

**ПЛАН
ЛАБОРАТОРНИХ ЗАНЯТЬ**

з дисципліни «**ФОРМАЛЬНІ МЕТОДИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**»
лабораторні заняття, годин – 14
Викладач – Блажко О.А.

Обсяг в годинах	Назва та стислий зміст роботи	Мета роботи
ЗМІСТОВИЙ МОДУЛЬ 2 – ДЕКЛАРАТИВНЕ ПРОГРАМУВАННЯ		
2	<p>Заняття 1. Декларативне програмування на основі регулярних виразів, у мові шляхів/виразів <i>Xpath</i> та запитів <i>Xquery</i></p> <p>1. Програмування обмежень цілісності БД на основі регулярних виразів з використанням синтаксису функції СКБД <i>Oracle, PostgreSQL</i></p> <p>2. Програмування функцій обробки текстових документів з використанням регулярних виразів</p> <p>3. Обробка 1є протирічливих даних мовами шляхів/виразів <i>Xpath</i> та запитів <i>Xquery</i>.</p>	<p>Навчитися декларативно програмувати бізнес-логіку програм на мові регулярних виразів на мові шляхів/виразів <i>Xpath</i> та запитів <i>Xquery</i>.</p> <p>Розуміти особливості вимог до програмного продукту, які визначають ефект від використання декларативного програмування.</p> <p>Розвинути навички декларативного програмування</p>
2	<p>Заняття 2. Основи генеративного програмування</p> <p>1. Проектування фізичної структури реляційних баз даних в стандарті <i>SQL</i> на основі семантичних моделей <i>UML</i> та мови обмежень цілісності <i>OCL</i>.</p> <p>2. Дослідження можливостей пакету <i>Dresden-OCL</i></p> <p>3. Розробка програми-генератора команд створення структури БД з урахуванням особливості СКБД <i>Oracle, PostgreSQL, MySQL</i></p>	<p>Навчитися проектувати реляційну базу даних в стандарті <i>SQL</i> на основі семантичних моделей <i>UML</i> та мови обмежень цілісності <i>OCL</i> з урахуванням особливості різних СКБД.</p> <p>Розуміти переваги генеративного програмування.</p> <p>Розвинути навички генеративного програмування.</p>
ЗМІСТОВИЙ МОДУЛЬ 3 – ДЕСКРИПЦІЙНА ЛОГІКА У СЕМАНТИЧНІ МЕРЕЖАХ ТА ОНТОЛОГІЯХ		
2	<p>Заняття 3 Генеративне програмування на основі семантичних мереж та онтологій</p> <p>1. Проектування фізичної структури реляційних баз даних обмеженнями цілісності у <i>SQL</i> на основі с семантичних мереж та онтологій моделей <i>OWL</i>.</p> <p>2. Розробка програми-генератора команд створення структури БД з урахуванням особливості СКБД <i>Oracle, PostgreSQL, MySQL</i></p> <p>3. Верифікація 1є протиріч обмежень цілісності.</p>	<p>Навчитися проектувати реляційну базу даних в стандарті <i>SQL</i> на основі семантичних мереж та онтологій моделей <i>OWL</i>.</p> <p>Розуміти переваги генеративного програмування на основі семантичних мереж.</p> <p>Розвинути навички використання семантичних мереж для вирішення виробничих завдань.</p>
ЗМІСТОВИЙ МОДУЛЬ 4 – Автоматне програмування та формальна верифікація		
2	<p>Заняття 4. Основи автоматного програмування</p> <p>1. Перетворення алгоритмічного представлення виконання задачі в автоматну форму.</p> <p>2. Включення автоматної форми виконання задачі у алгоритмічний каркас.</p> <p>3. Верифікація роботи автоматної програми</p>	<p>Навчитися проектувати автоматну модель на основі алгоритмічного представлення.</p> <p>Розуміти переваги автоматного програмування у процесі верифікації.</p> <p>Розвинути навички використання автоматного програмування.</p>

Обсяг в годинах	Назва та стислий зміст роботи	Мета роботи
ЗМІСТОВИЙ МОДУЛЬ 5 - ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ		
2	Заняття 5. Функціональне програмування мовою Haskell 1. Дослідження прикладів програмного коду мови Haskell та алгоритмічний опис дій. 2. Перетворення алгоритму у функціональне представлення мови Haskell	Навчитися аналізувати програмний код мови Haskell та програмувати на функціональній мові Haskell. Розуміти переваги функціонального програмування при аналізі функціональних вимог до програмного продукту. Розвинути навички використання функціонального програмування.
2	Заняття 6. Функціональне програмування в мовах загального призначення 1. Дослідження особливостей функціональної мови TypeScript як розширення мови JavaScript. 2. Програмування веб-форм з використанням TypeScript	Навчитися включати у загальні мови алгоритмічного програмування програмний код функціональної мови Розуміти переваги включення програмних модулів на мовах функціонального програмування. Розвинути навички використання функціонального програмування у звичайних алгоритмічних мовах.
2	Заняття 7. Основи реактивного програмування 1. Дослідження особливостей реактивного програмування 2. Дослідження особливостей функціонального реактивного програмування 3. Перетворення алгоритму у реактивне представлення мови.	Навчитися включати у загальні мови алгоритмічного програмування програмний код реактивної мови Розуміти переваги включення програмних модулів на мовах реактивного та реактивно-функціонального програмування. Розвинути навички використання реактивного та реактивно-функціонального програмування у звичайних алгоритмічних мовах.