

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА  
ШЕВЧЕНКА**

**Факультет радіофізики, електроніки та комп'ютерних систем  
Кафедра математики та теоретичної радіофізики**

**«ЗАТВЕРДЖУЮ»**

Заступник декана з навчальної роботи

\_\_\_\_\_ Наталія ГОРБОВЦОВА

« \_\_\_\_ » \_\_\_\_\_ 2023 року

***Робоча програма навчальної дисципліни***  
**Функціональне програмування**

**для студентів**

галузь знань 12 Інформаційні технології  
спеціальність 123 Комп'ютерна інженерія  
рівень вищої освіти перший  
освітня програма Інженерія комп'ютерних систем і мереж  
вид дисципліни вільного вибору

Форма навчання	денна
Навчальний рік	2023/2024
Семестр	7
Кількість кредитів ECTS	4
Мова викладання	українська
Форма заключного контролю	іспит

Викладач: Масютка Олександр Юрійович канд.фіз.-мат.наук, доцент кафедри математики та теоретичної радіофізики

Пролонговано: на 20\_\_/20\_\_ н. р. \_\_\_\_\_ ( \_\_\_\_\_ ) « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.  
на 20\_\_/20\_\_ н. р. \_\_\_\_\_ ( \_\_\_\_\_ ) « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Розробник:

**Масютка Олександр Юрійович**, канд.фіз.-мат.наук, доцент кафедри математики та теоретичної радіофізики

«ЗАТВЕРДЖЕНО»

*Зав. кафедри математики та теоретичної радіофізики*

\_\_\_\_\_ Володимир ВИСОЦЬКИЙ

Протокол № \_\_ від « \_\_ » \_\_\_\_\_  
2023 р.

«ЗАТВЕРДЖЕНО»

*Завідувач кафедри комп'ютерної інженерії*

\_\_\_\_\_ Юрій БОЙКО

Протокол № \_\_ від « \_\_ » \_\_\_\_\_  
\_\_\_\_\_ 2023 р.

Схвалено науково-методичною комісією факультету радіофізики, електроніки та комп'ютерних систем

Протокол № \_\_ від « \_\_ » \_\_\_\_\_ 2023 р.

Голова науково-методичної комісії

Сергій РАДЧЕНКО

« \_\_ » \_\_\_\_\_ 2023 р.

## ВСТУП

**1. Мета дисципліни:** ознайомлення студентів із основними особливостями функціонального підходу до розробки програмного забезпечення, значенням та місцем функціональних мов програмування у індустрії проектування програмних систем, основними методами та засобами функціонального програмування, зокрема із застосуванням мови Haskell.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни:**

вивчення дисципліни «Функціональне програмування» передбачає попереднє вивчення нормативних дисциплін „Програмування” та „Алгоритми та методи обчислень».

**3. Анотація навчальної дисципліни:** «Функціональне програмування» є дисципліною вільного вибору студента для спеціальності "комп'ютерна інженерія". Вона необхідна для формування у студентів загальних методологічних основ та практичних навичок розробки програмних систем з використанням функціонального підходу до програмування. "Функціональне програмування" включає ознайомлення з парадигмою функціонального програмування, теоретичними основами функціонального підходу до програмування та прикладами їх практичного застосування.

**4. Завдання навчальної дисципліни (навчальні цілі):**

- 1) надати основні теоретичні відомості функціонального програмування, які складають важливу частину підготовки студента-бакалавра за спеціальністю «Комп'ютерна інженерія»;
- 2) продемонструвати використання інструментальних засобів для створення програмного забезпечення на базі функціональних методів програмування;
- 3) застосування знань, умінь, навичок і комунікацій у професійній діяльності, розвиток логічного та аналітичного мислення студентів;
- 4) прищепити вміння використовувати функціональні методи при проектуванні прикладних програм.
- 5) Дисципліна спрямована на формування програмних компетентностей:

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

ФК13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

## 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			
<b>1</b>	студент повинен <b>знати</b> :	лекційні заняття, лабораторні заняття з використанням середовища Haskell Platform	оцінювання виконання лабораторних завдань та самостійно створених програм	до 45
1.1	основні поняття та терміни функціонального програмування			
1.2	основи роботи в середовищі Haskell Platform та базовий синтаксис мови Haskell			
1.3	основні керуючі конструкції мови Haskell			
1.4	правила створення методів та передачі параметрів методам			
<b>2</b>	студент повинен <b>вміти</b> :	лекційні заняття, лабораторні заняття з використанням середовища Haskell Platform	оцінювання виконання лабораторних завдань та самостійно створених програм	до 45
2.1	продумати алгоритм створення програми			
2.2	створювати проект консольного прикладення у відповідності до поставленої задачі			
2.3	аналізувати код програми, вміти налагодити його			
2.4	тестувати програму та відшукувати і виправляти в ній помилки			
<b>3</b>	<b>комунікація</b>	лекційні заняття, лабораторні заняття з використанням середовища Haskell Platform	оцінювання виконання лабораторних завдань та самостійно створених програм	до 5
3.1	здатність грамотно будувати комунікацію, виходячи з мети і ситуації спілкування – найкраще засвоєння матеріалу відбувається саме при спільному аналізі своїх програм			
<b>4</b>	<b>автономність та відповідальність</b>	лекційні заняття, лабораторні заняття з використанням середовища Haskell Platform	оцінювання виконання лабораторних завдань та самостійно створених програм	до 5
4.1	Здатність до самостійного пошуку технічної літератури або інших джерел інформації для пошуку шляхів усунення помилок студентом під час виконання лабораторних робіт			

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни Програмні результати навчання (назва)	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	4.1
ПРН1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.	+	+	+	+	+	+	+	+			
ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосунків, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання					+	+	+	+			+
ПРН18. Використовувати інформаційні технології для ефективного спілкування на професійному та соціальному рівнях.									+	+	
ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.									+	+	+

## 7. Схема формування оцінки

**7.1. Форми оцінювання студентів:** рівень досягнення всіх запланованих результатів навчання визначається за результатами виконання лабораторних робіт. Вклад результатів навчання у підсумкову оцінку, за умови їх опанування на належному рівні і успішної здачі всіх лабораторних робіт наступний:

- результати навчання 1.1 – 1.4 [**знання**] до 45 %;
- результат навчання 2.1 – 2.4 [**вміння**] – до 45%;
- результат навчання 3.1 [**комунікація**] – до 5%;
- результат навчання 4.1 [**автономність та відповідальність**] – до 5%.

Форми оцінювання студентів:

- **семестрове оцінювання:** передбачено проведення чотирьох лабораторних робіт за матеріалом занять. Виконання лабораторних робіт оцінюється максимум в 60 балів. По результатах семестрового оцінювання студент може отримати до 60 балів.
- **умови допуску до іспиту:** умовою допуску до іспиту є отримання студентом сумарно не менше, аніж *критично-розрахунковий мінімум 35 балів* за семестр. Студенти, які протягом семестру набрали сумарно меншу кількість балів, ніж критично-розрахунковий мінімум 35 балів, для одержання допуску до іспиту обов'язково повинні виконати на необхідну порогову кількість балів додаткову

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

- **підсумкове оцінювання (у формі іспиту):** форма іспиту – письмово-усна. Екзаменаційний білет іспиту складається із 4 питань:

1. теоретичне питання (оцінюється від 0 до 10 балів);
2. теоретичне питання (оцінюється від 0 до 10 балів);
3. практична задача на написання програми (оцінюється від 0 до 10 балів);
4. практична задача на написання програми (оцінюється від 0 до 10 балів);

Всього за іспит можна отримати від 0 до 40 балів. Умовою досягнення позитивної оцінки за дисципліну є отримання не менш ніж 60 балів.

## 7.2. Організація оцінювання:

*Оцінювання за формами контролю:*

	ЗМ-1	ЗМ-2
	бал (min-max)	бал (min-max)
Виконання студентом лабораторних робіт	0 – 20	0 – 40

*Орієнтований графік оцінювання:*

	<i>Орієнтовний період для здійснення відповідної форма оцінювання</i>
Виконання студентом лабораторних робіт	вересень-грудень
Добір балів/додаткова контрольна робота	грудень
Іспит	грудень

*Розрахунок балів, які студент отримує при успішній здачі іспиту:*

	Змістовий модуль 1	Змістовий модуль 2	Іспит	Підсумкова оцінка
<i>Мінімум</i>	10	25	25	60
<b>Максимум</b>	<b>20</b>	<b>40</b>	<b>40</b>	<b>100</b>

### 7.3. Шкала відповідності оцінок

Оцінка (за національною шкалою) / National grade	Рівень досягнень, % / Marks, %
<b>Відмінно</b> / Excellent	90-100%
<b>Добре</b> / Good	75-89%
<b>Задовільно</b> / Satisfactory	60-74%
<b>Незадовільно</b> / Fail	0-59%

### 8. Структура навчальної дисципліни. Тематичний план лекційних занять.

№	Тема	Кількість годин (лекцій)	Кількість годин (лабораторні заняття)	Контрольні роботи
<b>Змістовний модуль №1 .</b>				
1.	Парадигми програмування. Визначення та коротка історія функціонального програмування. Абстракція і декомпозиція при функціональному підході. Декларативне програмування.	2	–	
2.	Вступ до $\lambda$ -числення. Редукція. Мова функціонального програмування як $\lambda$ -редуктор. Функції декількох аргументів. Розгортання функції (currying). Умовний вираз. Визначення імен. Області видимості.	2	–	
3.	Циклічні конструкції. Зіставлення зі зразком. Рекурсія. Види рекурсії. Рекурсивні структури даних. Списки. Перестановки. Синтаксис породження списку (list comprehension).	2	2	
4.	Хвостова рекурсія. Функціональні структури даних. Представлення черги. Багатовимірні масиви.	2	2	
5.	Дерева. Дерева загального вигляду і бінарні дерева. Обхід дерева. Реалізація обходу за допомогою функції з відкладеним обчисленням. Дерева виразів і дерева пошуку. Продовження (continuations).	2	2	
6.	Основні моделі обчислень в $\lambda$ -численні. Синтаксис $\lambda$ -числення. Чисте і прикладне $\lambda$ -числення. Перетворення $\lambda$ -виразів.	2	2	
7.	Бета-редукція і заміна змінної. Нормальний і аплікативний порядок редукції. Відкладені та енергійні обчислення.	2	2	
<b>Змістовий модуль №2 .</b>				
8.	Механізми виклику і проблема поділу. Теорема Чьорча-Россера і теорема стандартизації. Екстенціональності..	2	4	
9.	Опис рекурсивних функцій. Оператор нерухомої точки. Комбінатори і комбінаторна логіка.	2	4	
10.	Представлення умовних виразів, списків і натуральних чисел в $\lambda$ -численні. Обчислюваність.	2	2	
11.	Еквівалентність алгоритмічних моделей. Перевизначення імен. Замикання. Мемоізація.	2	2	

12.	Типізація в мовах функціонального програмування. Класифікація мов програмування за видами типізації. Типізоване $\lambda$ -числення. Виведення типів.	2	2	
13.	Формальна семантика мов функціонального програмування. Класифікація формальних семантик. Семантика для найпростішої мови. Доведення властивостей програм. Доведення коректності програм на прикладах. Проблема самозастосовності.	2	2	
14.	Імперативне ядро в функціональних мовах. Монади. Асинхронні та паралельні обчислення. Сумісне використання розглянутих технологій.	2	2	
	<b>ВСЬОГО</b>	<b>28</b>	<b>28</b>	

Загальний обсяг **120 год.**, в тому числі:

Лекцій – **28 год.**

Лабораторні – **28 год.**

Самостійна робота **64 год.**

## 9. Рекомендована література:

### Основні джерела.

1. G. Hutton. Programming in Haskell. – Cambridge University Press, 2016.
2. H. Abelson, G.J. Sussman, J. Sussman. Structure and Interpretation of Computer Programs, 2nd Edition – MIT Press, 1996. [Електронний ресурс].- Режим доступу: URL: <http://mitpress.mit.edu/sicp/>.
3. R. Bird. Thinking functionally with Haskell. – Cambridge University Press, 2014.
4. M. Lipovaca. Learn you a Haskell for Great good. – No Starch Press, 2011.

### Додаткові джерела:

1. <http://haskell.org> – офіційний сайт мови Haskell.