

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньої програми



І. Б. Туркін
(підписи та прізвище)

« 30 » 08 2021 р.

**СИЛАБУС ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Логічне та функціональне програмування
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»
(код та найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Рівень вищої освіти: *перший (бакалаврський)*

Силабус введено в дію з 01.09.2021 року

Харків – 2021 р.

Розробник: Кузнецова Ю. А., доцент каф. 603, к.т.н., доцент 
(прізвище та ініціали, посада, науковий ступінь і вчене звання) (підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії
програмного забезпечення (№ 603)

(назва кафедри)

Протокол № 2 від « 31 » серпня 2021 р.

Завідувач кафедри, д.т.н., професор І. Б. Туркін
(науковий ступінь і вчене звання) (підпис) (ініціали та прізвище)

Погоджено з представником здобувачів освіти:

Представник студентського самоврядування Д. В. Колодій
(підпис) (ініціали та прізвище)

1. Загальна інформація про викладача



Кузнецова Юлія Анатоліївна доцент каф. 603, к.т.н., доцент.

Основні навчальні курси, що веде викладач: «Людино-машинна взаємодія» (24 години), «Інформаційні технології розроблення програмного забезпечення» (32 години), «Логічне та функціональне програмування» (32 години) щорічно. Розробила методичне забезпечення лекційного матеріалу, лабораторних і практичних робіт.

Має **79** публікацій, з них **74** наукових та **11** навчально-методичного характеру, у тому числі **15** наукових праць у фахових наукових виданнях України та **7** публікацій, які включено до наукометричних баз **Scopus** і **WebOfScience**.

Напрями наукових досліджень: інтернет речей, доповнена реальність, високонавантажені системи, контейнеризація та безсерверні обчислення, UI/UX дизайн, інтерактивне мультимедійне навчальне тестування.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 7 семестр.

Обсяг дисципліни: 4 кредити ЄКТС/ 120 годин, у тому числі аудиторних – 56 год, самостійної роботи здобувачів – 64 год.

Форма здобуття освіти – денна, заочна, дистанційна.

Дисципліна – вибіркова.

Види навчальної діяльності – лекції, лабораторні роботи, самостійна робота здобувача.

Види контролю – модульний контроль, іспит.

Мова викладання – українська.

Пререквізити – «Комп'ютерна дискретна математика», «Основи програмування».

Кореквізити – «Основи BIG Data», «Проектування виробничих експертних систем».

3. Мета та завдання навчальної дисципліни

Мета: вивчення та практичне засвоєння засобів функціонального та логічного програмування для розв'язування практичних та наукових задач.

Завдання: засвоєння концепцій та методології побудови аплікативного програмного забезпечення на базі системотехнічних, архітектурних, технологічних принципів і новітніх інструментальних засобів розробки, та логічних і функціональних мов програмування.

Компетентності, які набуваються:

Загальні компетентності:

- ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК02. Здатність застосовувати знання у практичних ситуаціях.
- ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.
- ЗК05. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Фахові компетентності:

- ФК07. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- ФК14. Здатність до алгоритмічного та логічного мислення.

Очікувані результати навчання:

- 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.
- ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовий модуль 1. Основи функціонального програмування.

Тема 1. Введення до функціонального програмування. *Історія розвитку функціонального програмування (ФП). Сучасний стан теорії ФП. Переваги та недоліки функціонального підходу. Основні властивості функціональних мов програмування. Приклади функціональних мов програмування. Типові задачі, що вирішуються методами ФП. Предмет вивчення і задачі дисципліни. Місце дисципліни в навчальному процесі.*

Тема 2. Базові принципи функціональної мови програмування Haskell. *Поняття списку у функціональному програмуванні. Базисні операції для роботи зі списками. Списки й облікові структури. Програмна реалізація списків у функціональних мовах. Списки в мові Haskell. Генератори списків і математичні послідовності. Коротке.*

Функція – основний об’єкт функціонального програмування. Угоди по іменуванню об’єктів у мові Haskell. Опис і визначення функцій мовою Haskell. Зразки й клози. Передача параметрів і повернення значень функціями. Виклики функцій. Використання λ -обчислень. Функція як об’єкт для передачі в інші функції.

Структури й типи даних. Синоніми типів. Типи функцій у функціональних мовах. Поліморфні типи. Часткове застосування. Ледачі (відкладені) обчислення мовою Haskell.

Охоронні вирази і конструкції. Розгалуження алгоритму. Двовимірний синтаксис. Локальні змінні для оптимізації коду функціональною мовою й мовою Haskell. Використання накопичуючого параметра (акумулятора) для оптимізації процесу обчислень. Головна й хвостова рекурсія.

Модулі й абстрактні типи даних. Модулі як способи структуризації й організації програм мовою Haskell. Імпорт і експорт даних за допомогою модулів. Приховування даних. Абстрактні типи даних і інтерфейси.

Тема 3. Класи та їх екземпляри у мові Haskell. *Симбіоз парадигм функціонального й об’єктно-орієнтованого програмування. Підтримка мовою Haskell об’єктно-орієнтованих механізмів й методів.*

Параметричний поліморфізм даних. Поняття класу і його реалізації в мові Haskell. Приклади параметричного поліморфізму в імперативних і функціональних мовах, а також у мові Haskell.

Класи в мові Haskell як спосіб абстракції дійсності. Розширений опис поняття класу в мові Haskell. Методи класу – шаблони функцій для реалізації обробки даних. Мінімальний опис методів класу й зв’язок методів. Визначення класів.

Спадкування й реалізація. Спадкування класів і спадкування методів. Екземпляри класів – реалізація інтерфейсів, надаваних реалізованим класом. Реалізація методів для обробки даних. Клас – шаблон типу, реалізація класу – тип даних. Реалізація для існуючих типів. Сорти типів.

Стандартні класи мови Haskell. Короткий опис всіх стандартних класів, розроблених для полегшення програмування мовою Haskell. Дерево спадкування стандартних класів. Типові способи використання стандартних класів мови Haskell. Реалізація стандартних класів – типи в мові Haskell.

Порівняння з іншими мовами програмування. Порівняння понять «клас» і «реалізація класу» у мові Haskell з об’єктно-орієнтованими мовами програмування (на прикладі мов C++ і Java, а також деяких інших мов). Глобальні відмінності поняття «клас» у функціональних і об’єктно-орієнтованих мовах.

Тема 4. Комбінаторна логіка й λ -обчислення. Комбінатори й обчислення за допомогою комбінаторів. Базиси в комбінаторній логіці. Використання базисних комбінаторів для вираження будь-яких обчислювальних процесів. Числа й інші математичні об'єкти у вигляді комбінаторів. Базові комбінатори.

Абстракція функцій як обчислювальних процесів. Функція – об'єкт математичного дослідження. Обчислювальний процес – функція. Опис функцій як λ -виражень. Вільні та зв'язані ідентифікатори. Застосування (аплікація) значень до λ -виражень. Редукція. Теза Черча-Тьюринга.

λ -обчислення як теоретична основа функціонального програмування. Інтенціонал і екстенціонал функцій. Правила виводу. Відповідності між обчисленнями функціональних програм і редукцією λ -виражень.

Кодування даних в λ -обчисленні. Механізм кодування даних в λ -обчисленні. Редукція й обчислення у функціональних мовах. Поняття редукції. Часткові обчислення з погляду редукції λ -виражень. Стратегія редукції й стратегія обчислень. Ледача редукція.

Модульний контроль

Модуль 2.

Змістовий модуль 2. Основи логічного програмування.

Тема 1. Методи логічного програмування. Предмет, об'єкт, цілі та задачі курсу. Теоретичні основи логічного програмування. Спектр задач, що вирішуються з використання парадигми логічного програмування. Поняття декларативного програмування. Структура логічної програми (основні конструкції, семантика, коректність). Мови логічного програмування. Практичне використання та перспективи розвитку.

Тема 2. Введення до мови ПРОЛОГ. Основні елементи мови. Структура ПРОЛОГ-програми. Використання предикатів. Факти та правила, запити та цілі. Прості об'єкти даних мови. Домени. Пошук та методи пошуку. Пошук усіх рішень. Стандартні предикати. Арифметичні розрахунки. Складені об'єкти даних. Функтори. Багаторівневі складені об'єкти даних. Налаштування програм.

Тема 3. ПРОЛОГ та рекурсивне програмування. Рекурсивне програмування. Рекурсивні структури даних. Списки та дерева. Дерева пошуку. Методи обходу дерев.

Тема 4. Робота з файлами та програмування баз даних у мові ПРОЛОГ. Створення динамічних баз даних. Занесення та видалення фактів з бази даних (знань). Запис фактів у файл. Текстові та бінарні файли. Робота з фактами з динамічної бази даних (знань) як з термами.

Тема 5. Представлення графів у ПРОЛОГ-програмах. Графи як складені структури даних. Представлення графів на мові ПРОЛОГ. Реалізація задач та методів теорії графів на мові ПРОЛОГ.

Тема 6. Стил ь програмування на мові ПРОЛОГ. ПРОЛОГ та процедурне програмування. Представлення фактів та правил у вигляді процедур. Модульне програмування. Розробка проектів. Рішення задач штучного інтелекту за допомогою мови ПРОЛОГ.

Модульний контроль

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

В таблицях 1 – 3 подано розподіл та обсяг аудиторної та самостійної робіт здобувачів.

Таблиця 1 – Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	Денна форма				
	Усього	У тому числі			
		л	п	лаб.	с.р.
1	2	3	4	5	6
Модуль 1					
Змістовий модуль 1. (Основи функціонального програмування)					
Тема 1. (Введення до функціонального програмування)	6	3		1	2
Тема 2. (Комбінаторна логіка й λ -обчислення)	20	5		5	10
Тема 3. (Класи та їх екземпляри у мові Haskell)	19	5		4	10
Тема 4. (Комбінаторна логіка й λ -обчислення)	15	3		2	10
Разом за змістовим модулем 1	60	16		12	32
Модуль 2					
Змістовий модуль 2. (Основи логічного програмування)					
Тема 1. (Методи логічного програмування)	5	2		2	1
Тема 2. (Введення до мови ПРОЛОГ)	5	2		2	1
Тема 3. (ПРОЛОГ та рекурсивне програмування)	15	3		2	10
Тема 4. (Робота з файлами та програмування баз даних у мові ПРОЛОГ)	15	3		2	10
Тема 5. (Представлення графів у ПРОЛОГ-програмах)	14	3		2	9
Тема 6. (Стиль програмування на мові ПРОЛОГ)	6	3		2	1
Разом за змістовим модулем 2	60	16		12	32
Усього годин	120	32		24	64

Таблиця 2 – Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Базові принципи функціональної мови програмування Haskell	7
2	Класи та їхні екземпляри у Haskell	7
3	Рекурсивне програмування у ПРОЛОГ	5
4	Робота з файлами та програмування баз даних у мові ПРОЛОГ	5
	Разом	24

Таблиця 3 – Теми для самостійної роботи

№ з/п	Назва теми	Кількість годин
1	Типові задачі, що вирішуються методами функціонального програмування	2
2	Локальні змінні для оптимізації коду функціональною мовою й мовою Haskell	10
3	Класи в мові Haskell як спосіб абстракції дійсності	10
4	λ -обчислення як теоретична основа функціонального програмування	5
5	Мови логічного програмування	3
6	Використання предикатів	4
7	Рекурсивні структури даних	10
8	Програмування баз даних у мові ПРОЛОГ	10
9	Реалізація задач та методів теорії графів на мові ПРОЛОГ	5
10	ПРОЛОГ та процедурне програмування	5
	Разом	64

5. Індивідуальні завдання

Не передбачено навчальним планом.

6. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: лабораторна робота, домашнє завдання.

2. За характером пізнавальної діяльності тих, хто навчається, – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.

3. За логікою пізнання – індуктивний, дедуктивний, метод аналогій, метод вивідних знань.

4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні питання до лабораторної роботи, тестування (традиційне та машинне).

Під час викладання дисципліни буде використовуватися презентаційна модель проведення лекційних занять з використанням проектору.

7. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.
4. Індивідуальна розрахункова робота (РР).
5. Форма підсумкового контролю успішності навчання: іспит (письмово).

8. Критерії оцінювання та розподіл балів, які отримують здобувачі

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	4...8	2	8...16
Модульний контроль	8...10	1	8...10
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 2			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	3...6	2	6...12
Модульний контроль	4...8	1	4...8
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з двох теоретичних питань (кожне питання 30 балів) та одного практичного питання (питання оцінюється в 40 балів).

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Здати індивідуальне завдання та основні лабораторні, написати модульні контрольні роботи. Знати основні принципи логічного та функціонального програмування, їх відмінність; базові поняття й визначення, що використовуються у логічному та функціональному програмуванні; методи й рівні подання даних, способи обробки й зберігання даних; основи технології програмування за допомогою програмних засобів, які використовуються у сучасних мовах логічного та функціонального програмування.

Добре (75-89). Твердо знати мінімум, здати всі лабораторні роботи, індивідуальне завдання та тестування. Досконало знати основні принципи логічного та функціонального програмування, їх відмінність; базові поняття й визначення, що використовуються у логічному та функціональному програмуванні; методи й рівні подання даних, способи обробки й зберігання даних; основи технології програмування за допомогою програмних засобів, які використовуються у сучасних мовах логічного та функціонального програмування. Досконало вміти орієнтуватися в сучасних мовах логічного та функціонального програмування, їхніх можливостях; обґрунтувати вибір мови програмування для рішення конкретних завдань; розробляти й тестувати програми із застосуванням програмних засобів, що використовуються у сучасних мовах логічного та функціонального програмування.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та вміти застосовувати їх.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

10. Методичне забезпечення

Підручники, навчальні посібники, навчально-методичні посібники, конспекти лекцій, методичні рекомендації з проведення лабораторних робіт тощо, які видані в Університеті, знаходяться за посиланням:

http://library.khai.edu/catalog?clear_all_params=0&mode=BookList&lang=rus&ext=no&theme_path=0%2C1665%2C14849%2C13426%2C10001&themes_basket=&ttp_themes_basket=&disciplinesearch=no&top_list=1&fullsearch fld=&author fld=&docname fld=&docname_cond=beginwith&theme_context=&theme_cond=all theme&theme_id=10001&is_ttp=0&combiningAND=0&step=20&tpage=1

Сторінка дисципліни знаходиться за посиланням:

<https://mentor.khai.edu/course/view.php?id=317>

11. Рекомендована література

Базова

Функціональне програмування:

1. Душкин Р. В. Функциональное программирование на языке Haskell // Р. В. Душкин. – М. : ДМК Пресс, 2009. – 608 с.
2. Роганова Н. А. Функциональное программирование: учеб. пособ. для студентов высших учебных заведений // Н. А. Роганова. – М. : ГИНФО, 2012. – 260 с.
3. Филд А. Функциональное программирование // А. Филд, П. Харрисон. – М. : Мир, 2013. – 638 с.

Логічне програмування:

1. Адаменко А. Н. Логическое программирование и Visual Prolog // А. Н. Адаменко, А. М. Кучуков. – СПб. : БХВ-Петербург, 2013. – 992 с.

2. Братко И. Алгоритмы искусственного интеллекта на языке PROLOG = Prolog Programming For Artificial Intelligence // И. Братко. – М. : Вильямс, 2014. – 640 с.
3. Ин Ц. Использование Турбо-Пролога // Ц. Ин., Д. Соломон. – М. : Мир, 2008. – 608 с.
4. Клоксин У. Программирование на языке Пролог // У. Клоксин, Д. Меллиш. – М. : Мир, 2007. – 336 с.
5. Стерлинг Л. Искусство программирования на языке Пролог // Л. Стерлинг, Э. Шапиро. – М. : Мир, 2010. – 235 с.

Додаткова

Функціональне програмування:

1. Городняя Л. В. Основы функционального программирования. Курс лекций – М. : Интернет-университет информационных технологий, 2014. – 280 с.
2. Филд А., Харрисон П. Функциональное программирование = Functional Programming. – М. : Мир, 2013. – 637 с.
3. John Harrison. Функциональное программирование. Курс лекций = Functional Programming. – 2007.

Логічне програмування:

1. Себеста Р. У. Основные концепции языков программирования = Concepts of programming languages. – 5-е изд. – М. : Вильямс, 2010.
2. ISO / IEC 13211-1:1995 / COR3 : 2017. Information technology – Programming languages – Prolog – Part 1: General core.
3. INCITS / ISO / IEC 13211-2:2000 [S2016]. Information technology – Programming languages – Prolog – Part 2: Modules.

12. Інформаційні ресурси

Функціональне програмування:

1. <http://www.intuit.ru/studies/courses/29/29/info>

Логічне програмування:

1. <http://www.intuit.ru/studies/courses/12333/1180/info>
2. <http://www.intuit.ru/studies/courses/51/51/info>