

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ
для студентів**

галузь знань **12 Інформаційні технології**
спеціальність **121 Інженерія програмного забезпечення**
освітній рівень **бакалавр**
освітня програма **Програмна інженерія**
блок вибору **Програмна інженерія**
вид дисципліни **вибіркова**

Форма навчання	денна
Навчальний рік	2022/2023
Семестр	7
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **к.ф.-м.н., доц. Галкін О.В.** (лекції, лабораторні заняття)
асп. Ількун О.П. (лабораторні заняття)

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

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

КИЇВ 2019

Розробник: Галкін Олександр Володимирович, к.ф.-м.н., доц.,
доцент кафедри інтелектуальних програмних систем

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

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

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«_____» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – вивчення функціонального програмування на прикладі мови Haskell та основ лямбда-числення.

2. Попередні вимоги до опанування навчальної дисципліни

1. **Знати:** парадигми програмування, зокрема процедурне та об'єктно-орієнтоване програмування; теорію алгоритмів.

2. **Вміти:** програмувати, використовуючи процедурний та об'єктно-орієнтований підходи.

3. Анотація навчальної дисципліни

Навчальна дисципліна «Функціональне програмування» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення освітньо-професійної програми „Програмна інженерія”.

Дисципліна належить до вибіркових дисциплін блоку вибору “Програмна інженерія”. Викладається у **7 семестрі 4 курсу в обсязі – 180 год. (6 кредитів ECTS)**, зокрема: лекції – 28 год., лабораторні заняття – 28 год., консультації – 4 год., самостійна робота – 120 год. У курсі передбачено 2 частини та 2 контрольні роботи. Завершується дисципліна – **іспитом у 7 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

знати теоретичні основи лямбда-числення; концепцію та прийоми функціонального програмування;

вміти програмувати на мові Haskell.

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Функціональне програмування» є набуття знань, умінь та навичок (компетенцій) на рівні новітніх досягнень в області функціонального програмування відповідно до освітньої кваліфікації бакалавр з програмної інженерії.

Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність розробляти архітектури, модулі та компоненти програмних систем (СК03).
- Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (СК13).
- Здатність до алгоритмічного та логічного мислення (СК14).
- Здатність застосовувати дискретні структури і сучасні методи дискретної математики під час аналізу, синтезу та проектування інформаційних систем різної природи (СК15.2).

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			
PH1.1	Знати концепцію та прийоми функціонального програмування	Лекція, лабораторне заняття, самостійна робота	Контрольна робота, іспит	20%
PH1.2	Знати теоретичні основи лямбда-числення	Лекція, самостійна робота	Контрольна робота, іспит	20%
PH2.1	Вміти програмувати на мові Haskell	Лабораторне заняття, самостійна робота	Контрольна робота, захист лабораторної роботи, іспит	45%
PH3.1	Обґрунтовувати власний погляд на розв'язання задачі, спілкуватися з колегами з питань проектування та розробки програм у функціональному стилі, складати письмові звіти	Лабораторне заняття, самостійна робота	Поточне оцінювання, захист лабораторної роботи	5%
PH4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Поточне оцінювання, захист лабораторної роботи	5%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторне заняття, самостійна робота	Захист лабораторної роботи	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Програмні результати навчання	Результати навчання дисципліни					
	PH1.1	PH1.2	PH2.1	PH3.1	PH4.1	PH4.2
ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	+	+			+	
ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.		+				
ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	+		+	+	+	+
ПРН10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.	+					

ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+	+	+			
ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.	+			+		+
ПРН25.2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для розв'язання завдань інженерії програмного забезпечення.	+	+		+	+	
ПРН27.2. Знати та вміти застосовувати сучасні технології та методи проектування та програмування.	+		+			

7. Схема формування оцінки.

7.1. Форми оцінювання студентів.

– **семестрове оцінювання (максимальна кількість балів):**

1. Контрольна робота 1: РН 1.1, РН 2.1 – **9/6 б.**
2. Контрольна робота 2: РН 1.2 – **9/6 б.**
3. Лабораторні роботи (3): РН 2.1, РН 3.1, РН 4.1-4.2 – **14/8 б. кожна (разом 42/24 б.)**

Підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН 1.1, РН 1.2, РН 2.1;
- форма проведення і види завдань: письмово-усна форма, 4 практичні завдання по 10 балів (25%) кожне.

Студент **допускається** до іспиту, якщо він під час семестру набрав **більше 36 балів**, причому лабораторні роботи було виконано **як мінімум на 60%**.

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

7.2. Організація оцінювання.

Терміни проведення форм оцінювання:

1. Контрольна робота 1: після лекції №8.
2. Контрольна робота 2: після лекції №14.
3. Лабораторна робота 1 (проект): лабораторне заняття №4.
4. Лабораторна робота 2 (проект): лабораторне заняття №9.
5. Лабораторна робота 3 (проект): лабораторне заняття №14.

Студент має право один раз перескласти контрольну роботу з можливістю отримати не більше 80% балів, призначених за роботу. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт 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 або невчасного виконання лабораторної роботи викладач має право не зарахувати завдання або знизити за нього бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою 10% балів за кожен тиждень, що пройшов від 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

8. Структура навчальної дисципліни. Тематичний план лекцій.

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабор. заняття	С/р
Частина 1. Мова Haskell				
1	Тема 1. Основні типи в мові Haskell. Основні функціональні патерни	2	2	11
2–3	Тема 2. Алгебраїчні типи даних. Вкладені форми в Haskell. Рекурсія та списки	4	4	13
4–6	Тема 3. Класи в Haskell. Строгі та нестрогі конструктори даних. Обробка виключень	6	6	23
7–8	Тема 4. Монади в Haskell. Абстрактні типи даних	4	4	17
Контрольна робота №1				2
		16	16	66
Частина 2. Лямбда-числення				
9	Тема 5. Вступ до лямбда-числення. Обчислення лямбда-виразів	2	2	11
10	Тема 6. Порядок редукцій та нормальні форми	2	2	11
11–12	Тема 7. Фундаментальні теореми лямбда-числення. Схеми редукції та механізми виклику	4	4	15
13–14	Тема 8. Чисте лямбда-числення. Типізоване лямбда-числення	4	4	15
Контрольна робота №2				2
		12	12	54
ВСЬОГО		28	28	120

Загальний обсяг– **180 год.**, у тому числі:

Лекцій – **28 год.**,

Консультацій – **4 год.**,

Лабораторних занять – **28 год.**,

Самостійна робота – **120 год.**

Тематика лабораторних проектів

1. Основи синтаксису мови Haskell.
2. Створення бізнес-додатку на мові Haskell.
3. Лямбда-числення.

Перелік питань для підготовки до іспиту

1. Основні типи мови Haskell.
2. Основні функціональні патерни.
3. Алгебраїчні типи даних.
4. Вкладені форми.
5. Класи в Haskell.
6. Поняття про строгі та нестрогі конструктори даних.
7. Обробка виключень в Haskell.
8. Монади в Haskell. Опис монад State, Map.
9. Абстрактні типи даних в Haskell.
10. Поняття про лямбда-числення.
11. Основні правила редукції в лямбда-численні.
12. Основний і аплікативний порядок редукції.
13. Поняття про нормальну форму. Слабка заголовочна нормальна форма.
14. Поняття про лінійні обчислення.
15. Чисте лямбда-числення.
16. Числення де Брейна.
17. Рекурсія в лямбда-численні. Y- комбінатор.
18. Теорема Чорча-Россера і основні її наслідки.
19. Теорема стандартизації.
20. Типізоване лямбда-числення
21. Поняття про неформальну перевірку типів.
22. Типізація за Чорчем.
23. Типізація за Каррі.
24. Поліморфізм. Let-поліморфізм.
25. Поняття про сильну нормалізацію.

9. Рекомендовані джерела

1. Підручник по Haskell:
<https://anton-k.github.io/ru-haskell-book/book/home.html>
2. Bryan O'Sullivan, Don Stewart, and John Goerzen. Real World Haskell, O'Reilly Media, 2009, 670 p.
3. Will Kurt. Get Programming with Haskell, Manning Publications, 2018, 616 p.
4. C. Hankin. An Introduction to Lambda Calculi for Computer Scientists, Kings College, 2004, 164 p.
5. X. Барендрегт. Лямбда-исчисление. Его синтаксис и семантика. – М.: Мир, 1985. – 606 с.
6. Математическая логика в программировании: сборник статей (ред. Захарьяшев М.В., Янова Ю.И.). – М.: Мир, 1991. – 408 с.