

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

СИЛАБУС

ФУНКЦІОНАЛЬНЕ ТА ЛОГІЧНЕ ПРОГРАМУВАННЯ /

FUNCTIONAL AND LOGICAL PROGRAMMING

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

Гарант освітньо-
професійної програми
«Інженерія програмного
забезпечення» другого
(магістерського) рівня
вищої освіти

Київ 2019

Завідувач
кафедри про-
грамної ін-
женерії та
кібербезпеки,
д.т.н., проф.
Дриворучко О.В.

**Розповсюдження і тиражування без офіційного дозволу КНТЕУ
заборонено**

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

Силабус розглянуто та схвалено на засіданні кафедри програмної інженерії та кібербезпеки 10 вересня 2019 р., протокол № 3.

СИЛАБУС

ФУНКЦІОНАЛЬНЕ ТА ЛОГІЧНЕ ПРОГРАМУВАННЯ / FUNCTIONAL AND LOGICAL PROGRAMMING

СИЛАБУС

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

1. Викладач:

1.1. Лектор: Савченко Тетяна Віталіївна,

- кандидат технічних наук, доцент кафедри програмної інженерії та кібербезпеки;
- педагогічний стаж – 21 рік;
- контактний телефон: +38(050)559-70-29;
- e-mail: *savchenko_tv@knu.edu.ua*
- наукові інтереси: інформаційні технології та системи, програмування, оптимізація;
- стажування та підвищення кваліфікації:
 - ✓ Національний авіаційний університет, Інститут аеропортів, кафедра комп'ютерних технологій будівництва, звіт про стажування, тема: «Комп'ютерна графіка», 09.2014 – 10.2014 р.
 - ✓ This is to certify that Tetyana Savchenko successfully completed and received a passing grade in «Build Your Own Chatbot» (CB0103EN, provided by Cognitive Class), a course on bdu.intela-edu.com. Powered by IBM Developer Skills Network. Issued by Intela-EDU. Issued on: April 23, 2019. Authenticity of this certificate can be validated by going to: <https://courses.bdu.intela-edu.com/certificates/e9b3aef3b72f46c4a7a7d1d43e52bda3>

2. Дисципліна: «Функціональне та логічне програмування»:

- рік навчання: 1 рік магістратури;
- семестр навчання: 1;
- кількість кредитів: 6;
- кількість годин за семестр:
 - лекційних: 28 год.;
 - лабораторних: 28 год.;
 - на самостійне опрацювання: 124 год.
- кількість аудиторних годин на тиждень:
 - лекційних: 2 год.;
 - лабораторних: 2 год.

3. Час та місце проведення:

- аудиторні заняття – відповідно до розкладу КНТЕУ, з урахуванням специфіки дисципліни передбачено аудиторіях: 510а, 514.
- поза аудиторна робота – самостійна робота студента, результат виконання якої висвітлено засобами Office 365.

4. Пререквізити та постреквізити навчальної дисципліни:

- **пререквізити:** «Об'єктно-орієнтоване програмування», «Архітектура та проектування програмного забезпечення», «Web-програмування та Web-дизайн», «Алгоритми та структури даних»;
- **постреквізити:** набуття навичок застосування функціонального та логічного програмування (зокрема, мов програмування високого рівня LISP та PROLOG) в майбутній професійній діяльності.

5. Характеристика дисципліни:

5.1. Призначення навчальної дисципліни: 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.

5.2. Мета вивчення дисципліни: 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.

5.3. Задачі вивчення дисципліни: є теоретична та практична підготовка майбутніх фахівців з таких питань:

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

5.4. Зміст навчальної дисципліни: відповідає навчальній та робочій програмі, що визначено науковими досягненнями як вітчизняних, так і закордонних вчених, а також запитам стейкхолдерів.

5.5. План вивчення дисципліни:

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
Набуття знань щодо значення навчальної	РОЗДІЛ 1. ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ		

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
<p>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.</p> <p>Оволодіння навичками практичного застосування основних команд та конструкцій мови програмування високого рівня LISP.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>Тема 1. Загальне уявлення про функціональне програмування та його застосування <i>План лекції 1</i></p> <ol style="list-style-type: none"> 1. Домінуючі парадигми програмування. 2. Функціональне та логічне програмування. 3. Концепція функціонального програмування. 4. LISP – мова програмування високого рівня. 5. Особливості функціонального програмування. <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11</p> <p>Самостійна робота: ознайомлення з основними парадигмами програмування: імперативне, директивне, структурне, непроцедурне, об'єктно-орієнтоване, декларативне програмування; підготовка до лабораторного заняття № 1.</p> <p>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.</p>	<p>2</p> <p>10</p> <p>2</p>	<p>3</p> <p>4</p>
<p>Набуття знань щодо основних понять та базових функцій обробки списків у LISP; розуміння предикативної форми запису функцій; способів</p>	<p>Тема 2. Елементарний LISP <i>План лекції 2</i></p> <ol style="list-style-type: none"> 1. Основні визначення. 2. Списки як засіб подання знань. 3. Вбудовані функції LISP. 4. Предикативна форма запису функцій. 5. Базові функції обробки списків. 	<p>2</p>	

Результати навчання	Навчальна діяльність*	Роб. час студ., год.	Оцінювання, бали
<p>опису та виклику нерекурсивних функцій.</p> <p>Набуття практичних навичок застосування базових функцій обробки списків для створення коду програми; перевірка працездатності програми в редакторі LISP.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>6. Елементарні функції над списками та S-виразами.</p> <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11</p> <p>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.</p>	8	3
	<p>Лабораторне заняття № 2</p> <p>Застосування функцій на мові LISP</p> <p>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.</p> <p>Завдання: вивчити роботу примітивних базових функцій; розробити текст програми згідно варіанту, записати в редакторі LISP та перевірити працездатність програми.</p>	2	4
<p>Набуття знань щодо конструювання списків у LISP; функцій вищого рівня, іменованих та неіменованих функцій; застосування <i>lambda</i>-виразів та <i>nlambda</i>-виразів.</p> <p>Набуття практичних навичок застосування функцій вищого рівня та структур розгалуження в LISP</p>	<p>Тема 3. Конструювання списків <i>План лекції 3</i></p> <ol style="list-style-type: none"> 1. Конструювання списків у LISP. 2. Функції вищого рівня. 3. Функції виділення елементів списку. 4. Неіменовані функції LISP. 5. Застосування <i>lambda</i>-виразів та <i>nlambda</i>-виразів. 6. Іменовані функції LISP. <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11</p> <p>Самостійна робота: ознайомлення з функціями конструювання списків у LISP (CONS, LIST, APPEND), виділення</p>	2	
		8	3

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
<p>з використанням конструкції COND; застосування <i>lambda</i>-виразів та <i>nlambda</i>-виразів та перевірка працездатності програми в редакторі LISP.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>елементів списку (LENGTH, REVERSE, LAST, NTH), функціями DEFUN та LABEL; підготовка до лабораторного заняття № 3.</p> <p>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.</p>	2	4
<p>Набуття знань щодо принципів застосування основних конструкцій мови LISP; використання числових функцій та керуючих структур.</p> <p>Набуття навичок розробки коду програми із застосуванням структур розгалуження та циклічних обчислень; використання редактора LISP для перевірки працездатності програми.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття</p>	<p>Тема 4. Числові функції. Керуючі структури <i>План лекції 4</i></p> <ol style="list-style-type: none"> Числові функції в LISP. Арифметичні, логічні, ірраціональні та трансцендентні функції. Тригонометричні функції. Керуючі структури. Структури розгалуження. Циклічні обчислення в LISP. <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5</p> <p>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.</p> <p>Лабораторне заняття № 4 Числові функції в LISP. Циклічні обчислення Мета: вивчення основних конструкцій мови LISP, функцій циклічних обчислень; отримання навичок</p>	2 8 2	 3 5

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
конструктивної критики, колективний тайм менеджмент	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.		
<p>Набуття знань щодо застосування простих рекурсій та рекурсій вищих порядків; засвоєння поняття функціонала та розуміння алгоритмів пошуку на LISP.</p> <p>Набуття навичок застосування правил побудови рекурсивних функцій; формулювання умови завершення рекурсії, опису формування результату функції та нових значень аргументу для рекурсивного виклику.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>Тема 5. Поняття рекурсії. Функціонал <i>План лекції 5</i></p> <ol style="list-style-type: none"> 1. Прості рекурсії та їх застосування. 2. Правила побудови рекурсивних функцій. 3. Рекурсії вищих порядків. 4. Поняття функціонала. 5. Алгоритми пошуку на LISP. <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11</p> <p>Самостійна робота: ознайомлення з призначенням рекурсивних функцій, аналіз та порівняння різних їх видів; застосування APPLY, FUNCALL, EVAL; підготовка до лабораторного заняття № 5.</p> <p>Лабораторне заняття № 5 Застосування простих рекурсивних функцій в мові LISP</p> <p>Мета: ознайомлення та засвоєння основних правил написання рекурсивних функцій у функціональній мові LISP.</p> <p>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.</p>	<p>2</p> <p>10</p> <p>2</p>	<p>3</p> <p>4</p>
Набуття знань щодо застосування	Тема 6. Застосування діалектів мови LISP	2	

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
<p>діалектів мови LISP; призначення та можливостей Visual LISP в системі AutoCAD.</p> <p>Набуття навичок застосування основних функцій програми на мові AutoLISP; програмування графічних примітивів в системі AutoCAD.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p><i>План лекції 6</i></p> <ol style="list-style-type: none"> 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. <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5 Інтернет-ресурси: 11, 12</p> <p>Самостійна робота: аналіз та порівняння практичного застосування діалектів мови LISP; наведення прикладів реалізації Common LISP, Emacs, Scheme, Visual LISP; підготовка до лабораторного заняття № 6.</p> <p>Лабораторне заняття № 6 Знайомство з середовищем Visual LISP</p> <p>Мета: ознайомлення з особливостями середовища Visual LISP; оволодіння навичками програмування графічних примітивів у системі AutoCAD.</p> <p>Завдання: вивчити основні функції та структуру програми на мові AutoLISP; розглянути застосування редактора Visual LISP в системі AutoCAD на простих прикладах, наданих викладачем.</p>	<p>8</p> <p>2</p>	<p>2</p> <p>5</p>
<p>Набуття знань щодо застосування експертних систем; розуміння основних функцій та базових команд CLIPS; розуміння основ методології розробки експертних систем.</p> <p>Закріплення практичних навичок</p>	<p>Тема 7. Вирішення задач, заснованих на знаннях <i>План лекції 7</i></p> <ol style="list-style-type: none"> 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. 	<p>2</p>	

Результати навчання	Навчальна діяльність*	Роб. час студ., год.	Оцінювання, бали
<p>застосування основних функцій програми на мові AutoLISP; програмування графічних примітивів в системі AutoCAD.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>6. Використання CLIPS в експертних системах.</p> <p>Список рекомендованих джерел: Основний: 1 Додатковий: 4, 5, 6</p> <p>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.</p> <p>Лабораторне заняття № 7 Створення програмного коду в редакторі Visual LISP</p> <p>Мета: набуття досвіду створення програми в редакторі Visual LISP; оволодіння навичками програмування графічних примітивів у системі AutoCAD.</p> <p>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.</p>	<p>10</p> <p>2</p>	<p>3</p> <p>4</p>
<p>Набуття знань щодо основних конструкцій логічного програмування; розуміння формалізації процесу логічного виведення.</p> <p>Закріплення практичних навичок розв'язання логічних задач засобами логіки висловлювань; створення структури програми</p>	<p>РОЗДІЛ 2. ЛОГІЧНЕ ПРОГРАМУВАННЯ Тема 8. Математичні основи логічного програмування <i>План лекції 8</i></p> <ol style="list-style-type: none"> 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. <p>Список рекомендованих джерел: Основний: 2 Додатковий: 7, 8</p>	<p>2</p>	<p>2</p>

Результати навчання	Навчальна діяльність*	Роб. час студ., год.	Оцінювання, бали
<p>на мові Prolog. SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>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.</p> <p>Лабораторне заняття № 8 Структура програми на мові Prolog Мета: вивчення основних конструкцій 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.</p>	<p>8</p> <p>2</p>	<p>4</p>
<p>Набуття знань щодо застосування штучного інтелекту; розуміння логіко-лінгвістичних моделей в системах управління.</p> <p>Набуття навичок реалізації логічних задач в робочих середовищах Turbo-Prolog та SWI-Prolog.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>Тема 9. Логічне програмування та штучний інтелект <i>План лекції 9</i></p> <ol style="list-style-type: none"> 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. <p>Список рекомендованих джерел: Основний: 2, 3 Додатковий: 7, 8</p> <p>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.</p> <p>Лабораторне заняття № 9</p>	<p>2</p> <p>10</p> <p>2</p>	<p>3</p> <p>4</p>

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
	<p align="center">Розв'язок логічних задач</p> <p>Мета: ознайомлення з логіко-лінгвістичними моделями в системах управління та використання штучного інтелекту для розв'язку логічних задач.</p> <p>Завдання: відповідно до завдання розглянути приклади реалізації логічних задач в різних робочих середовищах мови програмування Prolog.</p>		
<p>Набуття знань щодо застосування основних елементів Visual Prolog; розуміння понять аргументів і предикатів, стандартних типів даних; різниці між фактами і правилами.</p> <p>Набуття навичок розробки програми з використанням фактів і правил у середовищі Visual Prolog.</p> <p>SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики.</p>	<p align="center">Тема 10. Особливості мови Visual Prolog</p> <p align="center"><i>План лекції 10</i></p> <ol style="list-style-type: none"> 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. <p>Список рекомендованих джерел: Основний: 2, 3 Додатковий: 8, 9 Інтернет-ресурси: 13</p> <p>Самостійна робота: ознайомлення з робочим середовищем Visual Prolog, поняттям термів, стандартних типів даних (symbol, string, char, integer, real); підготовка до лабораторного заняття № 10.</p> <p align="center">Лабораторне заняття № 10</p> <p align="center">Управління пошуком рішень</p> <p>Мета: ознайомлення з основними елементами мови Visual Prolog; отримання навичок практичного використання фактів і правил у Visual Prolog.</p> <p>Завдання: відповідно до завдання розробити текст програми з використанням фактів і правил; записати код у середовищі Visual Prolog та перевірити працездатність програми.</p>	<p align="center">2</p> <p align="center">8</p> <p align="center">2</p>	<p align="center">3</p> <p align="center">4</p>

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
<p>Набуття знань щодо призначення основних розділів програми Visual Prolog; розуміння семантики програм, директив компілятора.</p> <p>Набуття навичок розробки програми з використанням повторювальних та рекурсивних обчислень; реалізації та перевірки працездатності програми у середовищі Visual Prolog.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики.</p>	<p>Тема 11. Структура програми Visual Prolog <i>План лекції 11</i></p> <ol style="list-style-type: none"> 1. Основні розділи програми Visual Prolog. 2. Директиви компілятора. 3. Призначення розділів Visual Prolog. 4. Семантика програм Visual Prolog. <p>Список рекомендованих джерел: Основний: 3 Додатковий: 8, 9 Інтернет-ресурси: 13</p> <p>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.</p> <p>Лабораторне заняття № 11 Повторювальні та рекурсивні обчислення</p> <p>Мета: ознайомлення на практиці з основними розділами програми Visual Prolog; отримання навичок практичного застосування рекурсивних обчислень у Visual Prolog.</p> <p>Завдання: відповідно до варіанту розробити текст програми з використанням повторювальних та рекурсивних обчислень; записати код та перевірити працездатність програми у середовищі Visual Prolog.</p>	<p>2</p> <p>10</p> <p>2</p>	<p>3</p> <p>4</p>
<p>Набуття знань щодо визначення предикатів; розуміння призначення відсікання та заперечення; застосування</p>	<p>Тема 12. Предикати у Visual Prolog <i>План лекції 12</i></p> <ol style="list-style-type: none"> 1. Визначення предикатів. 2. Модулі та їх структура. 3. Відсікання та заперечення. 4. Структура та реалізація циклу з відкатом. 5. Структура та реалізація рекурсії. 	<p>2</p>	

Результати навчання	Навчальна діяльність*	Роб.час студ., год.	Оцінювання, бали
<p>структури та реалізації циклу з відкатом.</p> <p>Набуття навичок практичного використання відсікання та заперечення при розробці програми в середовищі Visual Prolog.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики.</p>	<p>6. Предикати другого порядку та анонімні предикати.</p> <p>Список рекомендованих джерел: Основний: 3 Додатковий: 8, 9 Інтернет-ресурси: 13</p> <p>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.</p> <p>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.</p>	<p>8</p> <p>2</p>	<p>3</p> <p>5</p>
<p>Набуття знань подання списків у Visual Prolog; розуміння застосування предикатів для обробки списків та рекурсії.</p> <p>Набуття навичок розробки програми з використанням предикатів для обробки списків; застосування рекурсії та реалізації програми</p>	<p>Тема 13. Списки та їх обробка у Visual Prolog <i>План лекції 13</i></p> <ol style="list-style-type: none"> 1. Подання списків у Visual Prolog. 2. Структура списку. 3. Предикати для обробки списків. 4. Застосування рекурсії. 5. Застосування Visual Prolog. <p>Список рекомендованих джерел: Основний: 3 Додатковий: 8, 10 Інтернет-ресурси: 13</p> <p>Самостійна робота: практичне засвоєння предикатів для обробки списків: підрахунок числа елементів (довжини), додавання та видалення</p>	<p>2</p> <p>8</p>	<p>3</p>

Результати навчання	Навчальна діяльність*	Роб. час студ., год.	Оцінювання, бали
<p>в середовищі Visual Prolog.</p> <p>SoftSkills:</p> <p>комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент</p>	<p>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.</p> <p style="text-align: center;">Лабораторне заняття № 13 Робота зі списками</p> <p>Мета: вивчення структури списків у Visual Prolog та предикатів для обробки списків; отримання навичок практичного застосування набутих знань.</p> <p>Завдання: розробити текст програми з використанням предикатів для обробки списків згідно варіанту, записати код в редакторі Visual Prolog та перевірити працездатність програми.</p>	<p style="text-align: center;">\</p> <p style="text-align: center;">2</p>	<p style="text-align: center;">4</p>
<p>Набуття знань щодо створення консольних додатків; застосування графічного інтерфейсу; способів управління за допомогою елементів форми.</p> <p>Набуття практичних навичок застосування графічного інтерфейсу Visual Prolog для створення додатків; вміння довести важливість отриманих знань та навичок у подальшій професійній</p>	<p style="text-align: center;">Тема 14. Створення додатків у середовищі Visual Prolog <i>План лекції 14</i></p> <ol style="list-style-type: none"> 1. Створення консольних додатків. 2. Створення проекту. 3. Введення основного коду в програму. 4. Створення додатків за допомогою графічного інтерфейсу. 5. Управління за допомогою елементів форми. <p>Список рекомендованих джерел: Основний: 3 Додатковий: 8, 9, 10</p> <p>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.</p> <p style="text-align: center;">Лабораторне заняття № 14 Експертні системи у середовищі</p>	<p style="text-align: center;">2</p> <p style="text-align: center;">10</p> <p style="text-align: center;">2</p>	<p style="text-align: center;">3</p> <p style="text-align: center;">5</p>

Результати навчання	Навчальна діяльність*	Роб. час студ., год.	Оцінювання, бали
діяльності. SoftSkills: комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики.	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.		
Разом по дисципліні		180	100
Підсумковий контроль — екзамен			

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

6. 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 спеціальностей «Технологія машинобудування», «Інструментальне виробництво» денної та заочної форм навчання / О.В.Кобець. – Харків: НТУ «ХП», 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. Інтерпретація 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>.

7. **Контроль та оцінювання результатів навчання:** положення про оцінювання результатів навчання студентів і аспірантів наказ КНТЕУ № 2891 від 16.09.2019 р.

Під час вивчення дисципліни «Функціональне та логічне програмування» викладачем здійснюється поточний та підсумковий контроль. Поточний контроль та оцінювання передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (тестування за матеріалами лекції, який здійснюється на початку кожної наступної лекції);
- захист лабораторних робіт (проходить під час наступної лабораторної роботи);
- перевірка засвоєння матеріалу, що винесений на самостійне опрацювання під час фронтального опитування на лекції.

8. **Політика навчальної дисципліни:**

8.1. Відвідування лекційних та лабораторних занять: відвідування лекційних та лабораторних занять є обов'язковим. Допускаються пропуски занять з таких поважних причин, як хвороба (викладачу надається копія довідки від медичного закладу), участь в олімпіаді, творчому конкурсі тощо за попередньою домовленістю та згодою викладача за умови дозволу деканату (надаються документи чи інші матеріали, які підтверджують заявлену участь у діяльності студента).

8.2. Відпрацювання пропущених занять: відпрацювання пропущених занять є обов'язковим незалежно від причини пропущеного заняття. Лекційне заняття має бути відпрацьоване до наступної лекції на консультації викладача з використанням ПЗ 365 Office Teams. Відпрацювання лекційного матеріалу передбачає вивчення пропущеного теоретичного матеріалу та складання тесту за цим матеріалом. Лабораторне заняття відпрацьовується під час консультації викладача (розклад консультацій на сайті).

- 8.3. **Правила поведінки під час занять:** обов'язковим є дотримання техніки безпеки в комп'ютерних лабораторіях. Студенти повинні приймати активну участь в обговоренні навчального матеріалу, ознайомившись з ним напередодні (навчальний матеріал надається викладачем).
- 8.4. **За порушення академічної доброчесності** студенти будуть притягнені до такої академічної відповідальності:
- повторне проходження оцінювання (контрольна робота, іспит, залік тощо);
 - повторне проходження навчального курсу.