



ФУНКЦІЙНЕ ПРОГРАМУВАННЯ (HASKELL)

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	3 курс, весняний семестр
Обсяг дисципліни	4,0 кредити ECTS /120 годин (36 годин лекцій, 18 годин практичних занять, 66 годин — самостійна робота студента)
Семестровий контроль/ контрольні заходи	Залік, МКР, представлення робіт комп'ютерного практикуму
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент, Статівка Юрій Іванович, statyvka-yu@iit.kpi.ua Практичні заняття: к.т.н., доцент, Статівка Юрій Іванович, statyvka-yu@iit.kpi.ua
Розміщення курсу	https://ecampus.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою дисципліни є вивчення теоретичних основ та засобів створення програм в межах функційної парадигми; набуття студентами досвіду використання функційного програмування для розробки програмного забезпечення.

Предмет дисципліни — реалізація теоретичних основ функційного програмування у мові Haskell; набуття студентами досвіду використання мови програмування Haskell для розробки програмного забезпечення.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Основи програмування», «Комп'ютерна дискретна математика», «Алгоритми та структури даних».

Постреквізити дисципліни. Компетенції, отримані студентами в процесі вивчення дисципліни можуть бути використані при виконанні дипломної роботи.

3. Зміст навчальної дисципліни

Розділ 1. Вступ до функційного програмування

Тема 1.1. Вступ до ФП.

Тема 1.2. Інструментарій функційного програмування.

Тема 1.3. Модулі в Haskell.

Розділ 2. Основи мови Haskell

Тема 2.1. Мова програмування Haskell

Тема 2.2. Конструювання функції в Haskell.

Тема 2.3. Списки

Тема 2.4. Функції вищого порядку.

Тема 2.5. Типи

Тема 2.6. Контейнерні типи

Тема 2.7. Класи типів

Тема 2.8. Класи типів Semigroup, Monoid.

Тема 2.9. Класи типів Functor, Applicative.

Тема 2.10. Монади

Тема 2.11. Базові засоби введення/виведення.

Розділ 3. Розробка Haskell-застосунків

Тема 3.1. Робота з текстовими даними.

Тема 3.2. Безпечний доступ до баз даних.

Тема 3.3. Веб-застосунки.

Розділ 4. Концептуальні засади функційного програмування

Тема 4.1. Поняття про лямбда — числення.

4. Навчальні матеріали та ресурси

Основна література

1. Стативка Ю.І. Мова програмування Haskell. Базові типи і класи [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 Інженерія програмного забезпечення / Ю. І. Стативка ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 959 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 78 с. – Назва з екрана.
URI : <https://ela.kpi.ua/handle/123456789/63781>
2. Will Kurt Get Programming with Haskell. - Shelter Island: Manning Publications Co., 2018. – 616 p.
3. Alejandro Serrano Mena Practical Haskell: A Real World Guide to Programming (Second Edition) - Utrecht: Apress, 2019. – 600 p.

4. Vitaly Bragilevsky : CoverHaskell in Depth — Shelter Island: Manning Publications Co., 2021. – 664 p.

Додаткова література

5. Bryan O'Sullivan, Don Stewart, John Goerzen Real World Haskell – Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo: O'Reilly Media, 2009. — 712 p.

Інформаційні ресурси

6. Офіційна сторінка книги “Real World Haskell” <http://book.realworldhaskell.org/>
7. Офіційна сторінка Haskell - <https://www.haskell.org/>
8. Документація Haskell - <https://www.haskell.org/documentation>
9. Glasgow Haskell Compiler - <https://www.haskell.org/ghc>
10. CABAL - <https://www.haskell.org/cabal/>
11. Stack - <http://docs.haskellstack.org/>
12. Haskell-пакети - <https://hackage.haskell.org/>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Вступ до функційного програмування

Лекція 1. Вступ до ФП.

Історія функційного програмування. Основні властивості функційних мов програмування. Мова програмування Haskell та її реалізації. GHC. Робота з ghci. Сесії та сценарії (скрипти).

Лекція 2. Інструментарій функційного програмування.

GHC, Cabal (Common Architecture for Building Applications and Libraries), cabal-install, Haskell Platform, stack, Visual Studio Code, Haskero, Intero. Структура проекту.

Лекція 3. Модулі в Haskell.

Модулі. Структура модуля. Списки експорту. Імпортування. Простори імен. Модуль Prelude.

Розділ 2. Основи мови Haskell

Лекція 4. Мова програмування Haskell

Основні типи та конструкції мови Haskell. Синтаксис та двовимірний синтаксис. Відкладені обчислення. Поняття відкладених (ледачих) обчислень. Застосування відкладених обчислень.

Лекція 5. Конструювання функції в Haskell.

Рекурсивні функції. Означення функцій за допомогою шаблонів. Основні принципи використання рекурсії. Означення функцій шляхом зіставлення зі зразком (паттерном). Зразки на списках. Рекурсивні функції на списках. Використання накопичувача (акумулятора). Взаємна рекурсія.

Лекція 6. Списки

Конструювання списків. Генератори списків. Дот-нотація. Функції над списками. Data.List.

Лекція 7 Функції вищого порядку.

Поняття функції вищого порядку. Функції `map`, `filter`, `foldr`, `takeWhile`, `dropWhile`, `zipWith`, оператор аплікації. Каринг, часткова параметризація. Композиція функцій.

Лекція 8. Типи

Поняття типу. Множина значень та допустимих операцій. Означення нових типів даних: `data`, `newtype`, `type`. Успадкування методів класу типів (`deriving`). Побудова втілень класу типів (`instance`). Означення параметризованих (поліморфних) типів. Означення рекурсивних типів.

Лекція 9. Контейнерні типи

Типи даних `Maybe`, `Either`, списки, кортежі.

Лекція 10. Класи типів

Поняття класу типів. Основні класи типів: `Eq`, `Ord`, `Show`, `Read`, `Num`, `Integral`, `Fractional`, `Enum`. Означення нових класів типів. Оператори.

Лекція 11. `Semigroup`. `Monoid`.

Класи типів `Semigroup`. `Monoid`.

Лекція 12. `Functor`, `Applicative`

Класи типів `Functor`, `Applicative`.

Лекція 13. Монади

Поняття монади. Клас типів `Monad`. Методи та мінімальне повне визначення. Типові втілення класу типів. До-нотація.

Лекція 14. Базові засоби введення/виведення

Тип `IO`. Модуль `System.IO`. Робота з потоками введення/виведення. Робота з файлами. Обробка та створення винятків.

Розділ 3. Розробка Haskell-застосунків

Лекція 15. Доступ до баз даних

Засоби доступу до БД. З'єднання. Запити. Маніпуляція даними.

Лекція 16. Веб-застосунки.

Веб-екосистема Haskell. `Front-end` та `back-end` в Haskell-застосунках.

Лекція 17. Робота з текстовими даними.

Текстові типи даних. Створення та парсинг текстів.

Розділ 4. Концептуальні засади функційного програмування

Лекція 18. Поняття про лямбда — числення.

Основи лямбда - числення. Причини використання формалізації лямбда-числення. Поняття лямбда - виразу. Каррування. Вільні і зв'язані змінні, вирази. Лямбда-числення як формальна система. Підстановки. Конверсії. Рівність лямбда-термів. Екстенціональність. Редукція лямбда-термів. Редукційні стратегії. Комбінатори.

Практичні заняття

Практичне заняття 1. Підготовка інструментарію для програмування мовою Haskell. Stack. GHC. VS Code. Створення проекту. Конструювання функцій. Списки, кортежі, рекурсія.

Практичне заняття 2. Конструювання функцій.

Означення функцій шляхом зіставлення зі зразком (паттерном). Зразки на списках. Рекурсивні функції на списках. Використання накопичувача (акумулятора).

Практичне заняття 3. Конструювання функцій. Списки, кортежі, рекурсія.

Представлення роботи комп'ютерного практикуму № 1 (**КП 1**).

Практичне заняття 4. Функції вищого порядку

Представлення **КП 2**.

Практичне заняття 5. Типи даних. Створення власних нетривіальних типів даних.

Практичне заняття 6. Класи типів. Створення класів типів

Представлення **КП 3**.

Практичне заняття 7. Використання монад.

Програмування з використанням монадних типів.

Практичне заняття 8. Haskell-застосунки.

Представлення **КП 4**.

Практичне заняття 9. Підсумкове заняття.

Перелік робіт комп'ютерного практикуму

1. КП 1. Конструювання функцій. Списки, кортежі, рекурсія..
2. КП 2. Функції вищого порядку. Частково застосовані функції.
3. КП 3. Типи і класи типів..
4. КП 4. Розробка Haskell-застосунків.

Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного вирішення задач.

Модульна контрольна робота (МКР) виконується на шістнадцятому тижні семестру і проводиться у формі тестування.

Самостійна робота студента

Приблизний розподіл часу СРС

№ з/п	Вид самостійної роботи	Кількість	Кількість годин СРС
1	Виконання робіт комп'ютерного практикуму	4	24
2	Підготовка до практичних занять	9	36
3	Підготовка до МКР	1	6
Разом			66

Політика та контроль

6. Політика навчальної дисципліни (освітнього компонента)

Система вимог при вивченні дисципліни:

- правила відвідування занять: заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.
- правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
- політика дедлайнів та перескладань: якщо студент не проходив або не з'явився на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;
- політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Функційне програмування (Haskell)»;
- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.

7. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Система рейтингових (вагових) балів та критерії оцінювання

Поточний контроль: активність — питання і відповіді на лекційних заняттях, питання і відповіді та участь у ревію на практичних заняттях; МКР; виконання завдань до практичних занять; виконання та представлення робіт комп'ютерного практикуму.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу ("атестація").

Семестровий контроль: залік.

Умови допуску до семестрового контролю: семестровий рейтинг — не менше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
R < 40	Не допущено

- Активність

Активність (частота, змістовність) участі студента у процесі обговорення відповідних тем на заняттях — питання і відповіді на лекційних заняттях, питання і відповіді та участь у ревію на практичних заняттях; виконання завдань до практичних занять; оцінюється, максимум, 20 балами, які може отримати кожен студент за семестр.

- Роботи комп'ютерного практикуму

Максимальна кількість балів за виконання кожної роботи комп'ютерного практикуму становить 15 балів.

Критерії оцінювання:

Виконання робіт комп'ютерного практикуму:

- виконана у повному обсязі – 15 балів;
- виконана частково — відповідна частка від максимальної кількості балів.

- Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 20 балів.

Якість виконання роботи:

- виконана у повному обсязі з необхідними текстовими поясненнями дій та результатів – максимальна кількість балів;
- виконана частково з поясненнями — відповідна частка від максимальної кількості балів;
- виконана без текстових пояснень дій та результатів – не більше чотирьох балів.

- Творче завдання

Студент може обрати додаткове завдання творчого характеру, результати виконання якого можуть бути опубліковані у наукових виданнях, або повідомлені на студентській науковій конференції з публікацією тез. Максимальна кількість балів за виконання творчого завдання — 30 балів, за умови, що загальна кількість балів не перевищує 100 балів.

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{АКТ}} + r_{\text{КП}} + r_{\text{МКР}} + r_{\text{ТВЗ}} = 20 + 60 + 20 + (\text{до } 30) = 100 \text{ балів.}$$

Робочу програму навчальної дисципліни (силабус) «Функційне програмування (Haskell)»:

Склав доцент кафедри ІПЗЕ, к.т.н., доц. Стативка Юрій Іванович

Ухвалено кафедрою ІПЗЕ (протокол № 42 від 18 червня 2025 р.)

Погоджено Методичною комісією НН ІАТЕ ім.Ігоря Сікорського
(протокол № 8 від 20 червня 2025 р.)