

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., доцент.

Погоджено

Гарант освітньо-професійної програми:



Гришанович Т. О.

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

протокол № 2 від 17.09.2025 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 здобуття освіти	12 Інформаційні технології 122 Комп'ютерні науки Комп'ютерні науки та інформаційні технології бакалавр	Нормативна
Кількість годин/кредитів 120/4		Рік навчання 2
		Семестр 4-ий
		Лекції 30 год.
ІНДЗ: немає		Лабораторні 48 год.
		Самостійна робота 32 год.
		Консультації 10 год.
	Форма контролю: екзамен	
Мова навчання: українська		

II. Інформація про викладача ПП

Глинчук Людмила Ярославівна

Науковий ступінь кандидат фізико-математичних наук

Вчене звання доцент

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація ном. тел.: 0958904246, email: hlynchuk.ludmila@vnu.edu.ua

Дні занять <https://ps.vnu.edu.ua/cgi-bin/timetable.cgi?n=700>

III. Опис освітнього компонента

1. Анотація ОК.

Освітній компонент «Спеціалізовані мови програмування» відноситься до переліку дисциплін циклу професійної підготовки. Навчальна дисципліна присвячена сучасним технологіям програмування: логічного та функціонального. Дисципліна покликана формувати у здобувача освіти знання та вміння щодо мов програмування Scala та Prolog, і виробити техніку програмування для подальшого застосування при розробці програм.

2. Мета і завдання освітнього компонента:

метою ОК «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

ОК «Спеціалізовані мови програмування» є формування у здобувачів знань про етапи розробки програм; про методи логічного та функціонального програмування; про вбудовані типи даних у мовах програмування Scala, Prolog. 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. Компетентності. Програмні результати навчання. Soft skills.

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.

Soft skills.

Time management – уміння планувати час, розставляти пріоритети та виконувати завдання у встановлений термін.

Soft skills при вивченні мови програмування Scala.

Scala поєднує **об’єктно-орієнтоване** та **функціональне програмування**, тому:

Аналітичне мислення – необхідність мислити в категоріях функцій, колекцій, незмінності даних.

Структуроване вирішення задач – уміння розділяти складні задачі на модулі, застосовувати патерни проєктування.

Гнучкість мислення – здатність працювати одночасно з ООП і FP підходами.

Уважність до деталей – синтаксис Scala більш суворий і компактний, ніж Java чи Python, що вчить акуратності.

Креативність – створення виразного, елегантного коду завдяки лямбда-виразам та DSL.

Soft skills при вивченні Prolog

Prolog – це логічне програмування, тому розвиває інший набір навичок:

Логічне мислення – формулювання задач у вигляді фактів і правил, пошук рішень через дедукцію.

Критичне мислення – необхідність ретельно перевіряти логіку, уникати суперечностей.

Системне бачення – побудова моделей знань, пошук взаємозв’язків між об’єктами.

Робота з абстракціями – опис предметної області у вигляді бази знань.

Наполегливість та терпіння – дебаг у Prolog іноді складний, потрібно крок за кроком аналізувати правила.

Комунікація – пояснення своїх рішень команді, яка може не мати досвіду з логічним програмуванням.

4. Структура освітнього компонента.

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
Змістовий модуль 1. Спеціалізовані бібліотеки мови програмування Python						
Тема 1. Спеціалізовані бібліотеки мови Python для роботи з мережею та поштою, для візуалізації та побудови графіків. Асинхронність Python. Штучний інтелект і Python.	12	2	6	2	2	Звіт по лаб. роботі/9

Командний проєкт з використанням спеціалізованих бібліотек Python. Виконання та презентація.	10		4	4	2	РМГ/4
Разом за модулем 1	22	2	10	6	4	13
Змістовий модуль 2. 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	1	Звіт по лаб. роботі/7
Тема 3. Класи та об'єкти у мові Scala.	8	2	4	2		Звіт по лаб. роботі/7
Тема 4. Функціональне програмування. Вкладення функції у функцію. Аргументи по замовчуванню та іменовані аргументи. Змінна кількість аргументів. Процедури.	8	2	4	2	1	Звіт по лаб. роботі/6
Тема 5. Анонімні функції. Частковий виклик функцій. Замикання. Хвостова рекурсія. Функції вищих порядків. Каррінг.	8	2	2	2		Звіт по лаб. роботі/3
Тема 6. Співставлення із зразком (pattern matching). Case-класи та case-об'єкти.	6	2	2	2	1	Звіт по лаб. роботі/4
Тема 7. Лінійні обчислення у функціональній мові. Моноїди.	8	2	4	2	1	Звіт по лаб. роботі/3
Разом за модулем 2	50	14	20	12	4	30
Змістовий модуль 3. Основні методи та підходи у логічному програмуванні						
Тема 1. Вступ до логічного програмування. Мови логічного програмування.	6	2	2	2		Звіт по лаб. роботі/3
Тема 2. Синтаксичні особливості мови Prolog. Вбудовані предикати. Структура Prolog-програми та приклади	6	2	2	2		Звіт по лаб. роботі/5
Тема 3. Рекурсивне представлення даних і програм. Списки.	6	2	2	2		Звіт по лаб. роботі/3

Тема 4. Метод відсікання та метод повернення в логічному програмуванні.	6	2	2	2		Звіт по лаб. роботі/3
Тема 5. Методи організації повторень та рекурсії мовою Prolog.	8	2	4	2		Звіт по лаб. роботі/6
Тема 6. Побудова прикладних програм мовою логіки: створення бази даних.	7	2	2	2	1	Звіт по лаб. роботі/4
Тема 7. Побудова прикладних програм мовою логіки: побудова експертних систем.	9	2	4	2	1	Звіт по лаб. роботі/3
Разом за модулем 3	48	14	18	14	2	27
Види підсумкових робіт						Бал
Тестування						10
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
Всього годин/Балів	120	30	48	32	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. 8 год
Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль.
- Підготовка до лабораторних занять, виконання домашніх завдань. 12 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. 6 год
Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

№ з/п	Тема	Кількість годин
1	Аналіз великих даних у Scala. Паралельні обчислення. Монади. Функтори.	2

2	Основні можливості та особливості мови F#.	1
3	Мова програмування DataLog. Особливості та сфери застосування.	1
4	Сфера застосування видів програмування: функціональне, логічне, подійно-орієнтоване, реактивне, узагальнене програмування.	2
	Всього	6

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, 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 дистанційного курсу «Спеціалізовані мови програмування», розміщені на платформі дистанційного навчання Moodle, виконують всі домашні завдання. Прозвітуватися про виконання завдань можна, використовуючи вищезазначені дистанційний курс, або під час консультацій, одночасно при цьому з'ясувати незрозумілі моменти, задати запитання

викладачу або вживу, або через корпоративну пошту. Існує можливість використання форуму дистанційного курсу. Перескладання контрольних робіт та тестувань заборонено.

Можливість визнання результатів навчання, отриманих у формальній, неформальній та інформальній освіті

Під час вивчення освітнього компонента можливе визнання інших результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті. Порядок визнання результатів навчання для здобувачів вищої освіти, набутих у: формальній освіті (академічна мобільність студентів на території України чи поза її межами, для студентів, які переводяться, поновлюються з інших ЗВО (вітчизняних чи іноземних); неформальній та/або інформальній освіті здійснюється згідно «ПОЛОЖЕННЯ про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки».

Можливість отримати додаткові (бонусні) бали

Відповідно до пункту 4.5 Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Волинського національного університету імені Лесі Українки здобувачам освіти, які брали участь у роботі конференцій, підготовці наукових публікацій, в олімпіадах, конкурсах студентських наукових робіт, спортивних змаганнях, мистецьких конкурсах тощо й досягли значних результатів, може бути присуджено додаткові (бонусні) бали, які зараховуються як результати поточного контролю з відповідного ОК. Систему бонусних балів погоджує науково-методична комісія факультету (інституту). При цьому загальна кількість балів за поточну роботу, у випадку ОК, де передбачено екзамен, не може перевищувати 70 балів.

І так, здобувачі освіти мають можливість отримати додаткові бали за вказаний вид робіт з ОК «Спеціалізовані мови програмування» відповідно до таблиці витягу з протоколу № 1 засідання НМТ ФІТІМ ВНУ ім. Лесі Українки від 3.09.2025 р.

Системи бонусних балів для здобувачів освіти

Вид діяльності	Рівень / результат	Кількість бонусних балів
Студентські олімпіади	I місце	7
	II місце	5
	III місце	3
	Участь в олімпіаді	2
Конкурси студентських наукових робіт	Диплом I ступеня	7
	Диплом II ступеня	5
	Диплом III ступеня	3
Підготовка наукових публікацій	Публікація в WoS / Scopus	10
	Фахова стаття	7
	Нефахова стаття	5
	Публікація тез	2
Участь у конференціях	Виступ на конференції	2
Першість України з командного програмування	I місце	10
	II місце	8
	III місце	6
	Участь	4

V. Підсумковий контроль

Підсумковою формою контролю освітнього компонента «Спеціалізовані мови програмування» є іспит.

Оцінювання навчальних досягнень здійснюється за 100 бальною шкалою. Оцінка включає в себе поточний контроль та підсумковий модульний контроль. Максимальна кількість балів за поточний контроль з ОК, де форма контролю екзамен, становить 70 балів. Підсумковий модульний контроль здійснюється після завершення вивчення тем змістового модуля у формі виконання здобувачем освіти модульного контрольного завдання (контрольної роботи, тесту, колоквіуму тощо) та проводиться або під час навчального заняття (його частини), або поза розкладом згідно окремого графіка. Максимальний бал, отриманий за модульні контрольні роботи, становить, як правило, не більше як 30 балів.

З ОК, де формою контролю є екзамен, протягом поточної роботи здобувач освіти має набрати не менше 35 балів (як допуск до складання іспиту). Для отримання допуску здобувач освіти має відпрацювати практичні або лабораторні роботи до дати іспиту під час основної сесії. Оцінки за відпрацьовані роботи викладач виставляє в електронний журнал успішності поруч або замість «н». У випадку, якщо здобувач освіти отримав менше, ніж 35 балів, він не може бути допущеним до екзамену і повинен бути відрахований за академічну неуспішність. Під час складання екзамену здобувач може отримати від 0 до 30 балів.

Якщо з ОК, де передбачено іспит, підсумкові бали поточного контролю та підсумкового модульного контролю становлять не менше як 75 балів, то, за згодою здобувача освіти, набрані бали можуть бути зараховані як підсумкова оцінка з ОК.

Іспит складається у випадку якщо здобувач освіти бажає підвищити набрану кількість балів із цього ОК. У цьому випадку бали, набрані здобувачем за результатами модульних контрольних робіт (якщо вони передбачені силабусом ОК) анулюються.

Іспит передбачає виконання тестових завдань та розв'язування задач (розробка програми). 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.

Питання для іспиту

1. Спеціалізовані бібліотеки мови Python для роботи з мережею та поштою, для візуалізації та побудови графіків. Асинхронність Python. Штучний інтелект і Python.
2. Вступ до функціонального програмування. Мови функціонального програмування.
3. Вступ до мови Scala. Змінні, типи даних, основні конструкції (розгалуження, цикли, функції). Бібліотеки Math та Random.
4. Колекції: масиви, списки, тапли, множини, словники.
5. Класи та об'єкти у мові Scala. Приклади.
6. Функціональне програмування. Вкладення функції у функцію. Аргументи по замовчуванню та іменовані аргументи. Приклади.
7. Функціональне програмування. Змінна кількість аргументів. Процедури. Приклади.
8. Анонімні функції. Частковий виклик функцій. Замикання. Приклади.
9. Хвостова рекурсія. Функції вищих порядків. Каррінг. Приклади.
10. Співставлення із зразком (pattern matching). Приклади.
11. Case-класи та case-об'єкти. Приклади.
12. Лінійні обчислення у функціональній мові. Моноїди.

13. Аналіз великих даних у Scala. Мова Scala та паралельні обчислення. Монади. Функтори.
14. Основні можливості та особливості мови F#.
15. Вступ до логічного програмування. Мови логічного програмування.
16. Синтаксичні особливості мови Prolog. Вбудовані предикати. Структура Prolog-програми та приклади.
17. Мова Prolog: рекурсивне представлення даних і програм. Списки.
18. Мова Prolog: метод відсікання та метод повернення в логічному програмуванні.
19. Методи організації повторень та рекурсії мовою Prolog.
20. Побудова прикладних програм мовою логіки: створення бази даних.
21. Побудова прикладних програм мовою логіки: побудова експертних систем.
22. Мова програмування DataLog. Особливості та сфери застосування.
23. Сфера застосування видів програмування: функціональне, логічне, подійно-орієнтоване, реактивне, узагальнене програмування.

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.

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. Спеціалізовані мов програмування/Програмування 4 [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 Venner 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