

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ



Декан факультету інформаційних технологій
Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ
2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Кросплатформне програмування

Назва дисципліни

Призначення Робочої програми

Рівень вищої освіти

Мова навчання

Обсяг дисципліни, кредитів ЄКТС

Статус дисципліни

Факультет

Кафедра

Для освітніх програм різних спеціальностей

Перший (бакалаврський)

Українська

8

Вибіркова

Інформаційних технологій

Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	96	32	32	32		144	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена  ст.викладач Дмитро ДЕНИСЮК
Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем
Назва

Протокол від 18.08.2025 № 1.

Зав. кафедри кафедри комп'ютерної інженерії та інформаційних систем  Ольга ПАВЛОВА
Назва Ім'я, ПРІЗВИЩЕ

3. Пояснювальна записка

Дисципліна «Кроссплатформне програмування» є вибірковою та спрямована на формування у здобувачів вищої освіти знань і практичних навичок створення програмних продуктів мовою Java, що можуть функціонувати на різних апаратних і програмних платформах. Вивчення курсу сприяє поглибленню фахових умінь у сфері сучасної програмної інженерії, оволодінню методами та засобами кроссплатформної розробки із застосуванням стандартних бібліотек та додаткових інструментів Java, а також набуттю навичок оптимізації, налагодження і тестування програмного забезпечення для різних середовищ. Опанування дисципліни формує здатність до подальшої професійної спеціалізації, розвиває уміння працювати з технічною документацією, використовувати сучасні середовища розробки, інтегрувати зовнішні бібліотеки й фреймворки. У процесі навчання студенти також набувають важливих соціальних навичок (soft skills): комунікації, командної роботи, ініціативності, відповідальності та прийняття інженерних рішень у процесі виконання практичних і проєктних завдань.

Мета дисципліни. Формування у здобувачів вищої освіти системних теоретичних знань і практичних навичок із методів, засобів та технологій кроссплатформної розробки програмного забезпечення мовою Java. Дисципліна має на меті забезпечити здатність студентів застосовувати засоби мови Java, стандартні та додаткові бібліотеки, сучасні фреймворки й інтегровані середовища розробки для створення, тестування та оптимізації програмних продуктів, що функціонують у різних операційних системах і апаратних середовищах відповідно до вимог реальних проєктів.

Предмет дисципліни. Засоби, методи та технології кроссплатформної розробки програмного забезпечення мовою Java із застосуванням стандартних бібліотек, додаткових фреймворків і бібліотек. Предметом дисципліни є створення програмних продуктів, здатних функціонувати у різних операційних системах (Windows, Linux, macOS), з урахуванням особливостей середовищ виконання, взаємодії з інтерфейсами користувача, інтеграції з апаратними й мережевими ресурсами, а також вимог до продуктивності, надійності та безпеки.

Завдання дисципліни. Завданням дисципліни є формування у здобувачів практичних навичок розроблення кроссплатформних програмних продуктів із використанням мови Java та супровідних бібліотек і фреймворків, засвоєння основ побудови інтерфейсів користувача, організації взаємодії з подіями та даними, ознайомлення з процесами тестування, налагодження, оптимізації та розгортання програмних рішень. Дисципліна також спрямована на розвиток умінь інтегрувати сторонні бібліотеки та сервіси у кроссплатформні застосунки, формування аналітичного мислення, здатності працювати в команді, брати на себе відповідальність, приймати інженерні рішення та ефективно застосовувати сучасні засоби розробки в умовах реальних проєктів.

Результати навчання. Після вивчення дисципліни студент повинен знати основні концепції та принципи кроссплатформної розробки мовою Java, особливості використання стандартних бібліотек та фреймворків, підходи до створення інтерфейсів користувача, організації взаємодії з даними та подіями. Він має володіти інструментами проєктування, реалізації, тестування та налагодження програмних рішень, здатністю обґрунтовувати вибір інструментів і технологій, поєднувати теоретичні знання з практикою програмування та вирішувати прикладні завдання в умовах обмежених ресурсів. Студент повинен уміти працювати з технічною документацією, проводити дослідження, оцінювати результати й аргументовано захищати технічні рішення, а також проявляти ініціативність, здатність до самостійного навчання, ефективної комунікації, роботи в команді та прийняття відповідальних інженерних рішень.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
1. Основи Java та робота середовища виконання	4	4	4	14
2. Класи, об'єкти та ООП	4	4	4	14
3. Конструктори, абстракція, інтерфейси та вкладені класи	4	4	4	14
4. Масиви та колекції	4	4	4	16
5. Потоки введення/виведення та робота з файлами	4	4	4	16
6. Графічний інтерфейс Java (Swing)	4	4	4	16
7. Робота з базами даних та мережеві застосунки	4	4	4	28
8. Потоки, багатопотоковість і додаткові бібліотеки Java	6	6	6	26
Разом за семестр:	34	34	34	144

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	Тема 1. Основи Java та робота середовища виконання	4
1	Вступ до Java. Життєвий цикл. Робота JVM. Змінні, типи даних, приведення типів. Пріоритет операцій. Методи. Літ.: [1,3,5]	2
2	Робота із класами. Поняття класу та об'єкта. Методи створення об'єктів. Поля та методи об'єкта. Літ.: [1,3,4,5]	2
	Тема 2. Об'єктно-орієнтоване програмування у Java	4
3	Наслідування, поліморфізм та інкапсуляція. Приватні поля та методи доступу. Методи користувацьких класів. Літ.: [1,3-5]	2
4	Конструктори, абстрактні класи, інтерфейси. Вкладені класи. Клас Object і його методи. Літ.: [1,3-5]	2
	Тема 3. Масиви та колекції	4
5	Масиви: одновимірні й багатовимірні. Методи перебору та методи масивів. Інтерфейс Collection. Літ.: [1,3,5]	2
6	Колекції: ArrayList, LinkedList, HashSet, Dictionary. Літ.: [1,8]	2
	Тема 4. Потоки введення/виведення та робота з файлами	4
7	Потоки: InputStream, OutputStream, форматований вивід, PrintStream та PrintWriter. Літ.: [2,5]	2
8	Робота з файлами: FileInputStream, FileOutputStream, клас File, робота з каталогами. Літ.: [2,3,5]	2
	Тема 5. Графічний інтерфейс Java (Swing)	4
9	Вступ до Swing: графічні елементи, компоновання, панелі. Літ.: [1,3,5]	2
10	Елементи управління: кнопки, мітки, CheckBox, RadioButton, текстові поля. Події елементів. Літ.: [1,5]	2
	Тема 6. Java і бази даних	4
11	Основи баз даних. Технологія JDBC. Встановлення та базові налаштування MySQL. Літ.: [2,6]	2
12	Робота з даними у MySQL. Додавання, видалення, зміна даних. Методи executeUpdate та executeQuery. Літ.: [2,6]	2
	Тема 7. Мережеве програмування	4

13	Мережеві додатки 1: сокети, мережеві класи та інтерфейси. Літ.: [2,6-8]	2
14	Мережеві додатки 2: клас URLConnection. Клієнт-серверна взаємодія. Літ.: [2,6-8]	2
Тема 8. Потоки, багатопотоковість і додаткові бібліотеки		4
15	Клас Thread. Створення та виконання потоку. Завершення та переривання. Літ.: [1,4,5,7]	2
16	Багатопотоковість: синхронізація, synchronized, семафори, обмін даними. Додаткові бібліотеки Java: Math, BigInteger, BigDecimal, робота з датами. Літ.: [1,4,5,6,7]	2
Разом за семестр:		32

5.2 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
1	Вибір та налаштування середовища розробки для Java. Створення першої програми. Літ.: [1] с. 9–15; [3] с. 6–12	4
2	Класи та об'єкти у Java. Поля і методи. Створення та використання об'єктів. Літ.: [1] с. 40–58; [3] с. 13–25	4
3	Об'єктно-орієнтоване програмування: наслідування, інкапсуляція, поліморфізм. Літ.: [1] с. 70–85; [4] с. 20–35	4
4	Масиви та колекції. Робота з ArrayList, LinkedList, HashSet. Літ.: [1] с. 100–120; [8] с. 30–45	4
5	Потоки введення/виведення. Робота з файлами та каталогами. Літ.: [2] с. 50–63; [5] с. 22–33	4
6	Java Swing: створення графічних інтерфейсів. Події елементів управління. Літ.: [1] с. 130–150; [5] с. 40–55	4
7	Робота з базами даних. Технологія JDBC. Підключення до MySQL та виконання SQL-запитів. Літ.: [2] с. 70–90; [6] с. 15–28	4
8	Багатопотокове програмування. Створення потоків, синхронізація, обмін даними. Літ.: [1] с. 160–180; [7] с. 45–60	4
Разом за семестр		32

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
1	Лабораторна робота №1. Базові операції у Java. Робота зі змінними, методами та класами. Літ.: [1] с. 9–15; [3] с. 6–12	4
2	Лабораторна робота №2. Об'єктно-орієнтоване програмування: наслідування, інкапсуляція, поліморфізм. Літ.: [1] с. 70–85; [4] с. 20–35	4
3	Лабораторна робота №3. Масиви та колекції. Використання ArrayList, LinkedList, HashSet. Літ.: [1] с. 100–120; [8] с. 30–45	4
4	Лабораторна робота №4. Потоки введення/виведення. Робота з файлами та каталогами. Літ.: [2] с. 50–63; [5] с. 22–33	4
5	Лабораторна робота №5. Розробка графічного інтерфейсу (Swing). Обробка подій елементів. Літ.: [1] с. 130–150; [5] с. 40–55	4
6	Лабораторна робота №6. Підключення до бази даних (JDBC). Виконання SQL-запитів у MySQL. Літ.: [2] с. 70–90; [6] с. 15–28	4
7	Лабораторна робота №7. Мережеві застосунки у Java. Сокети, клієнт-серверна взаємодія. Літ.: [2] с. 95–110; [7] с. 45–60	4
8	Лабораторна робота №8. Багатопотокове програмування. Створення потоків, синхронізація, обмін даними. Літ.: [1] с. 160–180; [7] с. 61–75	4
Разом:		32

5.3 Зміст самостійної (у т.ч. індивідуальної) роботи здобувача вищої освіти

Самостійна робота з дисципліни «Кроссплатформне програмування» становить 138 годин і є важливою складовою освітнього процесу. Вона передбачає систематичне опрацювання лекційного матеріалу, вивчення рекомендованої літератури та електронних ресурсів, а також виконання індивідуальних завдань. У процесі самостійної роботи студенти готуються до практичних та лабораторних занять, відпрацьовують приклади коду, аналізують алгоритми, засвоюють синтаксичні конструкції Java та формують навички роботи з технічною документацією. Значна увага приділяється виконанню й захисту лабораторних робіт, розробці невеликих кроссплатформних програмних проєктів, тестуванню та налагодженню власних програмних рішень. Крім того, самостійна робота включає підготовку до поточного і підсумкового контролю знань. Для цього використовуються завдання у формі тестів, контрольних робіт, індивідуальних проєктів та дослідницьких завдань.

Номер тижня	Вид самостійної роботи	Кількість годин
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1 та ПЗ1. Підготовка до захисту ЛР1.	18
3-4	Опрацювання лекційного матеріалу. Підготовка до ЛР2 та ПЗ2. Підготовка до захисту ЛР2.	18
5-6	Опрацювання лекційного матеріалу. Підготовка до ЛР3 та ПЗ3. Підготовка до захисту ЛР3.	18
7-8	Опрацювання лекційного матеріалу. Підготовка до ЛР4 та ПЗ4. Підготовка до захисту ЛР4.	18
9-10	Опрацювання лекційного матеріалу. Підготовка до ЛР5 та ПЗ5. Підготовка до захисту ЛР5.	18
11-12	Опрацювання лекційного матеріалу. Підготовка до ЛР6 та ПЗ6. Підготовка до захисту ЛР6.	18
13-14	Опрацювання лекційного матеріалу. Підготовка до ЛР7 та ПЗ7. Підготовка до захисту ЛР7.	18
15-16	Опрацювання лекційного матеріалу. Підготовка до ЛР8 та ПЗ8. Підготовка до захисту ЛР8.	18
	Разом 1-й семестр:	144

Примітки: ЛР – лабораторна робота, ПЗ– практичне заняття.

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до практичних та лабораторних занять індивідуальні домашні завдання з відповідних тем курсу. Виконання таких завдань сприяє закріпленню знань з лекційного матеріалу, формуванню практичних навичок програмування та розвитку здатності застосовувати набуті знання у розробці програмних продуктів.

Керівництво самостійною роботою та контроль за виконанням індивідуальних завдань здійснюється викладачем згідно з розкладом консультацій у позаурочний час. Вимоги до виконання індивідуальних домашніх завдань та форми їх представлення викладені у Модульному середовищі для навчання на сторінці дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни «Кроссплатформне програмування» ґрунтується на поєднанні традиційних та сучасних методів навчання, що забезпечують ефективне засвоєння теоретичного матеріалу та формування практичних навичок.

Лекції проводяться з використанням мультимедійних презентацій, методів візуалізації, пояснення, проблемного та інтерактивного навчання. Викладач застосовує методи стимулювання та мотивації, інформаційно-комунікаційні технології, прийоми індивідуалізації та інтенсифікації навчання, що сприяє кращому розумінню матеріалу та залученню студентів до активної роботи на заняттях.

Практичні заняття організовуються у формі бесіди, інструктування, демонстрування та спостереження. Значна увага приділяється аналізу кейсів, розв'язанню ситуаційних завдань, презентаціям, дискусіям, діловим іграм, тренінговим вправам та методу «мозкового штурму». Це сприяє розвитку аналітичного мислення, комунікативних умінь та навичок командної роботи.

Лабораторні заняття спрямовані на набуття практичних навичок роботи з мовою програмування Java та середовищами розробки. Для цього застосовуються методи комп'ютерного моделювання, методи проєктної діяльності, тренінгові вправи, аналіз проблемних ситуацій, елементи дискусії та пояснення викладача. Виконання лабораторних робіт сприяє формуванню готовності студентів до вирішення прикладних завдань у реальних умовах.

Самостійна робота включає засвоєння теоретичного матеріалу, опрацювання рекомендованих джерел, підготовку до практичних та лабораторних занять, виконання індивідуальних та домашніх завдань, а також підготовку до поточного та підсумкового контролю знань. Для її організації використовуються інформаційно-комп'ютерні технології та технології дистанційного навчання, зокрема ресурси модульного середовища кафедри, де розміщені навчально-методичні матеріали та завдання для самоконтролю.

7. Методи контролю

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, з використанням у тому числі Модульного середовища для навчання.

До методів поточного контролю належать:

- усне опитування перед допуском до лабораторного заняття;
- тестовий контроль засвоєння теоретичного та практичного матеріалу з теми або розділу;
- оцінювання результатів роботи на практичних заняттях (опитування, розв'язування задач, участь у дискусіях та обговореннях);
- оцінювання результатів виконання та захисту лабораторних робіт;
- оцінювання результатів виконання індивідуальних завдань;
-

- оцінювання портфоліо здобувача (звіти, проєкти, результати програмування та їх захист).

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, вважається таким, який *має* академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію та навчально-методичне забезпечення освітнього процесу. Обов'язковою умовою є проходження інструктажу з техніки безпеки. Відвідування лекційних, практичних і лабораторних занять є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання може відбуватись в онлайн-режимі за погодженням із викладачем.

Успішне опанування дисципліни та формування фахових компетентностей передбачає обов'язкову підготовку до практичних і лабораторних занять: вивчення теоретичного матеріалу з теми, підготовку протоколів лабораторних робіт, а також готовність до усного опитування для допуску до заняття. Активна участь у практичних обговореннях і дискусіях, розв'язування задач і виконання індивідуальних завдань є невід'ємною частиною навчального процесу.

Оцінювання результатів навчання здійснюється за трьома основними складовими:

- виконання та захист лабораторних робіт;
- виконання практичних робіт (розв'язування задач, аналіз кейсів, робота в командах);
- два тестування, які перевіряють рівень засвоєння теоретичного матеріалу та практичних умінь з основних розділів дисципліни.

Здобувачі вищої освіти зобов'язані дотримуватись установлених термінів виконання всіх видів навчальної роботи. Захист лабораторної роботи вважається своєчасним, якщо студент захистив її на найближчому після виконання занятті. Пропущене практичне або лабораторне заняття студент повинен відпрацювати у термін, визначений викладачем, але не пізніше ніж за два тижні до завершення теоретичних занять у семестрі.

Засвоєння теоретичного матеріалу з дисципліни перевіряється під час практичних занять у формі усного опитування та у процесі тестування. Таким чином здійснюється як поточний, так і підсумковий контроль рівня підготовки.

Здобувач вищої освіти, виконуючи самостійну або індивідуальну роботу, має дотримуватись політики академічної доброчесності: забороняється списування, плагіат (у тому числі з використанням мобільних пристроїв). У разі виявлення порушення політики академічної доброчесності студент отримує незадовільну оцінку і зобов'язаний повторно виконати завдання з відповідної теми.

У межах вивчення дисципліни можливе визнання і зарахування результатів навчання, набутих у процесі неформальної освіти на відкритих освітніх платформах (наприклад, Coursera, Udemy, Prometheus), за умови відповідності їх змісту програмним результатам навчання. Такі результати враховуються згідно з «Положенням про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ».

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. При цьому кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>помилки</i> .

Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив 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 «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів *денної* форми здобуття освіти у семестрі

Аудиторна робота												Контрольні заходи	Семестровий контроль
Лабораторні заняття №:								Практичні заняття				Тестовий контроль	Залік
1*	2	3	4	5	6	7	8	1	2	3	4	Т 1-2	
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)													
5-8	5-8	5-8	5-8	5-8	5-8	5-8	5-8	3-4	3-4	3-4	3-4	5–10	За рейтингом
40–64								12–16				10–20	60-100**

Примітки: *За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання на практичних заняттях

Оцінка за практичні заняття виставляється з урахуванням таких елементів: перевірка знання теоретичного матеріалу з теми у формі короткого опитування; уміння аргументовано пояснювати вибір методів та засобів під час розв'язування завдань; активна участь у дискусіях та обговореннях. Результати практичних занять оцінюються в межах від 3 до 4 балів за кожне заняття. Загальна кількість балів за всі практичні заняття становить від 12 до 16 балів.

Оцінювання результатів захисту лабораторної роботи

Лабораторні роботи є ключовою складовою дисципліни та передбачають практичне закріплення теоретичних знань. При оцінюванні враховуються самостійність і правильність виконання, повнота відповіді, рівень володіння інструментами Java та здатність захистити результати роботи. Кожна лабораторна робота оцінюється в межах від 5 до 8 балів. Загальна кількість балів за всі лабораторні роботи становить від 40 до 64 балів. У разі якщо здобувач продемонстрував рівень знань, нижчий ніж 60% від максимальної кількості балів, лабораторна робота не зараховується. Студент зобов'язаний повторно опрацювати матеріал, виправити помилки та повторно захистити роботу у встановлений викладачем термін.

Оцінювання результатів тестового контролю

Протягом семестру проводяться два тестування, які перевіряють рівень засвоєння теоретичного матеріалу та здатність застосовувати знання на практиці. Кожен тест складається із 20 завдань однакової складності. Залежно від кількості правильних відповідей студент може отримати від 5 до 10 балів за тест. Таким чином, загальний результат тестового контролю за семестр становить від 10 до 20 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	0-11	12-13	14	15-16	17	18-20
Відсоток правильних відповідей	0-59	60-65	66-72	73-82	83-89	90-100
Кількість отриманих балів	0	6	7	8	9	10

На тестування відводиться 25 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. Також, студент може проходити тестування письмово, записуючи правильні відповіді у талоні відповідей. При отриманні негативної оцінки тест слід перездати до терміну *наступного* контролю.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий залік виставляється на останньому занятті за умови якщо загальна сума балів, яку накопичив здобувач з дисципліни (іншого освітнього компонента) за результатами *поточного* контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «*зараховано*», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення. Присутність здобувача у цьому випадку не є обов'язковою.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	<i>Відмінно/Excellent</i> – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		<i>Добре/Good</i> – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		
D	66-72		
E	60-65		<i>Задовільно/Satisfactory</i> – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
FX	40-59	Незараховано	<i>Незадовільно/Fail</i> – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		<i>Незадовільно/Fail</i> – Результати навчання відсутні

10. Питання для самоконтролю результатів навчання

1. Що таке життєвий цикл програми в Java і які його основні етапи?
2. Що таке Java Virtual Machine (JVM) і яку роль вона виконує?
3. Які бувають змінні в Java та як вони оголошуються?
4. Які основні типи даних існують у Java (примітивні та посилальні)?
5. Що таке явне приведення типів і коли воно використовується?
6. Що таке неявне приведення типів і як воно працює?
7. Як визначається пріоритет операцій у Java?
8. Що таке метод у Java і з чого складається його сигнатура?
9. Що таке клас та об'єкт у Java?
10. Що таке базовий клас і які його властивості?
11. Що таке поля та методи об'єкта?
12. Які ключові принципи ООП реалізовані у Java (інкапсуляція, наслідування, поліморфізм)?
13. Що таке модифікатори доступу та для чого вони використовуються?
14. Як здійснюється доступ до приватних полів класу (методи get/set)?
15. Що таке конструктори класів і для чого вони потрібні?
16. Чим відрізняються абстрактні класи від інтерфейсів у Java?
17. Що таке вкладені класи?
18. Яку роль виконує клас Object і які основні методи він має?
19. Що таке масиви в Java і які основні методи для роботи з ними існують?
20. Що таке колекції в Java і для чого вони використовуються?
21. Що таке інтерфейс Collection?
22. Які основні види колекцій існують у Java?
23. Які особливості класів ArrayList та LinkedList?
24. Які особливості класів HashSet та Dictionary?
25. Що таке потоки вводу/виводу (InputStream, OutputStream) у Java?
26. Як здійснюється форматований вивід у Java?
27. Для чого використовуються класи PrintStream та PrintWriter?
28. Які класи застосовуються для роботи з файлами у Java?
29. Які особливості використання FileInputStream та FileOutputStream?
30. Як здійснюється робота з каталогами у Java?
31. Які засоби існують для побудови графічних інтерфейсів у Java?
32. Які основні графічні елементи (компоненти) підтримуються в Java?
33. Що таке компонування елементів і які існують менеджери компоновки?
34. Як працюють кнопки та мітки (Labels, Buttons) у Java?
35. Що таке CheckBox та RadioButton і як вони використовуються?
36. Як відбувається обробка подій елементів інтерфейсу?
37. Що таке технологія JDBC?
38. Як здійснюється підключення до бази даних MySQL з Java?

39. Які основні операції можна виконувати з даними за допомогою Java?
40. Для чого використовуються методи `executeUpdate` та `executeQuery`?
41. Що таке сокети і для чого вони використовуються?
42. Які існують мережеві класи та інтерфейси у Java?
43. Хто такий клієнт і хто такий сервер у мережевому програмуванні?
44. Які є способи організації взаємодії між клієнтом і сервером?
45. Що таке клас `Thread` і як він використовується?
46. Як створюється і завершується потік у Java?
47. Що таке синхронізація потоків і для чого вона потрібна?
48. Що таке семафори і як вони працюють?
49. Як відбувається обмін даними між потоками?
50. Для чого використовуються класи `BigInteger` та `BigDecimal`?

11. Навчально-методичне забезпечення

Навчальний процес з дисципліни «Кросплатформне програмування» повністю забезпечений необхідними навчально-методичними матеріалами. У модульному середовищі для навчання розміщено:

- робочу програму дисципліни;
- методичні рекомендації до виконання лабораторних та практичних робіт;
- матеріали для самостійної роботи;
- тестові завдання для поточного та підсумкового контролю.

12. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Для організації навчального процесу з дисципліни «Кросплатформне програмування» необхідне таке матеріально-технічне забезпечення: персональні комп'ютери, ноутбуки або мобільні пристрої (планшети, смартфони) з доступом до мережі Інтернет, мультимедійний проєктор або інтерактивна дошка для проведення лекційних занять, а також обладнані навчальні аудиторії з робочими місцями для студентів.

Для виконання завдань студенти повинні мати доступ до:

1. Загальноживаних програм: пакети офісних програм (Microsoft Office, LibreOffice, Google Workspace), засоби для створення та перегляду презентацій.
2. Інструментів розробки: інтегрованих середовищ розробки (IntelliJ IDEA, Eclipse, NetBeans або Visual Studio Code).
3. Мов програмування: Java (JDK останніх версій), додатково можуть використовуватися бібліотеки та фреймворки для кросплатформної розробки.
4. Систем управління базами даних: MySQL або інші СУБД з підтримкою JDBC для виконання лабораторних та практичних завдань.
5. Інструментів для контролю версій: Git (GitHub, GitLab, Bitbucket) для роботи над колективними проєктами.

13. Рекомендована література:

1. Java® Notes for Professionals [Електронний ресурс]. – Goalkicker.com, 2018. – Режим доступу: <https://books.goalkicker.com/JavaBook/>
2. Urma, R.-G. Introducing Java 8 [Електронний ресурс] / Raoul-Gabriel Urma. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/introducing-java-8/9781491973157/>
3. Posta, C. Microservices for Java Developers [Електронний ресурс] / Christian Posta. – O'Reilly Media, 2019. – Режим доступу: <https://developers.redhat.com/books/microservices-java-developers>
4. Warburton, R. Object-Oriented vs. Functional Programming [Електронний ресурс] / Richard Warburton. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/object-oriented-vs-functional/9781491973096/>
5. Eisele, M. Modern Java EE Design Patterns [Електронний ресурс] / Markus Eisele. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/modern-java-ee/9781491978916/>
6. Evans, B. Java: The Legend [Електронний ресурс] / Benjamin Evans. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/java-the-legend/9781491932420/>
7. Gupta, A. Docker for Java Developers [Електронний ресурс] / Arun Gupta. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/docker-for-java/9781491983835/>

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL: <https://msn.khmn.edu.ua/course/view.php?id=7516>
2. Електронна бібліотека ХНУ. URL: <http://library.khmn.edu.ua/>
3. Інституційний репозитарій ХНУ. URL: <https://elar.khmn.edu.ua/home>

КРОСПЛАТФОРМНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	—
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни «Кросплатформне програмування» студент повинен знати принципи кросплатформної розробки програмного забезпечення, особливості використання мови Java для створення багатоплатформних застосунків, сучасні методи та інструменти програмної інженерії; уміти розробляти програмні продукти мовою Java з використанням стандартних бібліотек і додаткових інструментів, інтегрувати зовнішні бібліотеки та фреймворки, застосовувати сучасні середовища розробки для створення, налагодження, оптимізації та тестування програмного забезпечення на різних платформах, аналізувати технічну документацію, приймати обґрунтовані інженерні рішення під час виконання практичних і проєктних завдань; мати здатність до подальшої професійної спеціалізації у сфері програмної інженерії, ефективної комунікації у професійному середовищі, командної роботи з проявом ініціативності й відповідальності, організації спільної діяльності та розвитку соціальних навичок (soft skills).

Зміст навчальної дисципліни. Засоби мови Java для кросплатформної розробки; архітектура JVM і принципи «write once, run anywhere»; структура проєктів, бібліотеки та пакети. Використання IDE, довідкової літератури й специфікацій. Основи ООП у Java, робота з колекціями, потоками введення-виведення, винятками. Мережеве та асинхронне програмування, RESTful API, багатопоточність. Робота з базами даних (JDBC, Hibernate), інтеграція бібліотек і фреймворків (Maven, Gradle). Тестування (JUnit, TestNG), налагодження, профілювання та оптимізація. Розробка мобільних і десктопних застосунків, забезпечення сумісності між платформами. Організація командної роботи (Git) та практики безпеки.

Запланована навчальна діяльність: Запланована навчальна діяльність: мінімальний обсяг аудиторних навчальних занять в одному кредиті ЄКТС з дисципліни для першого (бакалаврського) рівня вищої освіти за денною формою становить 10 годин на 1 кредит ЄКТС. Решта годин відводиться на самостійну та індивідуальну роботу студентів.

Форми (методи) навчання: Навчальний процес з дисципліни «Кросплатформне програмування» включає лекційні, практичні, лабораторні та самостійні заняття. Лекції проводяться з використанням мультимедійних презентацій, інтерактивних елементів та проблемного навчання. Практичні заняття охоплюють аналіз прикладів, розв'язування завдань, обговорення кейсів і дискусії, що формує аналітичні та комунікативні вміння. Лабораторні роботи спрямовані на набуття навичок програмування й роботи з інструментами розробки. Самостійна робота передбачає опрацювання теорії, літератури, виконання індивідуальних завдань і підготовку до контролю з використанням ресурсів модульного середовища.

Форми оцінювання результатів навчання: Форми оцінювання результатів навчання з дисципліни «Кросплатформне програмування» включають оцінювання лабораторних і практичних робіт та проходження тестового контролю.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Java® Notes for Professionals [Електронний ресурс]. – Goalkicker.com, 2018. – Режим доступу: <https://books.goalkicker.com/JavaBook/>
2. Urma, R.-G. Introducing Java 8 [Електронний ресурс] / Raoul-Gabriel Urma. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/introducing-java-8/9781491973157/>
3. Posta, C. Microservices for Java Developers [Електронний ресурс] / Christian Posta. – O'Reilly Media, 2019. – Режим доступу: <https://developers.redhat.com/books/microservices-java-developers>
4. Warburton, R. Object-Oriented vs. Functional Programming [Електронний ресурс] / Richard Warburton. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/object-oriented-vs-functional/9781491973096/>
5. Eisele, M. Modern Java EE Design Patterns [Електронний ресурс] / Markus Eisele. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/modern-java-ee/9781491978916/>
6. Evans, B. Java: The Legend [Електронний ресурс] / Benjamin Evans. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/java-the-legend/9781491932420/>
7. Gupta, A. Docker for Java Developers [Електронний ресурс] / Arun Gupta. – O'Reilly Media, 2018. – Режим доступу: <https://www.oreilly.com/library/view/docker-for-java/9781491983835/>
8. Модульне середовище для навчання. URL: <https://msn.khmnu.edu.ua/course/view.php?id=7516>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL: <https://clar.khmnu.edu.ua/home>

Викладачі: старший викладач Денисюк Д.О.