

**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.

« ____ » _____ 2018 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**

Форма навчання	денна
Навчальний рік	2018/2019
Семестр	6
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	екзамен

Викладачі: **к.ф.-м.н., доц. Кузенко В.Ф.**

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2018

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

ЗАТВЕРДЖЕНО

В. о. зав. кафедри «Теорії та технології програмування»

_____ (Панченко Т.В.)
(підпис) (прізвище та ініціали)

Протокол № ____ від «____» _____ 20__ р.

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

Протокол від «____» _____ 20__ року № ____

Голова науково-методичної комісії _____ (Хусаїнов Д.Я.)
(підпис) (прізвище та ініціали)

«____» _____ 20__ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 20__ року № ____

Голова Вченої ради _____ Анісімов А.В.
(підпис) (прізвище та ініціали)

ВСТУП

1. Мета дисципліни – засвоєння базових знань щодо використання різних парадигм програмування: функціональної, логічної, декларативної, об'єктно-орієнтованої. 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 програми, 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:* програмування чи специфікації програм, що ґрунтуються на рекурсивних обчисленнях.

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”.

Дана дисципліна є навчальною дисципліною за вибором освітньо-професійної програми “Інформатика”.

Викладається у 6 семестрі 3 курсу в **обсязі – 120 год.**

(4 кредити ECTS) зокрема: *лекції – 38 год., лабораторні – 16 год., консультації – 2 год., самостійна робота – 64 год.* У курсі передбачено **3 частини** та **3 контрольні роботи (тести)**. Завершується дисципліна – екзаменом.

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 стосовно 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, на у Java та .Net.

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетенцій) на рівні сучасних досягнень у програмуванні відповідно до кваліфікації фахівців 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, лабораторні роботи	Тест, екзамен	20%
1.2	Знати суть і підгрунтя парадигми логічного програмування, знати засоби підтримки логічного стилю програмування.	Лекції, лабораторні роботи	Тест, екзамен	20%
1.3	Знати суть декларативних засобів стосовно об'єктно-орієнтованої парадигми, знати засоби декларативного програмування для .Net та Java.	Лекції, лабораторні роботи	Тест, екзамен	10%
2.1	Вміти застосовувати функціональний стиль програмування, використовуючи мову програмування Haskell.	лабораторні роботи (Haskell), самостійна робота	Захист лабораторних робіт, екзамен	20%
2.2	Вміти застосовувати логічне програмування, використовуючи Prolog-процесори, зокрема SWI-Prolog.	лабораторні роботи (SWI-Prolog), самостійна робота	Захист лабораторних робіт, екзамен	20%
2.3	Вміти використовувати декларативні засоби програмування для .Net та Java.	лабораторні роботи (Prolog), самостійна робота	Захист лабораторних робіт, екзамен	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	1.2	1.3	2.1	2.2	2.3
Програмні результати навчання (з опису освітньої програми)						
ПР8. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.	+	+	+	+	+	+
ВПРН2.2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, парадигми, технології, алгоритмічні і програмні рішення при проектуванні та розробці програмних систем.	+	+	+	+	+	+

* заповнюється за необхідністю, наприклад для практик, лабораторних курсів тощо.

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Тести: РН 1.1., РН 1.2, РН 1.3 — по 6/4 балів.
2. Лабораторні роботи : РН 2.1 — 14(4+4+6)/8 балів.
2. Лабораторні роботи : РН 2.2 — 18(5+5+8)/11 балів.
4. Лабораторні роботи : РН 2.3 — 10(3+3+4)/6 балів.

- підсумкове оцінювання (у формі екзамену):

- 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

Структура екзаменаційної роботи та критерії оцінювання:

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

Запитання для підготовки до екзамену

- 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*.
- Лінійні та енергійні обчислення. Приклади.
- Використання нескінчених списків. Списки арифметичних послідовностей.
- Лінійні та енергійні обчислення. Використання нескінчених списків на прикладі списку простих чисел.
- Лінійні 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, foldl1 foldr, foldr1*).
- 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 (\Rightarrow).
- Оператори аплікації. Оператор зв'язування *bind* ($\gg=$).
- 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 (\gg). Приклади використання у випадку *IO* монади.
- Ввід-вивід. Монадна функція *return*. Приклади.
-
- Скулемівські нормальні форми, множини диз'юнктив. 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 (COA). 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 екзамен не може бути меншою 24 балів

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів. Студент допускається до екзамену за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Тест РН 1.1., лабораторні роботи РН 2.1 : до 5 тижня семестру.
2. Тест РН 1.2., лабораторні роботи РН 2.2 : до 11 тижня семестру.
3. Тест РН 1.3., лабораторні роботи РН 2.3 : до 15 тижня семестру.

У випадку відсутності студента з поважних причин відпрацювання та перездачі МКР здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

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

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / 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).	2		3
2.	Тема 2. Функції. Типи функцій, визначення функцій рівняннями, застосування функцій. Автоматичне виведення типів. Функції від кількох аргументів. Карінг.	2		3
3.	Тема 3. Визначення функцій кількома рівняннями. Рекурсивні функції та теорема про нерухому точку.	2		3
4.	Тема 4. Поліморфні типи та поліморфні функції.	2	2	4

	Параметричний поліморфізм.			
5.	Тема 5. Лямбда-функції у Haskell. Функції вищих порядків.	2		4
6.	Тема 6. Лінійні та енергійні обчислення. Нескінчені списки. Списки арифметичних послідовностей.	2	2	3
7.	Тема 7. List comprehension. Приклади.	2		3
8.	Тема 8. Поняття класів у Haskell та поліморфізм класів.	2		3
9.	Тема 9. Проблеми вводу-виводу, поняття монадних обчислень та монад.	2		3
10.	Тема 10. Стандартні монади. Проектування монад. Монада Maybe.	2	2	3
Всього по частині 1		20	6	32
Частина 2. Парадигма логічного програмування				
11.	Тема 11. "Чистий" (недетермінований) Пролог. Синтаксис, логічна семантика.	2		3
12.	Тема 12. Операційна семантика Пролог-програм та метод резолюцій.	2	2	4
13.	Тема 13. Процесори для мови Пролог. Використання відкатів.	2		3
14.	Тема 14. Специфіка техніки програмування у Пролозі.	2	2	4
15.	Тема 15. Використання Прологу при реалізації експертних систем.	2		3
16.	Тема 16. Експертні системи з поясненнями. Експертні системи в умовах неповної визначеності.	2	2	4
Всього по частині 2		12	6	21
Частина 3. Парадигма об'єктно-орієнтованого програмування та декларативні засоби програмування				
17.	Тема 17. Об'єкти та метадані. Рефлексія об'єктів. Декларативний стиль програмування у Java. Анотації.	2		4
18.	Тема 18. Анотації, розробка анотацій. Приклад.	2	2	3
19.	Тема 19. Застосування декларативних засобів програмування у Java та .Net на прикладі технології веб-сервісів SOAP	2	2	4
Всього по частині 3		6	4	11
Консультація			2	
Екзамен				
ВСЬОГО		38	16	64

Загальний обсяг 120 год., в тому числі:

Лекцій – 38 год.

Лабораторні заняття - 16 год.

Консультації – 2 год.

Самостійна робота - 64 год.

9. Рекомендовані джерела:

Основні:

1. Липовача М. Изучай Haskell во имя добра! М.: ДМК Пресс, 2012.
2. Lipovača M. Learn You a Haskell for Great Good! Miran Lipovača. : No Starch Press», 2011.
3. Душкин Р. В. Функциональное программирование на языке Haskell. М.: ДМК Пресс, 2007.
4. Братко И. Программирование на языке PROLOG для искусственного интеллекта. М.: Мир, 1990.
5. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
6. Марселлус Д. Программирование экспертных систем на Турбо-Прологе, М., Финансы и статистика, 1994. 410с.
7. Хохгуртль Б., C# и Java: межплатформенные Web-сервисы, М., Кудиц-образ, 2004, 410с.
8. Машнин Т.С. Web-сервисы Java, БХВ-Петербург, 2012, 560 с.

Додаткові:

1. Ин Ц., Соломон Д. Использование Турбо-Пролога. М.: Мир, 1990.
2. Стерлинг Л., Шапиро Э. Искусство программирования на языке Пролог. М.: Мир, 1990.
3. Стобо Дж. Язык программирования Пролог. М.: Радио и связь, 1993.
4. Клоксин У., Меллиш К. Программирование на языке пролог. М.: Мир, 1987.
5. Янсон А. Турбо-Пролог в сжатом изложении. М.: Мир, 1991.
6. Душкин Р. В. Справочник по языку Haskell. М.: ДМК Пресс, 2008.
7. Душкин Р. В. Практика работы на языке Haskell. М.: ДМК Пресс, 2009.
8. Роганова Н. А. Функциональное программирование, 2002.
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. Учебник.