

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет імені Лесі Українки
Факультет інформаційних технологій і математики
Кафедра комп'ютерних наук та кібербезпеки

СИЛАБУС
нормативного освітнього компонента
СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ
підготовки здобувачів освіти
першого (бакалаврського) рівня
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми
Комп'ютерні науки та інформаційні технології

Силабус освітнього компонента «Спеціалізовані мови програмування» підготовки бакалавра, галузі знань 12 Інформаційні технології, спеціальності 122 Комп'ютерні науки, за освітньою програмою Комп'ютерні науки та інформаційні технології.

Розробники:

Глинчук Л. Я., доцент кафедри комп'ютерних наук та кібербезпеки, к.ф.-м.н.

Погоджено

Гарант освітньо-професійної програми:  Гришанович Т. О.

Силабус освітнього компонента затверджено на засіданні кафедри комп'ютерних наук та кібербезпеки

протокол № 2 від 25.09.2024 р.

Завідувач

кафедри:



Гришанович Т. О.

I. Опис освітнього компонента

Найменування показників	Галузь знань, спеціальність, освітньо-професійна /освітньо-наукова/освітньо-творча програма, освітній рівень	Характеристика освітнього компонента
Денна форма навчання	12 Інформаційні технології 122 Комп'ютерні науки Комп'ютерні науки та інформаційні технології бакалавр	Нормативна
Кількість годин/кредитів 150 / 5		Рік навчання 2
		Семестр 4-ий
ІНДЗ: є		Лекції 46 год.
		Лабораторні 50 год.
		Самостійна робота 44 год.
		Консультації 10 год.
Форма контролю: екзамен		
Мова навчання: українська		

II. Інформація про викладача ПП

Глинчук Людмила Ярославівна

Науковий ступінь кандидат фізико-математичних наук

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація hlynchuk.ludmila@vnu.edu.ua

Дні занять <http://194.44.187.20/>

III. Опис освітнього компонента

- Анотація курсу.** Освітній компонент «Спеціалізовані мови програмування» відноситься до переліку дисциплін циклу професійної підготовки. Навчальна дисципліна присвячена сучасним технологіям програмування: об'єктно-орієнтованого, логічного та функціонального. Дисципліна покликана формувати у здобувача знання та вміння щодо мов програмування Scala, Prolog, Kotlin та виробити техніку програмування для подальшого застосування при розробці програм.
- Мета і завдання освітнього компонента:** метою дисципліни «Спеціалізовані мови програмування» є вироблення у студентів навичок розробки програм із використанням різних парадигм програмування та мов, що підтримують ці парадигми. За результатами вивчення освітнього компонента здобувач повинен вміти реалізовувати алгоритми розв'язування задач в галузі комп'ютерних наук, самостійно вибираючи парадигму програмування та використовуючи засоби відповідних мов програмування. Завданнями дисципліни “Спеціалізовані мови програмування” є

формування у здобувачів знань про етапи розробки програм; про методи об'єктно-орієнтованого, подійно-орієнтованого, логічного та функціонального програмування; про вбудовані типи даних у мовах програмування Scala, Prolog, Kotlin. Також до завдань відноситься формування у здобувачів вмінь реалізовувати алгоритмічні конструкції, які підтримуються конкретними мовами програмування, вмінь реалізовувати алгоритм розв'язання задачі із застосуванням засобів мов програмування, вмінь реалізовувати типи даних користувача засобами конкретних мов програмування.

3. Результати навчання.

Загальні компетентності:

ЗК 1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ЗК 3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК 6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК 8. Здатність генерувати нові ідеї (креативність).

ЗК 10. Здатність бути критичним і самокритичним.

ЗК 11. Здатність приймати обґрунтовані рішення.

ЗК 12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ЗК 13. Здатність діяти на основі етичних міркувань.

ЗК 16. Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь яких інших проявів недоброчесності.

СК 1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК 8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК 12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

ПР 1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР 9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

4. Структура освітнього компонента.

Семестр 1

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
Змістовий модуль 1. Основні методи та підходи у функціональному програмуванні						
Тема 1. Вступ до функціонального програмування. Мови функціонального програмування.	2	2				ДС
Тема 2. Вступ до мови Scala. Змінні, типи даних, основні конструкції (розгалуження, цикли, функції). Бібліотеки Math та Random. Колекції: масиви, списки, тапли, множини, словники.	8	2	4	2		Звіт по лаб. роботі/4
Тема 3. Класи та об’єкти у мові Scala.	8	2	4	2		Звіт по лаб. роботі/4
Тема 4. Функціональне програмування. Вкладення функції у функцію. Аргументи по замовчуванню та іменовані аргументи. Змінна кількість аргументів. Процедури.	8	2	4	2		Звіт по лаб. роботі/4
Тема 5. Анонімні функції. Частковий виклик функцій. Замикання. Хвостова рекурсія. Функції вищих порядків. Каррінг.	8	2	2	2	2	Звіт по лаб. роботі/4
Тема 6. Співставлення із зразком (pattern matching). Case-класи та case-об’єкти.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 7. Лінійні обчислення у функціональній мові. Моноїди.	6	2	2	2		Звіт по лаб. роботі/2
Разом за модулем 1	50	14	18	14	4	20
Змістовий модуль 2. Основні методи та підходи у логічному програмуванні						
Тема 1. Вступ до логічного програмування. Мови логічного програмування.	6	2	2	2		Звіт по лаб. роботі/2
Тема 2. Синтаксичні особливості мови Prolog. Вбудовані предикати. Структура Prolog-програми та приклади	6	2	2	2		Звіт по лаб. роботі/2

Тема 3. Рекурсивне представлення даних і програм. Списки.	6	2	2	2		Звіт по лаб. роботі/2
Тема 4. Метод відсікання та метод повернення в логічному програмуванні.	6	2	2	2		Звіт по лаб. роботі/2
Тема 5. Методи організації повторень та рекурсії мовою Prolog.	8	2	4	2		Звіт по лаб. роботі/4
Тема 6. Побудова прикладних програм мовою логіки: створення бази даних.	6	2	2	2		Звіт по лаб. роботі/4
Тема 7. Побудова прикладних програм мовою логіки: побудова експертних систем.	8	2	2	2	2	Звіт по лаб. роботі/4
Разом за модулем 2	46	14	16	14	2	20
Змістовий модуль 3. Основи Kotlin для Android						
Тема 1. Вступ до Kotlin і Android. Огляд платформи Android. Встановлення середовища розробки (Android Studio). Створення першого проєкту Android на Kotlin.	4	2		2		ДС
Тема 2. Основи синтаксису Kotlin. Типи даних та змінні. Оператори, умовні вирази (if, when). Цикли (for, while). Функції та основи роботи з ними.	5	2	2		1	Звіт по лаб. роботі/2
Тема 3. Об'єктно-орієнтоване програмування в Kotlin. Класи та об'єкти. Конструктори, властивості та методи. Наслідування, абстрактні класи та інтерфейси. Уявлення про sealed та data клас.	7	2	2	2	1	Звіт по лаб. роботі/2
Тема 4. Колекції та робота з ними. Списки, множини та мапи. Операції над колекціями (filter, map, reduce). Іммутові та змінні колекції.	6	2	2	2		Звіт по лаб. роботі/2
Тема 5. Обробка винятків і Null безпечність. Модель обробки винятків в Kotlin. Nullable типи та оператори (safe call, Elvis). Оператор try-catch та використання finally.	7	2	2	2	1	Звіт по лаб. роботі/2
Тема 6. Функціональне програмування в Kotlin. Лямбда-функції та анонімні функції. Вищі функції. Inline-функції та кросінлайнові функції. Корутини в Kotlin: основи асинхронного програмування.	7	2	2	2	1	Звіт по лаб. роботі/2
Тема 7. Основи розробки інтерфейсу	6	2	2	2		Звіт по лаб. роботі/2

в Android. Основи XML та створення UI. Робота з Activity та Fragment. Взаємодія з елементами інтерфейсу (кнопки, текстові поля). Управління ресурсами та стилями.						
Тема 8. Архітектурні компоненти Android. Життєвий цикл Activity та Fragment. ViewModel та LiveData. DataBinding та робота з RecyclerView.	6	2	2	2		Звіт по лаб. роботі/2
Тема 9. Робота з мережами та базами даних. Використання бібліотеки Retrofit для API-запитів. Основи роботи з базою даних (Room). Кешування даних та робота з ресурсам.	6	2	2	2		Звіт по лаб. роботі/2
Разом за модулем 3	54	18	16	16	4	20
Види підсумкових робіт						Бал
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
Всього годин/Балів	150	46	50	44	10	100

Методи контролю*: ДС – дискусія, ДБ – дебати, Т – тести, ТР – тренінг, РЗ/К – розв’язування задач/кейсів, ІНДЗ/ІРС – індивідуальне завдання/індивідуальна робота здобувача освіти, РМГ – робота в малих групах, МКР/КР – модульна контрольна робота/ контрольна робота, Р – реферат, а також аналітична записка, аналітичне есе, аналіз твору тощо.

5. Завдання для самостійного опрацювання.

Самостійна робота здобувачів включає в себе:

Опрацювання лекційного матеріалу.	10 год
Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль.	
Підготовка до лабораторних занять, виконання домашніх завдань.	14 год
Перевірка здійснюється під час лабораторних занять.	
Систематизація вивченого матеріалу перед іспитом.	6 год
Перевірка здійснюється під час іспиту.	
Вивчення тем, що не розглядаються в курсі лекцій.	14 год

Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

№ з/п	Тема	Кількість годин
1	Аналіз великих даних у Scala.	2
2	Мова Scala та паралельні обчислення.	2
3	Монади. Функтори.	2
4	Типізація та Каррі. Типізація за Чьорчем.	2
5	Основні можливості та особливості мови F#.	2
6	Мова програмування DataLog. Особливості та сфери застосування.	2
7	Сфера застосування видів програмування: функціональне, логічне, подійно-орієнтоване, реактивне, узагальнене програмування.	2
	Всього	14

IV. Політика оцінювання Політика викладача щодо студента

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загальноприйнятих моральних принципів, правил поведінки та корпоративної культури; підтримувати атмосферу доброзичливості, відповідальності, порядності й толерантності. Атмосфера на заняттях повинна бути творчою, відкритою до конструктивної критики. Недопустимі запізнення на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття; списування. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Кожен студент повинен бути учасником дистанційного курсу <https://moodle.vnu.edu.ua/course/view.php?id=1383>). Завдання для практичного виконання (лабораторні роботи, ІНДЗ, самостійні роботи), завдання підсумкового контролю (тести, контрольні роботи, що передбачають розробку програм) здаються із використанням засобів дистанційного курсу.

Політика щодо академічної доброчесності

Під час навчання учасники освітнього процесу зобов'язані дотримуватися академічної доброчесності: етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової діяльності.

Дотримання академічної доброчесності здобувачами передбачає: самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю (для осіб з особливим освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей); посилення на джерела інформації у разі використання ідей, тверджень, відомостей; дотримання норм законодавства про авторське право; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності.

Під час оцінювання результатів навчання студенти не користуються забороненими засобами (мобільний телефон, планшет, конспект, навчальна література, інші джерела

інформації, в тому числі Інтернет-ресурси), самостійно виконують запропоновані завдання. При виконанні лабораторних робіт з курсу здобувачі мають право використовувати власні ноутбуки, якщо вони підтримують необхідне програмне забезпечення.

Політика щодо дедлайнів та перескладання

Якщо здобувач вищої освіти був відсутній на заняттях з будь-якої причини, він/вона вивчають теоретичний матеріал самостійно використовуючи навчальні посібники, конспекти лекцій, матеріали дистанційних курсів “Програмування 1”, “Програмування 2”, “Програмування 3”, “Програмування 4”, розміщених на платформі дистанційного навчання Moodle, виконують всі домашні завдання. Прозвітуватися про виконання завдань можна, використовуючи вищезазначені дистанційні курси, або під час консультацій, одночасно при цьому з’ясувати незрозумілі моменти, задати запитання викладачу. Існує можливість використання форуму дистанційного курсу. Перескладання контрольних робіт та тестувань заборонено.

V. Підсумковий контроль

Підсумковою формою контролю освітнього компонента “Спеціалізовані мови програмування” є іспит 4 семестрі.

Оцінювання навчальних досягнень здійснюється за 100 бальною шкалою. Оцінка включає в себе поточний контроль (оцінюється робота на парах, вчасне і якісне виконання домашніх завдань) та підсумковий контроль (самостійне виконання індивідуальних завдань, контрольні роботи, перевірка теоретичної підготовки у формі тестування, ІНДЗ). Максимальна кількість балів, яку може отримати здобувач під час поточного оцінювання за семестр – 40 балів. Максимальна кількість балів, яку може отримати здобувач за підсумковий контроль за семестр складає 60 балів.

Якщо за результатами семестру накопичено не менше 75 балів і студент погоджується із цим результатом, то оцінка за семестр може виставлятися без складання іспиту. В іншому випадку студент складає іспит; максимальна кількість балів, яку можна отримати на іспиті – 60 балів. Вони замінюють бали модульного семестрового контролю, поточний семестровий контроль при цьому зберігається.

Іспит передбачає виконання тестових завдань (30 питань по одному балу) та розв’язування задач (розробка програми) – 3 задачі по 10 балів. Викладач залишає за собою право ставити уточнюючі питання під час відповіді студента та просити прокоментувати програмний код. Студент повинен використати засоби тієї мови програмування, яка вказана у завданні.

Передбачається виконання індивідуальних завдань. Варіант ІНДЗ включає себе набір задач, що охоплюють одну або кілька близьких тем. Або одне завдання, розв’язання якого вимагає самостійного опрацювання невеликих тем.

Приклад ІНДЗ:

Розробити програму, що реалізує функціональну обробку даних клієнтів. Система повинна приймати список клієнтів, виконувати різні операції (фільтрація, трансформація, агрегація) і повертати результати в структурованому вигляді.

Вимоги до системи:

Клас Client:

Створити клас Client з властивостями:

name: String

age: Int

balance: Double

Функції:

Реалізувати функцію `filterAdults(clients: List[Client]): List[Client]`, яка повертає список клієнтів старше 18 років.

Реалізувати функцію `transformBalances(clients: List[Client]): List[Double]`, яка повертає список балансів клієнтів, помножених на 1.2 (для обліку податків).

Реалізувати функцію `totalBalance(clients: List[Client]): Double`, яка обчислює загальний баланс усіх клієнтів.

Композиція функцій:

Створити функцію `processClients(clients: List[Client]): (List[Client], List[Double], Double)`, яка об'єднує вищеописані функції та повертає кортеж, що містить:

Список повнолітніх клієнтів.

Список трансформованих балансів.

Загальний баланс.

Приклад використання:

Написати код для створення списку клієнтів і виклику функції `processClients`, а також вивести результати у зрозумілому форматі.

Питання для іспиту

1. Вступ до функціонального програмування. Мови функціонального програмування.
2. Вступ до мови Scala. Змінні, типи даних, основні конструкції (розгалуження, цикли, функції). Бібліотеки Math та Random.
3. Колекції: масиви, списки, тапли, множини, словники.
4. Класи та об'єкти у мові Scala. Приклади.
5. Функціональне програмування. Вкладення функції у функцію. Аргументи по замовчуванню та іменовані аргументи. Приклади.
6. Функціональне програмування. Змінна кількість аргументів. Процедури. Приклади.
7. Анонімні функції. Частковий виклик функцій. Замикання. Приклади.
8. Хвостова рекурсія. Функції вищих порядків. Каррінг. Приклади.
9. Співставлення із зразком (pattern matching). Приклади.
10. Case-класи та case-об'єкти. Приклади.
11. Лінійні обчислення у функціональній мові. Моноїди.
12. Аналіз великих даних у Scala.
13. Мова Scala та паралельні обчислення.
14. Монади. Функтори.
15. Типізація та Каррі. Типізація за Чьорчем.
16. Основні можливості та особливості мови F#.
17. Вступ до логічного програмування. Мови логічного програмування.
18. Синтаксичні особливості мови Prolog. Вбудовані предикати. Структура Prolog-програми та приклади.
19. Мова Prolog: рекурсивне представлення даних і програм. Списки.
20. Мова Prolog: метод відсікання та метод повернення в логічному програмуванні.
21. Методи організації повторень та рекурсії мовою Prolog.

22. Побудова прикладних програм мовою логіки: створення бази даних.
23. Побудова прикладних програм мовою логіки: побудова експертних систем.
24. Мова програмування DataLog. Особливості та сфери застосування.
25. Сфера застосування видів програмування: функціональне, логічне, подійно-орієнтоване, реактивне, узагальнене програмування.
26. Вступ до Kotlin і Android. Огляд платформи Android. Встановлення середовища розробки (Android Studio). Створення першого проєкту Android на Kotlin.
27. Основи синтаксису Kotlin. Типи даних та змінні. Оператори, умовні вирази (if, when).
28. Цикли (for, while). Функції та основи роботи з ними.
29. Об'єктно-орієнтоване програмування в Kotlin. Класи та об'єкти. Конструктори, властивості та методи.
30. Наслідування, абстрактні класи та інтерфейси. Уявлення про sealed та data клас.
31. Колекції та робота з ними. Списки, множини та мапи. Операції над колекціями (filter, map, reduce). Іммутовані та змінні колекції.
32. Обробка винятків і Null безпечність. Модель обробки винятків в Kotlin.
33. Nullable типи та оператори (safe call, elvis). Оператор try-catch та використання finally.
34. Функціональне програмування в Kotlin. Лямбда-функції та анонімні функції. Вищі функції.
35. Inline-функції та кросінлайнові функції. Корутини в Kotlin: основи асинхронного програмування.
36. Основи розробки інтерфейсу в Android. Основи XML та створення UI. Робота з Activity та Fragment.
37. Взаємодія з елементами інтерфейсу (кнопки, текстові поля). Управління ресурсами та стилями.
38. Архітектурні компоненти Android. Життєвий цикл Activity та Fragment.
39. ViewModel та LiveData. DataBinding та робота з RecyclerView.
40. Робота з мережами та базами даних. Використання бібліотеки Retrofit для API-запитів.
41. Основи роботи з базою даних (Room). Кешування даних та робота з ресурсам.

Приклад завдання для іспиту:

1. Засобами мови Scala реалізувати анонімні функції з коротким записом. Використати методи map, foreach, filter. А також, використати метод reduceLeft/reduceRight. Отже, дано певний список елементів. Відфільтрувати (filter + анонімна функція) від'ємні елементи з того списку. Змінити всі отримані елементи помноживши кожен на певне число (map + анонімна функція). Знайти суму отриманих елементів (foreach + анонімна функція) та середнє арифметичне. Всі отримані значення вивести.
2. Засобами мови Prolog реалізувати програму для: дано список виду [a, 1, b, 2, c, 3, d, 4, ...]. Написати предикат, який цифри збирає у першій половині списку, а літери у другий.
3. Розробити Android-додаток, що реалізує простий калькулятор для виконання базових арифметичних операцій (додавання, віднімання, множення, ділення).

VI. Шкала оцінювання

Шкала оцінювання знань здобувачів освіти з освітніх компонентів, де формою контролю є іспит

Оцінка в балах	Лінгвістична оцінка	Оцінка за шкалою ECTS	
		оцінка	пояснення
90–100	Відмінно	A	відмінне виконання
82–89	Дуже добре	B	вище середнього рівня
75–81	Добре	C	загалом хороша робота
67–74	Задовільно	D	непогано
60–66	Достатньо	E	виконання відповідає мінімальним критеріям
1–59	Незадовільно	Fx	необхідне перескладання

VI. Рекомендована література та інтернет-ресурси

1. Глинчук Л.Я. Програмування 3 [Електронний ресурс]: електронний курс навчальної дисципліни, URL: <https://moodle.vnu.edu.ua/course/view.php?id=1383>
2. Глинчук Л.Я., Дослідження використання циклу, звичайної рекурсії та хвостової у мові програмування SCALA. *Стан, досягнення та перспективи інформаційних систем і технологій*: матеріали XXIV Всеукраїнської наук.-техн. конференції молодих вчених, аспірантів та студентів. (Одеса, 18-19 квітня 2024 р.). Одеса, Видавництво ОНТУ, 2024 р. С. 221-223.
3. Глинчук Л.Я. Аналіз онлайн-середовищ розробки для вивчення логічної мови програмування Prolog. *Математика. Інформаційні технології. Освіта*: матеріали XIII міжн. наук.-практ. конф. (Луцьк-Світязь, 31 травня - 2 червня 2024 р.). Луцьк: Волинський національний університет імені Лесі Українки, Авторська редакція, 2024. С.98-100.
URL: <https://drive.google.com/file/d/16GSq2WkNNsnEs6KRBkfCGdcYCKwi1bwO/view>
4. Глинчук Л.Я. Побудова прикладних програм мовою логіки: приклад створення бази даних та робота з нею. *Математика. Інформаційні технології. Освіта*: збірник статей. 2024, № 11, С.30-39.
URL: https://drive.google.com/file/d/1szWQhkQHZW3YVhcZbf_en0s3MyYyKqk9/view
5. Старченко В. В. Логічне програмування (Prolog) : метод. вказівки / В. В. Старченко. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2023. 48 с. (Методична серія ; вип. 414)
6. Шевченко І.В. Функціональне та логічне програмування (Частина 1. Функціональне програмування): Навч. посібник з виконання лабораторних робіт. Харків: Нац. аерокосм. ун-т «Харк.авіац. ін-т », 2020. 92с.

7. Заяць В. М. Логічне і функціональне програмування. Системний підхід : підручник / В. М. Заяць, М. М. Заяць. Рівне : НУВГП, 2018. 422 с.
8. Методичні рекомендації до виконання лабораторних робіт з курсу «Основи логічного та функціонального програмування» / укл.: С.В.Шаров. Мелітополь: РВЦ МДПУ, 2018. 52 с.
9. Документація Scala. URL: <https://docs.scala-lang.org/uk/>
10. Функціональне програмування – це... | FoxmindED. *FoxmindEd*. URL: <https://foxminded.ua/funktsionalne-prohramuvannia/>
11. Програмування на Scala, 3-тє видання. Martin Odersky, Lex Spoon, Bill Venners version 3.0, 2016-12 URL: <https://ac2epsilon.github.io/TRANS/SCALA/pis3-adoc/>
12. Get a Random Number in Scala | Baeldung on Scala. Baeldung on Scala. URL: <https://www.baeldung.com/scala/random-number>
13. Scala/Coursera/Структури даних – Вікіпідручник. *Вікіпідручник*. URL: https://uk.wikibooks.org/wiki/Scala/Coursera/Структури_даних#Stream
14. Природня розробка на платформі Andtoid [Електронний ресурс]: <https://developer.android.com/ndk/guides>
15. Ukrainian W. Уроки від W3Schools українською онлайн. Kotlin підручник. *W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів*. URL: <https://w3schoolsua.github.io/kotlin/index.html#gsc.tab=0>.