



ІМПЕРАТИВНІ МОВИ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Третій (доктор філософії)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>КОМП'ЮТЕРНІ НАУКИ</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весінній семестр</i>
Обсяг дисципліни	<i>2 кредити (60 годин)</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Перший та другий тиждень: Лекція/лабораторна робота, четвер, 10-25, on-line</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i> Практичні : <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i>
Розміщення курсу	

Програма навчальної дисципліни

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

Дисципліна «Імперативні мови програмування» (ІВ2-2) є нормативною дисципліною навчального плану підготовки докторів філософії з спеціальності «Комп'ютерні науки» і грає важливу роль у підготовці фахівців.

Метою навчальної дисципліни є вивчення та застосування імперативної парадигми програмування, зокрема за допомогою мови програмування Java, під час програмної реалізації прикладних завдань в області інформаційних систем і технологій та у майбутній професійній діяльності. Основними завданнями дисципліни є набуття теоретичних знань з основ програмування мовою Java, реалізації основних принципів об'єктно-орієнтованого програмування мовою Java; здатностей застосовувати мову програмування Java та її особливостей для реалізації алгоритмів розв'язування різнотипних завдань; проектувати та розробляти програмне забезпечення мовою Java із застосуванням структурного та об'єктно-орієнтованого, з відповідними моделями, методами та алгоритмами обчислень, структурами даних (колекціями) і механізмами управління.

Метою кредитного модуля є формування у аспірантів загальних та спеціальних (фахових, предметних) компетентностей:

- ЗК01 – здатність до абстрактного мислення, аналізу та синтезу;

- ЗК03 — здатність працювати в міжнародному контексті.
- ЗК04 – здатність розробляти проекти та управляти ними;
- СК02. Здатність усно і письмово презентувати та обговорювати результати наукових досліджень з комп’ютерних наук та/або інноваційних розробок державною та іноземною (англійською або іншими) мовами, глибоке розуміння іншомовних;
- СК03 – здатність створювати і застосовувати сучасні інформаційні технології, архітектури і спеціалізоване програмне забезпечення у науковій та освітній діяльності, керувати інформаційними ресурсами, інформаційними системами та цифровими сервісами;
- СК05 – здатність розвивати теоретичні засади, створювати моделі інформаційних технологій, проектувати та створювати інформаційні системи і цифрові сервіси та їх прототипи;
- СК06 – здатність застосовувати сучасні методи дослідження, синтезу, проектування інформаційних систем і технологій у науковій та науково-педагогічній діяльності;

Програмні результати навчання:

Згідно з вимогами програми навчальної дисципліни аспіранти після засвоєння кредитного модуля мають продемонструвати такі результати навчання:

- РН01. Мати передові концептуальні та методологічні знання з комп’ютерних наук і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.
- РН04. Розробляти та досліджувати концептуальні, математичні і комп’ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у комп’ютерних науках та дотичних міждисциплінарних напрямках.
- РН06. Застосовувати сучасні інструменти і технології пошуку, оброблення та аналізу інформації, зокрема, статистичні методи аналізу даних великого обсягу та/або складної структури, спеціалізовані бази даних та інформаційні системи.
- РН07. Розробляти та реалізовувати наукові та/або інноваційні інженерні проекти, які дають можливість переосмислити наявне та створити нове цілісне знання та/або професійну практику і розв’язувати значущі наукові та технологічні проблеми комп’ютерної науки з дотриманням норм академічної етики і врахуванням соціальних, економічних, екологічних та правових аспектів.
- РН09. Вивчати, узагальнювати та впроваджувати в навчальний процес інновації комп’ютерних наук.
- РН10. Відшукувати, оцінювати та критично аналізувати інформацію щодо поточного стану та трендів розвитку, інструментів та методів досліджень, наукових та інноваційних проектів з комп’ютерних наук.

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

Пререквізити: здатність застосовувати знання у практичних ситуаціях, навички використання інформаційних і комунікаційних технологій. Цей курс базується на таких забезпечуючих дисциплінах: тенденції розвитку інформаційних систем та технологій, сучасні концепції створення інтелектуальних систем, проектування інформаційних систем.

Постреквізити: Перелік напрямків діяльності, що забезпечуються: педагогічна практика, захист дисертаційної роботи.

3. Зміст навчальної дисципліни

Розділ 1. Вступ до дисципліни.

Тема 1.1. Огляд основних парадигм програмування.

Тема 1.2. Використання імперативного підходу програмування.

Тема 1.3. Особливості мови програмування Java.

Розділ 2. Введення в програмування на мові Java.

Тема 2.1. Історія розвитку та застосування.

Тема 2.2. Структура, функції та робота JVM.

Розділ 3. Лексична структура мови програмування Java.

Тема 3.1. Основні оператори.

Тема 3.2. Типи.

Тема 3.3. Масиви.

Тема 3.4. Робота з рядками.

Розділ 4. ООП в Java.

Тема 4.1. Принципи ООП.

Тема 4.2. Абстрактні класи.

Тема 4.3. Інтерфейси.

Тема 4.4. Модифікатори доступу.

Тема 4.5. Створення і використання пакетів.

Розділ 5. Виключні ситуації та узагальнене програмування.

Тема 5.1. Виключні ситуації.

Тема 5.2. Узагальнене програмування на мові Java.

Розділ 6. Колекції.

Тема 6.1. Використання колекцій.

Тема 6.2. Ієрархія колекцій.

Тема 6.3. Реалізації колекцій.

Розділ 7. Потоки вводу/виводу.

Тема 7.1. Організація потоків.

Тема 7.2. Серіалізація.

Тема 7.3. Робота з файлами.

Розділ 8. Потоки виконання.

Тема 8.1. Потоки та JVM.

Тема 8.2. Багатопоточність.

Тема 8.3. Взаємодія потоків.

Тема 8.4. Синхронізація.

Тема 8.5. Організація конкурентного доступу.

4. Навчальні матеріали та ресурси

Базова

1. Schildt Н. Java: The Complete Reference. Twelfth Edition - NY: McGraw Hill Professional, 2021. - 1573 р.
2. Роберт Мартін. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І.Бондар-Терещенко. Харків : Видво «Ранок» Ж Фабула. 2021. 448 с
3. Cheng F. Exploring Java 9. Build Modularized Applications in Java. - Berkeley: Apress, 2018. - 174 р.
4. Галкін О.В., Катеринич Л.О., Шкільняк О.С. Програмування на Java 8: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. – Київ: ЛОГОС, 2017. – 186 с.
5. Кадомський К.К., Ніколюк П.К. Java. Теорія і практика : Навч. посіб. - Вінниця: Донну, 2019. 197 с. 5. Васильєв О.М. Програмування мовою Java. — Тернопіль: Богдан, 2019. — 696 с.

Додаткова

6. Josh Juneau, Luciano Manelli. Java 17 Recipes: A Problem-Solution Approach. 2022 589 p. DOI: <https://doi.org/10.1007/978-1-4842-7963-2>
7. Mastering Java. A Beginner's Guide. Edited by Sufyan bin Uzayr. Taylor & Francis Group, LLC. 491 p. DOI: 10.1201/9781003229063

Навчальний контент

5. Методика опанування навчальної дисципліни(освітнього компонента)

№ з/п	Назва теми лекції та перелік основних питань (дидактичні матеріали: презентація PowerPoint, відеолекції.)
1	2
1	<p style="text-align: center;">Розділ 1. Вступ до дисципліни.</p> <p>Лекція 1. Імперативна парадигма програмування на прикладі мови Java</p> <p>Основні концепції та терміни. Перелік сучасних парадигм програмування. Переваги та недоліки різних підходів. Концепцій імперативної парадигми: послідовність команд, змінні, стан програми. Основні характеристики мови Java. Огляд найпоширеніших IDE для написання програм мовою Java та їх встановлення. Література: 1-3. Самостійна робота: Ознайомлення з мовою програмування Java для розробки прикладного програмного забезпечення. Підготовка до лабораторної роботи.</p>

	Розділ 2. Введення в програмування на мові Java.
2	<p>Лекція 2. Функції та робота JVM.</p> <p>Поняття віртуальної машини Java. Ролі та принципи роботи Java Virtual Machine. Особливості запуску та виконання Java bytecode. Поняття JVM, JDK та JRE. Runtime Data Areas та Execution Engine, їх роль у виконанні Java-програм. Процесу виконання Java-програм на JVM. Управління пам'яттю. Just-In-Time компіляція. Порівняння JVM з іншими віртуальними машинами. Література: 1-3. Самостійна робота: Встановлення jdk. Запуск java застосунку. Підготовка до лабораторної роботи.</p>
	Розділ 3. Лексична структура мови програмування Java.
3	<p>Лекція 3. Синтаксис та вбудовані типи даних</p> <p>Синтаксис java мови. Вбудовані типи даних і літерали. Змінні та оператори. Вирази. Поняття масиву. Оголошення та ініціалізація. Особливості роботи з елементами масиву. Оператор циклу foreach. Клас Arrays. Робота з рядками та її особливості. Лінійній програми. Консольне введення та виведення даних. Оператор умовного переходу if-else. Оператор множинного вибору switch. Enum. Цикли. Оператор циклу з передумовою (while) та післяумовою (do-while). Оператор циклу з параметром for. Функції в Java. Параметри функцій. Параметри змінної довжини. Оператор Return. Результат функції. Рекурсивні функції. Література: 1-5. Самостійна робота: створення проекту та запуск власного java застосунку. Підготовка до лабораторної роботи.</p>
	Розділ 4. ООП в Java.
4	<p>Лекція 4. Об'єктно орієнтоване програмування.</p> <p>Ідеологія та основні принципи ООП. Поняття класа та об'єкта. Створення об'єкту. Абстракція, наслідування, поліморфізм та інкапсуляція. Поля та методи класу. Конструктори. Модифікатори доступу. Робота з об'єктами. Методи get та set. Об'єкти, як параметри методів. Метод toString(). Використання абстрактних класів в Java. Поняття</p>

	інтерфейсу та ключові відмінності від абстрактних класів. Класи-обгортки примітивних типів. Композиція, агрегація. Поняття this і super. Порівняння об'єктів instanceof. Поліморфізм. Перевизначення (override) та перевантаження (overload) Література: 1-5. Самостійна робота: використання IDE для запуску java застосунку. Створення власної ієрархії класів.
	Розділ 5. Виключні ситуації та узагальнене програмування.
5	Лекція 5. Обробка виключних ситуацій та узагальнені типи в Java. Поняття виключних ситуацій. Використання конструкцій Try-catch, Try-finally, Try-catch-finally. Клас Exception. Ієрархія класів помилок. Поняття checked та unchecked типів помилок. Обробка декількох винятків. Параметризовані класи. Узагальнення (Generics). Наслідування узагальнень. Лямбда-вирази. Лямбда як параметри і результати методів. Вбудовані функціональні інтерфейси. Література: 1-6. Самостійна робота: використання узагальнених типів даних в прикладних програмах. Підготовка до лабораторної роботи.
	Розділ 6. Колекції.
6	Лекція 6. Collections Framework в Java. Java Collections Framework. Основні типи колекцій. Основні інтерфейси Collections API. Інтерфейс Collection. Тип колекції «Список». Інтерфейс List та його реалізація: класи ArrayList, LinkedList. Ітерування елементів колекції (інтерфейси Iterator, ListIterator). Методи equals, hashCode. Інтерфейси Comparator, Comparable. Тип колекції «Множина». Інтерфейс Set та його реалізація класи HashSet, LinkedHashSet Інтерфейси SortedSet, NavigableSet. Клас TreeSet. Тип колекції «Черга». Інтерфейс Queue та його реалізації (LinkedList, ArrayDeque). Колекції типу «Асоціативний масив (ключ:значення)». Інтерфейс Map і його реалізація: класи TreeMap, LinkedHashMap та HashMap. Література: 1, 4-5. Самостійна робота: використання типів колекцій у власному застосунку. Підготовка до лабораторної роботи.
	Розділ 7. Потoki вводу/виводу.
7	Лекція 7. Робота з потоками вводу/виводу. Байтові потоки введення і виведення. FileInputStream і FileOutputStream. Закриття потоків. Класи ByteArrayInputStream і ByteArrayOutputStream. Буферизація потоків байтів: BufferedInputStream і BufferedOutputStream Форматоване виведення: PrintStream і PrintWriter Класи DataOutputStream і DataInputStream. Читання і запис текстових файлів. Буферизація символних потоків. BufferedReader і BufferedWriter. Сериалізація об'єктів. Інтерфейс Serializable. Клас File. Робота з файлами і каталогами. Архівування. Робота з JAR та ZIP-архівами Література: 1, 3-6. Самостійна робота: Створення застосунку для читання та виведення на консоль даних з файлу. Підготовка до лабораторної роботи.
	Розділ 8. Потoki виконання.
8	Лекція 8. Багатопотоковість в Java Створення потоків із використанням класу Threads. Створення потоків із використанням інтерфейсу Runnable. Синхронізація потоків. Оператор synchronized. Демони потоки. Взаємодія потоків. Об'єкт блокування. Методи wait, notify, notifyAll(). Модифікатор volatile. Java Memory Model. Література: 1, 3-7. Самостійна робота: створення багатопоточного застосунку.

Практичні заняття

№ з/п	Назва практичної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	<i>Практична робота 1. Створення проекту за допомогою IDE. Створення простого застосунку та його запуск. Демонстрація</i>	2

	можливостей зміни значень за допомогою <i>debug</i> . Література: 1, 3-5.	
2	Практична робота 2. Читання файлів і аварійна зупинка виконання програми. Обробка помилок. Література: 1-3, 7.	2

6. Самостійна робота аспіранта

№ з/п	Назви тем і питань, що виносяться на самостійне опрацювання та посилання на навчальну літературу	Кількість годин СРС
1	2	2
1	Створення проекту за допомогою Maven та публікація його в системі GitHub [1-3]	4
2	Парсинг JSON з файлу та написання тестів на це [2-4]	4
3	Парсинг даних з XML файлу [2-4]	4
4	Підключення до бази даних SQL та отримання даних із таблиці [2-5]	4
5	Огляд фреймворків для створення веб-сервісів [2-5]	4
6	Обробка даних, які отримуються з потоку kafka [2-5]	4
7	Створення веб сервісу з підтримкою Web Socket [3-6]	6
8	Написання програми рішення two-sum problem	6
9	Написання програми симуляції ходів коня на шаховій дошці [2-7]	6

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять

Відвідування лекцій переконливо рекомендується, але штрафних санкцій за пропуски лекцій не передбачено. Відвідування занять комп'ютерного практикуму необхідно в обсязі, достатньому для виконання вимог викладача щодо виконання і своєчасної здачі лабораторних робіт та індивідуального завдання.

Пропущені контрольні заходи

Лабораторні роботи можна здавати у відведений за розкладом час лабораторних занять, як до, так і після встановленого терміну здачі конкретної роботи. Додаткові години для здачі індивідуального завдання призначаються викладачем в межах часу лабораторних занять. Завідомості поважних причин пропуску (медична довідка тощо) штрафні бали не нараховуються.

Процедура оскарження результатів контрольних заходів

Здобувачі мають можливість підняти будь-яке питання, яке стосується процедур контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: виконання індивідуального завдання на проектування інформаційних систем, виконання комп'ютерних практикумів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: мінімально позитивна оцінка за індивідуальне завдання /зарахування усіх практичних робіт/семестровий рейтинг більше 30 балів.

Рейтинг аспіранта з дисципліни складається з двох складових: стартової – призначена для оцінювання заходів поточного контролю впродовж семестру та екзаменаційної – призначена для оцінювання окремих запитань (завдань) на екзамені і формується з балів, що він отримує за:

- 1) індивідуальне завдання на проектування інформаційної системи;
- 2) виконання комп'ютерних практикумів;
- 3) відповідь на екзамені.

1. Індивідуальне завдання

Ваговий бал – 25. Кожний аспірант виконує індивідуальне завдання щодо розробки інформаційної системи (програмного забезпечення) в рамках свого дисертаційного дослідження, яке передбачає використання всієї низки програмно-технічного інструментарію, що вивчається в рамках курсу.

2. Робота на комп'ютерних практичних заняттях

Ваговий бал – 5. Максимальна кількість балів на всіх практичних заняттях дорівнює $5 \cdot 9 = 45$ балів.

3. Відповідь на заліку

Кількість балів по відповіді на кожне питання визначається викладачем з врахуванням складності питання та якості відповіді. Максимальна кількість балів 30.

Штрафні та заохочувальні бали за:

- виконання завдань із удосконалення дидактичних матеріалів з дисципліни надається від 2 до 5 заохочувальних балів.

Розрахунок шкали (R) рейтингу

Сума вагових балів контрольних заходів протягом семестру складає:

$$RC = 70 + 30 = 100 \text{ балів.}$$

Для отримання аспірантом відповідних оцінок (ECTS та традиційних) його рейтингова оцінка *R* переводиться згідно з таблицею відповідності рейтингових балів оцінкам за університетською шкалою.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за тематикою дисципліни «Новітні програмно-технічні засоби проектування інформаційних систем»;
- інша інформація для аспірантів щодо особливостей опанування навчальної дисципліни: Застосовуються стратегії активного і колективного навчання, які визначаються наступними методами і технологіями:

- 1) кредитно-модульна технологія навчання;
- 2) особистісно-орієнтовані (розвиваючі) технології, засновані на активних формах і методах навчання («аналіз ситуацій» ділові, імітаційні ігри, дискусія, експрес-конференція, навчальні дебати.);
- 3) інформаційно-комунікаційні технології, що забезпечують проблемно-дослідницький характер процесу навчання та активізацію самостійної роботи аспірантів (електронні презентації для лекційних занять, використання аудіо-, відео-підтримки навчальних занять, розробка і

застосування на основі комп'ютерних і мультимедійних засобів творчих завдань, доповнення традиційних навчальних занять засобами взаємодії на основі мережевих комунікаційних можливостей).

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н. Кравцовим Григорієм Олексійовичем

Ухвалено: Вченою радою ІПМЕ ім. Г.Є. Пухова НАН України (протокол №15 від 27.11.2025 р.)