

# СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

## 1. Загальна інформація про навчальну дисципліну

<b>Повна назва навчальної дисципліни</b>	Функціональне програмування
<b>Повна офіційна назва закладу вищої освіти</b>	Сумський державний університет
<b>Повна назва структурного підрозділу</b>	Факультет електроніки та інформаційних технологій. Кафедра комп'ютерних наук
<b>Розробник(и)</b>	Москаленко В`ячеслав Васильович
<b>Рівень вищої освіти</b>	Другий рівень вищої освіти, НРК – 7 рівень, QF-LLL – 7 рівень, FQ-EHEA – другий цикл
<b>Семестр вивчення навчальної дисципліни</b>	8 тижнів протягом 2-го семестру
<b>Обсяг навчальної дисципліни</b>	Обсяг становить 5 кред. ЄКТС, 150 год. Для денної форми навчання 32 год. становить контактна робота з викладачем (16 год. лекцій, 16 год. лабораторних занять), 118 год. становить самостійна робота.
<b>Мова викладання</b>	Українська

## 2. Місце навчальної дисципліни в освітній програмі

<b>Статус дисципліни</b>	Вибіркова навчальна дисципліна для освітньої програми "Інформатика"
<b>Передумови для вивчення дисципліни</b>	Передумови для вивчення відсутні
<b>Додаткові умови</b>	Додаткові умови відсутні
<b>Обмеження</b>	Обмеження відсутні

## 3. Мета навчальної дисципліни

досягнення студентами сучасного конструктивного, фундаментального мислення та комплексу спеціальних знань з функціонального програмування.

## 4. Зміст навчальної дисципліни

Тема 1 Типи даних, функції та модулі

Особливості функціональної парадигми програмування. Структура програми на мові Haskell. Примітивні типи даних. Типи синоніми. Розробка програм на мові Haskell в середовищі Haskell Platform. Рекурсивні та анонімні функції. Сторожові, контрольні та локальні вирази. Спискові включення та діапазони.

<p>Тема 2 Функції вищого порядку та функціональні ланцюжки</p> <p>Карінг та декарінг. Часткове застосування функцій. Функції як аргументи та результати. Лінійні обчислення. Використання функцій вищого порядку для обробки строкових даних. Функції композиції та функції застосування. Параметричний поліморфізм. Функції <code>map</code>, <code>filter</code>, <code>fold</code> та <code>reduce</code></p>
<p>Тема 3 Типи користувача</p> <p>Алгебраїчні типи даних. Використання алгебраїчних типів даних. Рекурсивні алгебраїчні типи даних. Класи типів, наслідування, екземпляри класів та контекст типу. Використання користувацьких класів типів та визначення екземплярів</p>
<p>Тема 4 Контейнери, функтори і монади</p> <p>Ізоморфний тип <code>newtype</code>. Контейнер <code>Maybe</code>. Моноїди. Функтори та аплікативні функтори. Реалізація функторів та аплікативних функторів. Монади. Введення-виведення даних. Обробка виключень. Реалізація монад та використання монади IO</p>

## 5. Очікувані результати навчання навчальної дисципліни

Після успішного вивчення навчальної дисципліни здобувач вищої освіти зможе:

PH1	Вміти здійснювати попередню оцінку можливості застосування функціонального програмування для вирішення поставленої задачі
PH2	Вміти здійснювати підготовку та первинну обробку даних та складати вхідний математичний опис алгоритмічного і програмного забезпечення, що розробляється в рамках парадигми функціонального програмування
PH3	Знати основні концепції та особливості функціонального програмування
PH4	Вміти обґрунтовувати вибір конкретного стилю функціонального програмування для вирішення відповідних практичних задач
PH5	Володіти сучасними програмними мовами функціонального програмування

## 7. Роль освітнього компонента у формуванні соціальних навичок

Загальні компетентності та соціальні навички, формування яких забезпечує навчальна дисципліна:

## 8. Види навчальних занять

<p><b>Тема 1. Типи даних, функції та модулі</b></p> <p>Лк1 "Особливості функціональної парадигми програмування. Структура програми на мові Haskell. Примітивні типи даних. Типи синоніми" (денна)</p> <p>Ця лекція охоплює ключові аспекти функціональної парадигми програмування, зокрема в контексті мови Haskell. Основна увага приділяється структурі програми на Haskell, включно з її особливостями та відмінностями від імперативних мов. Лекція також розглядає примітивні типи даних у Haskell, детально розбираючи їх особливості та використання. Окрім цього, звертається увага на типи-синоніми, їх важливість та застосування в програмуванні. Ці теми забезпечують фундаментальне розуміння функціональної парадигми та ефективного використання Haskell.</p>
--

Лк2 "Рекурсивні та анонімні функції. Сторожові, контрольні та локальні вирази. Спискові вклучення та діапазони" (денна)

Ця лекція охоплює ключові концепції, включаючи рекурсивні та анонімні функції, роз'яснюючи їх важливість та використання у програмуванні. Крім того буде зосереджена увага на сторожових, контрольних та локальних виразах, які забезпечують гнучкість у коді, та завершимо обговоренням спискових вклучень та діапазонів, що спрощують роботу з колекціями даних. Ця лекція не тільки збагатить ваше розуміння Haskell, але й надасть практичні навички для ефективного програмування.

Лб1 "Розроблення програми на мові Haskell в середовищі Haskell Platform" (денна)

Ця лабораторна охоплює вивчення та застосування функцій у мові програмування Haskell. Вона вклучає теми, такі як написання простих визначень Haskell, роботу з локальними визначеннями та правилами верстки, а також використання кортежів та шаблонів. Студенти навчаються використовувати інтерактивний інтерпретатор Haskell, виконувати різні вправи з написання функцій, та розуміти основи функціонального програмування. Лабораторна сприяє розвитку навичок роботи з функціями та розуміння особливостей мови Haskell.

Лб2 "Розроблення рекурсивних функцій на мові Haskell з використанням сторожових, контрольних та локальних виразів" (денна)

Лабораторна робота зосереджується на розробці рекурсивних функцій у мові Haskell. Вона охоплює теми, такі як використання шаблонів, рекурсія з цілими числами та списками, а також вирази реєстрів. Студенти вчаться писати рекурсивні функції, зокрема, з використанням базових та рекурсивних випадків, а також вправляються у використанні виразів реєстру. Ця лабораторна робота має на меті підвищити навички програмування в Haskell та глибше зрозуміти концепції функціонального програмування.

## **Тема 2. Функції вищого порядку та функціональні ланцюжки**

Лк3 "Карінг та декарінг. Часткове застосування функцій. Функції як аргументи та результати. Лінійні обчислення" (денна)

У цій лекції ми дослідимо концепції каррінгу та декаррінгу, розглянемо часткове застосування функцій, яке дозволяє гнучко маніпулювати аргументами, та з'ясуємо, як функції можуть виступати як аргументи чи результати інших функцій. Особлива увага буде приділена лінійним обчисленням, що є важливою характеристикою Haskell, дозволяючи ефективно управляти ресурсами та виконанням програми. Ця лекція надасть глибше розуміння ключових аспектів Haskell та покаже, як ці особливості можуть бути застосовані для підвищення ефективності програмування.

Лк4 "Функції композиції та функції застосування. Параметричний поліморфізм. Функції map, filter, fold та reduce" (денна)

Ця лекція зосереджується на функціях композиції та застосування, розкриваючи їхню сутність і важливість у створенні ефективного коду. Ми також оглянемо концепцію параметричного поліморфізму, яка розширює гнучкість та повторне використання функцій. Особлива увага буде приділена функціям map, filter, fold, та reduce, що є ключовими для обробки колекцій даних. Крізь ці теми, лекція підкреслює силу абстракції та композиції у функціональному програмуванні, демонструючи, як ці підходи можуть поліпшити читабельність, підтримуваність та ефективність коду в Haskell.

Лб3 "Використання функцій та параметричного поліморфізму для оброблення списків" (денна)

Лабораторна робота зосереджена на вивченні функцій та параметричного поліморфізму в Haskell для оброблення списків. Вона включає теми, такі як використання каррінг, композиції функцій, лямбда-абстракцій, операторів списків, а також згортки (fold). Студенти навчаються створювати різноманітні функції для ефективної роботи зі списками та розуміння основ функціонального програмування. Ці навички є ключовими для розробки більш складних програм на Haskell і глибшого розуміння функціональних парадигм.

Лб4 "Використання функцій вищого порядку для оброблення строкових даних" (денна)

Лабораторна робота зосереджується на використанні функцій вищого порядку для оброблення строкових даних у мові Haskell. Вона охоплює теми, такі як налагодження функціональних програм, рекурсивний стиль програмування, розуміння списків, поліморфізм та рекурсивне програмування. Студенти навчаються ефективним методам налагодження коду, розробці рекурсивних функцій, розумінню та застосуванню виразів розуміння списків, а також розвивають навички поліморфного програмування, важливих для роботи з різними типами даних.

### **Тема 3. Типи користувача**

Лк5 "Алгебраїчні типи даних" (денна)

У цій лекції ми поглибимо знання про алгебраїчні типи даних, важливу концепцію, яка дозволяє визначати складні типи через комбінацію існуючих. Ми розглянемо як сумові (union types), так і добуткові типи (product types), досліджуючи їхню роль у структуруванні та представленні даних у програмах. Лекція надасть практичні приклади та покаже, як алгебраїчні типи можуть бути застосовані для створення більш гнучкого, безпечного та виразного коду, демонструючи їхню ефективність у рішенні реальних задач програмування.

Лк6 "Класи типів, наслідування, екземпляри класів та контекст типу" (денна)

Ця лекція розкриває концепції класів типів, наслідування, екземплярів класів та контексту типу, які є ключовими для розуміння типової системи мови. Ми досліджуємо, як класи типів дозволяють визначати поведінку певних типів, а також розглядаємо, як створення екземплярів цих класів забезпечує конкретну реалізацію цієї поведінки. Лекція також пояснює, як наслідування в класах типів та контекст типу впливають на поліморфізм та переважання функцій. Через ці теми, студенти набудуть розуміння засобів Haskell для створення більш гнучкого, модульного та експресивного коду.

Лб5 "Використання алгебраїчних типів даних. Рекурсивні алгебраїчні типи даних" (денна)

Лабораторна робота присвячена вивченню алгебраїчних типів даних у мові програмування Haskell. Основні теми включають поліморфізм і класи типів, типи даних, визначені користувачем, та тип даних Maybe. Студенти навчаються використовувати ці концепції для розробки більш гнучких та потужних програм. Робота з алгебраїчними типами даних дозволяє студентам глибше зрозуміти структури даних і способи їх ефективної обробки в мові Haskell.

Лб6 "Використання користувачьких класів типів та визначення екземплярів" (денна)

Лабораторна робота фокусується на використанні користувачьких класів типів та визначенні екземплярів. Метою роботи є навчити студентів створювати та використовувати власні класи типів і визначати екземпляри цих класів для різних типів даних. Основні підтеми включають основи класів типів в Haskell, створення користувачьких класів, визначення екземплярів, а також практичне застосування в програмуванні. Лабораторна робота має на меті розвивати навички у використанні принципів поліморфізму та створенні універсальних інтерфейсів для різних типів даних.

#### **Тема 4. Контейнери, функтори і монади**

Лк7 "Ізоморфний тип newtype. Контейнер Maybe. Моноїди. Функтори та аплікативні функтори" (денна)

У цій лекції вивчається ізоморфний тип newtype, який дозволяє ефективно визначати нові типи даних. Розглядається контейнер Maybe, який використовується для безпечного представлення відсутніх або необов'язкових значень. Переходячи до моноїдів, розглядається їх важливість у структуруванні операцій. Також детально описується концепції функторів та аплікативних функторів, які розширюють можливості маніпуляції та комбінації функцій. Ця лекція допоможе студентам глибше зрозуміти ці ключові аспекти мови Haskell, показуючи, як вони можуть бути застосовані для створення більш потужного та виразного коду.

Лк8 "Монади. Введення-виведення даних. Обробка виключень" (денна)

Ця лекція присвячена одному з найважливіших понять функціонального програмування – монадам. Ми розглянемо монади не тільки як могутній інструмент для управління побічними ефектами, але й зосередимося на їх використанні у введенні-виведенні даних (I/O) у Haskell. Окрім цього, буде розібрана обробка виключень за допомогою монад, що дозволяє ефективно впоратися з несподіваними ситуаціями чи помилками в програмі. Лекція покликана дати глибше розуміння цих концепцій, показуючи, як вони сприяють написанню більш структурованого, безпечного та читабельного коду.

Лб7 "Реалізація функторів та аплікативних функторів" (денна)

Лабораторна робота зосереджується на реалізації функторів та аплікативних функторів. Метою є навчання студентів створювати та використовувати функтори та аплікативні функтори для ефективної роботи з різними структурами даних. Ключові елементи включають введення у функтори, їх реалізацію в Haskell, особливості аплікативних функторів, а також практичне застосування цих концепцій. Лабораторна робота охоплює теоретичні відомості, зокрема закони функторів та аплікативних функторів, а також практичні завдання на реалізацію та застосування цих концептів.

Лб8 "Реалізація монад та використання монади IO" (денна)

Лабораторна робота охоплює основи роботи з IO монадами, обробку виключень та роботу з потоками в Haskell. Студенти вивчають монади, особливості IO монади, та реалізують практичні завдання, які включають роботу з IO монадами для вводу/виводу, обробку виключень та реалізацію многопоточності. Ця лабораторна робота розширює розуміння студентів щодо функціонального програмування, зокрема використання монад для управління побічними ефектами в Haskell.

## **9. Стратегія викладання та навчання**

## 9.1 Методи викладання та навчання

Дисципліна передбачає навчання через:

МН1	Лекційне навчання
МН2	Проблемне навчання
МН3	Практикоорієнтоване навчання

Лекції надають студентам теоретичні матеріали з тем дисципліни, що є основою для проблемного навчання здобувачів вищої освіти. Лекції доповнюються практичними заняттями, що надають студентам можливість застосовувати теоретичні знання на практичних прикладах. Проблемному навчанню сприятиме підготовка до лекцій. В особливих умовах застосовуються методи та засоби електронного навчання на базі платформи [mix.sumdu.edu.ua](http://mix.sumdu.edu.ua)

Під час проведення занять студенти отримують навички комунікації, вміння працювати в команді, здатність логічно і системно мислити, аргументовано висловлювати свої думки. Виконання лабораторних робіт допоможе студентам розвивати та реалізувати навички логічного та системного мислення, тайм-менеджменту, самостійного опрацювання матеріалу.

## 9.2 Види навчальної діяльності

НД1	Інтерактивні лекції
НД2	Підготовка до лекцій
НД3	Виконання лабораторних робіт

## 10. Методи та критерії оцінювання

### 10.1. Критерії оцінювання

Визначення	Чотирибальна національна шкала оцінювання	Рейтингова бальна шкала оцінювання
Відмінне виконання лише з незначною кількістю помилок	5 (відмінно)	$90 \leq RD \leq 100$
Вище середнього рівня з кількома помилками	4 (добре)	$82 \leq RD < 89$
Загалом правильна робота з певною кількістю помилок	4 (добре)	$74 \leq RD < 81$
Непогано, але зі значною кількістю недоліків	3 (задовільно)	$64 \leq RD < 73$
Виконання задовольняє мінімальним критеріям	3 (задовільно)	$60 \leq RD < 63$
Можливе повторне складання	2 (незадовільно)	$35 \leq RD < 59$
Необхідний повторний курс з навчальної дисципліни	2 (незадовільно)	$0 \leq RD < 34$

### 10.2 Методи поточного формативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок
--	----------------	----------------	-------------------

МФО1 Експрес-тестування	Експрес-тестування призначене для проміжного оцінювання рівня засвоєння теоретичного матеріалу. Проводиться на початку наступної лекції з використанням платформи електронного навчання МІХ. Результати тестування обговорюються протягом виконання завдання	Протягом лекційного заняття	<a href="https://mix.sumdu.edu.ua">https://mix.sumdu.edu.ua</a> , Google meet
МФО2 Опитування та усні коментарі викладача за його результатами	Призначені для контролю засвоєння теоретичних знань поточної і минулих лекцій. Проводиться протягом дискусій і обговорень проблематики лекційного заняття	Протягом лекційного заняття	Google meet
МФО3 Проміжне оцінювання виконання лабораторних завдань	Призначено для перевірки теоретичних та практичних знань, отриманих протягом модуля. Тестові питання та завдання для виконання рефакторингу, реінжинірингу і верифікації програмного забезпечення	згідно графіку навчального процесу	<a href="https://mix.sumdu.edu.ua">https://mix.sumdu.edu.ua</a> , особистий кабінет

### 10.3 Методи підсумкового сумативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок
МСО1 Проміжний модульний контроль	Проміжний модульний контроль призначений для перевірки рівня засвоєння теоретичного матеріалу. Проводиться у форматі тестування засобами системи <a href="https://mix.sumdu.edu.ua">mix.sumdu.edu.ua</a> . Оцінка за проміжний модульний контроль не перескладається.	згідно графіку навчального процесу	<a href="https://mix.sumdu.edu.ua/">https://mix.sumdu.edu.ua/</a>

<p>МСО2 Оцінювання участі в дискусії</p>	<p>Участь в дискусії не є обов'язковим видом завдання, але є необхідним для отримання максимальної оцінки за курс. Дискусії та обговорення направлені на отримання навичок пошуку, аналізу інформації, формулювання висновків та висловлення власної позиції щодо оголошених проблемних питань державною (чи англійською) мовою. Для отримання максимальної оцінки студент має не лише написати власний пост, а й прокоментувати, принаймі, два пости інших студентів</p>	<p>В кінці лекції</p>	<p>Google Meet, <a href="https://mix.sumdu.edu.ua">https://mix.sumdu.edu.ua</a></p>
<p>МСО3 Оцінювання звіту за результатами виконання лабораторних робіт</p>	<p>Звіт за результатами виконання лабораторних робіт повинен містити ілюстрацію основних результатів виконання завдань відповідно методичних вказівок. Для успішного зарахування необхідно виконати мінімальний рівень завдання та оформити звіт відповідно вимогам до оформлення звітів. Оцінка зі звіту може бути один раз підвищення за умови усунення зауважень до представлених результатів і надсилання звіту у вказані терміни. В разі затримки термінів виконання оцінка не може бути підвищена.</p>	<p>До початку наступного лабораторного заняття</p>	<p>Google Meet, <a href="https://mix.sumdu.edu.ua">https://mix.sumdu.edu.ua</a></p>

Контрольні заходи:

		Максимальна кількість балів	Мінімальна кількість балів	Можливість перекладання з метою підвищення оцінки
<b>2 семестр</b>		<b>100 балів</b>		
МСО1. Проміжний модульний контроль		<b>20</b>		
	2x10	20	5	Ні
МСО2. Оцінювання участі в дискусії		<b>32</b>		
	8x4	32	4	Ні
МСО3. Оцінювання звіту за результатами виконання лабораторних робіт		<b>48</b>		



	8x6	48	4	Ні
--	-----	----	---	----

До диференційованого заліку необхідно виконати всі лабораторні роботи на мінімальний рівень складності завдань. При успішному (отримання персоніфікованого сертифікату із вказівкою рівня успішності) вивченні масових відкритих онлайн курсів можуть бути реалізовані наступні варіанти перезарахувань частини кредитів: 1. Для курсу <https://www.udemy.com/course/learn-haskell> можуть бути перезараховані тема 1 в обсязі 15 годин/0,5 кредит (10 балів); 2. Для курсу <https://www.udemy.com/course/learning-path-haskell-functional-programming-and-haskell/> може бути перезарахована тема 2 в обсязі 10 годин/0,5 кредит (10 балів); 3. Для курсу <https://www.udemy.com/course/master-haskell-from-scratch-a-basic-to-advanced-course/> може бути перезарахована тема 3 та 4 в обсязі 15 годин/0,5 кредит (10 балів). Також викадачем можуть бути розглянуті інші масові відкриті онлайн курси за умови попереднього аналізу структури курсу.

## 11. Ресурсне забезпечення навчальної дисципліни

### 11.1 Засоби навчання

ЗН1	Бібліотечні фонди
ЗН2	Комп'ютери, комп'ютерні системи та мережи
ЗН3	Мультимедіа, відео- і звуковідтворювальна, проєкційна апаратура (відеокамери, проєктори, екрани, смартдошки тощо)
ЗН4	Прикладне програмне забезпечення (Haskell Platform )

### 11.2 Інформаційне та навчально-методичне забезпечення

<b>Основна література</b>	
1	Функціональне та логічне програмування (Частина 1. Функціональне програмування) / Шевченко І. В., Кузнецова Ю. А., Сьомочкін М. О.. – Навч. посібник з виконання лабораторних робіт. – Харків: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2021. – 98 с.
2	Функціональне програмування: навчальний посібник / Верес М.М., Галкін О.В., Демківський Є.О. та ін. – Київ, 2022. – 178 с. - <a href="http://csc.knu.ua/media/filer_public/81/fc/81fc0d0a-d1e1-40ce-9f11-42df4a47b25a/funktsionalne_programuvannia.pdf">http://csc.knu.ua/media/filer_public/81/fc/81fc0d0a-d1e1-40ce-9f11-42df4a47b25a/funktsionalne_programuvannia.pdf</a>
<b>Допоміжна література</b>	
3	Q. Miller. A Tutorial Introduction to Haskell / Miller Q. // 2019. – <a href="https://www.some.ox.ac.uk/wp-content/uploads/2019/08/HaskellTutorial.pdf">https://www.some.ox.ac.uk/wp-content/uploads/2019/08/HaskellTutorial.pdf</a> .
4	J. P. Mueller. Functional Programming For Dummies / J. P. Mueller // John Wiley & Sons, Inc., Hoboken, New Jersey, 2019. <a href="http://dl.booktolearn.com/ebooks2/computer/programming/9781119527503_Functional_Programming_For_Dummies_b8d7.pdf">http://dl.booktolearn.com/ebooks2/computer/programming/9781119527503_Functional_Programming_For_Dummies_b8d7.pdf</a> .
<b>Інформаційні ресурси в Інтернеті</b>	
5	<a href="https://tryhaskell.org/">https://tryhaskell.org/</a>

