

Силабус навчальної дисципліни

№	Назва поля	Детальний контент, коментарі
1.	Назва факультету	Факультет Комп'ютерних наук
2.	Рівень вищої освіти	Перший (бакалаврський)
3.	Код і назва спеціальності	F3 Комп'ютерні науки
4.	Тип і назва освітньої програми	ОПП «Штучний інтелект»
5.	Назва дисципліни (інформація з ЦІСТ)	Крос-платформне програмування
6.	Кількість ЄКТС кредитів	5
7.	Структура дисципліни (розподіл за видами та годинами навчання)	Лекції – 24 годин, практичні заняття – 24 годин, консультації – 8 годин, самостійна робота – 94 годин. Семестровий контроль – залік.
8.	Графік (терміни) вивчення дисципліни	2-й рік, 4-й семестр
9.	Передумови для навчання за дисципліною	Раніше мають бути вивчені дисципліни: Дискретна математика, Алгоритмізація та програмування, Об'єктно-орієнтоване програмування, Організація баз даних та знань, Web-технології та web-дизайн
10.	Анотація (зміст) дисципліни	<p>Обов'язкова дисципліна циклу професійної та практичної підготовки за спеціальністю, містить змістові модулі:</p> <p>1. Основи кросплатформових технологій</p> <p>Тема 1. Огляд кросплатформових технологій та їх історія (порівняння Java, .Net) Історія кросплатформових технологій: розвиток від перших спроб до сучасних рішень. Переваги та недоліки кросплатформових технологій порівняно з нативними рішеннями. Порівняння різних кросплатформових підходів: гібридні додатки, веб-додатки, нативні обгортки, розвиток на мовах програмування, що використовуються в браузерах. Тенденції розвитку кросплатформових технологій: аналіз сучасних тенденцій та прогнозування майбутнього розвитку.</p> <p>Тема 2. Основи мови Java Основні поняття Java. Просте приклад Java-прикладань, его компіляція та запуск. Види Java-прикладань. Об'єктно-орієнтоване програмування в Java. Пакети. Структура Java-проектів. Основні стандартні пакети Java API. Стандартні типи. Ціла та дійсна арифметика. Класи-обгортки (wrapper classes)). Приведення типів. Керуючі конструкції - цикли, оператори розгалуження, і.т.п. Виключення (exceptions), їх класифікація та обробка. Assertions.</p>

Тема 3. Багатопоточне програмування в Java

Основні поняття та принципи багатопоточного програмування в Java: потоки виконання, синхронізація, взаємодія потоків. Створення потоків в Java: використання класу Thread, інтерфейсу Runnable та інших класів, що дозволяють створювати потоки. Синхронізація потоків: використання synchronized, ReentrantLock, CountdownLatch та інших механізмів для управління доступом до спільних ресурсів. Використання механізмів wait, notify та notifyAll для координації роботи потоків. Розподіл завдань між потоками в Java: використання ThreadPoolExecutor та інших інструментів для ефективного використання ресурсів.

2. Мережева комунікація прикладань, візуальні інтерфейси, бази даних

Тема 4. Мережева взаємодія в Java (Socket, WebSocket)

Мережева взаємодія в Java. Різниця між Java Socket та WebSocket та коли використовувати кожний з них. Побудова взаємодії на основі Java Socket: робота зі з'єднаннями, обробка запитів та відповідей. Використання класу WebSocket: обробка з'єднань, обмін повідомленнями та додаткові налаштування. Асинхронний підхід до мережевої взаємодії: використання CompletableFuture та інших асинхронних інструментів. Протоколи мережевої взаємодії: HTTP, HTTPS, TCP, UDP. Налаштування мережевої взаємодії: використання засобів логування, моніторингу трафіку та дебагінгу.

Тема 5. Створення візуальних інтерфейсів JavaFX, обробка подій (Scene Builder)

JavaFX та Scene Builder: Огляд інструментів та їх основні можливості. Компоненти JavaFX та їх використання: Кнопки, поля введення, списки, таблиці та інші компоненти інтерфейсу. Обробка подій: Додавання обробників подій до компонентів JavaFX, включаючи кліки миші, введення тексту та інші події. Створення віджетів JavaFX за допомогою Scene Builder: Розробка візуальних елементів інтерфейсу за допомогою Scene Builder та їх інтеграція з JavaFX. Налаштування та стилізація інтерфейсу: Налаштування функціональності та вигляду елементів інтерфейсу, зміна фону, шрифтів та ін.

Тема 6. Використання баз даних у Java (SQL, Hibernate, Firebase)

Взаємодія з базами даних у Java: Підключення до баз даних з допомогою JDBC, створення та виконання запитів. ORM засоби у Java: Огляд ORM засобів (Object-Relational Mapping) у Java, таких як Hibernate, JPA, та їх використання для роботи з базами даних. Розробка додатків на основі Hibernate: Створення моделей даних, налаштування підключення до баз даних, робота з сутностями та виконання запитів. Firebase: Взаємодія з Firebase Realtime Database та Cloud Firestore, збереження даних, налаштування прав доступу, виконання запитів.

3. Інструменти для ефективної розробки програмного забезпечення в Java

Тема 7. Git, сховище програмного коду (Git, GitHub, GitLab, SourceTree, Araxis)

Git, його функції, Найважливіші команди Git, такі як ініціалізація репозиторію, додавання та збереження змін, створення та злиття гілок. Робота з репозиторіями на Git: Створення репозиторіїв на Git, клонування та імпортування існуючих репозиторіїв, збірка проекту з використанням Git. Робота зі сховищами коду: Робота з різними сховищами коду, такими як GitHub, GitLab, Bitbucket тощо, включаючи створення проекту та імпортування коду. Використання клієнтів Git: Огляд клієнтів Git, таких як SourceTree включаючи налаштування та роботу з ними.

Тема 8. Використання бібліотек, Apache Maven
Apache Maven, його функції, переваги та області використання. Конфігурація проекту з використанням Apache Maven: Створення нового проекту, додавання залежностей та інші налаштування Maven у файлі pom.xml. Залежності та репозиторії: Додавання залежностей у файл pom.xml, використання репозиторіїв для зберігання залежностей. Розгортання проекту: Розгортання проекту за допомогою Maven, включаючи створення артефактів та дистрибутивів.

4. Поглиблене вивчення кросплатформової технології Java

Тема 9. Розробка серверних програм у Java (Standalone server, Apache Tomcat)

Розробка серверних програм у Java. Розробка standalone-сервера на базі Java Socket. Огляд Apache Tomcat і його можливостей. Розробка сервлетів для Apache Tomcat. Використання JSP.

Тема 10. Використання колекцій у Java

Огляд основних класів колекцій у Java, їх особливості та призначення. Робота зі списками: ArrayList, LinkedList, Vector та їх використання у програмуванні. Робота з множинами: HashSet, TreeSet, LinkedHashSet та їх використання у програмуванні. Робота з мапами: HashMap, TreeMap, LinkedHashMap та їх використання у програмуванні. Робота з чергами: PriorityQueue, ArrayDeque, LinkedList та їх використання у програмуванні. Порівняння колекцій: Колекції, які можна використовувати для різних задач і їх відмінності. Алгоритми роботи з колекціями: Реалізація алгоритмів з використанням колекцій у Java.

Тема 11. Шаблони проектування в Java

Шаблони створення, використовуються для створення об'єктів, включаючи Factory Method, Abstract Factory, Singleton і т.п. Шаблони структури, вони визначають способи організації класів і об'єктів, включаючи Adapter, Bridge, Composite, Decorator і т.п. Шаблони поведінки, ці шаблони визначають способи співпраці між об'єктами, включаючи Strategy, Observer, Template Method, Command і т.п. Проектування прикладань з використанням шаблонів проектування в Java Model-View-Controller (MVC)

5. Створення Instant messenger з використанням технології P2P

Тема 12. P2P технології для передачі відео та аудіо інформації (WebRTC)

Визначення технології WebRTC та її особливості. Переваги та недоліки WebRTC порівняно з іншими технологіями передачі відео та аудіо.

Архітектура WebRTC та її компоненти. Інтеграція WebRTC з іншими технологіями. Приклади застосування WebRTC, включаючи відеоконференції, передачу файлів та стрімінг мультимедіа.

11.	Компетентності, знання, вміння, розуміння, якими оволодіє здобувач вищої освіти в процесі навчання	<p>Дисципліна сприяє формуванню та розвитку наступних компетентностей:</p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК8. Здатність генерувати нові ідеї (креативність).</p> <p>ЗК9. Здатність працювати в команді.</p> <p>ФК8. Здатність проєктувати 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.</p> <p>ФК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.</p>
12.	Результати навчання здобувача вищої освіти	<p>ПР9. 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.</p> <p>ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проєктувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.</p>
13.	Система оцінювання відповідно до кожного завдання для складання заліку/екзамену	<ol style="list-style-type: none"> 1. Індивідуальне завдання, що розкриває знання кожної теми дисципліни. 2. Для підсумкового контролю у формі заліку для оцінювання роботи студента протягом семестру використовують підсумкову рейтингову оцінку $R_{п} = \text{Осем}$. Оцінку за семестр обчислюють як суму оцінок за різні види занять. Підсумкова рейтингова оцінка за семестр обчислюється у 100-бальній системі.
14.	Якість освітнього процесу	<p>Дотримання принципів академічної доброчесності (http://lib.nure.ua/plagiat). Оновлення робочої програми дисципліни – 2022 р. Використовуються: Сучасна мова кросплатформового програмування Java .</p>

15.	Методичне забезпечення	1. Комплекс навчально-методичного забезпечення навчальної дисципліни «Крос-платформне програмування» для студентів усіх форм навчання за спеціальністю 122 Комп'ютерні науки, ОПІ «Штучний інтелект» [Електронний ресурс] / ХНУРЕ; розроб. Шевченко О.Ю. -Х. ХНУРЕ, 2017.- 75с.
16.	Розробник силябусу (посада, ПБ, ел. пошта)	О.Ю. Шевченко, доцент каф. ШІ, к.т.н. e-mail: oleksandr.shevchenko@nure.ua