема 1. Загальне уявлення про функціональне програмування та його застосування.

Домінуючі парадигми програмування. Типи парадигми. Імперативне (процедурне програмування). Директивне та структурне програмування. Непроцедурне програмування. Об'єктно-орієнтоване програмування. Декларативне програмування. Функціональне та логічне програмування. Концепція функціонального програмування. LISP – мова програмування високого рівня. Функціональна модель обчислень. Суперпозиція функцій і рекурсія як основні методи програмування. Вирішення задач нечислового характеру. Особливості функціонального програмування.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5
Інтернет-ресурси: 11

Тема 2. Елементарний LISP

Основні визначення. Символи, атомарні об'єкти, списки, S-вирази. Списки як засіб подання знань. Точкова пара. Вбудовані функції LISP. Предикативна (префіксна) форма запису функцій. Функції QUOTE (блокування) і EVAL (зняття блокування). Арифметичні функції в LISP. Базові функції обробки списків. Функції CAR і CDR. Функція-конструктор CONS. Предикати ATOM, EQ, EQUAL. Застосування багатократних CAR-CDR. Елементарні функції над списками та S-виразами.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5
Інтернет-ресурси: 11

Тема 3. Конструювання списків

Конструювання списків у LISP. Функції вищого рівня: CONS, LIST, APPEND. Функції виділення елементів списку: LENGTH, REVERSE, LAST, NTH. Неіменовані функції LISP. Застосування *lambda*-виразів та *nlambda*-виразів. Іменовані функції LISP. Створення власних функцій. Функція DEFUN. Конструктор функцій LABEL.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5
Інтернет-ресурси: 11

Тема 4. Числові функції. Керуючі структури

Числові функції в LISP. Арифметичні, логічні, булеві, ірраціональні та трансцендентні функції. Функції збільшення та зменшення. Тригонометричні функції. Керуючі структури. Функції присвоювання: SET, SETQ, LET. Опис структур. Послідовність обчислень. Структури розгалуження. Застосування функцій COND, IF, WHEN і UNLESS. Циклічні обчислення в LISP. Використання функцій DO, DOTIMES та LOOP.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5

Тема 5. Поняття рекурсії. Функціонал

Прості рекурсії. Рекурсії за значенням і за аргументами. Застосування простих рекурсій. Правила побудови рекурсивних функцій. Паралельна рекурсія. Взаємна рекурсія. Рекурсії вищих порядків. Приклади застосування рекурсій більш високого порядку.

Поняття функціонала. Функціональний аргумент. Аплікативний функціонал. Застосування APPLY, FUNCALL, EVAL. Алгоритми пошуку на LISP.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5
Інтернет-ресурси: 11

Тема 6. Застосування діалектів мови LISP

Реалізації стандарту Common LISP. Інтерпретатор Common LISP. Текстовий редактор Emacs. Розробка програм в середовищі Scheme. Елементи мови програмування Scheme. Визначення процедур. Символьні вирази. Процедури вищого порядку. Присвоювання. Використання Visual LISP в системі автоматизованого проектування AutoCAD. Призначення та можливості AutoLISP. Основні команди системи AutoCAD в мові AutoLISP. Робота з базами даних в AutoLISP.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5
Інтернет-ресурси: 11, 12

Тема 7. Вирішення задач, заснованих на знаннях

Області застосування експертних систем (ЕС). Основи методології розробки експертних систем. Експертні системи реального часу. Основні властивості ЕС реального часу. Архітектура ЕС реального часу. CLIPS – програмне середовище для розробки ЕС. Можливості мови CLIPS. Використання Бекусо-Наурова (БНФ)-нотації для визначення конструкцій мови CLIPS. Компоненти CLIPS. Цикл виконання правил. Конструктори. Факти і шаблони. Основні функції та базові команди CLIPS. Використання CLIPS в експертних системах.

Список рекомендованих джерел:
Основний: 1
Додатковий: 4, 5, 6

РОЗДІЛ 2. ЛОГІЧНЕ ПРОГРАМУВАННЯ

Тема 8. Математичні основи логічного програмування

Концепція логічного програмування. Основні конструкції логічного програмування. Поняття формальної системи. Логіка висловлювань. Логічні зв'язки. Формули логіки висловлювань. Логічний висновок у логіці висловлювань. Рівнозначність логічних формул. Відношення логічного висновку. Скорочений метод перевірки аргументів. Розв'язання логічних задач засобами логіки висловлювань. Числення

предикатів першого порядку. Недостатність логіки висловлень. Поняття предиката. Квантори загальності та існування. Заперечення речень з кванторами. Інтерпретація формул числення предикатів першого порядку. Нормалізація формул числення предикатів першого порядку. Формалізація процесу логічного виведення.

Список рекомендованих джерел:
Основний: 2
Додатковий: 7, 8

Тема 9. Логічне програмування та штучний інтелект

Поняття штучного інтелекту. Логіко-лінгвістичні моделі в системах управління. Штучний інтелект і теорія пошуку висновку. Сучасний стан штучного інтелекту. Експертні системи. Робототехніка. Автономні агенти. Чат-роботи. Перспективи та тенденції розвитку штучного інтелекту. Нейронні мережі. Еволюційні обчислення. Напрямки розвитку штучного інтелекту. Перспективи логічного програмування. Області застосування мови Prolog. Робоче середовище Turbo-Prolog. SWI-Prolog – реалізація мови програмування Prolog.

Список рекомендованих джерел:
Основний: 2, 3
Додатковий: 7, 8

Тема 10. Особливості мови Visual Prolog

Особливості застосування Visual Prolog. Основні елементи мови Visual Prolog. Поняття термів. Стандартні типи даних (symbol, string, char, integer, real). Позначення констант та змінних у Visual Prolog. Факти та правила у Visual Prolog. Поняття аргументів та предикатів. Логічні формули. Призначення запитів у Prolog. Робота механізму пошуку з поверненням. Дерева пошуку рішень.

Список рекомендованих джерел:
Основний: 2, 3
Додатковий: 8, 9
Інтернет-ресурси: 13

Тема 11. Структура програми Visual Prolog

Основні розділи програми Visual Prolog (CONSTANTS – опис констант; DOMAINS – опис доменів; FACTS – опис предикатів внутрішньої бази даних; PREDICATES – опис предикатів; CLAUSES – опис тверджень; GOAL – опис внутрішньої цілі). Директиви компілятора (trace, include). Призначення розділів Visual Prolog. Семантика програм Visual Prolog.

Список рекомендованих джерел:
Основний: 3
Додатковий: 8, 9
Інтернет-ресурси: 13

Тема 12. Предикати у Visual Prolog

Визначення предикатів. Вбудовані предикати Visual Prolog. Модулі. Структура модуля. Відсікання та заперечення. Принцип роботи відсікання. Область видимості відсікання. Використання відсікань. Зелені та червоні відсікання. Динамічне відсікання. Заперечення. Цикли з відкатом. Структура циклу з відкатом. Реалізація циклів з відкатом. Рекурсія. Структура рекурсії. Реалізація рекурсії. Предикати другого порядку та анонімні предикати.

Список рекомендованих джерел:
Основний: 3
Додатковий: 8, 9
Інтернет-ресурси: 13

Тема 13. Списки та їх обробка у Visual Prolog

Подання списків у Visual Prolog. Оголошення списків. Список як рекурсивний складений об'єкт. Структура списку (голови, хвости). Використання та друк списків. Підрахунок числа елементів списку (довжини). Предикати для обробки списків. Додавання та видалення елемента в списку. Зчеплення (конкатенація) списків. Видалення зі списку повторювальних елементів. Обернення списків у Visual Prolog (reverse). Застосування рекурсії. Предикат перевірки, чи є список паліндромом (palindrom). Предикат одержання елемента по номеру у списку (n_element). Видалення зі списку всіх входжень заданого

значення (delete_all, delete_one). Обчислення суми елементів списку чисел. Пошук максимального елемента. Застосування Visual Prolog.

Список рекомендованих джерел:
Основний: 3
Додатковий: 8, 10
Інтернет-ресурси: 13

Тема 14. Створення додатків у середовищі Visual Prolog

Створення консольних додатків. Управління за допомогою меню головного вікна задач. Створення проекту. Створення модального діалогу. Зміна меню програми. Зміна панелі інструментів. Введення основного коду в програму. Інкапсуляція інтерактивного коду. Створення додатків за допомого графічного інтерфейсу. Управління за допомогою елементів форми. Створення нового класу. Технології створення програми в режимі роботи з графічним інтерфейсом.

Список рекомендованих джерел:
Основний: 3
Додатковий: 8, 9, 10

3. СТРУКТУРА ДИСЦИПЛІНИ «ФУНКЦІОНАЛЬНЕ ТА ЛОГІЧНЕ ПРОГРАМУВАННЯ» ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ (ТЕМАТИЧНИЙ ПЛАН)

Назва теми	Кількість годин				Форми контролю
	Усього год/кредитів	Лекції	Лабораторні заняття / МК	Самостійна робота студ.	
РОЗДІЛ 1. ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ					
Тема 1. Загальне уявлення про функціональне програмування та його застосування	14	2	2	10	УО, Т, ЛР
Тема 2. Елементарний LISP	12	2	2	8	УО, Т, ЛР
Тема 3. Конструювання списків	12	2	2	8	УО, Т, ЛР
Тема 4. Числові функції. Керуючі структури	12	2	2	8	УО, Т, ЛР
Тема 5. Поняття рекурсії. Функціонал	14	2	2	10	УО, Т, ЛР
Тема 6. Застосування діалектів мови LISP	12	2	2	8	УО, Т, ЛР
Тема 7. Вирішення задач, заснованих на знаннях	14	2	2	10	УО, Т, ЛР
РОЗДІЛ 2. ЛОГІЧНЕ ПРОГРАМУВАННЯ					
Тема 8. Математичні основи логічного програмування	12	2	2	8	УО, Т, ЛР
Тема 9. Логічне програмування та штучний інтелект	14	2	2	10	УО, Т, ЛР
Тема 10. Особливості мови Visual Prolog	12	2	2	8	УО, Т, ЛР
Тема 11. Структура програми Visual Prolog	14	2	2	10	УО, Т, ЛР
Тема 12. Предикати у Visual Prolog	12	2	2	8	УО, Т, ЛР
Тема 13. Списки та їх обробка у Visual Prolog	12	2	2	8	УО, Т, ЛР
Тема 14. Створення додатків у середовищі Visual Prolog	14	2	2	10	УО, Т, ЛР
Разом	**180/6**	**28**	**28**	**124**	
Підсумковий контроль – екзамен					

Примітка: УО – опитування; Т – тестування; ЛР – захист лабораторних робіт.

4. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ЛАБОРАТОРНИХ ЗАНЯТЬ І САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо значення навчальної дисципліни «Функціональне та логічне програмування» для своєї професійної підготовки; основних понять та концепцій програмування; місця функціонального та логічного програмування серед домінуючих парадигм; особливостей функціонального програмування. Оволодіння навичками практичного застосування основних команд та конструкцій мови програмування високого рівня LISP.	**РОЗДІЛ 1. ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ** **Тема 1. Загальне уявлення про функціональне програмування та його застосування** *План лекції 1* 1. Домінуючі парадигми програмування. 2. Функціональне та логічне програмування. 3. Концепція функціонального програмування. 4. LISP – мова програмування високого рівня. 5. Особливості функціонального програмування. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11	2	
	Самостійна робота: ознайомлення з основними парадигмами програмування: імперативне, директивне, структурне, непроцедурне, об'єктно-орієнтоване, декларативне програмування; підготовка до лабораторного заняття № 1.	10	3
	Лабораторне заняття № 1 **Основні конструкції мови LISP** **Мета:** ознайомлення та засвоєння основних команд та конструкцій мови LISP; набуття навичок практичного застосування знань. **Завдання:** скласти програму обчислення значення функції згідно з варіантом; засвоїти основні команди та конструкції мови LISP.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо основних понять та базових функцій обробки списків у LISP; розуміння предикативної форми запису функцій; способів опису та виклику нерекурсивних функцій. Набуття практичних навичок застосування базових функцій обробки списків для створення коду програми; перевірка працездатності програми в редакторі LISP.	**Тема 2. Елементарний LISP** *План лекції 2* 1. Основні визначення. 2. Списки як засіб подання знань. 3. Вбудовані функції LISP. 4. Предикативна форма запису функцій. 5. Базові функції обробки списків. 6. Елементарні функції над списками та S-виразами. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11	2	
	Самостійна робота: ознайомлення з основними функціями обробки списків у мові LISP: функції CAR і CDR, функція-конструктор CONS, предикати ATOM, EQ, EQUAL; підготовка до лабораторного заняття № 2.	8	3
	Лабораторне заняття № 2 **Застосування функцій на мові LISP** **Мета:** вивчення базових функцій організації та обробки списків, а також способів опису та виклику нерекурсивних функцій у мові LISP; отримання навичок роботи з інтерпретатором LISP. **Завдання:** вивчити роботу примітивних базових функцій; розробити текст програми згідно варіанту, записати в редакторі LISP та перевірити працездатність програми.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо конструювання списків у LISP; функцій вищого рівня, іменованих та неіменованих функцій; застосування *lambda*-виразів та *nlambda*-виразів. Набуття практичних навичок застосування функцій вищого рівня та структур розгалуження в LISP з використанням конструкції COND; застосування *lambda*-виразів та *nlambda*-виразів та перевірка працездатності програми в редакторі LISP.	**Тема 3. Конструювання списків** *План лекції 3* 1. Конструювання списків у LISP. 2. Функції вищого рівня. 3. Функції виділення елементів списку. 4. Неіменовані функції LISP. 5. Застосування *lambda*-виразів та *nlambda*-виразів. 6. Іменовані функції LISP. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11	2	
	Самостійна робота: ознайомлення з функціями конструювання списків у LISP (CONS, LIST, APPEND), виділення елементів списку (LENGTH, REVERSE, LAST, NTH), функціями DEFUN та LABEL; підготовка до лабораторного заняття № 3.	8	3
	Лабораторне заняття № 3 **Основні конструкції мови LISP. Структури розгалуження** **Мета:** вивчення основних конструкцій мови LISP, функцій розгалуження; отримання навичок використання конструкції COND. **Завдання:** вивчити роботу функцій розгалуження; розробити текст програми з використанням конструкції COND згідно варіанту, записати в редакторі LISP та перевірити працездатність програми.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо принципів застосування основних конструкцій мови LISP; використання числових функцій та керуючих структур. Набуття навичок розробки коду програми із застосуванням структур розгалуження та циклічних обчислень; використання редактора LISP для перевірки працездатності програми.	**Тема 4. Числові функції. Керуючі структури** *План лекції 4* 1. Числові функції в LISP. 2. Арифметичні, логічні, ірраціональні та трансцендентні функції. 3. Тригонометричні функції. 4. Керуючі структури. 5. Структури розгалуження. 6. Циклічні обчислення в LISP. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5	2	
	Самостійна робота: ознайомлення з функціями присвоювання SET, SETQ, LET; функціями розгалуження COND, IF, WHEN і UNLESS; функціями циклічних обчислень DO, DOTIMES, LOOP; підготовка до лабораторного заняття № 4.	8	3
	Лабораторне заняття № 4 **Числові функції в LISP. Циклічні обчислення** **Мета:** вивчення основних конструкцій мови LISP, функцій циклічних обчислень; отримання навичок практичного використання циклічних конструкцій. **Завдання:** вивчити роботу циклічних функцій; розробити текст програми з використанням конструкції з циклічними обчисленнями згідно варіанту, записати в редакторі LISP та перевірити працездатність програми.	2	5

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо застосування простих рекурсій та рекурсій вищих порядків; засвоєння поняття функціонала та розуміння алгоритмів пошуку на LISP. Набуття навичок застосування правил побудови рекурсивних функцій; формулювання умови завершення рекурсії, опису формування результату функції та нових значень аргументу для рекурсивного виклику.	**Тема 5. Поняття рекурсії. Функціонал** *План лекції 5* 1. Прості рекурсії та їх застосування. 2. Правила побудови рекурсивних функцій. 3. Рекурсії вищих порядків. 4. Поняття функціонала. 5. Алгоритми пошуку на LISP. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11	2	
	Самостійна робота: ознайомлення з призначенням рекурсивних функцій, аналіз та порівняння різних їх видів; застосування APPLY, FUNCALL, EVAL; підготовка до лабораторного заняття № 5.	10	3
	Лабораторне заняття № 5 **Застосування простих рекурсивних функцій в мові LISP** **Мета:** ознайомлення та засвоєння основних правил написання рекурсивних функцій у функціональній мові LISP. **Завдання:** виконати завдання згідно з варіантом, використовуючи правила написання рекурсивних функцій; навчитися формулювати умову завершення рекурсії, описувати формування результату функції та нових значень аргументів для рекурсивного виклику.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо застосування діалектів мови LISP; призначення та можливостей Visual LISP в системі AutoCAD. Набуття навичок застосування основних функцій програми на мові AutoLISP; програмування графічних примітивів в системі AutoCAD.	**Тема 6. Застосування діалектів мови LISP** *План лекції 6* 1. Реалізації стандарту Common LISP. 2. Текстовий редактор Emacs. 3. Розробка програм в середовищі Scheme. 4. Використання Visual LISP в системі AutoCAD. 5. Призначення та можливості AutoLISP. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11, 12	2	
	Самостійна робота: аналіз та порівняння практичного застосування діалектів мови LISP; наведення прикладів реалізації Common LISP, Emacs, Scheme, Visual LISP; підготовка до лабораторного заняття № 6.	8	2
	Лабораторне заняття № 6 **Знайомство з середовищем Visual LISP** **Мета:** ознайомлення з особливостями середовища Visual LISP; оволодіння навичками програмування графічних примітивів у системі AutoCAD. **Завдання:** вивчити основні функції та структуру програми на мові AutoLISP; розглянути застосування редактора Visual LISP в системі AutoCAD на простих прикладах, наданих викладачем.	2	5

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо застосування експертних систем; розуміння основних функцій та базових команд CLIPS; розуміння основ методології розробки експертних систем. Закріплення практичних навичок застосування основних функцій програми на мові AutoLISP; програмування графічних примітивів в системі AutoCAD.	**Тема 7. Вирішення задач, заснованих на знаннях** **Набуття практичних навичок програмування графічних примітивів у системі AutoCAD.** *План лекції 7* 1. Області застосування експертних систем. 2. Основи методології розробки експертних систем. 3. Експертні системи реального часу. 4. CLIPS – програмне середовище для розробки ЕС. 5. Основні функції та базові команди CLIPS. 6. Використання CLIPS в експертних системах. **Список рекомендованих джерел:** Основний: 1 Додатковий: 4, 5, 6	2	
	Самостійна робота: ознайомлення з програмним середовищем CLIPS та його практичним застосуванням; використання Бекусо-Наурова (БНФ)-нотації для визначення конструкцій мови CLIPS; підготовка до лабораторного заняття № 7.	10	3
	Лабораторне заняття № 7 **Створення програмного коду в редакторі Visual LISP** **Мета:** набуття досвіду створення програми в редакторі Visual LISP; оволодіння навичками програмування графічних примітивів у системі AutoCAD. **Завдання:** написати програмний код на мові AutoLISP згідно варіанту, виданого викладачем, застосовуючи знання та навички програмування графічних примітивів у редакторі Visual LISP в системі AutoCAD.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо основних конструкцій логічного програмування; розуміння формалізації процесу логічного виведення. Закріплення практичних навичок розв'язання логічних задач засобами логіки висловлювань; створення структури програми на мові Prolog.	**РОЗДІЛ 2. ЛОГІЧНЕ ПРОГРАМУВАННЯ** **Тема 8. Математичні основи логічного програмування** *План лекції 8* 1. Концепція логічного програмування. 2. Основні конструкції логічного програмування. 3. Логіка висловлювань. 4. Розв'язання логічних задач засобами логіки висловлювань. 5. Формалізація процесу логічного виведення. **Список рекомендованих джерел:** Основний: 2 Додатковий: 7, 8	2	
	Самостійна робота: ознайомлення з практичним застосуванням конструкцій логічного програмування та логіки висловлювань; вивчення формул логіки висловлювань та скороченого методу перевірки аргументів; підготовка до лабораторного заняття № 8.	8	2
	Лабораторне заняття № 8 **Структура програми на мові Prolog** **Мета:** вивчення основних конструкції логічного програмування; набуття досвіду розв'язання логічних задач засобами логіки висловлювань. **Завдання:** вивчити основні функції та структуру програми на мові Prolog; виконати завдання згідно з варіантом, використовуючи правила розв'язання логічних задач засобами логіки висловлювань.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо застосування штучного інтелекту; розуміння логіко-лінгвістичних моделей в системах управління. Набуття навичок реалізації логічних задач в робочих середовищах Turbo-Prolog та SWI-Prolog.	**Тема 9. Логічне програмування та штучний інтелект** *План лекції 9* 1. Поняття штучного інтелекту. 2. Логіко-лінгвістичні моделі в системах управління. 3. Штучний інтелект і теорія пошуку висновку. 4. Робоче середовище Turbo-Prolog. 5. SWI-Prolog – реалізація мови програмування Prolog. **Список рекомендованих джерел:** Основний: 2, 3 Додатковий: 7, 8	2	
	Самостійна робота: ознайомлення з напрямками розвитку штучного інтелекту: експертними системами, робототехнікою, автономними агентами, чат-роботами, нейронними мережами; вивчення робочого середовища Turbo-Prolog та SWI-Prolog; підготовка до лабораторного заняття № 9.	10	3
	Лабораторне заняття № 9 **Розв'язок логічних задач** **Мета:** ознайомлення з логіко-лінгвістичними моделями в системах управління та використання штучного інтелекту для розв'язку логічних задач. **Завдання:** відповідно до завдання розглянути приклади реалізації логічних задач в різних робочих середовищах мови програмування Prolog.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо застосування основних елементів Visual Prolog; розуміння понять аргументів і предикатів, стандартних типів даних; різниці між фактами і правилами. Набуття навичок розробки програми з використанням фактів і правил у середовищі Visual Prolog.	**Тема 10. Особливості мови Visual Prolog** *План лекції 10* 1. Основні елементи мови Visual Prolog. 2. Стандартні типи даних. 3. Факти та правила у Visual Prolog. 4. Поняття аргументів та предикатів. 5. Призначення запитів у Prolog. 6. Дерева пошуку рішень. **Список рекомендованих джерел:** Основний: 2, 3 Додатковий: 8, 9 Інтернет-ресурси: 13	2	
	Самостійна робота: ознайомлення з робочим середовищем Visual Prolog, поняттям термів, стандартних типів даних (symbol, string, char, integer, real); підготовка до лабораторного заняття № 10.	8	3
	Лабораторне заняття № 10 **Управління пошуком рішень** **Мета:** ознайомлення з основними елементами мови Visual Prolog; отримання навичок практичного використання фактів і правил у Visual Prolog. **Завдання:** відповідно до завдання розробити текст програми з використанням фактів і правил; записати код у середовищі Visual Prolog та перевірити працездатність програми.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо призначення основних розділів програми Visual Prolog; розуміння семантики програм, директив компілятора. Набуття навичок розробки програми з використанням повторювальних та рекурсивних обчислень; реалізації та перевірки працездатності програми у середовищі Visual Prolog.	**Тема 11. Структура програми Visual Prolog** *План лекції 11* 1. Основні розділи програми Visual Prolog. 2. Директиви компілятора. 3. Призначення розділів Visual Prolog. 4. Семантика програм Visual Prolog. **Список рекомендованих джерел:** Основний: 3 Додатковий: 8, 9 Інтернет-ресурси: 13	2	
	Самостійна робота: практичне ознайомлення з розділами Visual Prolog: CONSTANTS – опис констант; DOMAINS – опис доменів; FACTS – опис предикатів внутрішньої БД; PREDICATES – опис предикатів; CLAUSES – опис тверджень; GOAL – опис внутрішньої цілі; підготовка до лабораторного заняття № 11.	10	3
	Лабораторне заняття № 11 **Повторювальні та рекурсивні обчислення** **Мета:** ознайомлення на практиці з основними розділами програми Visual Prolog; отримання навичок практичного застосування рекурсивних обчислень у Visual Prolog. **Завдання:** відповідно до варіанту розробити текст програми з використанням повторювальних та рекурсивних обчислень; записати код та перевірити працездатність програми у середовищі Visual Prolog.	2	4

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо визначення предикатів; розуміння призначення відсікання та заперечення; застосування структури та реалізації циклу з відкатом. Набуття навичок практичного використання відсікання та заперечення при розробці програми в середовищі Visual Prolog.	**Тема 12. Предикати у Visual Prolog** *План лекції 12* 1. Визначення предикатів. 2. Модулі та їх структура. 3. Відсікання та заперечення. 4. Структура та реалізація циклу з відкатом. 5. Структура та реалізація рекурсії. 6. Предикати другого порядку та анонімні предикати. **Список рекомендованих джерел:** Основний: 3 Додатковий: 8, 9 Інтернет-ресурси: 13	2	
	Самостійна робота: практичне засвоєння понять модуля, структури та предикатів у Visual Prolog; вивчення принципів роботи відсікання: область видимості відсікання, зелені та червоні відсікання, динамічне відсікання; підготовка до лабораторного заняття № 12.	8	3
	Лабораторне заняття № 12 **Застосування відсікання** **Мета:** ознайомлення з основними структурами та предикатами другого порядку; отримання навичок практичного застосування відсікання та заперечення у Visual Prolog. **Завдання:** відповідно до завдання розробити текст програми з використанням відсікання та заперечення; записати код програми та перевірити працездатність у середовищі Visual Prolog.	2	5

Продовження таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань подання списків у Visual Prolog; розуміння застосування предикатів для обробки списків та рекурсії. Набуття навичок розробки програми з використанням предикатів для обробки списків; застосування рекурсії та реалізації програми в середовищі Visual Prolog.	**Тема 13. Списки та їх обробка у Visual Prolog** *План лекції 13* 1. Подання списків у Visual Prolog. 2. Структура списку. 3. Предикати для обробки списків. 4. Застосування рекурсії. 5. Застосування Visual Prolog. **Список рекомендованих джерел:** Основний: 3 Додатковий: 8, 10 Інтернет-ресурси: 13	2	
	Самостійна робота: практичне засвоєння предикатів для обробки списків: підрахунок числа елементів (довжини), додавання та видалення елемента, зчеплення (конкатенація) списків, видалення повторювальних елементів, обернення списків (reverse), перевірки, чи є список паліндромом (palindrom), одержання елемента по номеру у списку (n_element), видалення всіх входжень заданого значення (delete_all, delete_one); підготовка до лабораторного заняття № 13.	8	3
	Лабораторне заняття № 13 **Робота зі списками** **Мета:** вивчення структури списків у Visual Prolog та предикатів для обробки списків; отримання навичок практичного застосування набутих знань. **Завдання:** розробити текст програми з використанням предикатів для обробки списків згідно варіанту, записати код в редакторі Visual Prolog та перевірити працездатність програми.	2	4

Закінчення таблиці

Результати навчання	Навчальна діяльність*	Роб. час студ., год	Оцінювання, бали
Набуття знань щодо створення консольних додатків; застосування графічного інтерфейсу; способів управління за допомогою елементів форми. Набуття практичних навичок застосування графічного інтерфейсу Visual Prolog для створення додатків; вміння довести важливість отриманих знань та навичок у подальшій професійній діяльності.	**Тема 14. Створення додатків у середовищі Visual Prolog** *План лекції 14* 1. Створення консольних додатків. 2. Створення проекту. 3. Введення основного коду в програму. 4. Створення додатків за допомого графічного інтерфейсу. 5. Управління за допомогою елементів форми. **Список рекомендованих джерел:** Основний: 3 Додатковий: 8, 9, 10	2	
	Самостійна робота: ознайомлення з елементами форми, технологіями створення програми в режимі роботи з графічним інтерфейсом та способами управління у Visual Prolog; підготовка до лабораторного заняття № 14; підготовка до тестування.	10	3
	Лабораторне заняття № 14 **Експертні системи у середовищі Visual Prolog** **Мета:** ознайомлення з практичним використанням графічного інтерфейсу Visual Prolog та застосуванням елементів форми для створення додатків. **Завдання:** відповідно до завдання та варіанту створити додаток у середовищі Visual Prolog за допомогою графічного інтерфейсу та елементів форми.	2	5
Разом по дисципліні		**180**	**100**
Підсумковий контроль — екзамен			

***** *всі лабораторні завдання виконуються на основі інтерактивних методів навчання у комп'ютерному середовищі*

5. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Заяць В.М. Логічне та функціональне програмування. Системний підхід : підручник / В.М. Заяць. – Рівне : НУВГП, 2018. – 422 с.
2. Месюра В.І. Математичні основи логічного програмування : навч. посіб. / В.І. Месюра, Н.В. Лисак, О.І. Суприган. – Вінниця : ВНТУ, 2013. – 94 с.
3. Шумейко О.О. Visual Prolog. Опануй на прикладах : навч. посіб. / О.О. Шумейко, В.М. Кнуренко. – Дніпропетровськ : Біла К.О., 2014. – 404 с.

Додатковий

4. Кургаєв О.П. Методи та системи штучного інтелекту : конспект лекцій для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форми навчання / О.П. Кургаєв – К. : НУХТ, 2014. – 279 с.
5. Кургаєв О.П. Методи та системи штучного інтелекту : лабораторний практикум для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форми навчання / О.П. Кургаєв – К. : НУХТ, 2015. – 219 с.
6. Кобець О.В. Методичні вказівки до виконання практичних робіт «Програмування на AutoLISP» з курсу «Основи програмування інженерних систем графіки» для студентів машинобудівних спеціальності «Технологія машинобудування», «Інструментальне виробництво» денної та заочної форм навчання / О.В.Кобець. – Харків: НТУ «ХПІ», 2004. – 55 с.
7. Шекета В.І. Логічне програмування : метод. вказівки / В. І. Шекета. – Івано-Франківськ : ІФНТУНГ, 2009. – 91 с.
8. Марков В.Н. Современное логическое программирование на языке Visual Prolog 7.5: учебник. / В.Н. Марков – СПб.: БХВ-Петербург, 2016. – 544 с.
9. Юрчишин В.М. Пролог – мова логічного програмування : конспект лекцій / В. М. Юрчишин, В. І. Шекета, Л. М. Гобир. – Івано-Франківськ : ІФНТУНГ, 2005. – 138 с.
10. Eduardo Costa. Visual Prolog 7.3 for Tyros. / Eduardo Costa. – New York: Springer-Verlag, 2010. – 270 p.

Інтернет-ресурси

11. Кристиан Кеннек. Интерпретация Лиспа и Scheme. URL: http://blog.ilammy.net/lisp/.
12. Д. Троицкий. Программирование на AutoLISP. URL: http://www.cad.dp.ua/kurs/index.html#CONTENTS.
13. О.П. Солдатова, И.В. Лёзина. Логическое программирование на языке Visual Prolog. URL: https://epdf.tips/-visual-prolog-.html.