КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

КАФЕДРА ТЕОРІЇ ТА ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

«ЗАТВЕРДЖУЮ»
Заступник декана
з навчальної роботи
_____ Кашпур О.Ф.
«28» 09 2020 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
"ПАРАДИГМИ ТА ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ"
для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)
освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Інформатика»**
(назва освітньої програми)
вид дисципліни **вибіркова**
вибірковий блок **«Теорія та технологія програмування»**

Форма навчання	**заочна**
Навчальний рік	**2020/2021**
Семестр	**7**
Кількість кредитів ECTS	**4**
Мова викладання, навчання та оцінювання	**українська**
Форма заключного контролю	**іспит**

Пролонговано: на 2021/2022 н.р. _____ (_____) «28» 02 2021 р.
(підпис, ПІБ, дата)
на 20__/20__ н.р. _____ (_____) «___» ___ 20__р.
(підпис, ПІБ, дата)

КИЇВ – 2020

Розробник: Кузенко Володимир Федорович, к.ф.-м.н., доцент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО
Зав. кафедри теорії та технології програмування
_____ (Нікітченко М.С.)
(підпис) (прізвище та ініціали)

Протокол № 1 від «28» серпня 2020 р.

Схвалено Гарантом освітньо-професійної програми «Інформатика»
_____(Омельчук Л.Л.)
(підпис)
«28» серпня 2020 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «28» серпня 2020 року №1
Голова науково-методичної комісії _____ (Омельчук Л.Л.)
(підпис) (прізвище та ініціали)

«28» серпня 2020 року

1. Мета дисципліни – засвоєння базових знань щодо таких парадигм програмування: як функціональна, логічна, декларативної у порівнянні об'єктно-орієнтованою парадигмою. Оволодіння навичками проектування та розробки програмних систем із застосуванням різноманітних парадигм програмування з відповідними структурами даних, механізмами управління та технологіями.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (*за наявності*):

1. Знати: загальні поняття мов програмування (синтаксис, семантика програм, об'єктно-орієнтований стиль програмування), основи теорії алгоритмів (рекурсивні функції та рекурсивні обчислення), основні поняття математичної логіки, зокрема, поняття резольвенти, резолютивного виводу.

2. Вміти: використовувати різноманітні середовища розробки, редагування та виконання програм.

3. Володіти елементарними навичками: специфікації та програмування рекурсивних обчислень.

3. Анотація навчальної дисципліни (*до 700 символів*):

Навчальна дисципліна "Парадигми та технології програмування" є складовою програми підготовки фахівців за першим (*бакалаврським*) рівнем вищої освіти *галузі знань* 12 „Інформаційні технології" зі *спеціальності 122* „Комп'ютерні науки", *освітньо-професійної програми* –„Інформатика", вибірковий блок «Теорія та технологія програмування».

Дана дисципліна є навчальною дисципліною за вибором освітньо-професійної програми ***"Інформатика"***.

Викладається у 7 семестрі 4 курсу в **обсязі – 120 год.**

(4 кредити ECTS) зокрема: *лекції – 7 год., лабораторні заняття - 3 год., консультації – 1 год., самостійна робота – 109 год.* У курсі передбачено **3** *частини*. Завершується дисципліна – іспитом.

В результаті вивчення навчальної дисципліни студент повинен

знати: суть і підґрунтя парадигми логічного програмування та парадигми функціонального програмування, основні конструкції мов програмування *Prolog* та *Haskell*, суть декларативних засобів для об'єктно-орієнтованої парадигми;

вміти:

- застосовувати логічний та функціональний стилі програмування при розв'язуванні програмістських задач;
- використовувати відомі та створювати власні декларативні засоби програмування стосовно об'єктно-орієнтованої парадигми, зокрема, для платформ *Java* та *.Net*.

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетенцій) на рівні сучасних досягнень у програмуванні відповідно до кваліфікації фахівець з інформаційних технологій. Зокрема, розвивати:

- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність до алгоритмічного та логічного мислення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	**Результат навчання**			
1.1	Знати суть і підгрунтя парадигми функціонального програмування, знати засоби підтримки функціонального стилю програмування та їх подання у мові програмування *Haskell*.	Лекції, самостійна робота	Контрольна робота, іспит	20%
1.2	Знати суть і підгрунтя парадигми логічного програмування, знати засоби підтримки логічного стилю програмування.	Лекції, самостійна робота	Контрольна робота, іспит	20%
1.3	Знати суть декларативних засобів стосовно об'єктно-орієнтованої парадигми, знати засоби декларативного програмування для *.Net* та *Java*.	Лекції, самостійна робота	Контрольна робота, іспит	10%
2.1	Вміти застосовувати функціональний стиль програмування, використовуючи мову програмування *Haskell*.	Лекції, самостійна робота	Контрольна робота, іспит	20%
2.2	Вміти застосовувати логічне програмування, використовуючи Prolog-процесори, зокрема SWI-Prolog.	Лекції, самостійна робота	Контрольна робота, іспит	20%
2.3	Вміти використовувати декларативні засоби програмування для *.Net* та *Java*.	Лекції, самостійна робота	Контрольна робота, іспит	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни **Програмні результати навчання**	**1.1**	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**
(з опису освітньої програми)						
ПРН16. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення			+			+
ПРН18.2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, парадигми, технології, алгоритмічні і програмні рішення при проектуванні та розробці програмних систем.	+	+	+	+	+	+

Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота : РН 1.1, РН 2.1 — 24/14,4 балів.

2. Контрольна робота: РН 1.2, РН 2.2 — 24/14,4 балів.

3. Контрольна робота: РН 1.3, РН 2.3 — 12/7,2 балів.

- підсумкове оцінювання (у формі іспиту)*:*

- максимальна кількість балів, які можуть бути отримані студентом: 40 балів;

- результати навчання, які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН2.2, РН2.3;

- форма проведення і види завдань: письмова робота.

Структура екзаменаційної роботи та критерії оцінювання:

2 теоретичних питання (по 8 балів), 2 задачі (по 12 балів)

Запитання для підготовки до іспиту

- Парадигма функціонального програмування. Мова *Haskell*. Типи даних (елементарні та складені).
- Мова *Haskell*. Функції. Типи функцій, каррінг.
- Поліморфні типи та поліморфні функції. Параметричний поліморфізм. Виведення типів.
- Визначення функцій рівняннями. Зіставлення зі зразком. Приклади.
- Визначення функцій рівняннями. Рекурсивні функції. Приклади.
- Мова *Haskell*. Інфіксні оператори та бінарні функції. Операторна форма бінарних функцій. Функціональна форма інфіксних операторів .
- Мова *Haskell*. Умовні конструкції при визначенні функцій. Використання двовимірної структуризації.
- Каррінгові та кортежні функції. Функції як аргументи. Функції вищих порядків *curry* та *uncurry.*
- Мова *Haskell*. Типи користувача. Конструктори. Приклади.
- Рекурсивні типи користувача. “Власний” списковий тип.
- Рекурсивні типи користувача. Тип *Tree.*
- Ліниві та енергійні обчислення. Приклади.
- Використання нескінчених списків. Списки арифметичних послідовностей.
- Ліниві та енергійні обчислення. Використання нескінчених списків на прикладі списку простих чисел.
- Ліниві та енергійні обчислення. Використання нескінчених списків на прикладі списку чисел Фібоначчі.
- *List comprehension.* Приклади. Функція швидкого сортування.
- Функції вищих порядків. Приклади (*map*, *filter*, *zip*, *zipWith*).
- Функції вищих порядків. Приклади (*foldl*, *foldl*1 *foldr*, *foldr*1).
- Мова *Haskell*. Класи типів. Клас типів рівності (*Eq*). Розширення класів (*class extension*). Клас типів *Ord*. Варіанти втілення класів *Eq* та *Ord* у типі *Tree.*
- Мова *Haskell*. Похідні втілення класів типів. Похідні втілення *Eq* та *Ord* у типі *Tree.*
- Мова *Haskell*. Клас типів *Show*. Варіанти втілення класу *Show* у типі *Tree*.
- Суперпозиція функцій та композиція (.) у мові *Haskell.*
- Композиції та монадні композиції. Оператор монадної композиції (>=>).
- Оператори аплікації. Оператор зв’язування *bind* (>>=).

- Клас типів *Monad.* Монадна композиція та приємна (прозора) версія монадних законів. Дві версії монадних законів.
- Інтеграція монадних функцій зі «звичайними» функціями. Монадна функція *return.*
- Не всюди визначені функції. Тип *Maybe.* Монада *Maybe* (монадне втілення *Maybe*).
- Компонування *Maybe* монадних функцій. Приклади. Монада *Maybe* та монадні закони.
- Інтеграція *Maybe* монадних функцій зі «звичайними» функціями. Приклади.
- *Do*-нотація для монадних функцій. Перетворення *do*-нотації до виразу з монадною композицією. Приклади.
- Ввід-вивід. *IO* монада. Деякі функції вводу-виводу. Приклади.
- Оператор монадної послідовності (>>). Приклади використання у випадку *IO* монади.
- Ввід-вивід. Монадна функція *return.* Приклади.
- Скулемівські нормальні форми, множини диз'юнктів. Метод резолюцій для логіки висловлювань.
- Уніфікація. Алгоритм уніфікації. Метод резолюцій для логіки предикатів.
- Чистий (недетермінований) Пролог. Синтаксис. Логічна (декларативна) семантика.
- Чистий (недетермінований) Пролог. Операційна семантика. Приклади.
- Пролог. Списковий тип. Приклади.
- Приклади використання Прологу (задача про 8 ферзів, родинні зв'язки).
- Приклади використання Прологу (задача Ейнштейна).
- Пролог-процесори. Огляд особливостей (стратегія обчислень, відкати).
- Пролог-процесори. Техніка програмування (управління відкатами, моделювання циклів, "повтори" на основі предиката *repeat*, повтори з лічильником).
- Моделювання стеків та черг.
- Експертні системи та стратегії пошуку відповіді. Експертні системи із прямою стратегією.
- Експертні системи з оберненою стратегією. Подання знань у вигляді правил Прологу.
- Подання знань в експертних системах. Мережі знань.
- Експертні системи з поясненнями. Питання "Навіщо?" ("*Why*?").
- Експертні системи з поясненнями. Питання "Як?" ("*How*?").
- Декларативні засоби Java-програмування на прикладі веб-служб.
- Декларативні засоби *.Net* програмування на прикладі веб-служб.
- Веб-служби (*Web Services*) та сервісно-орієнтована архітектура (СОА). Стандарти веб-служб. Протокол *SOAP*. Конверт *SOAP*-повідомлення.
- Документування веб-служб: генерація документації для сприйняття людиною (з використанням веб-браузерів), генерація документації, орієнтованої на використання програмами – *wsdl*-файли.
- Розробка *.Net* веб-служб. Тест-форми веб-служб.
- Розробка клієнтських *.Net* програм для веб-служб. Утиліта *Wsdl.exe*.
- Розробка веб-служб на *Java*.
- Розробка клієнтських *Java* -програм для веб-служб.

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів. Студент допускається до іспиту за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота : РН 1.1, РН 2.1: до останнього лекційного заняття.
2. Контрольна робота: РН 1.2, РН 2.2: до останнього лекційного заняття.
3. Контрольна робота: РН 1.3, РН 2.3: до останнього лекційного заняття.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб. занят.	Самост. робота
Частина 1. Парадигма функціонального програмування.				
1.	**Тема 1.** Основні концепції функціональної парадигми. Мова *Haskell*. Елементарні типи даних. Вбудовані складені типи (списки, кортежі, функції).	1		3
2.	**Тема 2.** Функції, типи функцій. Визначення функцій рівняннями, застосування функцій. Автоматичне виведення типів. Функції від кількох аргументів. Карінгові функції.	1		3
3.	**Тема 3.** Визначення функцій кількома рівняннями. Рекурсивні функції та теорема про нерухому точку.			3
4.	**Тема 4.** Поліморфні типи та поліморфні функції. Параметричний поліморфізм.			3
5.	**Тема 5.** Лямбда-функції у *Haskell*. Функції вищих порядків.			3
6.	**Тема 6.** Ліниві та енергійні обчислення. Нескінчені списки. Списки арифметичних послідовностей.			4
7.	**Т**			4
8.	**Тема 8.** Поняття класів у *Haskell.* Поліморфізм класів.			4
9.	**Тема 9.** Поняття монадних типів, монадних функцій та клас типів *Monad*.			4

10.	**Тема 10.** Поняття монад на прикладі монади *Maybe*. Спискова монада.			4
11.	**Тема 11.** Проблеми вводу-виводу та монада *IO Monad `*			4
	Контрольна робота 1			10
	Частина 2. Парадигма логічного програмування			
12.	**Тема 12.** "Чистий" (недетермінований) Пролог. Синтаксис, логічна семантика.	1		4
13.	**Тема 13.** Операційна семантика Пролог-програм та метод резолюцій.	1		4
14.	**Тема 14.** Процесори для мови Пролог. Поняття відкатів та їх використання. Управління відкатими.			4
15.	**Тема 15.** Специфіка техніки програмування у Прологі.			4
16.	**Тема 16.** Використання Прологу для прикладних задач (синтаксичні аналізатори, експертні системи).			4
17.	**Тема 17.** Експертні системи з поясненнями. Експертні системи в умовах неповної визначеності.	1		4
	Контрольна робота 2			10
	Частина 3. Парадигма об'єктно-орієнтованого програмування та декларативні засоби програмування			
18.	**Тема 18.** Об'єкти та метадані. Рефлексія об'єктів. Декларативний стиль програмування у *Java*. Анотації.	1		4
19.	**Тема 19.** Анотації, розробка анотацій. Приклад.	1		4
20.	**Тема 20.** Застосування декларативних засобів програмування у *Java* та *.Net* на прикладі технології веб-сервісів *SOAP*.			4
21.	**Тема 21.** Використання архітектурного стилю *Rest* на платформах *Java* та *.Net*.			4
	Контрольна робота 3			10
	Консультація - 1 години			
	Іспит			
	ВСЬОГО	7	3	109

Загальний обсяг ***120*** *год.,* в тому числі:
Лекцій **–** ***7*** *год.*
Лабораторні – **3** *год.*
Консультації – ***1*** *год.*
Самостійна робота **- 109** *год.*

9. Рекомендовані джерела:

Основні:

1. https://classroom.google.com/u/0/c/MjY4Mjg5NjUzMzQw - презентації та умови завдань лабораторного практикуму.
2. Lipovača M. Learn You a Haskell for Great Good! Miran Lipovača. : No Starch Press», 2011, 383p. (Липовача М. Изучай Haskell во имя добра! М.: ДМК Пресс, 2012, 490с.).
3. Bratko I., Prolog Programming for Artificial Intelligence (4th Edition), 2011, 442p. (Братко И. Программирование на языке PROLOG для искусственного интеллекта. М.: Мир, 1990, 559с..
4. Марселлус Д. Программирование экспертных систем на Турбо-Прологе, М., Финансы и статистика, 1994. 410с.
5. Snell J., Tidwell D., Kulchenko P., Programming Web Services With SOAP, 2002, 237p.
6. O'Sullivan B., Goerzen J., Stewart D. Real World Haskell 1st Edition, 2008, 671p.

Додаткові:

1. Барендрегт Х. Ламбда-исчисление. Его синтаксис и семантика, 1985, 606с.
2. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983, 360с.
3. Ин Ц., Соломон Д. Использование Турбо-Пролога. М.: Мир, 1990.
4. Стерлинг Л., Шапиро Э. Искусство программирования на языке Пролог. М.: Мир, 1990.
5. Стобо Дж. Язык программирования Пролог. М.: Радио и связь, 1993.
6. Sahay R., Pragmatic WCF Paperback, 2015, 329p.
7. Клоксин У., Меллиш К. Программирование на языке пролог. М.: Мир, 1987.
8. Янсон А. Турбо-Пролог в сжатом изложении. М.: Мир, 1991.
9. Филд А., Харрисон П. Функциональное программирование. М.: Мир, 1993.
10. Хендерсон П. Функциональное программирование. Применение и реализация. М.: Мир, 1983.
11. Хьюдак П., Петерсон Дж., Джозеф Фасел Дж. Мягкое введение в Haskell.
12. Машнин Т.С. Web-сервисы Java, БХВ-Петербург, 2012, 560 с.