

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Робоча програма
навчальної дисципліни
"КРОСПЛАТФОРМОВІ ТА БАГАТОЗВЕННІ
ТЕХНОЛОГІЇ"**

**для студентів спеціальності 7.05010101
"Інформаційні управляючі системи та технології"
всіх форм навчання**

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 7 від 14.12.2011 р.

Укладачі: Парфьонов Ю. Е.
Поляков А. О.

P58 Робоча програма навчальної дисципліни "Кросплатформові та багатозвенні технології" для студентів спеціальності 7.05010101 "Інформаційні управляючі системи та технології" всіх форм навчання / укл. Парфьонов Ю. Е., Поляков А. О. – Х. : Вид. ХНЕУ, 2012. – 40 с. (Укр. мов.)

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами, вміщено плани лекцій і лабораторних занять, матеріал щодо закріплення знань, а саме: самостійну роботу, контрольні запитання, а також методичні рекомендації та оцінювання знань студентів.

Рекомендовано для студентів спеціальності 7.05010101 "Інформаційні управляючі системи та технології".

Вступ

У сучасних умовах ефективно управління як в державній так і в господарській діяльності не можливе без використання інформаційних систем. Бізнес-вимоги до таких систем постійно знаходяться у стані удосконалення, розширення, змінення та породження нових для підтримки адекватності сучасного стану. Інфраструктура таких систем часто повинна бути географічно розподіленою, користувачі системи знаходяться в мережі Інтернет, вузли підтримуються різними операційними системами, використовується різні технології обробки та зберігання даних.

При розробленні таких систем необхідно враховувати такі вимоги, як легке масштабування системи, кросплатформна реалізація компонентів, узгодженість компонентів архітектури, чітку специфікацію компонентів, виділення рівнів: подання даних, бізнес-логіки, доступу до даних, збереження даних. Такий підхід до розроблення інформаційних систем дасть змогу зробити більш ефективним подальше використання системи, модифікування та реалізацію нових бізнес-вимог.

Метою навчальної дисципліни є засвоєння необхідних знань щодо основних сучасних технологій створення розподілених додатків, які базуються на платформі Java EE, а також удосконалення практичних навичок з використання мови Java при розробленні розподілених додатків.

Предметом вивчення дисципліни є клієнтська технологія JavaFX та серверні технології JavaServer Pages, JavaServer Faces, Servlets, Web Services, серверні компоненти бізнес-логіки (EJB), стандартизований інтерфейс JPA для об'єктно-реляційного відображення, а також сучасний Java-фреймворк для об'єктно-реляційного відображення Hibernate.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з технічною літературою та сучасним програмними засобами розроблення програм.

Структура навчальної дисципліни "Кросплатформові та багатозвенні технології" наведена в табл.1.

У процесі навчання студенти отримують необхідні знання під час проведення аудиторних занять: лекційних та лабораторних. Також велике значення в процесі вивчення та закріплення знань має самостійна та індивідуальна робота студентів. Усі ці види занять розроблені відповідно до положень кредитно-модульної системи організації навчального процесу.

Структура навчальної дисципліни

Характеристика дисципліни: підготовки спеціалістів	Напрямок, галузь знань, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів, відповідних до ECTS – 4; у тому числі: змістовних модулів – 2; самостійна робота	Шифр та назва галузі знань: 0501 "Інформатика та обчислювальна техніка"	Обов'язкова. Рік підготовки: 5. Семестр: 9
Кількість годин: усього – 144; змістовними модулями: модуль 1 – 84 год.; модуль 2 – 60 год.	Шифр та назва спеціальності: 7.05010101 "Інформаційні управляючі системи і технології"	Лекції – 34 години. Лабораторні заняття – 34 години. Самостійна робота – 76 годин (у тому числі: поточні консультації – 8. Іспит + передекзаменаційна консультація – 4
Кількість тижнів викладання дисципліни: 17. Кількість годин за тиждень: 4	Освітньо-кваліфікаційний рівень:– спеціаліст	Вид контролю: іспит

1. Кваліфікаційні вимоги до студентів

Навчальна дисципліна є фахово-спрямованою для підготовки фахівців зі спеціальності 7.05010101 "Інформаційні управляючі системи і технології".

Необхідна навчальна база перед початком вивчення дисципліни: з метою найкращого засвоєння матеріалу студенти повинні до початку вивчення дисципліни засвоїти теоретичні знання та опанувати практичні вміння з дисциплін "Кросплатформове програмування", "Web-програмування", "Організація баз даних та знань", а також мати навички роботи в інтегрованих середовищах NetBeans та Eclipse.

У результаті вивчення навчальної дисципліни студенти повинні знати:

- основні поняття кросплатформного розроблення розподілених систем;

- модель взаємодії відкритих систем (OSI) та TCP/IP;
- види адресація рівнів в моделі OSI та TCP/IP;
- архітектуру взаємодії "клієнт-сервер";
- модель сокетів Берклі;
- структуру сокетів TCP та UDP;
- архітектуру клієнт-серверного додатка;
- класи та інтерфейси пакету java.net;
- компоненти розподілених систем та моделі їх взаємодії;
- вимоги до розподілених систем;
- архітектуру платформи Java Enterprise Edition;
- призначення серверів додатків та Web-контейнерів;
- серверні Web-технології на платформі Java EE;
- фази "життєвого циклу" сервлета;
- призначення сервлетних фільтрів та їх "життєвий цикл";
- фази "життєвого циклу" JSP;
- модель Model-View-Controller (MVC);
- механізми відслідковування сесій користувача;
- способи використання перенаправлення HTTP-запитів;
- компонентно-орієнтований підхід реалізації Web-додатків на основі JSF;
- етапи "життєвого циклу" та вимоги до контролерів JSF;
- вимоги до бізнес-компонентів EJB;
- типи компонентів, що підтримуються специфікацією EJB 3.0 та 3.1;
- основи об'єктно-реляційного відображення;
- відображення таблиць та зв'язків на POJO-об'єкти;
- мета, принципи та елементи сервіс-орієнтованої архітектури;
- архітектуру JavaFX Rich Internet Application;
- "життєвий цикл" додатка JavaFX;
- основи роботи з графом сцени JavaFX;
- основи 2D- і 3D-трансформацій, переміщень та анімації;
- архітектуру фреймворку Hibernate;
- основні класи та інтерфейси фреймворку Hibernate;
- механізм впровадження залежностей (dependency injection).

Практичні навички, які мають отримати студенти при вивченні навчальної дисципліни:

- працювати з інтегрованими середовищами NetBeans і Eclipse.
- реалізовувати взаємодію клієнта та сервера, використовуючи сокети TCP і UDP;
- реалізовувати Web-додатки за шаблоном MVC на базі сервлетів та JSP-сторінок;
- розробляти архітектуру багаторівневого програмного додатка з виділенням рівнів бізнес-логіки та рівня збереження даних;
- розробляти бізнес-сутності з збереженням та без збереження стану;
- описувати метаданими POJO-класи сутностей для об'єктно-реляційного відображення таблиць бази даних;
- проектувати Web-орієнтовані програмні додатки на базі технології JSF;
- розробляти RIA-додатки з простою анімацією на базі Java FX;
- проектувати рівень збереження даних з використанням фреймворку Hibernate;

Професійні компетентності, якими мають володіти студенти після вивчення навчальної дисципліни:

- здатність застосовувати технології, методи проектування та інструменти для розроблення розподілених додатків на платформах Java SE, Java EE;
- здатність критично вивчати, аналізувати і оцінювати з різних точок зору технології, процеси, методи та інструменти для проектних робіт, пов'язаних з розробленням розподілених програмних систем на платформах Java SE, Java EE, особисті знання і потребу в інших знаннях, перспективи та загальні тенденції щодо розподілених програмних систем;
- здатність вивчати нові технології, методи та прийоми щодо розроблення розподілених програмних систем, а також критично аналізувати їх для професійної роботи.

2. Тематичний план навчальної дисципліни

При вивченні дисципліни "Кросплатформові та багатозвенні технології" студент має ознайомитися з програмою дисципліни, її структурою, формами та методами навчання, видами та методами контролю знань.

Тематичний план дисципліни "Кросплатформові та багатозвенні технології" складається з двох модулів, кожний з яких об'єднує у собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками.

Структура тематичного плану навчальної дисципліни наведена у табл. 2.

Таблиця 2

Структура тематичного плану навчальної дисципліни

Тема	Кількість годин		
	Лекції	Лабораторні заняття	Самостійна робота
Модуль 1. Використання Java-технологій для розроблення Web-додатків на платформі Java EE			
Тема 1. Основи мережної взаємодії	2	4	6
Тема 2. Основи Java EE	2	–	6
Тема 3. Сервлети та JSP	6	6	12
Тема 4. Рівень бізнес-логіки в Java EE	2	8	10
Тема 5. Об'єктно-реляційне відображення в Java EE	2		6
Тема 6. Основні елементи технології JSF	4		8
Разом годин за модулем	18	18	48
Модуль 2. Використання Java-технологій для розроблення розподілених додатків			
Тема 7. Особливості використання технології JSF	2	–	6
Тема 8. Web-служби на платформі Java EE	2	10	4
Тема 9. Розроблення Rich Internet Applications	6		6
Тема 10. Огляд сучасних Java-технологій розроблення багатозвенних додатків	6	6	12
Разом годин за модулем	16	16	28
Всього годин	34	34	76

3. Зміст навчальної дисципліни за модулями та темами

Модуль 1. Використання Java-технологій для розроблення Web-додатків на платформі Java EE

Тема 1. Основи мережної взаємодії

Розподілені системи, компоненти розподіленої системи, вимоги до розподілених систем.

Модель OSI та стек протоколів TCP/IP. Рівні моделі OSI: фізичний, каналний, мережний, транспортний, сесійний, представницький, прикладний. Інкапсуляція даних, адресація мереж, протокол IP (IPv4, IPv6), протоколи транспортного рівня TCP та UDP.

Архітектура "клієнт-сервер", рівні та моделі архітектури "клієнт-сервер", модель сервісу.

Сокети TCP та UDP, TCP-сервер, TCP-клієнт, UDP-сервер, UDP-клієнт, пакет java.net, використання потоків введення-виведення.

Пакет java.nio, канали.

Література: основна [1; 6; 8 – 10]; ресурси мережі Інтернет [15; 20; 21; 30].

Тема 2. Основи Java EE

Рівні компонент розподіленої системи, проміжне середовище та її сервіси, гетерогенна розподілена система.

Розподілені об'єкти, віддалений виклик процедур RMI, синхронні та асинхронні виклики, віддалений інтерфейс, реалізація віддаленого об'єкту.

Платформа Java EE, багатозвенна архітектура Java EE, Java EE-компоненти, Java EE-клієнти, Web-компоненти, бізнес-компоненти, Java EE-контейнери, пакетування EJB, JAR, WAR.

Метадані та java-код, використання анотацій в Java SE та Java EE.

Література: основна [1 – 3; 5; 6; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 25; 31; 33].

Тема 3. Сервлети та JSP

Протокол HTTP 1.0 та 1.1, запити GET, POST, HTML 5, CSS.

Структура Web-додатка, застосування анотацій у Web-додатках, файл web.xml.

Сервлет, "життєвий цикл" сервлета, HTTP-сервлет, "життєвий цикл" HTTP-сервлета, конфігурування сервлета – анотації та web.xml, відстеження сесій, фільтри, "слухачі" подій.

Вирази JSP, дії JSP, директиви JSP, JSP Standard Tag Library, Expression Language.

Література: основна [1; 3; 6; 7; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 25; 31; 33].

Тема 4. Рівень бізнес-логіки в Java EE

EJB-контейнер, сесійні компоненти EJB, компоненти-сутності (Entity Beans), компоненти, що управляються повідомленнями (Message Driven Beans).

Склад EJB-компонента, інтерфейси EJB 3.0, бізнес-інтерфейс, клас EJB-компонента.

Види сесійних компонентів. Сесійні компоненти без підтримки стану (Stateless Session Beans), сесійні компоненти з підтримкою стану (Stateful Session Beans). Життєвий цикл сесійних компонентів.

Література: основна [5; 11; 14; 23]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 31].

Тема 5. Об'єктно-реляційне відображення в Java EE

Реляційна модель даних. Об'єктно-реляційне відображення (Object Relation Mapping). Архітектура Java Persistence API.

Сутність (entity). POJO-клас. Метадані сутності. Доступ до стану сутності.

Об'єктно-реляційне відображення простих типів, первинного ключа, зв'язків таблиць.

Література: основна [14]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Тема 6. Основні елементи технології JSF

Архітектура JSF. Використання моделі Model-View-Controller (MVC).

Компоненти Managed Bean. Компонентная архітектура Java Beans. Вимоги до компонента Managed Bean. Склад компонента Managed Bean. Використання анотацій у компонентах Managed Bean.

Обмін даними між JSF-сторінкою та компонентом Managed Bean. Використання мови Expression Language.

Facelets. Мова XHTML. Стандартні бібліотеки тегів. Використання шаблонів JSF.

Розроблення JSF-дodatка. Сервлет FacesServlet.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Модуль 2. Використання Java-технологій для розроблення розподілених додатків

Тема 7. Особливості використання технології JSF

Використання конверторів та валідаторів.

Події JSF. Оброблення подій. Слухачі подій.

Навігація в JSF.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Тема 8. Web-служби на платформі Java EE

Поняття сервіс-орієнтованої архітектури (SOA). Мета, принципи та елементи SOA.

Web-служби. Universal Description, Discovery and Integration (UDDI). Мова опису Web-служб - Web Services Description Language (WSDL); Протокол Simple Object Access Protocol (SOAP). Транспортні протоколи Web-служб.

Види розміщення Web-служб.

Архітектура JAX-WS. Вимоги до кінцевих точок JAX-WS

Архітектура Representational State Transfer (REST). Переваги REST. Кореневі класи ресурсів RESTful.

Особливості створення Web-методів. Створення програмних клієнтів для взаємодії з Web-службами. Робота з Web-посиланнями в середовищі програмування.

Література: основна [3; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 28; 29; 30; 31; 33].

Тема 9. Розроблення Rich Internet Applications

Основи технології JavaFX. Архітектура JavaFX. Структура сцени. Графічна підсистема JavaFX. 2D і 3D перетворення. Візуальні ефекти.

Розгортання додатку JavaFX.

Використання мови FXML.

Елементи управління. Прив'язка до даних. Події. Використання стилів, графіки, трансформацій та анімації в додатках JavaFX.

Використання CSS в JavaFX; Діаграми. Медіа.

Література: ресурси мережі Інтернет [19].

Тема 10. Огляд сучасних Java-технологій розроблення багатозвенних додатків

Основи Hibernate. Архітектура Hibernate та її основні елементи.

Підтримка діалектів SQL в Hibernate.

Класи-сутності.

Конфігурація і об'єктно-реляційне відображення бази даних. Конфігураційні файли Hibernate.

Базове об'єктно-реляційне відображення. Відображення первинного ключа. Асоціації відображень: many-to-one, one-to-one, one-to-many.

Фабрики сесій (SessionFactory). Клас HibernateUtil. Область сесій і транзакції.

Створення запитів до баз даних: мова SQL, мова запитів Hibernate query language та Criteria API.

Основи Spring. Архітектура Spring. Контейнер Inversion of Control. Фреймворк аспектно-орієнтованого програмування. Фреймворк віддаленого доступу. Фреймворк управління транзакціями.

Література: основна [12; 13]; ресурси мережі Інтернет [15; 20; 21; 22; 30; 32].

4. Плани лекцій

Модуль 1. Використання Java-технологій для розроблення Web-додатків на платформі Java EE

Тема 1. Основи мережної взаємодії

1.1. Розроблення розподілених додатків з використанням мережних бібліотек Java SE.

Фізичний та каналний рівні. MAC-адреса.

Мережний рівень. Протокол IP. Адресація мереж. Маршрутизація.

Транспортний рівень. Протоколи TCP і UDP.

Характеристики протоколів TCP і UDP.

Прикладний рівень. Доменна адресація.

Пакети java.net і java.nio.

Організація мережної взаємодії за протоколом UDP.

Організація мережної взаємодії за протоколом TCP.

Література: основна [1; 6; 8 – 10]; ресурси мережі Інтернет [15; 20; 21; 30].

Тема 2. Основи Java EE

2.1. Вступ до технології Java EE.

Метадані і java-код.

Анотації.

Компоненти розподіленої системи.

Вимоги до розподілених систем.

Рівні компонент розподіленої системи.

Проміжне середовище та його сервіси.

Розподілені об'єкти.

Багатоланкова архітектура Java EE.

Java EE-компоненти.

Java EE-клієнти.

Web-компоненти.

Бізнес-компоненти.

Java EE-контейнери.

Пакування.

Література: основна [1 – 3; 5; 6; 8; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 25; 30; 31; 33].

Тема 3. Сервлети та JSP

3.1. Використання технології сервлетів.

Сервери додатків.

Типова архітектура WEB-орієнтованого Java EE-додатка.

Серверні Java-технології.

Web-контейнер.

Сервлети, їх продуктивність та властивості.

"Життєвий цикл" сервлета на HTTP-сервлета та його методи.

Цикл обробки запита сервлетом.

Бібліотека javax.servlet та сервера TomCat, GlassFish.

Анотації @WebServlet, @WebInitParam та дескриптор розгортання сервлета web.xml.

Структура Web-додатка.

HTTP-запит та його параметри, об'єкт HttpServletRequest. Методи доступу до даних запиту.

Управління запитом: redirect & forward.

Обробка виключень і передача параметрів в сервлетах та JSP.

3.2. Основні елементи технології JSP.

Архітектура Java Server Pages.

Компіляція та "життєвий цикл" JSP.

Співвідношення сервлетів та JSP.

Структура JSP-сторінки. Вбудовані об'єкти сторінки.

Типи конструкцій JSP. Елементи. Дані шаблону.

Бібліотеці користувальницьких тегів JSP.

3.3. Особливості використання технології сервлетів.

Відстеження "життєвого циклу" клієнтських сесій. Використання "кукі". Перезапис URL. Використання скритих полів форми.

Відстеження сесій за допомогою API сервлетів. Основні елементи інтерфейсу javax.servlet.http.HttpSession.

Модель оброблення подій у Web-додатку. Слухачі подій.

Фільтри: призначення, "життєвий цикл", використання.

Література: основна [1; 3; 6; 7; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 24 – 26; 30; 31; 33].

Тема 4. Рівень бізнес-логіки в Java EE

4.1. Основи технології EJB.

Архітектура Java EE.

Структура та рівні додатка Java EE.

Рівень бізнес-логіки.

Типи EJB-компонентів.

Життєвий цикл SSB- і SFEB-компонентів.

Види сеансових EJB-компонентів.

Склад EJB-компонента.

Інтерфейси EJB-компонентів та вимоги до них.

Локальні та віддалені компоненти.

Одержання посилання на EJB-компонент: JNDI та "ін'єкція залежностей".

Розроблення та використання EJB-компонентів.

Література: сновна [5; 11; 14; 23]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 31].

Тема 5. Об'єктно-реляційне відображення в Java EE

5.1. Основи використання Java Persistence API.

Рівень збереження даних Java EE.

Реляційна та об'єктно-реляційна моделі даних.

Об'єктно-реляційне відображення.

Призначення Java Persistence API.

POJO-класи. Компоненти-сутності та вимоги до них.

Управління сутностями. Зберігання, пошук та видалення сутностей.

Анотації, що використовуються в Java Persistence API. Анотація іменованих запитів.

Література: основна [14]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 31; 33].

Тема 6. Основні елементи технології JSF

6.1. Основи технології JSF.

Призначення та архітектура JSF.

Структура каталогів JSF-додатка.

Склад типового JSF-додатка.

Презентаційні технології для JSF-додатків.

Бібліотеки тегів.

Етапи розроблення JSF-додатка.

6.2. Використання серверних елементів управління.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Модуль 2. Використання Java-технологій для розроблення розподілених додатків

Тема 7. Особливості використання технології JSF

7.1. Особливості використання технології JSF.

Огляд конверторів та валідаторів. Стандартні конвертори та валідаторі, їх використання.

Події JSF. Оброблення подій. Слухачі подій.

Архітектура системи навігації в JSF.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Тема 8. Web-служби на платформі Java EE

8.1. Web-служби на платформі Java EE.

Web-служби та сервіс-орієнтована архітектура.

Принципи SOA.

Елементи архітектури SOA.

Стандарті Web-сервісів.

Розміщення WEB-сервісів.

Архітектура JAX-WS.

Анотації @WebService, @WebMethod, @WebParam, @WebResult.

Вимоги кінцевих точок JAX-WS.

Representation State Transfer (RESTful) Web Service.

"Життєвий цикл" запиту RESTful-ресурсу.

Методи доступу до RESTful-ресурсів.

Література: основна [3; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 28; 29; 30; 31; 33].

Тема 9. Розроблення Rich Internet Applications

9.1. Основи технології JavaFX.

Сучасні технології Rich Internet Applications.

Архітектура JavaFX.

Робота з графом сцени JavaFX.

Інтерфейс користувача в JavaFX.

Робота з візуальними елементами управління.

Обробка подій в JavaFX.

Пакетування та розгортання JavaFX-додатків.

JavaFX та Swing.

9.2. Використання графічних можливостей технології JavaFX.

Графічна підсистема JavaFX. 2D і 3D перетворення. Візуальні ефекти.

Застосування CSS.

Додавання HTML-контенту.

Ефекти, трансформації та анімація.

9.3. Використання мови FXML.

Література: ресурси мережі Інтернет [19].

Тема 10. Огляд сучасних Java-технологій розроблення багатозвенних додатків

10.1, 10.2. Використання Hibernate.

Архітектури Hibernate.

Діалекти баз даних в Hibernate.

Класи-сутності в Hibernate.

Конфігурація і об'єктно-реляційне відображення бази даних.

Файли конфігурації Hibernate та їх використання.

Створення фабрики сесій.

Транзакції.

10.3. Основи Spring.

Архітектура Spring.

Інверсія управління (IoC). Контейнер IoC.

Аспектно-орієнтоване програмування.

Література: основна [12; 13]; ресурси мережі Інтернет [15; 20 – 22; 30 – 32].

5. Плани лабораторних занять

Лабораторне заняття – це організаційна форма навчального заняття, на якому студенти під керівництвом викладача формують уміння й навички з практичного застосування основних теоретичних положень навчальної дисципліни шляхом індивідуального виконання завдань до лабораторних робіт.

Проведення лабораторного заняття ґрунтується на попередньо підготовленому методичному матеріалі – запитаннях для виявлення ступеня оволодіння студентами необхідними теоретичними положеннями, наборі завдань різної складності для розв'язання їх студентами на занятті.

Лабораторні заняття з дисципліни "Кросплатформові та багатозвенні технології" проводяться в спеціально обладнаному навчальному класі з використанням комп'ютерного устаткування, пристосованого до навчального процесу.

Лабораторне заняття включає проведення попереднього контролю знань, вмінь і навичок студентів, постановку загальної проблеми викладачем та її обговорення за участю студентів, розв'язування завдань з їх обговоренням, розв'язування контрольних завдань, їх перевірку, оцінювання.

З метою підвищення якості навчального процесу під час проведення лабораторного заняття призначається ще один викладач і навчальна група ділиться на дві підгрупи. Кожний студент працює самостійно, виконуючи індивідуальне завдання для лабораторного дослідження.

За результатами виконаної на занятті лабораторної роботи студенти оформлюють індивідуальні звіти з її виконання та захищають їх перед викладачем. Результати виконання лабораторних досліджень оцінюються викладачем.

Оцінки, отримані студентом за окремі лабораторні заняття, враховуються при виставленні поточної модульної (практичний модульний контроль) оцінки з даної навчальної дисципліни. Перелік тем лабораторних занять наведено в табл. 3.

Таблиця 3

Перелік тем лабораторних занять

Теми лабораторних занять	Кількість годин	Література
Модуль 1. Використання Java-технологій для розроблення Web-додатків на платформі Java EE		
1. Використання сокетів у розподілених додатках	4	Основна [1; 6; 8 – 10]; Ресурси мережі Інтернет [15; 20; 21; 30]
2. Розроблення Web-додатка з використанням технологій сервлетів та JSP	6	Основна [1; 3; 6; 7; 11]; Ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 25; 31; 33]
3. Розроблення додатків з використанням основних елементів платформи Java EE	8	Основна [5; 11; 14; 23]; Ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 31]
Модуль 2. Використання Java-технологій для розроблення розподілених додатків		
4. Розроблення Java EE-додатка з використанням JavaFX та Web-служби	6	Основна [3; 11]; Ресурси мережі Інтернет [15; 17 – 21; 28; 29; 30; 31; 33]
5. Розроблення додатка з використанням графічних можливостей технології JavaFX.	4	Ресурси мережі Інтернет [19]
6. Розроблення додатка з використанням технології Hibernate	6	Основна [1 – 3; 5 – 7; 11; 14]; Ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 26; 30; 31; 33]
Усього годин	34	

6. Самостійна робота студентів

6.1. Основні форми самостійної роботи студентів

Для ефективного вивчення дисципліни "Кросплатформові та багатозвенні технології" окрім лекційних і лабораторних занять значну увагу слід приділяти самостійній роботі студентів.

Основні форми самостійної роботи студента:
вивчення лекційного матеріалу;
вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
вивчення основних термінів та понять за темами дисципліни;
підготовка до лабораторних занять;
контрольна перевірка кожним студентом знань за питаннями для самодіагностики;
підготовка доповіді для виступу на лабораторному занятті;
підготовка до проміжного та підсумкового модульного контролю;
систематизація вивченого матеріалу перед іспитом;
оформлення звітів з лабораторних робіт;
робота з опрацювання та вивчення рекомендованої літератури.

6.2. Питання для самостійного опрацювання

Модуль 1. Використання Java-технологій для розроблення Web-додатків на платформі Java EE

Тема 1. Основи мережної взаємодії

1. Рівні модель OSI. Призначення та властивості.
2. Фізичний рівень.
3. Канальний рівень. Адресація та протоколи.
4. Мережний рівень. Адресація, протоколи, маршрутизація.
5. Транспортний рівень. Протоколи TCP і UDP та їх властивості.
6. Сеансовий рівень.
7. Представницький рівень. Кодування даних.
8. Прикладний рівень. Протоколи та сервіси, доменна адресація.
9. Підтримка мережної моделі в Java.
10. Пакет `java.net`, класи `Inet6Address`, `Proxy`, `MulticastSocket`.
11. Пакет `javax.net` та `javax.net.ssl`. Класи підтримки мережної автентифікації та захищеного з'єднання.
12. Блокування та тайм-аут.
13. Буферизація та TCP.

Література: основна [1; 6; 8 – 10]; ресурси мережі Інтернет [15; 20; 21; 30].

Тема 2. Основи Java EE

1. Розподілені системи.
2. Сучасні кросплатформові технології.
3. Модульність та компонентність розподіленої системи.
4. Сервери додатків.
5. Можливості та особливості сервера Oracle Web Logic.
6. Можливості та особливості сервера IBM WebSphere.
7. Можливості та особливості сервера Apache Tomcat.
8. Можливості та особливості сервера Oracle GlassFish.
9. Видалений виклик процедур RMI.
10. Архітектура RMI.
11. Платформа Java Enterprise Edition.
12. Архітектура Java EE 6
13. Анотації.

Література: основна [1 – 3; 5 ;6;11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23 – 25; 31; 33].

Тема 3. Сервлети та JSP

1. Огляд серверних Web-технологій.
2. Компоненти Web-серверів.
3. Архітектура сервера додатків Oracle Web Logic
4. Архітектура сервера додатків IBM WebSphere.
5. Архітектура сервера додатків Apache Tomcat.
6. Архітектура сервера додатків Oracle GlassFish.
7. Налаштування домена сервера.
8. Особливості Web-дodatка за технологією Java EE.
9. Серверні Java-технології.
10. Механізми безпеки WEB-контейнерів.
11. Контекст сервлетів. Область дії, ініціалізація, конфігурація.
12. Запити параметри та атрибути.
13. Механізми відстеження сесії користувача.
14. Архів WAR пакету розгортання Web-дodatка.
15. JSP-компіляція та "життєвий цикл" JSP.
16. JavaBeans в Java Server Pages.
17. Обробка помилок та виключень.
18. JavaBean и JSP Expression language.
19. Користувальницькі бібліотеки тегів.

20. Інтернаціоналізація.
21. Фільтри для зміни функціональності додатка.
22. Безпека та конфіденційність з'єднання.

Література: основна [1; 3; 6; 7; 11]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 24 – 26; 31; 33].

Тема 4. Рівень бізнес-логіки в Java EE

1. Рівень бізнес логіки в Java EE.
2. Клієнти сесійних компонентів.
3. Підтримка іменування та каталогів (JNDI).
4. Сесійний компонент без збереження стану (Stateless Session Bean).
5. Сесійний компонент з збереженням стану (Stateful Session Bean).
6. Сесійний компонент-"одиночка" (Singleton Session Bean).
7. Message-Driven Beans. Контракт з контейнером.
8. Управління транзакціями в EJB 3.0

Література: основна [5; 11; 14; 23]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 31].

Тема 5. Об'єктно-реляційне відображення в Java EE

1. EJB і рівень збереження даних.
2. Стратегії об'єктно-реляційного відображення.
3. Переваги та недоліки об'єктно-реляційного відображення.
4. Управління класами-сутностями: EntityManager.
5. Життєвий цикл JPA.
6. Наслідування сутностей.
7. Одна таблиця на ієрархію класів.
8. Відображення таблиці на відповідний клас.
9. Відображення таблиці на підклас.
10. Запити, критерії та JPA QL.
11. Об'єктні зворотні виклики та слухачі.

Література: основна [14]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 33].

Тема 6. Основні елементи технології JSF

1. Компоненти Managed Bean.
2. Компонентная архітектура Java Beans.

3. Вимоги до компонента Managed Bean.
4. Склад компонента Managed Bean.
5. Використання анотацій у компонентах Managed Bean.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Модуль 2. Використання Java-технологій для розроблення розподілених додатків

Тема 7. Особливості використання технології JSF

1. Статична та динамічна навігація.
2. Правила навігації.
3. Розроблення користувальницьких конверторів.
4. Розроблення користувальницьких валідаторів.
5. Фази подій.
6. Системні події.

Література: основна [4]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 27; 30; 31; 33].

Тема 8. Web-служби на платформі Java EE

1. Функції SOA.
2. Елементи архітектури SOA.
3. Enterprise Service Bus.
4. Передача даних. Специфікація SOAP.
5. Стандарт передачі даних JSON.
6. Організація реєстру сервісів. Стандарт UDDI.
7. Мова опису сервісів WSDL.
8. Архітектура JAX-WS.
9. Підтримка типів в JAX-WS.
10. JAXB-сумісність типів параметрів та повернень.
11. Класи та інтерфейси javax.jws і java.jws.soap.
12. Representation State Transfer (RESTful) Web Service.
13. Інтеграція JAX-RS с EJB.
14. Підтримка методів GET і PUT HTTP запиту.
15. Витягування даних з запиту.
16. Класи та інтерфейси javax.ws, java.ws.core і java.ws.ext.

Література: основна [14]; ресурси мережі Інтернет [15; 17; 18; 20; 21; 23; 30; 33].

Тема 9. Розроблення Rich Internet Applications

1. Adobe Flash.
2. Silverlight.
3. HTML 5.
4. JavaFX.
5. Компоненти архітектури JavaFX.
6. Інтеграція з платформою Java.
7. Розроблення додатка JavaFX.
8. Дерево графа сцени в JavaFX.
9. Спеціалізовані колекції в JavaFX.
10. Інтерфейс користувача.
11. Робота з макетами.
12. Додавання тексту.
13. Механізм Drag і Drop.
14. 2D і 3D трансформації.
15. Робота з Doclet.
16. Додавання медіа-ресурсів в додатки JavaFX.

Література: ресурси мережі Інтернет [19].

Тема 10. Огляд сучасних Java-технологій розроблення багатозвенних додатків

1. Hibernate.
2. Пул підключень Hibernate.
3. Інтеграція Hibernate з сервером додатків Java EE.
4. Типи Hibernate.
5. Альтернативи метаданих: XDoclet і Annotation.
6. Колекція відображень Hibernate.
7. Мова запитів Hibernate.
8. Критерії запитів у Hibernate.
9. Пакетна обробка запитів у Hibernate.
10. Модулі фреймворку Spring.
11. Контейнер IoC. "Ін'єкції залежності". Конфігурування залежностей.
12. Ресурси Spring. Вбудована реалізація ресурсів.
13. Валидація в Spring, зв'язування даних і перетворення типів.
14. Аспектно-орієнтоване програмування в Spring.

Література: основна [12; 13]; ресурси мережі Інтернет [15; 20 – 22; 30; 31].

6.3. Тематика доповідей

Для перевірки ступеня оволодіння студентами необхідними теоретичними положеннями дисципліни на початку лабораторного заняття на протязі 10 хвилин заслуховуються міні-доповіді студентів тривалістю до 3-х хвилин. Доповідь супроводжується демонструванням презентації (2 – 3 слайда).

1. Історія розвитку технологій Java EE.
2. Встановлення з'єднання за протоколом TCP.
3. Взаємодія за протоколом UDP.
4. Підходи до розроблення мережних багатопоточних додатків.
5. Переваги використання селекторів у мережних додатках.
6. Асинхронна робота мережних додатків.
7. Використання сокетів Берклі.
8. Інтерфейс передачі повідомлень (MPI).
9. Реалізація протоколів прикладного рівня.
10. Підходи до реалізації Proxy-сервера.
11. Підходи до реалізації SendMail-сервера.
12. Роль RMI в архітектурі Java EE.
13. Обробка запитів сервлетами з використанням захищеного протоколу HTTPS.
14. Механізми відстеження сесій в розподіленому Web-додатку.
15. Механізми програмної підтримки безпеки в сервлетах та JSP.
16. Шифрування, архівування та кешування даних в сервлетах.
17. Управління станом в Web-додатку на базі сервлетів та JSP.
18. Використання JSP 2.0 Expression language.
19. Підходи до реалізація простого Web-сервера.
20. Анотації та "ін'єкція залежностей" (Dependency Injection) в EJB 3.0.
21. Управління транзакціями в EJB 3.0.
22. Використання EJB 3.0 Timer Services в додатках Java EE.
23. Використання сервісів Java Authentication і Authorization.
24. Клієнти сесійних компонентів: локальний, видалений та Web-служба.
25. Переваги сервіс-орієнтованої архітектури інформаційної системи підприємства.
26. Сучасні підходи до створення ESB.
27. Програмне забезпечення за вимогою.

28. Робота з сервісами Google.
29. Робота з сервісами Amazon.
30. Програмне забезпечення як послуга.
31. Порівняння JWS-RS і JXA-WS Web-служб.
32. Використання RIA-додатків в бізнесі.
33. Використання RIA-додатків в іграх.
34. Використання RIA-додатків у навчальному процесі.
35. Продуктивність 2D графіки в RIA-додатках.
36. Продуктивність 3D графіки в RIA-додатках.
37. Спеціалізовані потоки в JavaFX.
38. Використання потокового відео та аудіо в додатках JavaFX.
39. Ефективність використання об'єктно-реляційного відображення в програмному забезпеченні рівня підприємства.
40. Відміна Hibernate від інших фреймворків об'єктно-реляційного відображення.
41. Інтеграція Hibernate з технологіями Java EE.
42. Реалізація зв'язку між таблицями в Hibernate.
43. Підключення Hibernate до серверів ApacheTomcat та GalassFish.
44. Hibernate Envers: аудіювання записів.
45. Критерії користувача в Hibernate.
46. Типи користувача в Hibernate.
47. Використання Spring в розподілених додатках.
48. Web-патерн MVC фреймворку Spring.
49. Можливості посередницького рівня Data Access у фреймворку Spring.
50. Інтеграція Spring з Enterprise Java Bean.

7. Контрольні запитання для самодіагностики

1. У чому полягає призначення рівнів модель OSI?
2. Стек протоколів TCP/IP.
3. IP-адресація. Класи адресації.
4. Призначення протоколу TCP.
5. Призначення протоколу UDP.
6. Основні відмінності протоколів TCP і UDP.
7. Що таке порт?
8. Що таке сокет?
9. У чому полягає суть сокетів Бекрлі?

10. Призначення класу InetAddress.
11. Робота з класом URLConnection.
12. Призначення класів Socket, ServerSocket.
13. Призначення класів DatagramPacket, DatagramSocket.
14. Розкрийте сутність роботи метода accept() та його призначення в програмі.
15. Опишіть архітектуру розподіленого програмного додатка.
16. Описати алгоритм функціонування серверного додатка.
17. Описати алгоритм функціонування клієнтського додатка.
18. Навести каркас коду серверного додатка.
19. Навести каркас коду клієнтського додатка.
20. Назвіть основні класи й інтерфейси пакту java.net.
21. У чому полягає особливість бібліотеки java.nio?
22. Призначення мови HTML та її недоліки.
23. Призначення каскадних таблиць стилів.
24. Призначення методу GET протоколу HTTP та особливості його застосування.
25. Призначення методу POST протоколу HTTP та особливості його застосування.
26. Що розуміється під сервлетом?
27. Що таке Web-контейнер?
28. Що становить сервер додатків? Наведіть приклади серверів додатків.
29. Архітектура платформи Java EE.
30. Перелічіть технології, що підтримуються Web-контейнерами.
31. Призначення та особливості сервлетів.
32. Призначення та особливості JSP.
33. Що розуміється під JSP-компіляцією?
34. Розкрийте сутність методів "життєвого циклу" сервлета.
35. "Життєвий цикл" JSP-сторінки.
36. Поняття Web-додатка та його структура.
37. Поняття про дескриптор розгортання Web-додатка.
38. Яким чином виконується передача даних між сервлетами та JSP?
39. Поясніть роботу forward та redirect.
40. Що таке фільтри і як вони працюють?
41. Що, становить бізнес-логіка додатка?

42. Поясніть, що реалізує бізнес-рівень додатка?
43. Якою технологією Java EE реалізується бізнес-логіка?
44. Поясніть архітектуру компонента EJB.
45. Що становить EJB-контейнер?
46. Чим відрізняється Web-контейнер від EJB-контейнера?
47. Які існують види компонентів EJB?
48. Які існують типи сесійних компонентів EJB?
49. Які етапи "життєвого циклу" сесійних компонентів?
50. Призначення сесійного компонента без збереження стану.
51. Призначення сесійного компонента з збереженням стану.
52. У чому полягає різниця між компонентами Stateless та Stateful?
53. Що таке бізнес-інтерфейс?
54. Призначення локального та віддаленого інтерфейсів.
55. Якими анотаціями описуються сесійні компоненти EJB?
56. Якими анотаціями описується компонент EJB на боці клієнта?
57. Розкрийте суть компонента Message Driven Bean.
58. Розкрийте суть об'єктно-реляційного відображення.
59. Як подаються таблиці бази даних при об'єктно-реляційному відображенні?
60. Що таке POJO-класи?
61. Який компонент EJB описує сутності?
62. Опишіть "життєвий цикл" сутності.
63. Призначення рівня збереження даних.
64. Якими анотаціями описуються класи-сутності?
65. Яким чином відображається первинний ключ на клас сутності?
66. Що таке пул підключень до бази даних?
67. Що таке Web-служба?
68. Що визначає протокол SOAP?
69. Що визначає мова WSDL?
70. Для чого призначена інструмент UDDI?
71. Принципи SOA.
72. У чому полягає сутність SOA?
73. Архітектура WEB-служби JXA-WS.
74. Що означає поняття "кінцева точка"?
75. Якими анотаціями описується Web-служба JXA-WS?
76. Що становить Web-служба RESTful?
77. У чому полягає різниця між Web-службами JXA-WS та RESTful?

78. Який зв'язок між протоколом HTTP та RESTful Web-службою?
79. Що становить RIA-додаток?
80. Які сучасні RIA-технології існують?
81. Яка архітектура технології JavaFX?
82. Що таке граф сцени?
83. "Життєвий цикл" додатка JavaFX.
84. Що становить мова FXML?
85. Особливості колекцій JavaFX?
86. Яким чином виконується трансформація об'єктів сцени в JavaFX?
87. Які меді-ресурси доступні для використання в JavaFX і яким чином?
88. Що таке фреймворк?
89. Архітектура фреймворку Hibernate?
90. Яким чином реалізується об'єктно-реляційне відображення в Hibernate?
91. Який існує зв'язок між JPA та Hibernate?
92. Що таке XDoclet?
93. Як в Hibernate реалізується зв'язок між таблицями бази даних?
94. Для чого призначений фреймворк Spring?
95. Архітектура фреймворку Spring?
96. Призначення контейнера Inversion of Control?
97. Що таке аспектно-орієнтоване програмування?
98. Модель MVC у фреймворку Spring.

8. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, що винесені на поточний контроль тощо.

З теоретичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

1) індивідуальних консультацій, на яких студент отримує відповідь від викладача на конкретні запитання або пояснення певних теоретичних положень чи аспектів їх практичного застосування;

2) групових консультацій, на яких викладач розглядає типові приклади з використання концепцій об'єктно-орієнтованого програмування.

З практичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

1) індивідуальних консультацій, на яких викладач розглядає лабораторні завдання, стосовно яких виникли запитання у студента;

2) групових консультацій, на яких викладач розглядає практичні ситуації, які потребують колективного обговорення.

Індивідуально-консультативна робота для комплексної оцінки засвоєння програмного матеріалу проводиться у вигляді:

1) індивідуального захисту самостійних та індивідуальних завдань;

2) підготовки рефератів для виступу на науковому семінарі;

3) підготовки рефератів для виступу на науковій конференції.

9. Методики активізації процесу навчання

При викладанні навчальної дисципліни "Кросплатформові та багатозвені технології" для активізації навчально-пізнавальної діяльності студентів передбачено застосування сучасних навчальних технологій, таких, як: проблемні лекції, міні-лекції, робота в малих групах, презентації.

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни наведено в табл. 4.

Проблемні лекції спрямовані на розвиток логічного мислення студентів. Коло питань теми лекції обмежується двома – трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов відображення в підручниках, використовується досвід закордонних навчальних закладів з роздаванням студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор задає запитання, які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку потрібно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати лекційний матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

**Розподіл форм та методів активізації навчання за темами
навчальної дисципліни**

Тема	Практичне застосування навчальних технологій
Тема 1. Основи мережної взаємодії	Міні-лекція: "Сучасні технології програмування взаємодії клієнт-сервер". Робота в малих групах: "Спільне розроблення клієнт-серверного додатка"
Тема 3. Сервлети та JSP	Проблемна лекція на тему: "Сучасні Java-технології розроблення Web-додатків". Міні-лекція: "Програмування простого HTTP-сервлета"
Тема 5. Об'єктно-реляційне відображення в Java EE	Проблемна лекція на тему: "Сучасні підхід до розробки зв'язку з БД у розподіленому програмному забезпеченні. Persistence Tier – JPA"
Тема 9. Розроблення Rich Internet Applications	Проблемна лекція на тему: "Сучасні Rich Internet Applications та тенденції їх розвитку"
Тема 10. Огляд сучасних Java-технологій розроблення багатозвенних додатків	Міні-лекція: "Програмування рівня збереження даних з використанням Hibernate"

Міні-лекції передбачають викладання навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження.

На початку проведення міні-лекції за вказаними раніше темами лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані в плані лекцій, але викладаються вони стисло. Лекційне заняття, проведене в такий спосіб, пробуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача.

Проблемні лекції та міні-лекції доцільно поєднувати з такою формою активізації навчального процесу, як робота в малих групах.

Робота в малих групах дає змогу структурувати лекційні або лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

Після висвітлення проблеми (при використанні проблемних лекцій) або стислого викладання матеріалу (при використанні міні-лекцій) студентам пропонується об'єднуватися у групи по 5 – 6 чоловік і презентувати наприкінці заняття своє бачення та сприйняття матеріалу.

10. Система поточного та підсумкового контролю знань студентів

Система оцінювання знань, вмінь та навичок студентів передбачає виставлення оцінок за усіма формами проведення занять.

Перевірка та оцінювання знань студентів може проводитись у таких формах:

1. Оцінювання роботи студентів в процесі лабораторних занять.
2. Проведення поточно-модульного контролю.
3. Проведення підсумкового іспиту.

Загальна модульна оцінка складається з поточної оцінки, яку студент отримує під час лабораторних занять, та оцінки за виконання модульної контрольної роботи.

Порядок оцінювання знань студентів під час лабораторних занять проводиться за 12-бальною шкалою за такими критеріями:

- 1) розуміння, ступінь засвоєння матеріалу навчальної дисципліни;
- 2) ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;
- 3) уміння поєднувати теорію з практикою при розробці програм, розв'язанні задач, проведенні розрахунків при виконанні завдань, винесених для самостійного опрацювання, та завдань, винесених на розгляд в аудиторії;
- 4) логіка, структура, якість оформлення, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" (10 – 12 балів) ставиться за умови відповідності звіту з виконаного лабораторного завдання студента та його усної відповіді при захисті завдання усім чотирьом зазначеним критеріям. Відсутність тієї або іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу).

Проведення поточно-модульного контролю

Поточно-модульний контроль здійснюється та оцінюється за двома складовими: практичний модульний контроль і теоретичний модульний контроль. Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час лабораторних занять. Теоретичний модульний контроль здійснюється у письмовій формі.

Для підведення підсумків роботи студентів із змістовного модуля виставляється підсумкова оцінка з поточно-модульного контролю, яка враховує оцінки за практичний та теоретичний модульний контроль.

Проведення підсумкового іспиту

Іспит здійснюється у на комп'ютерах за екзаменаційними білетами. Екзаменаційний білет складається з двох завдань. Результаті іспиту оцінюються за 12-бальною шкалою. Підсумкова оцінка за іспит є сумою оцінок за кожне завдання. Кожне завдання оцінюється від 0 до 6 балів відповідно до такої шкали (табл. 5):

Шкала оцінювання підсумкового іспиту

Бал	Критерії
6	Завдання виконано в повному обсязі. Програма працює правильно. Інтерфейс та вихідний код програми задовольняють встановленим вимогам
5	Завдання виконано. Програма працює правильно, але одна з її функціональних можливостей реалізована з порушенням вимог, вказаних в завданні
4	Завдання в основному виконано. Програма працює, але дві з її функціональних можливостей реалізовані з порушенням вимог, вказаних в завданні
3	Завдання виконано, але не в повному обсязі. Програма працює, але не реалізована одна з функціональних вимог, вказаних в завданні, або три з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні
2	Завдання не виконано. Програма запускається, але не реалізовані дві з функціональних вимог, вказаних в завданні, або чотири з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні

1	2
1	Програма запускається та частково відповідає постановці завдання, але не реалізовані більше двох з функціональних вимог, вказаних в завданні, або більш чотирьох з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні. Програма не запускається або завершується аварійно, але є програмний код, розроблений студентом, який відповідає постановці завдання
0	Програма відсутня. Програма не містить програмного коду, розробленого студентом. Програма не відповідає постановці завдання. Програма має явні ознаки несаможитності її розробки

Невиконання або суттєве порушення будь-якої з загальних вимог до програми, наведених далі, зніжує кількість балів по завданню на 1.

Загальні вимоги до програми

Вимоги до інтерфейсу користувача

1. Інтерфейс має відповідати постановці завдання.
2. Інтерфейс повинен мати текстову інформацію, яка відносяться до введення та виведення основних та допоміжних даних.
3. Назви елементів інтерфейсу повинні бути виконані російською або українською мовою.

Вимоги до вихідного тексту

1. Наявність коментарів (для класів – призначення класу; для методів – призначення методу, опис параметрів та значення, що повертається) з обов'язковим використанням відповідних документальних тегів JavaDoc.
2. Додержання принципу інкапсуляції щодо рівнів доступу до полів, властивостей та методів класів.
3. Вихідний код кожного з класів програми повинен міститися в окремому файлі.

Для підведення підсумків роботи студента з навчальної дисципліни виставляється загальна оцінка, яка враховує оцінки з кожного виду контролю (дві оцінки поточно-модульного контролю за роботу протягом семестру та оцінку за результатами іспиту):

$$П = 0,6 \cdot E + 0,4 \cdot (M1 + M2),$$

де П – підсумкова оцінка з навчальної дисципліни;

Е – екзаменаційна оцінка;

M1, M2 – оцінки за модулями.

Підсумкова оцінка з дисципліни згідно з методикою переведення показників успішності знань студентів Університету в систему оцінювання за шкалою ECTS конвертується в підсумкову оцінку за шкалою ECTS (табл. 6).

Таблиця 6

**Переведення показників успішності знань студентів
у систему оцінювання за шкалою ECTS**

Відсоток студентів, які зазвичай успішно досягають успішної оцінки	Оцінка за шкалою ECTS		Оцінка за бальною шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
10	відмінне виконання	A	12 – 11	відмінно
25	вище середнього рівня	B	10	
30	взагалі робота правильна, але з певною кількістю помилок	C	9 – 7	добре
25	непогано, але із значною кількістю помилок	D	6	задовільно
10	виконання задовольняє мінімальні критерії	F	5 – 4	
–	потрібне повторне перекладання	FX	3	незадовільно
–	повторне вивчення дисципліни	F	2 – 1	

Приклад екзаменаційного білета з дисципліни

Міністерство освіти і науки, молоді та спорту України
Харківський національний економічний університет

Спеціальність 7.05010101

Дисципліна: "Кросплатформові та багатозвенні технології" (5 курс)

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

Завдання 1

Розробити додаток JavaFX, який дозволяє малювати одну з заданих геометричних фігур в клієнтській області його головного вікна.

Вимоги до програми:

1. Геометричні фігури – коло (Circle) та квадрат (Rectangle).
2. Фігура повинна малюватися по кожному подвійному клацанню правою кнопкою миші, а координати "центру" фігури повинні збігатися з координатами курсору миші у момент "клацання".
3. Для забезпечення можливості зміни типу геометричної фігури, що малюється, головне вікно додатка повинне використовувати елемент управління "Список" (ChoiceBox) з двома пунктами. Після вибору першого пункту "Списку" повинні малюватися тільки кола, а після вибору другого пункту – тільки квадрати.

Завдання 2

З використанням технологій Java Server Pages (JSP) та сервлетів розробити Web-додаток клієнт-сервер, який надає інформацію про середню оцінку студента за результатами сесії.

Web-інтерфейс додатку має бути сторінкою JSP, яка дозволяє вибрати прізвище студента із списку та відправити його сервлету.

Бізнес-логіка додатку повинна реалізовуватися в сервлеті, який:

1. Містить асоціативний масив "ключ-значення", де ключами є прізвища студентів, а значеннями – їх середні оцінки.
2. Обчислює середні оцінки студентів, використовуючи оцінки, одержані з трьох дисциплін. Оцінка за дисципліною – випадкове число, рівномірно розподілене на інтервалі [7;9].

3. Додає до контейнера дані про середні оцінки декількох студентів.

4. Має метод, який за прізвиськом, отриманим від Web-інтерфейсу, шукає в контейнері середню оцінку та повертає її.

5. Генерує HTML-сторінку, де відображується результат пошуку.

Розроблені до кожного завдання проекти (кожний в своїй папці з іменами "Task 1", "Task 2") зберегти разом в окремій папці на диску.

Затверджено на засіданні кафедри "Інформаційних систем"
протокол № 7 від 14.12.2011 р.

Зав. кафедри _____ проф. В. С. Пономаренко

Екзаменатор _____ доц. Ю. Е. Парфьонов

11. Рекомендована література

11.1. Основна

1. Блинов И. Н. Java. Промышленное программирование : практ. пособ. / И. Н. Блинов, В. С. Романчик. – Мн : УниверсалПресс, 2007. – 768 с.

2. Дейтел Х. М. Технологии программирования на Java 2: Книга 2. Распределенные приложения / Х. М. Дейтел, П. Дж. Дейтел. ; пер. с англ. – М. : ООО "Бином-Пресс", 2003. – 464 с. : ил.

3. Дейтел Х. М. Технологии программирования на Java 2: Книга 3. Корпоративные системы / Х. М. Дейтел, П. Дж. Дейтел, С. И. Самтри ; пер. с англ. – М. : ООО "Бином-Пресс", 2003. – 672 с. : ил.

4. Дэвид М. Гери, JavaServer Faces. Библиотека профессионала. JavaServer Faces. CORE / Дэвид М. Гери, Кей С. Хорстманн. – 3-е изд. – М. : Издательский дом "Вильямс", 2011. – 544 с.

5. Монсон Хейфел Р. Enterprise JavaBeans / Р. Хейфел Монсон ; пер. с англ. – 3-е изд. – СПб. : СимволПлюс, 2002. – 672 с. : ил.

6. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010. – 944 с. : ил.

7. Перри Б. Java сервлеты и JSP: сборник рецептов // Б. Перри ; пер с англ. – М. : Кудиц-пресс, 2006. – 768 с.

8. Таменбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. – СПб. : Питер, 2003. – 877 с. : ил.

9. Эккель Б. Философия Java. Библиотека программиста. / Б. Эккель. – 4-е изд. – СПб. : Питер, 2009. – 640 с. : ил.

10. Calvert K. L. TCP/IP Sockets in Java Practical Guide for Programmers / K. L. Calvert, M. J. Donahoo. – 2nd ed. – Burlington : Morgan Kaufmann, 2007. – 193 p.

11. Heffelfinger D. Java EE 6 with GlassFish 3 Application Server / D. Heffelfinger – Packt Publishing.

12. Linwood J. Beginning Hibernate, Second Edition / J. Linwood, D. Minter ; 2nd ed. – NY. : Apress, 2010. – 401 p.

13. Mak G. Spring Enterprise Recipes: A Problem-Solution Approach / G. Mak, J. Long; 2nd ed. – NY. : Springer, 2009. – 1104 p.

14. Mike Keith, ol. Pro JPA 2. Mastering the Java™ Persistence API / Mike Keith. Merrick Schincari. – New York : Apress, 2009. – 238 p.

11.2. Ресурси мережі Internet

15. Всесвітня щорічна конференція з Java технологій [Електронний ресурс]. – Режим доступу : <http://www.oracle.com/javaone/index.html>.

16. Документація Java™ Platform, Standard Edition 7 API Specification [Електронний ресурс]. – Режим доступу : <http://docs.oracle.com/javase/7/docs/api/>.

17. Документація Java™ Platform, Enterprise Edition 6 API Specification [Електронний ресурс]. – Режим доступу : <http://docs.oracle.com/javaee/6/api/>.

18. Документація Java™ 2 Platform Enterprise Edition, 5.0. API Specifications [Електронний ресурс]. – Режим доступу : <http://docs.oracle.com/javaee/5/api/>.

19. Офіційна документація JavaFX [Електронний ресурс]. – Режим доступу : <http://docs.oracle.com/javafx/>.

20. Статті всесвітніх експертів з Java-технологій [Електронний ресурс]. – Режим доступу : <http://www.javaworld.com>.

21. Статті експертів компанії IBM з Java-технологій [Електронний ресурс]. – Режим доступу : <http://www.ibm.com/developerworks/ru/java/>.
22. Універсальний фреймворк з відкритим вихідним кодом для Java-платформи. The Spring Framework [Електронний ресурс]. – Режим доступу : <http://www.springsource.org/>.
23. JSR 318: Enterprise JavaBeans™, Version 3.1. EJB Core Contracts and Requirements. Sun Microsystems. – November 5, 2009. – 626 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=318>
24. JSR 315: Java™ Servlet Specification, Version 3.0. Oracle. – February 6, 2011. – 230 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=315>.
25. JSR 245: JavaServer Pages™ Specification, Version 2.2. Maintenance Release 2. Sun Microsystems. – December 10, 2009. – 594 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=245>.
26. JSR 245: Expression Language Specification, Version 2.2. Maintenance Release. A component of the JavaServer™ Pages Specification. Version 2.2. – December 10, 2009. 594 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=245>.
27. JSR 314: JavaServer™ Faces Specification, Version 2.1. Maintenance Release 2. Oracle. – November 8, 2010. – 468 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=314>.
28. JSR 311: JAX-RS: Java™ API for RESTful Web Services, Version 1.1. Final Release. Sun Microsystems. – September 17, 2009. – 51 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=311>.
29. JSR 224: The Java API for XML-Based Web Services (JAX-WS) 2.2 Rev a. Maintenance Release. Oracle. – May 13, 2011. – 181 p. [Electronic resource]. – Access mode : <http://jcp.org/en/jsr/detail?id=224>.
30. Internet- інститут інформаційних технологій [Електронний ресурс]. – Режим доступу : – <http://www.intuit.ru>.
31. NetBeans IDE. Учебная карта по Java EE и Java Web. [Електронний ресурс]. – Режим доступу : http://netbeans.org/kb/trails/java-ee_ru.html.
32. ORM Hibernate [Електронний ресурс]. – Режим доступу : <http://www.hibernate.org/>.
33. The Java EE 6 Tutorial [Електронний ресурс] // Oracle Corporation – July 2011. – P. 906. – Режим доступу: <http://download.oracle.com/javasee/6/tutorial/doc/javaeetutorial6.pdf>

Зміст

Вступ.....	3
1. Кваліфікаційні вимоги до студентів.....	4
2. Тематичний план навчальної дисципліни.....	6
3. Зміст навчальної дисципліни за модулями та темами.....	8
4. Плани лекцій	11
5. Плани лабораторних занять.....	16
6. Самостійна робота студентів	17
6.1. Основні форми самостійної роботи студентів.....	17
6.2. Питання для самостійного опрацювання.....	18
6.3. Тематика доповідей.....	23
7. Контрольні запитання для самодіагностики	24
8. Індивідуально-консультативна робота	27
9. Методики активізації процесу навчання	28
10. Система поточного та підсумкового контролю знань студентів.....	30
11. Рекомендована література.....	35
11.1. Основна	35
11.2. Ресурси мережі Інтернет.....	36

НАВЧАЛЬНЕ ВИДАННЯ

**Робоча програма
навчальної дисципліни
"КРОСПЛАТФОРМОВІ ТА БАГАТОЗВЕННІ
ТЕХНОЛОГІЇ"**

**для студентів спеціальності 7.05010101
"Інформаційні управляючі системи та технології"
всіх форм навчання**

Укладачі: **Парфьонов** Юрій Едуардович
Поляков Андрій Олександрович

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Бутенко В. О.**

Коректор **Бриль В. О.**

План 2012 р. Поз. № 277.

Підп. до друку Формат 60×90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 2,5. Обл.-вид. арк. 3,13. Тираж прим. Зам. №

Видавець і виготівник – видавництво ХНЕУ, 61166, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*

**Робоча програма
навчальної дисципліни
"КРОСПЛАТФОРМОВІ
ТА БАГАТОЗВЕННІ ТЕХНОЛОГІЇ"
для студентів спеціальності
7.05010101 "Інформаційні управляючі
системи та технології"
всіх форм навчання**