

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
 Декан факультету ІТ
Говорущенко Т.О.
 1 вересня 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Функційне програмування


Назва

Статус дисципліни: вибіркова

Факультет – Факультет інформаційних технологій


Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
			Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС			Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття						
Д		парний	8.0	240	108	36	36	36		132	-	-	+	-
Разом			8.0	240	108	36	36	36		132			1	


Програма складена  Лисенком С.М.
 Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол № 28 від 2 серпня 2024 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем  Засорнова І.О.
 Підпис Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради  Говорущенко Т.О.
 Підпис Ініціали, прізвище

ВСТУП

Мета викладання дисципліни. Дисципліна “Функційне програмування” відноситься до циклу вибіркових дисциплін та забезпечує підготовку студентів з програмування з використанням концепцій та парадигм функційного програмування.

Метою курсу є вивчення основних концепцій функційного програмування на прикладі мови F#, таких як концепція незмінного стану, функцій вищого порядку, чистих функцій, опціональних типів і співставлення зі зразком, відкладених обчислень і нескінченних структур даних, а також лямбда-обчислень, та застосування цих концепцій при вирішенні прикладних задач із залученням методів штучного інтелекту.

Предмет дисципліни. Основи програмування мовою F# з використанням концепцій та парадигм функційного програмування та їх застосування при вирішенні прикладних задач із залученням методів штучного інтелекту.

Завдання дисципліни:

- навчити навичкам програмування мовою F# із застосуванням парадигм функційного програмування;
- навчити здійснювати вибір засобів функційного програмування та використовувати переваги функційного підходу до розробки і реалізації прикладних програмних систем;
- ознайомитися з технікою програмування задач штучного інтелекту з використанням парадигм функційного програмування.

Після вивчення дисципліни «Функційне програмування» студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

знати:

- теоретичні положення, що лежать в основі парадигм функційного програмування; основні концепції функційного програмування, такі як концепція незмінного стану, функцій вищого порядку, чистих функцій, опціональних типів і співставлення зі зразком, відкладених обчислень і нескінченних структур даних, а також лямбда-обчислень; принципи програмування задач штучного інтелекту з використанням парадигм функційного програмування.

вміти:

- здійснювати пошук, оброблення та аналіз інформації з різних джерел для розв'язання прикладних задач з використанням парадигм функційного програмування;
- застосовувати знання основних концепцій функційного програмування, таких як концепція незмінного стану, функцій вищого порядку, чистих функцій, опціональних типів і співставлення зі зразком, відкладених обчислень і нескінченних структур даних, а також лямбда-обчислень при розв'язуванні задач аналізу та синтезу програмних систем;
- застосовувати знання теоретичних положень, що лежать в основі парадигм функційного програмування, для ідентифікації, формулювання і розв'язування прикладних задач, використовуючи основні концепції функційного програмування, що є найбільш придатними для досягнення поставлених цілей;
- поєднувати теорію і практику функційного програмування, а також приймати оптимальні рішення при виробленні стратегії розв'язку прикладних задач з урахуванням виробничих інтересів;
- виконувати експериментальні дослідження з метою пошуку оптимальних шляхів розв'язку прикладних задач з використанням парадигм функційного програмування;
- системно мислити та застосовувати творчі здібності до формування нових ідей при програмуванні задач штучного інтелекту з використанням парадигм функційного програмування;
- оцінювати отримані результати та аргументовано захищати прийняті рішення при виборі засобів функційного програмування для розв'язку прикладних задач;

бути здатним:

- до абстрактного мислення, аналізу і синтезу при побудові розв'язків прикладних задач з використанням парадигм функційного програмування;
- до навчання та оволодіння сучасними знаннями з метою вироблення стратегії побудови розв'язків прикладних задач з використанням парадигм функційного програмування та застосовувати одержані знання у практичних ситуаціях;
- до розуміння концепцій та парадигм функційного програмування;
- застосовувати знання концепцій та парадигм функційного програмування у практичних ситуаціях;
- використовувати поняття концепцій функційного програмування, а також принципи програмування задач штучного інтелекту з використанням парадигм функційного програмування при розробленні програмного забезпечення на сучасних мовах програмування, зокрема F#;
- аргументувати вибір методів розв'язування прикладних задач та засобів функційного програмування, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;
- розв'язувати складні задачі під час розробки прикладного програмного забезпечення, що передбачає застосування концепцій та парадигм функційного програмування і характеризується комплексністю та невизначеністю умов.

ФУНКЦІЙНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Кредити ЄКТС	8,0
Форми здобуття освіти	Очна денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: знати теоретичні положення, що лежать в основі парадигм функційного програмування; основні концепції функційного програмування, такі як концепція незмінного стану, функцій вищого порядку, чистих функцій, опціональних типів і співставлення зі зразком, відкладених обчислень і нескінченних структур даних, а також лямбда-обчислень; принципи програмування задач штучного інтелекту з використанням парадигм функційного програмування; вміло застосовувати знання теоретичних положень, що лежать в основі парадигм функційного програмування, для ідентифікації, формулювання і розв'язування прикладних задач, використовуючи основні концепції функційного програмування, що є найбільш придатними для досягнення поставлених цілей; поєднувати теорію і практику функційного програмування, а також приймати оптимальні рішення при виробленні стратегії розв'язку прикладних задач; виконувати експериментальні дослідження з метою пошуку оптимальних шляхів розв'язку прикладних задач з використанням парадигм функційного програмування; використовувати поняття концепцій функційного програмування, а також принципи програмування задач штучного інтелекту з використанням парадигм функційного програмування при розробленні програмного забезпечення на сучасних мовах програмування, зокрема F#.

Зміст навчальної дисципліни. Основні концепції функційного програмування. Концепції незмінного стану, функцій вищого порядку, чистих функцій, опціональних типів, співставлення зі зразком, відкладених обчислень, нескінченних структур даних, лямбда-обчислень. Переваги функційного програмування. Основи функційного програмування. Реалізація ідей функційного програмування мовами програмування. Мова програмування F#. Структура програми. Оператори, арифметика, типи F#. Виведення типів. Функції. Каррінг. Рекурсія. Композиція функцій. Конвеєризація. Лямбда-вирази. Відкладені обчислення. Зіставлення зі зразком. Списки. Послідовності. КORTEжі. Розмічені об'єднання. Масиви. Основні стратегії розв'язування задач. Розв'язування задач із залученням методів штучного інтелекту. Машинне навчання.

Запланована навчальна діяльність: лекції – 17 год., лабораторні заняття – 68 год., практичні заняття – 17 год., самостійна робота – 138 год.; разом – 240 год.

Методи навчання: словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, частково-пошукові (практичні та лабораторні заняття), практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: захист лабораторних та практичних робіт, тестовий контроль.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – Springer, 2018. – 738 p.
2. Dixon, J. Mastering .NET Machine Learning / J. Dixon. – Packt Publishing Ltd, 2019. – 358 p.
3. Masood, A. Learning F# Functional Data Structures and Algorithms. / A. Masood. – Packt Publishing Ltd, 2020. – 206 p.
4. Mohri, M., Rostamizadeh, A., Talwalkar A. Foundations of Machine Learning / M. Mohri, A. Rostamizadeh, A. Talwalkar. – The MIT Press, 2020. – 432 p.
5. Mukherjee, S. F# for Machine Learning Essentials / S. Mukherjee. – Packt Publishing Ltd, 2018. – 194 p.
6. Murphy, K. P., Bach, F. Machine Learning: A Probabilistic Perspective / K. P. Murphy, F. Bach. – The MIT Press, 2019. – 1104 p.
7. Shalev-Shwartz, S., Ben-David, S. Understanding Machine Learning: From Theory to Algorithms / S. Shalev-Shwartz, S. Ben-David. – Cambridge University Press, 2020. – 449 p.
8. F# Language Reference [© Microsoft, 2020]. Доступ до ресурсу: <https://docs.microsoft.com/en-us/dotnet/fsharp/language-reference/>
9. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua/course/view.php?id=546>
10. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/page_lib.php

Викладач: кандидат техн. наук, доцент Бобровнікова К.Ю.

1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Денна форма			
	Лекції	Лабораторні роботи	Практичні роботи	СРС
<i>Шостий семестр</i>				
Тема 1. Основи функційного програмування. Введення в F#.	4	16	4	32
Тема 2. Колекції F#.	4	8	2	16
Тема 3. Засоби функційного програмування.	6	8	4	16
Тема 4. Методи штучного інтелекту. Функційне програмування та машинне навчання..	2	32	6	74
Підсумкове заняття.	1	4	1	-
Разом за 6-й семестр:	17*	68	17*	138

Примітка. * по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять)

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Шостий семестр</i>	
1.	Тема 1. Основи функційного програмування. Введення в F#. Лекція № 1. Загальний огляд мови F#. Побудова програми на мові F#. Значення пробільних символів. Функції. Узагальнені функції. Рекурсивні функції. Каррінг. Лямбда-вирази. Композиція функцій. Конвеєризація. Типи мови F#. Синтаксис типів. Функції перетворення типів. Арифметика F#. Бітові операції. Символи. Рядки. Умовні вирази. Вираз while ... do. Прості цикли for. Вираз for ... to. Цикли-перерахування for. Вираз for ... in. Літ. [2, 4-6]	2
2.	Тема 1. Основи функційного програмування. Введення в F#. Лекція № 2. Функційний підхід до програмування. Алгебраїчні типи і зіставлення зі зразком. Реалізація обчислення коренів квадратного рівняння: приклад на мові C# з використанням списку коренів; приклад на мові C# з використанням перерахування; приклад на мові F# з використанням алгебраїчного типу; приклад на мові C# з використанням інтерфейсу і успадкованих класів; приклад на мові F# з використанням інтерфейсу і успадкованих класів. Літ. [4, 5]	2
3.	Тема 2. Колекції F#. Лекція № 3. Кортежі. Колекції F#. Списки. Кортежі. Списки. Діапазони списків. Генератори списків. Функції модуля List. Агрегатні оператори модуля List. Згортки. Літ. [2, 4, 5]	2
4.	Тема 2. Колекції F#. Лекція № 4. Колекції F#. Послідовності. Масиви. Тип option. Послідовності. Вирази послідовності. Функції модуля Seq. Агрегатні оператори модуля Seq. Масиви. Зрізи масивів. Способи створення масивів. Функції модуля Array. Агрегатні оператори модуля Array. Багатомірні масиви. Прямокутні масиви. Невирівняні масиви. Літ. [4-6]	2
5.	Тема 3. Засоби функційного програмування. Лекція № 5. Засоби функційного програмування. Функції, які повертають функції. Взаємна рекурсія. Символьні оператори. Іменовані зразки. Зіставлення з літералами. Обмеження when. Групування зразків. Зіставлення структур даних. Необов'язкові значення. Зіставлення з груповими символами. Альтернативний синтаксис лямбда-виразів. Літ. [1-6]	2
6.	Тема 3. Засоби функційного програмування. Лекція № 6. Засоби функційного програмування. Розмічені об'єднання. Використання розмічених об'єднань для створення деревовидних структур. Методи і властивості розмічених об'єднань. Записи. Типи Lazy. Літ. [4]	2
7.	Тема 3. Засоби функційного програмування. Лекція № 7. Прикладне функційне програмування. Одиниці виміру. Активні шаблони. Використання модулів. Перетворення модулів в класи. Робота зі списками. Хвостова рекурсія. Програмування з застосуванням функцій. Функційні шаблони проектування. Літ. [4, 6]	2
8.	Тема 4. Методи штучного інтелекту. Функційне програмування та машинне навчання. Лекція № 8. Задачі теорії розпізнавання образів. Підходи до розпізнавання образів. Процедура розпізнавання. Розроблення системи розпізнавання. Методи розпізнавання. Перетворення зорових образів у цифровий код. Літ. [1, 3]	2
9.	Підсумкове заняття.	1
Разом за 6-й семестр:		17

2.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<i>Шостий семестр</i>		
1.	<i>Лабораторна робота №1.</i> Створення консольного проекту в F#. Створення функцій. Композиція функцій і конвеєризація. Каррінг. Типи даних F#.	8
2.	<i>Лабораторна робота №2.</i> Цикли та розгалуження в F#.	8
3.	<i>Лабораторна робота №3.</i> Кортежі. Колекції F#: списки, послідовності, масиви.	8
4.	<i>Лабораторна робота №4.</i> Засоби функційного програмування.	8
5.	<i>Лабораторна робота №5.</i> Застосування методів машинного навчання в задачах розпізнавання образів. Формування вибірки.	8
6.	<i>Лабораторна робота №6.</i> Застосування методів машинного навчання в задачах розпізнавання образів. Аргументація методів формування залежностей.	8
7.	<i>Лабораторна робота №7.</i> Застосування методів машинного навчання в задачах розпізнавання образів. Реалізація методу.	8
8.	<i>Лабораторна робота №8.</i> Застосування методів машинного навчання в задачах розпізнавання образів. Тестування і модифікація.	8
9.	Підсумкове заняття.	4
<i>Разом за 6-й семестр:</i>		68

2.3 Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
<i>Шостий семестр</i>		
1.	<i>Практична робота №1.</i> Основи F#. Створення функцій. Композиція і конвеєризація. Каррінг.	2
2.	<i>Практична робота №2.</i> Циклічні конструкції та розгалуження в F#.	2
3.	<i>Практична робота №3.</i> Колекції F#. Функції модуля List. Функції модуля Seq. Функції модуля Array.	2
4.	<i>Практична робота №4.</i> Алгебраїчні типи і зіставлення зі зразком.	2
5.	<i>Практична робота №5.</i> Задачі на розстановку фігур на шаховій дошці. Задача про вісім ферзів.	2
6.	<i>Практична робота №6.</i> Застосування методів машинного навчання в задачах розпізнавання образів. Формування вибірки.	2
7.	<i>Практична робота №7.</i> Створення програмного додатку розпізнавання рукописних цифр. Реалізація методу.	2
8.	<i>Практична робота №8.</i> Бінарна класифікація за допомогою методу k-NN. Розпізнавання зображень.	2
9.	Підсумкове заняття.	1
<i>Разом за 6-й семестр:</i>		17

2.4 Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту практичних та лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань тощо.

Номер тижня	Вид самостійної роботи	К-ть годин
	<i>Шостий семестр</i>	
1.	Опрацювання лекційного матеріалу. Підготовка до ЛР1.	8
2.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР1.	8
3.	Опрацювання лекційного матеріалу. Підготовка до ЛР2.	8
4.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР2.	8
5.	Опрацювання лекційного матеріалу. Підготовка до ЛР3.	8
6.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР3.	8
7.	Опрацювання лекційного матеріалу. Підготовка до ЛР4.	8
8.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР4.	8
9.	Опрацювання лекційного матеріалу. Підготовка до ЛР5.	8
10.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР5.	8
11.	Опрацювання лекційного матеріалу. Підготовка до ЛР6.	8
12.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР6.	8
13.	Опрацювання лекційного матеріалу. Підготовка до ЛР7.	8
14.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР7.	8
15.	Опрацювання лекційного матеріалу. Підготовка до ЛР8.	8
16.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР8.	8
17.	Опрацювання лекційного матеріалу. Підготовка до підсумкового лабораторного заняття. Підготовка до тестування.	10
	<i>Разом за 6-й семестр:</i>	138

3. МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням словесних, наочних, проблемно-пошукових методів; практичні та лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі заліку. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Процес оцінювання підготовленості студента можна розділити на етапи. Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань предметної області, вміє розширити їх. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо концепцій функційного програмування, здатна застосовувати базові засоби функційного програмування при розробленні програм, зокрема при розв'язуванні задач із залученням методів штучного інтелекту.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації.

Кожний вид роботи з дисципліни оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (тестування), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється під час проведення лабораторних занять; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінка „зараховано” А виставляється студенту, який глибоко засвоїв методи розв'язування практичних задач та вміє їх раціонально застосувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „зараховано” В отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „зараховано” С отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки „зараховано” D заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок застосування методів розв'язування практичних задач, але припустився неточностей. Вагається при відповіді на видозмінене запитання,

разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки „зараховано” Е заслугове студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички застосування технологій розв’язування практичних задач.

Оцінка „незараховано” FХ виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань.

Як правило, оцінка „незараховано” F виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота		Самостійна, індивідуальна робота	Семестровий контроль (залік)
VI семестр			
Лабораторні роботи №:	Практичні роботи №:	Тестовий контроль	
1-8	1-8	Т 1-4	за рейтингом
ВК:	0,7	0,1	0,2
			0

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–18	19-20
Оцінка	2	3	4	5

Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в онлайн режимі в модульному середовищі MOODLE. Викладач виставляє результати тестування згідно журналу оцінок модульного середовища MOODLE.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Інституцій на шкала балів	Інституцій на оцінка	Критерії оцінювання	
A	4,75-5,00	5	Зараховано	
B	4,25-4,74	4		Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків.
C	3,75-4,24	4		Добре – повне знання навчального матеріалу з кількома незначними помилками.
			Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками.	

D	3,25-3,74	3		Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією.
E	3.00-3,24	3		Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	Незараховано	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2		Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Побудова програми на мові F#.
2. Модулі. Вкладені модулі.
3. Простори імен.
4. Управління файлами з вихідними кодами F#.
5. Значення пробільних символів.
6. Взаємодія з бібліотеками .NET.
7. Прив'язка let.
8. Функції.
9. Параметри функцій. Виведення типів.
10. Узагальнені функції.
11. Область видимості.
12. Каррінг.
13. Рекурсивні функції.
14. Функційні значення.
15. Лямбда-вирази.
16. Композиція функцій. Прямий оператор композиції. Зворотний оператор композиції.
17. Конвеєризація. Прямий конвеєрний оператор. Зворотний конвеєрний оператор.
18. Типи мови F#. Синтаксис типів.
19. Функції перетворення типів.
20. Арифметика F#.
21. Бітові операції.
22. Символи.
23. Рядки.
24. Порівняння і рівність.
25. Булеві значення.
26. Умовні вирази.
27. Вираз while ... do.
28. Прості цикли for. Вираз for ... to.
29. Цикли-перерахування for. Вираз for ... in.
30. Зіставлення зі зразком.
31. Кортежі.
32. Списки.
33. Діапазони списків.
34. Генератори списків.
35. Функції модуля List.
36. Агрегатні оператори модуля List.
37. Згортки.
38. Тип option.
39. Послідовності.
40. Вирази послідовності.
41. Функції модуля Seq.
42. Агрегатні оператори модуля Seq.
43. Масиви.
44. Зрізи масивів.
45. Способи створення масивів.
46. Функції модуля Array.
47. Агрегатні оператори модуля Array.
48. Багатовимірні масиви. Прямокутні масиви. Невирівняні масиви.
49. Функції, які повертають функції.
50. Взаємна рекурсія.
51. Символьні оператори.

52. Іменовані зразки.
53. Зіставлення з літералами.
54. Обмеження when.
55. Групування зразків.
56. Зіставлення структур даних.
57. Необов'язкові значення.
58. Зіставлення з груповими символами.
59. Альтернативний синтаксис лямбда-виразів.
60. Розмічені об'єднання.
61. Використання розмічених об'єднань для створення деревовидних структур.
62. Методи і властивості розмічених об'єднань.
63. Записи.
64. Типи Lazy.
65. Методи штучного інтелекту.
66. Машинне навчання.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Функційне програмування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою, зокрема в модульному середовищі для навчання.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Abraham, Isaac. Get Programming with F#: A guide for .NET developers. Simon and Schuster, 2018.
2. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – Springer, 2018. – 738 p.
3. Brink, H., Richards, J., Fetherolf, M. Real-World Machine Learning / H. Brink, J. Richards, M. Fetherolf. – Manning Publications, 2019. – 264 p.
4. Dixon, J. Mastering .NET Machine Learning / J. Dixon. – Packt Publishing Ltd, 2019. – 358 p.
5. Liu, T. F# for C# Developers / T. Liu. – Pearson Education, 2018. – 640 p.
6. Marsland S. Machine Learning: An Algorithmic Perspective / S. Marsland. – Chapman and Hall/CRC, 2020. – 457 p.
7. Masood, A. Learning F# Functional Data Structures and Algorithms. / A. Masood. – Packt Publishing Ltd, 2020. – 206 p.
8. Mohri, M., Rostamizadeh, A., Talwalkar A. Foundations of Machine Learning / M. Mohri, A. Rostamizadeh, A. Talwalkar. – The MIT Press, 2020. – 432 p.
9. Mukherjee, S. F# for Machine Learning Essentials / S. Mukherjee. – Packt Publishing Ltd, 2018. – 194 p.
10. Murphy, K. P., Bach, F. Machine Learning: A Probabilistic Perspective / K. P. Murphy, F. Bach. – The MIT Press, 2019. – 1104 p.
11. Sestoft, P. Programming language concepts. Springer, 2017.
12. Shalev-Shwartz, S., Ben-David, S. Understanding Machine Learning: From Theory to Algorithms / S. Shalev-Shwartz, S. Ben-David. – Cambridge University Press, 2020. – 449 p.
13. F# Language Reference [© Microsoft, 2020]. Доступ до ресурсу: <https://docs.microsoft.com/en-us/dotnet/fsharp/language-reference/>

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету.