

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

Технологія розробки програмного забезпечення - 2

Методичні вказівки до виконання лабораторних робіт

Для студентів напрямку 6.050103 “Програмна інженерія”
спеціальності 7.05010301 “Програмне забезпечення систем”
кафедри обчислювальної техніки денної форми навчання

Рекомендовано
Вченою радою факультету
інформатики та обчислювальної
техніки НТУУ «КПІ»
Протокол № ____ від __.__.2017 р.

Київ
2017

Зміст

Вступ	6
1. Мета та основні завдання роботи	7
2. Порядок виконання робіт	8
3. Оформлення звіту та порядок його подання	9
4. Заходи безпеки під час виконання комп'ютерного практикуму в комп'ютерному класі	10
5. Основні теоретичні відомості	13
5.1. Методи роботи із сервером застосувань	13
5.1.1. Встановлення серверу застосувань Glassfish	13
5.1.2. Запуск/зупинка серверу застосувань Glassfish	13
5.1.3. Запуск адміністративної веб-консолі Glassfish	14
5.1.4. Налаштування з'єднання серверу застосувань із базою даних	14
5.1.5. Налаштування реалму безпеки для серверу застосувань	20
5.2. Налаштування інтегрованого середовища розробки	23
5.2.1. Встановлення Netbeans IDE	23
5.2.2. Підключення серверу застосувань Glassfish до Netbeans IDE ...	23
5.2.3. Запуск/зупинка серверу застосувань із Netbeans IDE	26
5.2.4. Розробка веб-застосувань із використанням Netbeans IDE	27
5.2.5. Запуск веб-застосування із Netbeans IDE	27
5.2.6. Запуск веб-застосування із Netbeans у режимі відладки	27
6. Комп'ютерний практикум	29
6.1. Комп'ютерний практикум №1	29
6.1.1. Тема	29
6.1.2. Теоретичні відомості	29
6.1.3. Вимоги до виконання роботи	30
6.1.4. Завдання	30
6.1.5. Контрольні питання	31

6.2. Комп'ютерний практикум №2	33
6.2.1. Тема	33
6.2.2. Теоретичні відомості	33
6.2.3. Вимоги до виконання роботи	33
6.2.4. Завдання	33
6.2.5. Контрольні питання	34
6.3. Комп'ютерний практикум №3	35
6.3.1. Тема	35
6.3.2. Теоретичні відомості	35
6.3.3. Вимоги до виконання роботи	35
6.3.4. Завдання	35
6.3.5. Контрольні питання	36
6.4. Комп'ютерний практикум №4	37
6.4.1. Тема	37
6.4.2. Теоретичні відомості	37
6.4.3. Вимоги до виконання роботи	38
6.4.4. Завдання	38
6.4.5. Контрольні питання	39
6.5. Комп'ютерний практикум №5	40
6.5.1. Тема	40
6.5.2. Теоретичні відомості	40
6.5.3. Вимоги до виконання роботи	40
6.5.4. Завдання	40
6.5.5. Контрольні питання	41
6.6. Комп'ютерний практикум №6	42
6.6.1. Тема	42
6.6.2. Теоретичні відомості	42
6.6.3. Вимоги до виконання роботи	42

6.6.4. Завдання	43
6.6.5. Контрольні питання	43
6.7. Комп'ютерний практикум №7	44
6.7.1. Тема.....	44
6.7.2. Теоретичні відомості	44
6.7.3. Вимоги до виконання роботи	44
6.7.4. Завдання	44
6.7.5. Контрольні питання	44
Література	45
Додаток 1 Шаблон звіту	47

ВСТУП

Комп'ютерний практикум виконується студентами 4 курсу напряму 6.050103 “Програмна інженерія” спеціальності 7.05010301 “Програмне забезпечення систем” денної форми навчання.

Основна мета виконання комп'ютерного практикуму – це закріплення знань, які отримані на лекційних заняттях та в результаті самостійної роботи студента. Лабораторні роботи призначені для отримання студентом досвіду із застосування технологій Java Enterprise Edition (JEE).

1. МЕТА ТА ОСНОВНІ ЗАВДАННЯ РОБОТИ

Розробка платформо-незалежних програм є актуальною практичною задачею і здобуття відповідних умінь і досвіду є важливим етапом підготовки майбутніх фахівців в галузі розробки комп'ютеризованих систем.

Розробка застосувань корпоративного типу, які характеризуються великою кількістю користувачів та високим навантаженням, є актуальним напрямом у галузі розробки комп'ютеризованих систем.

Мета комп'ютерного практикуму – допомогти студентам навчитися розробляти платформо-незалежні корпоративні веб-орієнтовані застосувань на базі технологій платформи Java EE.

Основним завданням роботи є засвоєння стеку технологій Java EE, а саме:

1. Засвоїти технологію JDBC по роботі з базами даних;
2. Засвоїти базові технології розробки веб-орієнтованих застосувань – технології сервлетів та JSP.
3. Засвоєння технології JSF для розробки веб-орієнтованих застосувань.
4. Засвоїти технологію JPA для роботи з базами даних на основі об'єктно-реляційного відображення.
5. Засвоїти технологію EJB для розробки бізнес-логіки застосування.
6. Засвоїти методи забезпечення аутентифікації та авторизації для платформи Java EE.
7. Засвоїти технології веб-сервісів JAX-WS та JAX-RS.

2. ПОРЯДОК ВИКОНАННЯ РОБІТ

При виконанні комп'ютерного практикуму необхідно дотримуватись наступних вимог:

1. Необхідно застосовувати угоди щодо кодування для мови програмування Java (Java Code Convention) [1], які стосуються іменування та програмних практик .
2. Необхідно застосовувати коментарі щодо класів програми, методів класів та окремих частин програми.
3. Для кожної роботи має бути підготовлений та зданий звіт. Вимоги до звіту надано в розділі 3 даних методичних вказівок.

При виконанні комп'ютерного практикуму в лабораторії чи комп'ютерному класі слід дотримуватись правил техніки безпеки.

3. ОФОРМЛЕННЯ ЗВІТУ ТА ПОРЯДОК ЙОГО ПОДАННЯ

Для кожного комп'ютерного практикуму має бути підготовлений звіт. Без звіту студент не допускається до здачі.

Звіт має містити:

- 1) формулювання теми роботи;
- 2) постановку завдання, яке необхідно виконати;
- 3) текст програми;
- 4) висновки щодо роботи програми.

Шаблон звіту наведено у додатку 1.

Текст звіту виконується у відповідності до вимог стандартів машинним (на пристроях ЕОМ) способом. Формат аркуша - **A4**. Параметри сторінки (поля): верхнє – **25** мм, нижнє – **15** мм, ліве – **25** мм, праве – **10** мм. Основний текст набирається шрифтом **Times New Roman** розмір **12**, міжрядковий інтервал - **1,0**.

У висновках необхідно зазначити, чи вирішує програма поставлене завдання, вказати труднощі, з якими студент стикнувся під час виконання завдання.

4. ЗАХОДИ БЕЗПЕКИ ПІД ЧАС ВИКОНАННЯ КОМП'ЮТЕРНОГО ПРАКТИКУМУ В КОМП'ЮТЕРНОМУ КЛАСІ

До виконання лабораторних робіт допускаються студенти, які пройшли інструктаж з техніки безпеки з наступним записом про це в спеціальному журналі.

Загальні положення:

- До виконання лабораторних робіт допускаються студенти, які пройшли інструктаж з техніки безпеки і правил поведінки в комп'ютерному класі.
- Робота студентів в комп'ютерному класі дозволяється лише у присутності викладача.

Перед початком роботи необхідно:

- Переконалися у відсутності видимих пошкоджень на робочому місці;
- Розмістити на столі зошити, навчальні посібники так, щоб вони не заважали роботі на комп'ютері;

При роботі в комп'ютерному класі категорично забороняється:

- Знаходитися в класі у верхньому одязі;
- Класти одяг, сумки та інші речі на клавіатуру, монітор і системний блок;
- Знаходитися в класі з напоями та їжею;
- Розташовуватися збоку або ззаду від включеного монітора;
- Приєднувати або від'єднувати кабелі, торкатися роз'ємів, оголених дротів і розеток;

- Пересувати комп'ютери і монітори;
- Відкривати системний блок;
- Включати і вимикати комп'ютери самостійно;
- Намагатися самостійно усувати несправності в роботі обладнання;
- Перекривати вентиляційні отвори на системному блоці та моніторі;
- Видаляти і переміщати чужі файли;
- Приносити і запускати комп'ютерні ігри.

Знаходячись в комп'ютерному класі, учні зобов'язані:

- Дотримуватись тиші і порядку;
- Виконувати вимоги викладача;
- Знаходячись в мережі працювати лише під своїм ім'ям і паролем;
- Дотримуватись режиму роботи (згідно Санітарних правил і норм);
- При появі різі в очах, різкому погіршенні видимості, неможливості сфокусувати погляд або навести його на різкість, появи болю в пальцях і кистях рук, посилення серцебиття негайно покинути робоче місце, повідомити про це, викладача і звернутися до лікаря;
- Після закінчення роботи завершити всі активні програми і коректно вимкнути комп'ютер;
- Залишити робоче місце чистим.

Вимоги безпеки в аварійних ситуаціях:

- При появі програмних помилок або збоях обладнання студент повинен негайно звернутися до викладача.
- При появі запаху гару, незвичайного звуку негайно припинити роботу, і повідомити викладача.

.

5. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

5.1. Методи роботи із сервером застосувань

Для виконання комп'ютерного практикуму необхідно встановити сервер застосувань, який підтримує специфікацію Java EE 6 або вище.

Рекомендується використовувати сервер застосувань Glassfish v3. Але на вибір студента може бути використаний будь-який інший сервер застосувань, який підтримує Java EE 6. В даних методичних вказівках дані інструкції щодо Glassfish v3.

5.1.1. Встановлення серверу застосувань Glassfish

Встановити сервер застосувань Glassfish v3. Встановлення серверу здійснюється згідно інструкцій в інсталяційному пакеті продукту.

Інсталяція продукту може бути отримана наступним чином:

- 1) Окрема інсталяція (<http://glassfish.java.net/downloads/3.1.2-final.html>);
- 2) Інсталяція в пакеті з Netbeans IDE (<http://netbeans.org/downloads/index.html>, _bundled server GlassFish Server Open Source Edition 3.1.2).

5.1.2. Запуск/зупинка серверу застосувань Glassfish

Для того, щоб запустити та зупинити Glassfish v3 може бути використана утиліта asadmin, яка працює із командного рядка.

Запуск сервера застосувань:

```
PATH_TO_GLASSFISH/glassfish/bin/asadmin start-domain domain1
```

Зупинка серверу застосувань:

```
PATH_TO_GLASSFISH/glassfish/bin/asadmin stop-domain domain1
```

Запуск та зупинка серверу застосувань також можуть бути здійснені із інтегрованого середовища розробки (див. розділ 5.2.3).

5.1.3. Запуск адміністративної веб-консолі Glassfish

Адміністративна веб-консоль дозволяє здійснювати адміністрування серверу застосувань Glassfish із веб-інтерфейсу.

Для запуску адміністративної веб-консолі Glassfish необхідно відкрити браузер та відкрити адресу <http://localhost:4848>

5.1.4. Налаштування з'єднання серверу застосувань із базою даних

Для налаштування JDBC з'єднання серверу застосувань Glassfish із БД необхідно виконати наступні дії:

- 1) підключити до серверу застосувань необхідний JDBC драйвер;
- 2) створити пул з'єднань JDBC (JDBC connection pool);
- 3) створити джерело з'єднань JDBC (JDBC datasource).

Для підключення JDBC-драйвера необхідно:

- 1) знайти JDBC драйвер для бази даних, що використовується
 - a. Для БД MySQL: mysql-connector-java-5.1.18-bin.jar
 - b. Для БД Oracle: ojdbc14.jar
 - c. Якщо використовується інша база даних, то відповідний JDBC драйвер.
- 2) скопіювати JDBC-драйвер для обраної БД у папку
PATH_TO_GLASSFISH\glassfish\domains\domain1\lib\
- 3) перепустити сервер застосувань (зупинити та запустити – див. розділ 5.1.2).

Для створення пулу з'єднань JDBC необхідно:

- 1) відкрити адміністративну веб-консоль Glassfish
(<http://localhost:4848>);

2) вибрати в лівому боковому пункті «Resources | JDBC | Connection Pools» (рис. 5.1);

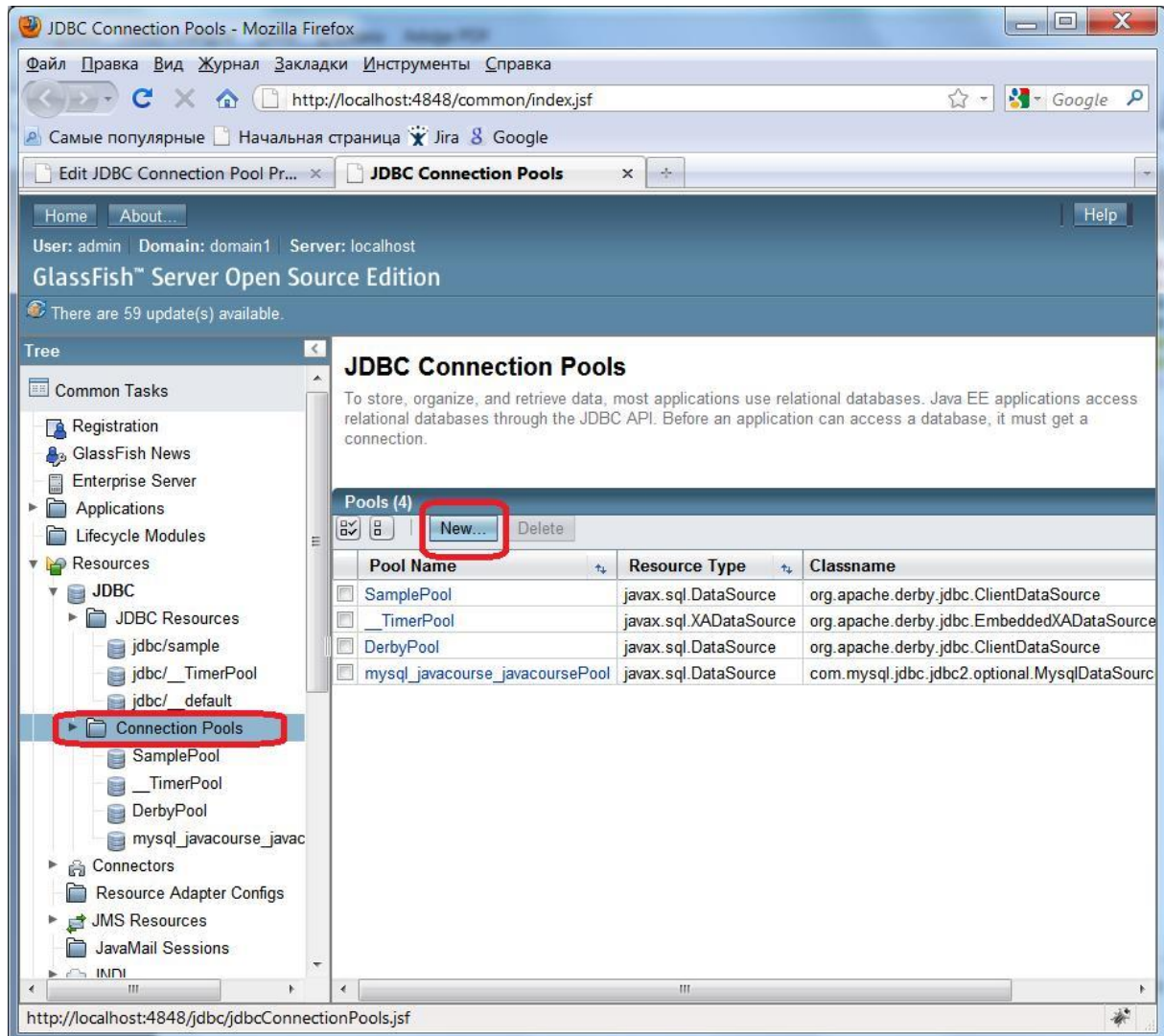


Рис. 5.1

3) вказати назву пулу ресурсів, його тип та тип бази даних (рис. 5.2);

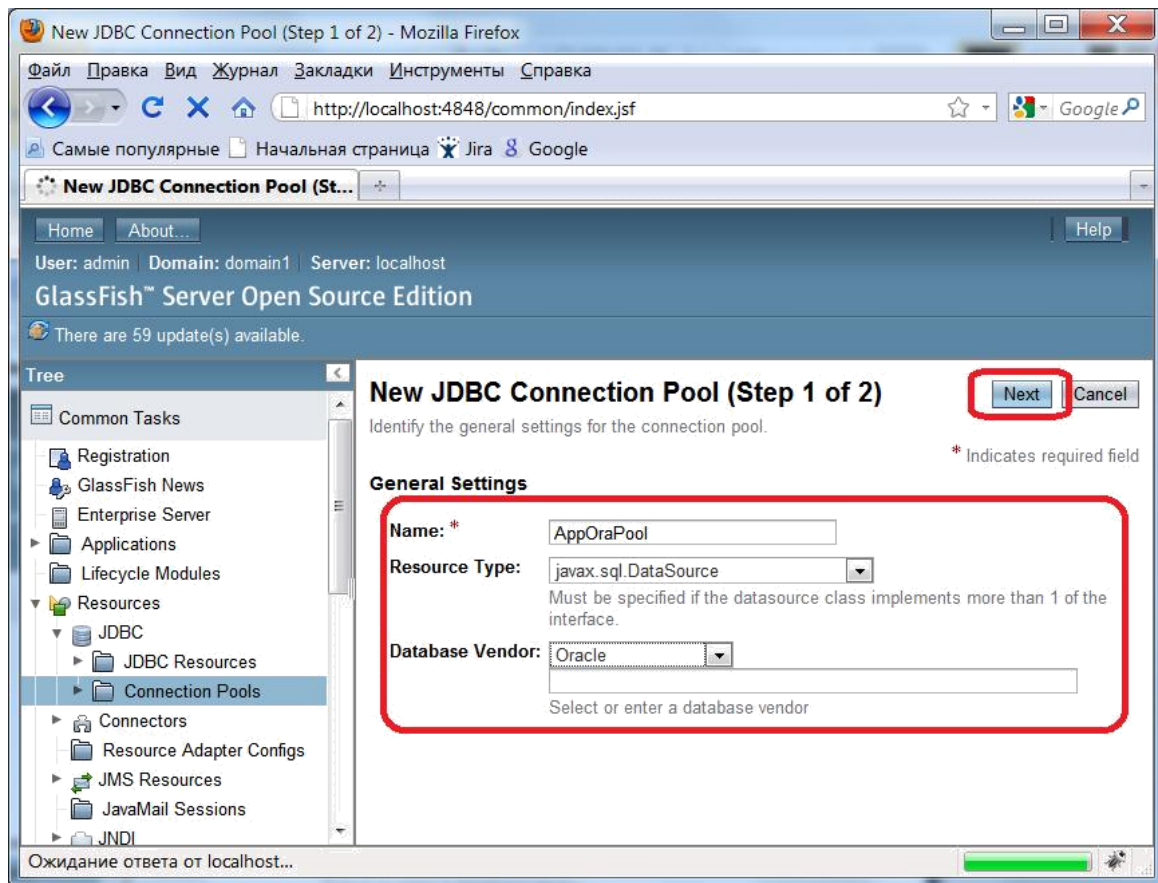


Рис. 5.2

4) вказати реквізити підключення (рис. 5.3):

a. URL.

i. Для Oracle рядок URL має наступний вигляд:
jdbc:oracle:thin:@//host:port/database

ii. Для MySQL рядок URL має наступний вигляд:
jdbc:mysql://host:port/database

b. User

c. Password

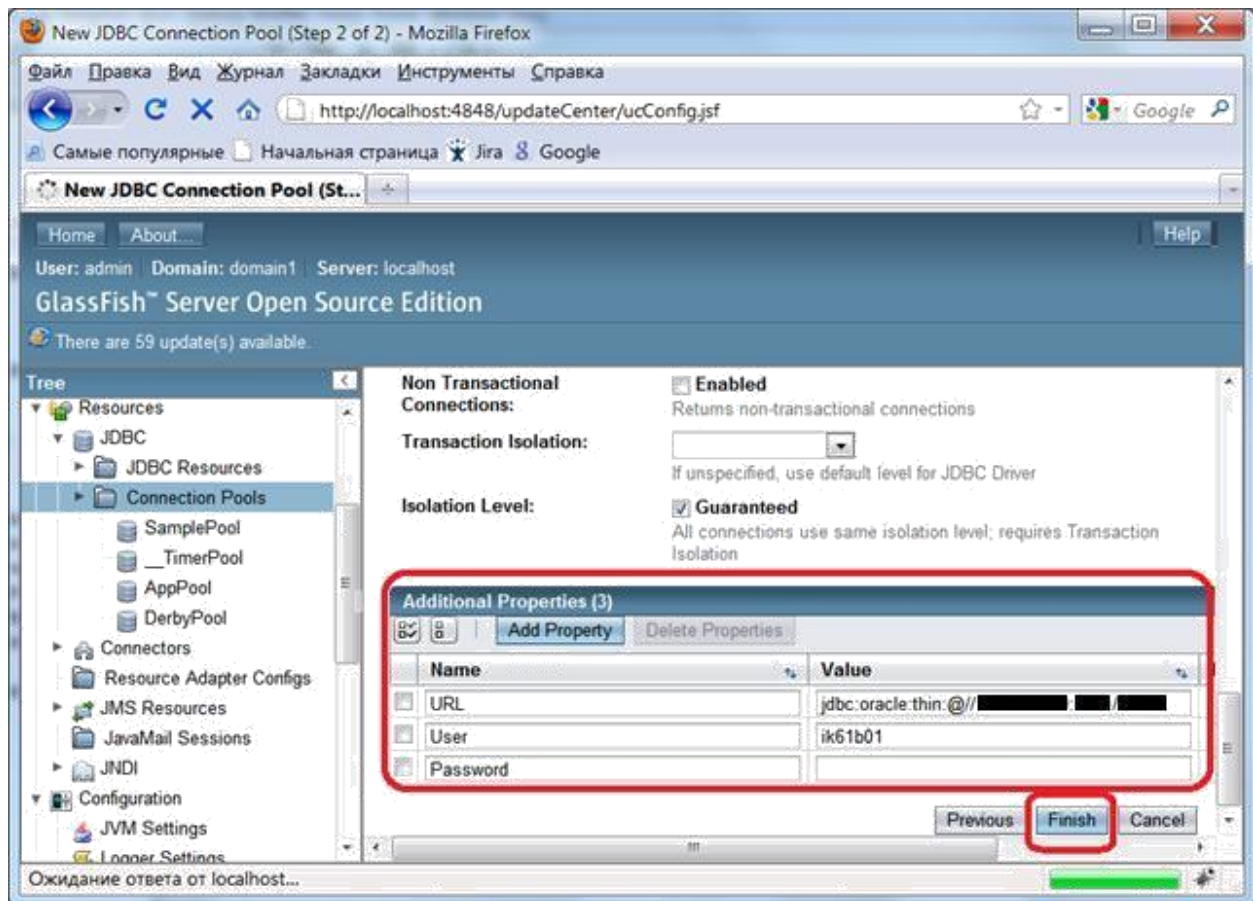


Рис. 5.3

5) Перевірити підключення. Для цього для вибраного пулу підключень натиснути кнопку Ping (рис. 5.4):

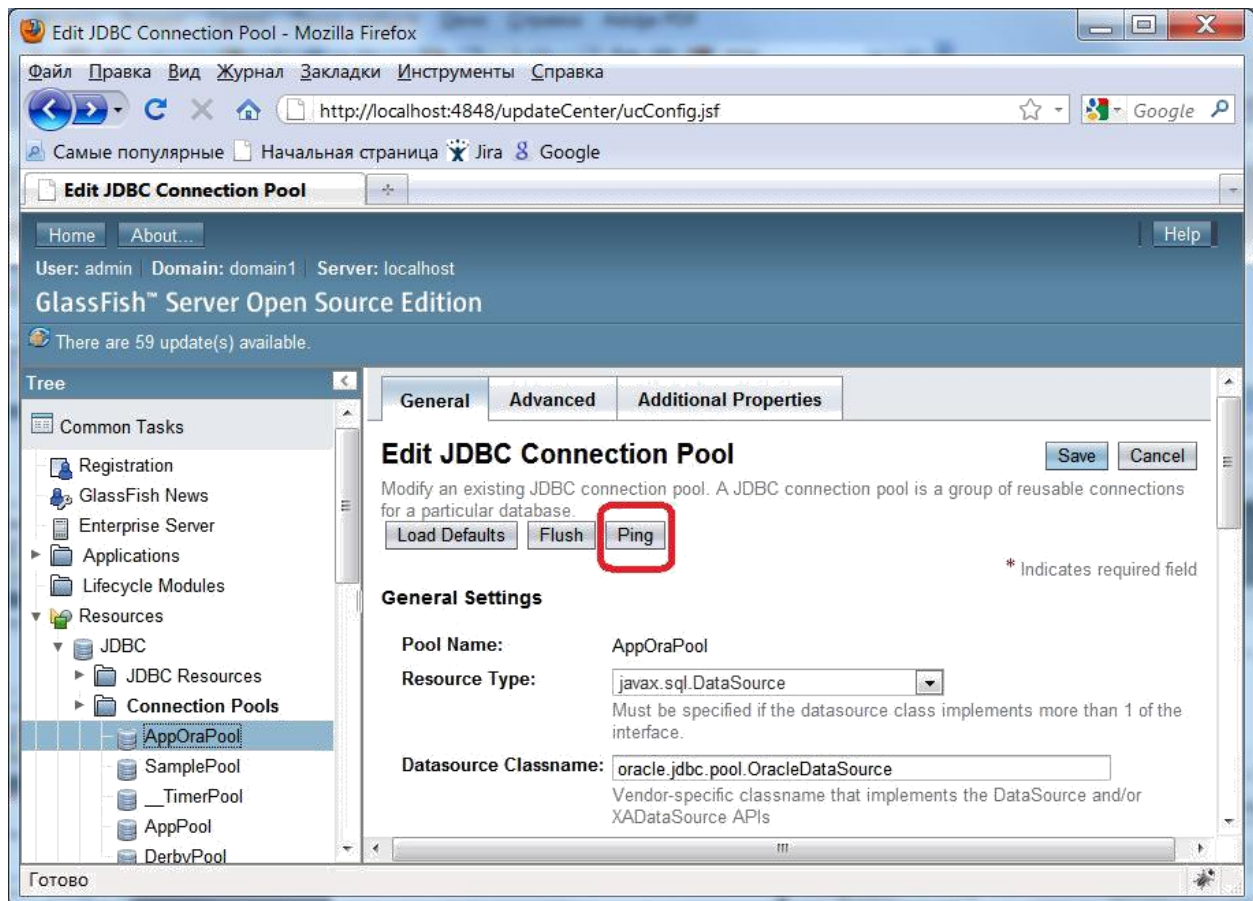


Рис. 5.4

Для створення джерела з'єднань необхідно:

- 1) відкрити адміністративну веб-консоль Glassfish (<http://localhost:4848>);
- 2) вибрати в лівому боковикі адміністративної веб-консолі пункт “Resources | JDBC | JDBC Resources” (рис. 5.5);

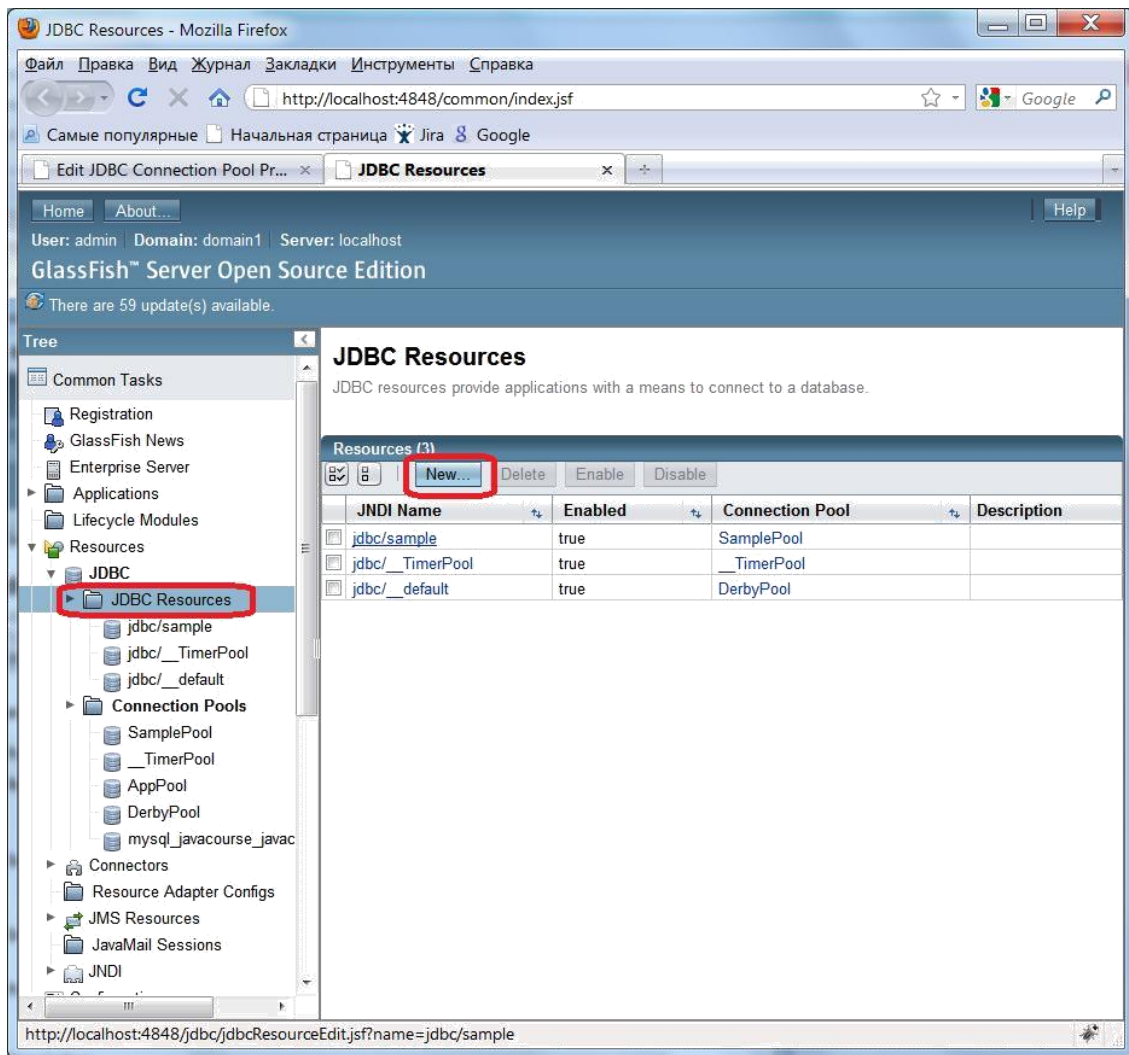


Рис. 5.5

- 3) вказати наступні параметри: JNDI-name – назва для джерела даних, яку можна буде використовувати при розробці; назву пулу з'єднань (рис. 5.6).

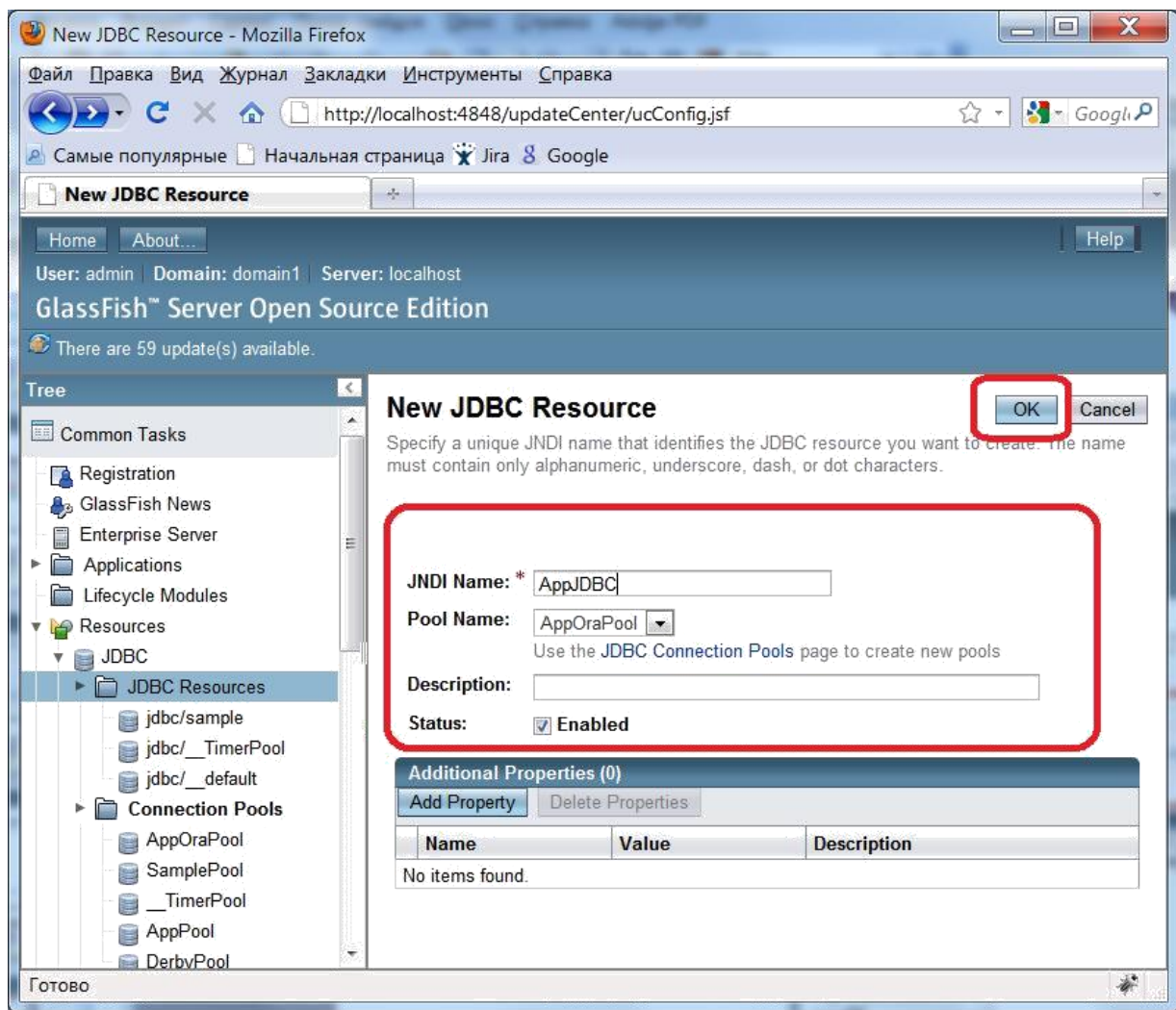


Рис. 5.6

5.1.5. Настройка реалму безпеки для серверу застосувань

Для настройки реалу безпеки необхідно відкрити адміністративну веб-консоль серверу застосувань Glassfish <http://localhost:4848> та виконати наступні дії:

- 1) у лівому боковому меню необхідно вибрати пункт меню «Security | Realms | New Realm» (рис. 5.7);

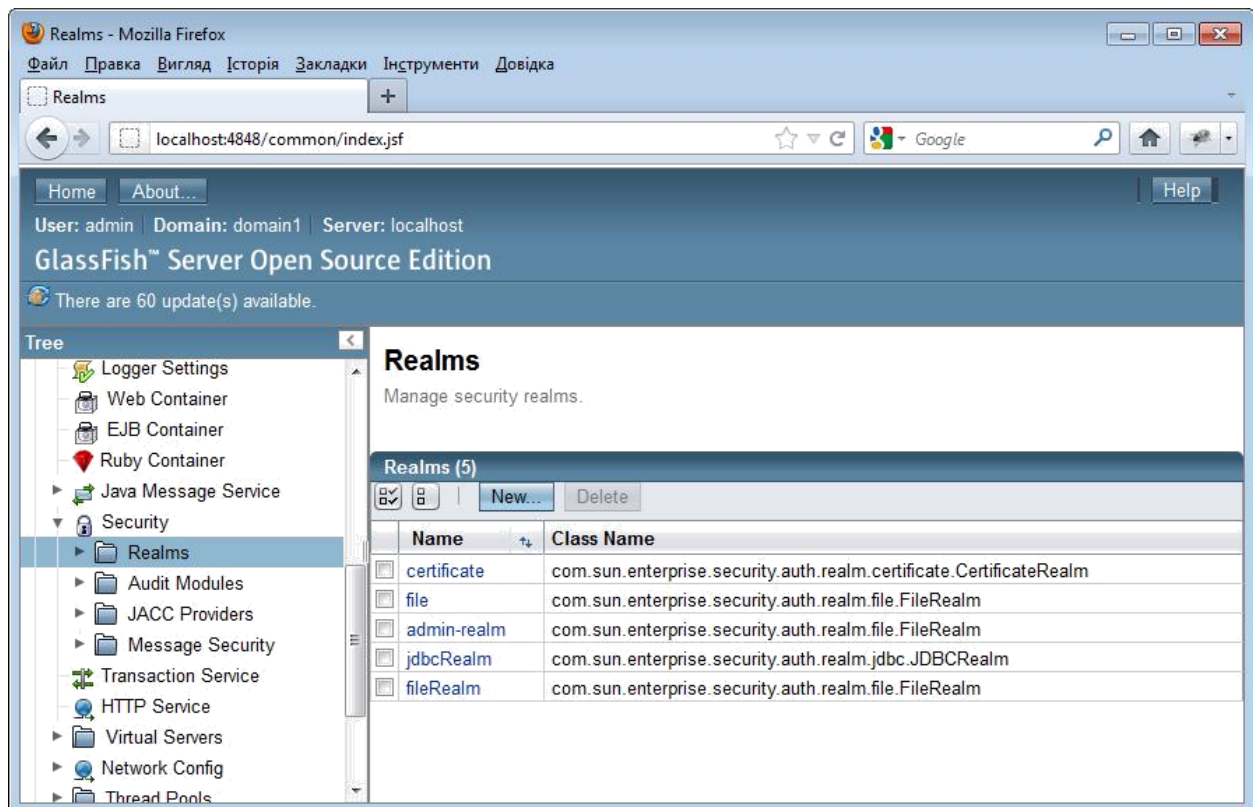


Рис. 5.7

2) вказати властивості реалму:

- a. JNDI = JNDI-name джерела даних до БД, де зберігаються дані про користувачів. Для настройки джерела даних див. розділ 5.1.4
- b. Jaas Context = jdbcRealm
- c. User Table = назва таблиці, де зберігаються логіни та паролі користувачів
- d. User Name Column = назва колонки, де зберігається логін користувача
- e. Password Column = назва колонки, де зберігається пароль користувача
- f. Group Table = назва таблиці, де зберігаються групи користувачів

g. Group Name Column = назва колонки, де вказана група користувача

h. Digest Algorithm = алгоритм хешування паролю (якщо не використовується жодний, то необхідно вказати NONE)

Далі наведено приклад заповнення полів реалму.

Нехай у таблиці із назвою USERTABLE зберігаються дані про користувачів, у таблиці GROUPTABLE зберігаються дані про групи користувачів.

Таблиця **USERTABLE**

USERID	PASSWORD
Ivanov	****

Таблиця **GROUPTABLE**

USERID	GROUPID
Ivanov	Student

Тоді настройки реалму мають бути наступними:

User Table	USERTABLE
User Name Column	USERID
Password Column	PASSWORD
Group Table	GROUPTABLE
Group Name Column	GROUPID
Digest Algorithm	NONE

5.2. Налаштування інтегрованого середовища розробки

При виконанні лабораторних робіт для рекомендується використовувати як інтегроване середовище розробки NetBeans IDE версії 6.9 або вище, редакція Java EE.

Але на вибір студента може бути використане будь-яке інше інтегроване середовище розробки (Eclipse, Idea).

В даних методичних вказівках дані інструкції для Netbeans IDE. Додаткові теоретичні відомості щодо використання Netbeans IDE для розробки Java EE застосувань вказані в [2].

5.2.1. Встановлення Netbeans IDE

Встановлення Netbeans IDE відбувається згідно інструкцій і інсталяційному пакеті продукту.

Інсталяція продукту може бути отримана за адресою <http://netbeans.org/downloads/index.html>.

Необхідно вибрати для завантаження одну з наступних редакцій:

- Java EE;
- У пакеті із GlassFish Server Open Source Edition 3.1.2.

5.2.2. Підключення серверу застосувань Glassfish до Netbeans IDE

Якщо інсталяція Netbeans IDE та Glassfish виконувалась із об'єднаного пакету, то подальші дії, що описані в цьому розділі, виконувати не потрібно.

Після встановлення Netbeans IDE необхідно підключити до Netbeans IDE сервер застосувань. Для цього необхідно виконати наступні дії:

1. Вибрати пункт меню Tools/Servers.
2. Натиснути кнопку “Add Server”

- а. Вказати тип серверу застосувань - Glassfish Server 3 (рис. 5.8)

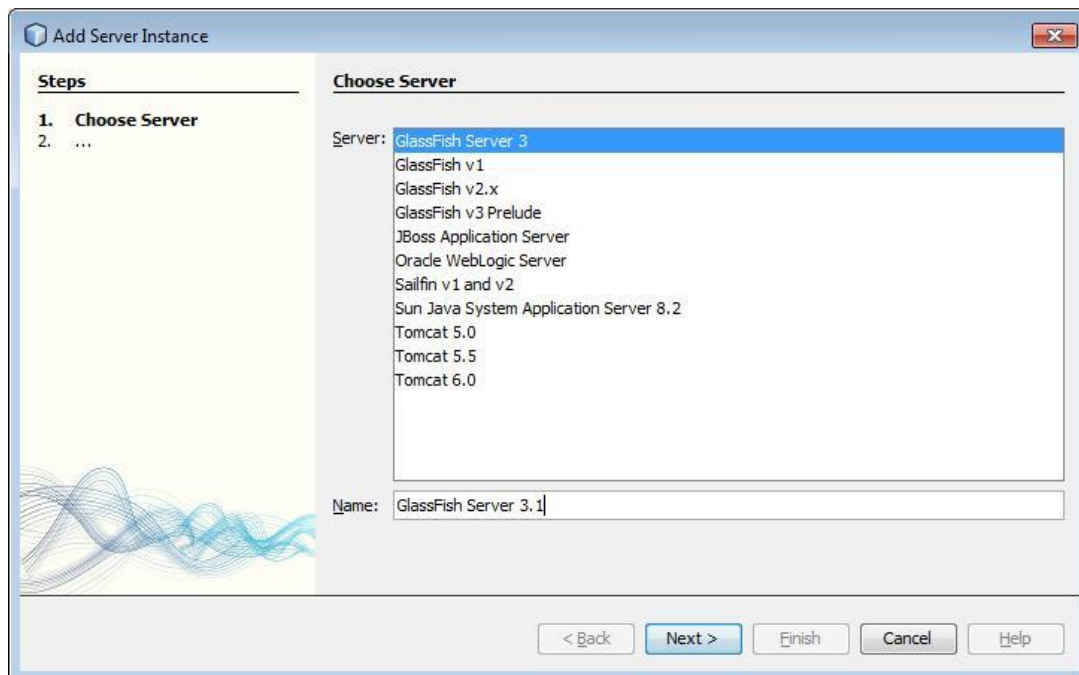


Рис. 5.8

- б. Вибрати директорію на диску, куди був встановлений сервер застосувань (рис. 5.9):

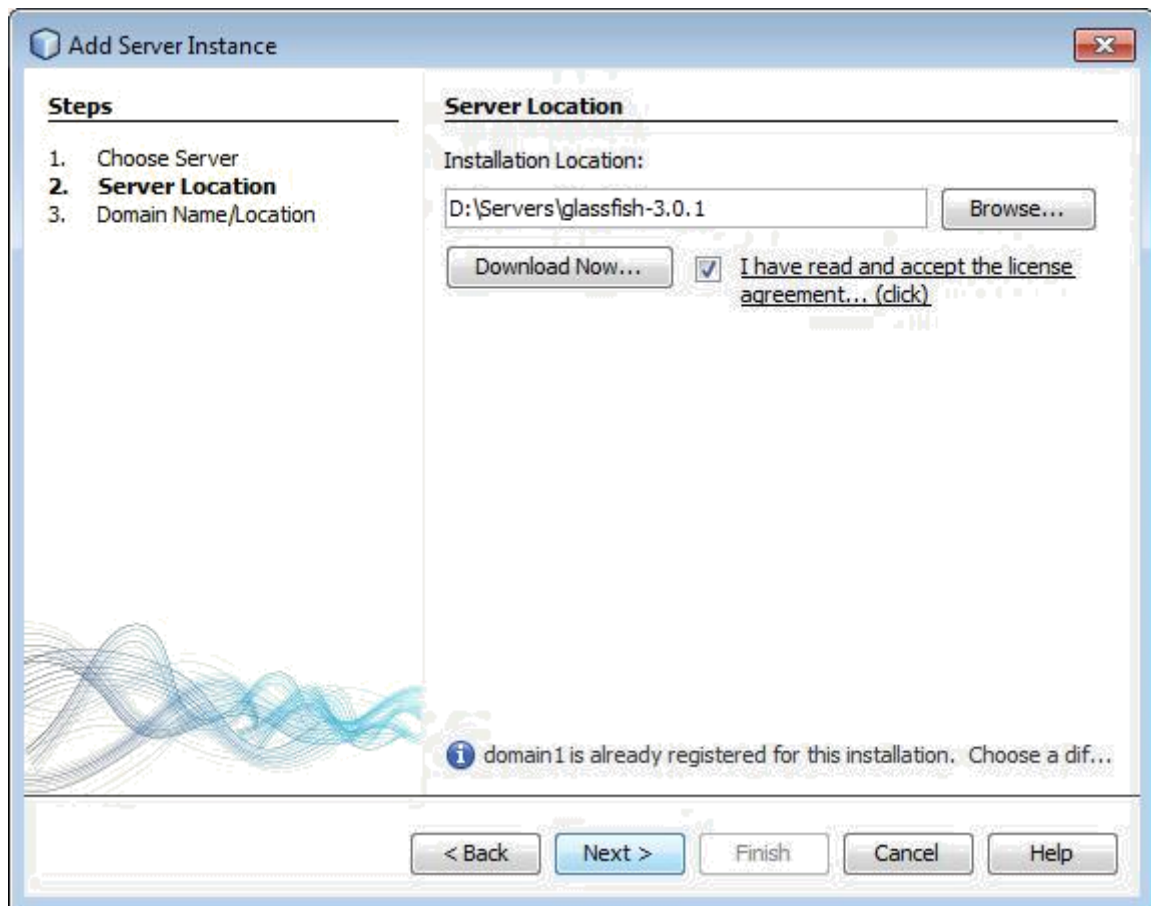


Рис. 5.9

с. Зареєструвати домен серверу застосувань (рис. 5.10):

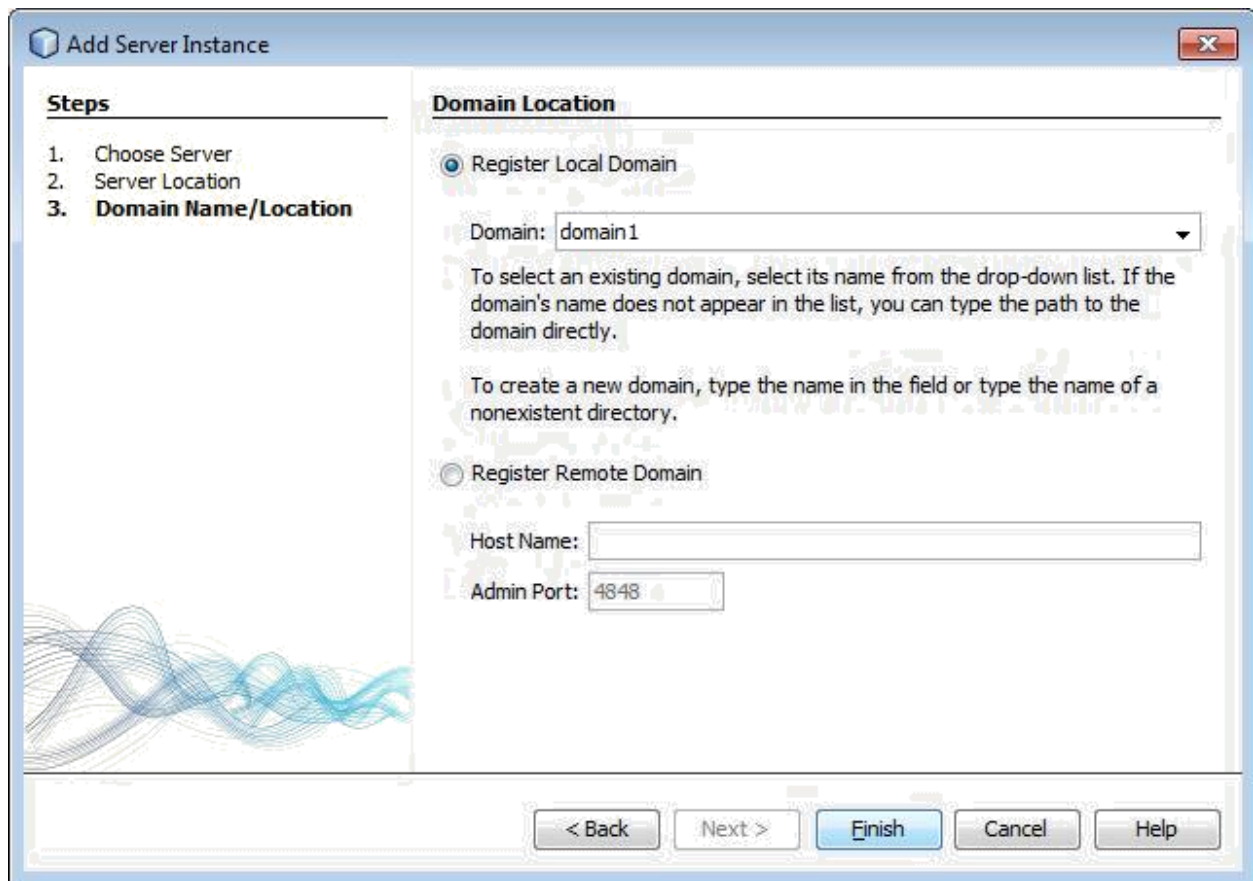


Рис. 5.10

d. Натиснути кнопку Finish (рис. 5.10).

5.2.3. Запуск/зупинка серверу застосувань із Netbeans IDE

Для того, щоб виконати старт/зупинку серверу застосувань необхідно:

- 1) Відкрити закладку Services (пункт меню Window/Services)
- 2) Вибрати елемент Servers, розкрити його вміст (рис. 5.11), вибрати сервер застосувань, натиснути на праву кнопку миші та вибрати необхідну дію (Start або Stop).

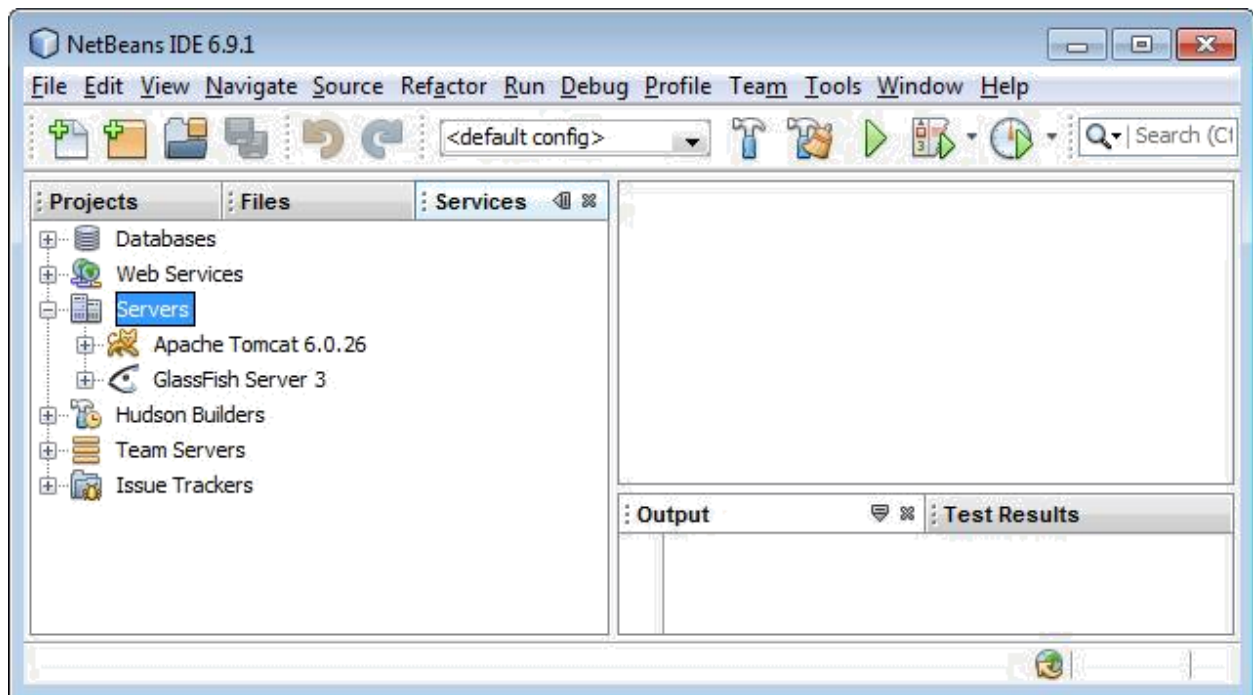


Рис. 5.11

5.2.4. Розробка веб-застосувань із використанням Netbeans IDE

Для створення веб-застосування необхідно за допомогою Netbeans IDE необхідно виконати наступні кроки:

- 1) створити проект: File | New Project | Java Web | Web Application;
- 2) вказати сервер застосувань, в який буде відбуватися завантаження застосування.

5.2.5. Запуск веб-застосування із Netbeans IDE

Відкрити закладку “Projects” (за допомогою пункту меню Window/Projects), вибрати необхідне веб-застосування, натиснути праву кнопку миші та виконати команду Run. Якщо сервер застосувань не був запущений, то після виконання цієї команди із запуску застосування, він буде запущений автоматично.

5.2.6. Запуск веб-застосування із Netbeans у режимі відладки

Відкрити закладку “Projects” (за допомогою пункту меню Window/Projects), вибрати необхідне веб-застосування, натиснути праву

кнопку миші та виконати команду Debug. Якщо сервер застосувань не був запущений, то після виконання цієї команди із запуску застосування, він буде запущений автоматично у режимі відладки.

Далі необхідно встановлювати в коді застосування точки зупинки (breakpoint). Netbeans IDE автоматично призупинить виконання, коли потік виконання дійде до точки breakpoint.

6. ЛАБОРАТОРНІ РОБОТИ

6.1. Лабораторна робота №1

6.1.1. Тема

Засвоєння технології JDBC (Java Database Connectivity) для роботи із базою даних.

6.1.2. Теоретичні відомості

Теоретичні відомості щодо технології JDBC надано в дидактичних матеріалах до дисципліни [3] (лекція «Технологія JDBC») та [4] (глава 4).

Для виконання лабораторної роботи необхідно виконати наступні підготовчі дії:

- 1) Створити Java-застосування (Standalone Java Application).
- 2) Підключити JDBC драйвер до застосування. Для цього у Netbeans IDE, відкрити вікно Project Properties, вибрати в списку Libraries, натиснути на кнопку «Add Jar/Folder» та вказати місцезнаходження JDBC-драйверу.
 - a. Для БД MySQL: mysql-connector-java-5.1.18-bin.jar
 - b. Для БД Oracle: ojdbc14.jar
 - c. Якщо використовується інша база даних, то відповідний JDBC драйвер.
- 3) Підготувати реквізити для з'єднання з БД:
 - a. URL.
 - i. Для Oracle рядок URL має наступний вигляд:
jdbc:oracle:thin:@//host:port/database

- ii. Для MySQL рядок URL має наступний вигляд:
jdbc:mysql://host:port/database
- b. User
- c. Password

Якщо робота виконується в лабораторії чи комп'ютерному класі, то реквізити з'єднання мають бути отримані у викладача.

6.1.3. Вимоги до виконання роботи

Доведення робото-спроможності програми буде відбуватися **на основі модульних тестів**. У складі програми повинні бути реалізовані модульні тести для перевірки коректності роботи програми:

- Перевірити додавання інформації (перевірка відбувається наступним чином: викликати функцію додавання інформації і переконатися шляхом виклику функції отримання інформації, що функція додавання працює вірно).
- Перевірити редагування інформації (перевірка відбувається наступним чином: вичитати об'єкт із БД шляхом виклику функції отримання інформації, змінити його поля та зберегти в БД за допомогою реалізованої функції редагування. Повторно вичитати та пересвідчитись, що дані змінено).

Для задачі роботи необхідно підготувати звіт. Вимоги до звіту надані в розділі 3.

6.1.4. Завдання

Вибрати завдання згідно варіанту.

Варіант 1. «Парк автомобілів». В БД зберігається інформація про автомобілі (марка, номер, тощо).

Варіант 2. «Вакансія». В БД зберігається про вакансію (компанія, посада, вимоги тощо).

Варіант 3. «Фільмотека». В БД зберігається інформація про фільми (назва, режисер, тривалість, рік випуску, короткий опис тощо).

Власний варіант. В БД зберігається інформація про об'єкти, вибрані для власного варіанту виконання лабораторної роботи.

Для вибраного варіанту необхідно реалізувати наступне:

- Створити БД та створити таблиці.
- Реалізувати отримання інформації із БД на базі технології JDBC.
 - Функція приймає на вхід критерії пошуку та повертає список об'єктів предметної області (наприклад, для варіанту «Парк автомобілів» функція повертає список автомобілів).
- Реалізувати додавання інформації до БД на базі технології JDBC.
 - Функція приймає на вхід об'єкт предметної області (наприклад, для варіанту «Парк автомобілів» функція приймає на вхід об'єкт «Автомобіль»).
- Реалізувати редагування інформації в БД на базі технології JDBC.
 - Функція приймає на вхід об'єкт предметної області (наприклад, для варіанту «Парк автомобілів» функція приймає на вхід об'єкт «Автомобіль»).
- Реалізувати видалення інформації в БД на базі технології JDBC.
- Організацію з'єднання з базою даних винести в окремий клас, метод якого повертає з'єднання.

6.1.5. Контрольні питання

1. Назвіть призначення технології JDBC.
2. Назвіть основні класи технології JDBC та їх призначення.
3. Опишіть алгоритм отримання з'єднання із БД.

4. Вкажіть різницю в призначення класів Statement та PreparedStatement.
5. Вкажіть, як необхідно передавати параметри в запити.

6.2. Лабораторна робота №2

6.2.1. Тема

Засвоєння технологій сервлетів та JSP (Java Server Pages) для розробки веб-застосунків.

6.2.2. Теоретичні відомості

Теоретичні відомості щодо застосування технологій сервлетів та JSP надані в дидактичних матеріалах до дисципліни [3] (лекції «Технології сервлетів» та «Технологія JSP»), [5] (глава 1).

Теоретичні відомості щодо створення веб-застосунка у середовищі Netbeans IDE наведені в розділі 5.2.4.

6.2.3. Вимоги до виконання роботи

Для задачі лабораторної роботи необхідно підготувати звіт.
Вимоги до звіту надані в розділі 3.

6.2.4. Завдання

Завдання базується на базі варіанту, вибраного для лабораторної роботи №1.

На базі обраного варіанту та на базі виконаної лабораторної роботи №1 виконати наступне:

1. Створити веб-застосунок.
2. Розробити веб-інтерфейс для виводу на веб-сторінку об'єктів, що зберігаються в БД (для варіанту «Парк автомобілів» - це автомобілі, для варіанту «Фільмотека» - це фільми тощо).
 - а. Для цього може застосовуватись технологія сервлетів або JSP (на вибір).
3. Розробити веб-інтерфейс для створення об'єкту:

- a. розробити JSP-сторінку для вводу даних;
- b. розробити сервлет, який буде здійснювати отримання даних із сторінки вводу даних та викликати функцію створення об'єкту в БД, яка була розроблена в рамках лабораторної роботи 1.

6.2.5. Контрольні питання

1. Вкажіть призначення технології сервлетів.
2. Опишіть відмінність технології сервлетів від CGI.
3. Опишіть призначення технології JSP.
4. Вкажіть, яким чином досягти зниження кількості скриптлетів на JSP-сторінках.

6.3. Лабораторна робота №3

6.3.1. Тема

Засвоєння технології JPA (Java Persistence API) для роботи з базою даних на основі об'єктно-реляційного відображення.

6.3.2. Теоретичні відомості

Теоретичні відомості щодо застосування технології JPA наведено в дидактичних матеріалах до дисципліни [3] (лекції «Технологія JPA (1)» та «Технологія JPA (2)») а також у [6- 7].

Теоретичні відомості щодо настройки серверу застосувань із базою даних вказані в розділі 5.1.4.

6.3.3. Вимоги до виконання роботи

Для здачі лабораторної роботи необхідно підготувати звіт із наступними даними:

- ER-діаграма моделі даних
- Роздруківка діаграми фізичної моделі даних
- SQL скрипти із створення об'єктів БД
- Тексти програми:
 - класи сутностей, що зберігаються
 - фрагменти коду, в якому застосовується JPA EntityManager.

Інші загальні вимоги до звіту надані в розділі 3.

6.3.4. Завдання

Завдання виконується на базі комп'ютерного лабораторної №2.

1) Налаштувати з'єднання серверу застосувань із базою даних.

- 2) У веб-застосуванні, яке було розроблене при виконанні роботи №2, замінити функції для роботи із БД, розроблені на базі технології JDBC, на такі, що розроблені на основі технології JPA та використовують EntityManager.

6.3.5. Контрольні питання

1. Вкажіть призначення технології JPA.
2. Назвіть, як в технології JPA реалізуються основні стратегії із відображення наслідування в реляційних таблицях.
3. Опишіть основні функції менеджера сутностей EntityManager.
4. Опишіть призначення контексту збереження об'єктів. Вкажіть основні стани об'єктів в цьому контекстів.
5. Основні методи формування запитів в технології JPA.

6.4. Лабораорна робота №4

6.4.1. Тема

Засвоєння технології JSF (Java Server Faces) для розробки веб-застосунків.

6.4.2. Теоретичні відомості

Теоретичні відомості щодо застосування технології JSF наведені у дидактичних матеріалах до дисципліни (лекції «Технологія JSF (1)» та «Технологія JSF (2)»), а також [8- 11].

Для підключення технології JSF до веб-застосунка у середовищі Netbeans IDE необхідно:

- 1) Вибрати у закладці Projects веб-застосування, викликати настройки проекту за допомогою правою кнопки миші.
- 2) Вибрати категорію “Frameworks” та додати фреймворк “Java Server Faces” (рис. 6.1).

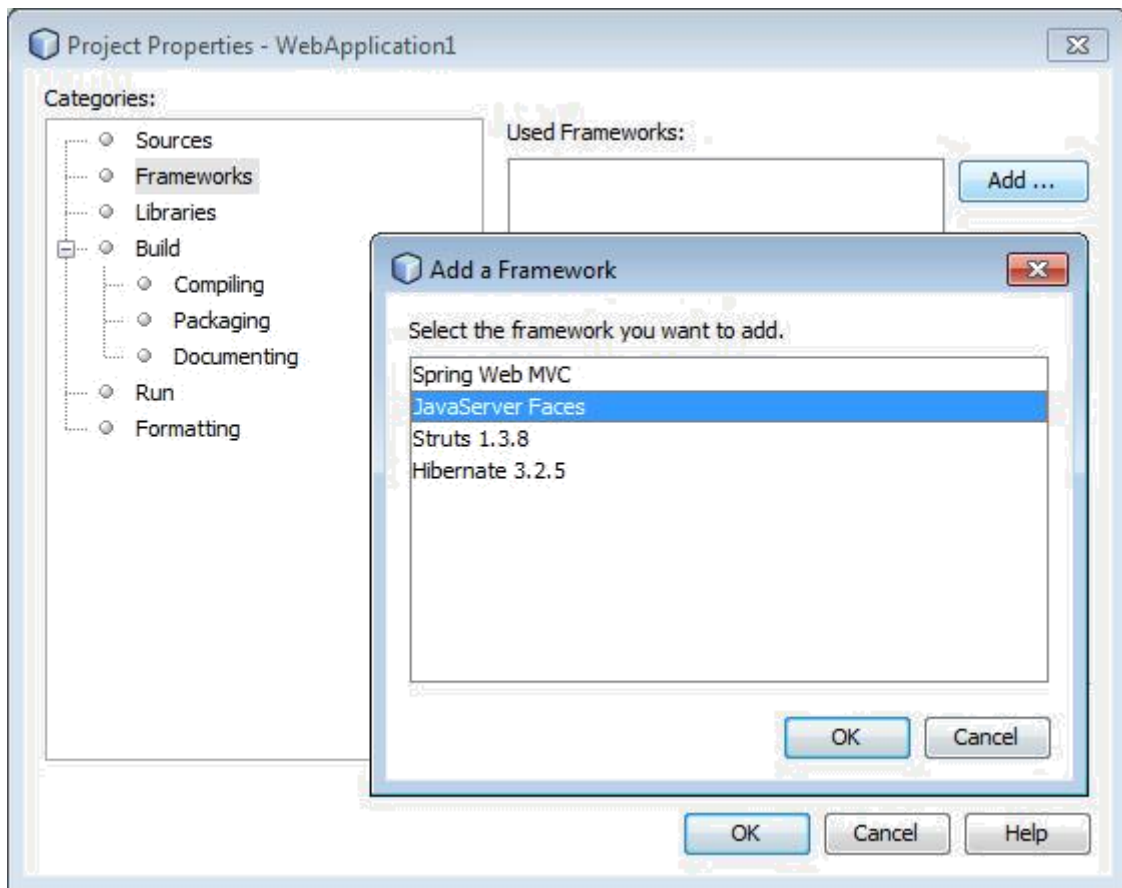


Рис. 6.1

6.4.3. Вимоги до виконання роботи

Для задачі роботи необхідно підготувати звіт.

Вимоги до звіту надані в розділі 3.

У звіті у розділі «Тексти програми» мають бути вказані:

- фрагменти веб-сторінок;
- фрагменти JSF backing bean.

6.4.4. Завдання

Завдання виконується на базі лабораторних робіт №№2-3.

У веб-застосуванні, яке було розроблене при виконанні лабораторних робіт №2 та №3, замінити графічний інтерфейс користувача, розроблений на базі технологій сервлетів та JSP, на графічний інтерфейс, розроблений на базі технології JSF.

Розробити веб-інтерфейс для наступних операцій:

- 1) вивід списку об'єктів;
- 2) створення об'єкту;
- 3) редагування об'єкту;
- 4) видалення об'єкту.

6.4.5. Контрольні питання

1. Вкажіть призначення технології JSF.
2. Опишіть життєвий цикл обробки запиту в технології JSF.
3. Вкажіть, із чого складається веб-застосування, розроблене на базі технології JSF.
4. Вкажіть області існування об'єктів, які підтримується технологією JSF.
5. Вкажіть основні системні задачі, які вирішуються фреймворком JSF.
6. Опишіть технологію конверторів в JSF.
7. Підтримка AJAX технологію JSF.

6.5. Лабораторна робота №5

6.5.1. Тема

Засвоєння технології EJB (Enterprise Java Beans) для розробки бізнес-логіки корпоративного застосування.

6.5.2. Теоретичні відомості

Теоретичні відомості щодо застосування технології EJB наведені у Дидактичних матеріалах до дисципліни [3] (лекції «Технологія EJB» та «Підтримка транзакцій в Java EE»), а також в [13] (глава 8).

6.5.3. Вимоги до виконання роботи

Для здачі лабораторної роботи необхідно підготувати звіт.

Вимоги до звіту надані в розділі 3.

В розділі «Текст програми» мають міститися фрагменти тексту EJB-компонентів.

6.5.4. Завдання

Завдання виконується на базі лабораторної роботи №4.

Розробити EJB-компонент(и), який містить наступні методи:

1. отримання списку об'єктів;
2. створення об'єкту;
3. редагування об'єкту;
4. видалення об'єкту.

Приділити увагу підтримці транзакцій контейнером для методів EJB-компонентів.

Для виконання роботи використовувати EJB-компоненти типу Stateless Bean. При цьому використовувати один із наступних типів взаємодії:

- 1) локальний тип взаємодії із локальним бізнес-інтерфейсом (local interface), або
- 2) локальний тип взаємодії без бізнес-інтерфейсу (no interface view).

При використанні local interface необхідно:

- 1) створити EJB-модуль та помістити в нього EJB-компонент;
- 2) створити Enterprise-модуль та помістити в нього EJB-модуль, створений на кроці 1 та веб-модуль, створений в рамках лабораторної роботи №4.

При використанні no interface view не потрібно створювати окремий EJB-модуль для EJB-компонентів. У цьому випадку є можливість помістити EJB-компонент всередину веб-модуля, створеного в рамках лабораторної роботи №4.

6.5.5. Контрольні питання

1. Основне призначення технології EJB.
2. Вкажіть, який контейнер управляє життєвим циклом EJB-компонентів.
3. Назвіть основні типи EJB-компонентів.
4. Назвіть призначення EJB-компонентів типу Stateless Session Bean.
5. Які enterprise-можливості забезпечує контейнер для EJB-компонентів.

6.6. Лабораторна робота №6

6.6.1. Тема

Засвоєння методів забезпечення аутентифікації та авторизації для платформи Java EE.

6.6.2. Теоретичні відомості

Теоретичні відомості щодо забезпечення аутентифікації та авторизації платформою Java EE наведено в дидактичних матеріалах до дисципліни [3] (лекція «Технології для забезпечення безпека Java EE-застосувань») та в [13] (глава 9).

Відомості щодо настройки реалму безпеки для серверу застосувань Glassfish наведено в розділі 5.1.5.

6.6.3. Вимоги до виконання роботи

Для здачі лабораторної роботи необхідно підготувати звіт.

Вимоги до звіту надані в розділі 3.

У розділі «Текст програми» має міститися наступна інформація:

- опис реалму безпеки для аутентифікації (це може бути знімок екрану із сторінкою настройки реалму);
- фрагменти тексту програми, які відповідають за настройку та функціонування аутентифікації та авторизації:
 - дескриптори розгортання web.xml, sun-web.xml (у разі використання Glassfish або інший спеціалізований дескриптор розгортання у разі використання іншого серверу застосувань);
 - фрагменти коду.

6.6.4. Завдання

Завдання виконується на базі лабораторних робіт №4 та №5.

- 1) Налаштувати реалм безпеки для серверу застосувань.
- 2) Забезпечити аутентифікацію та авторизацію для веб-модуля застосування.

6.6.5. Контрольні питання

1. Чим відрізняється аутентифікація від авторизації.
2. Які компоненти мають бути захищені в Java EE.
3. Опишіть механізми безпеки, які надаються платформою Java EE.
4. Опишіть, як Java EE сервер застосувань здійснює аутентифікацію користувача.
5. Опишіть, як здійснюється авторизація Java EE сервером застосувань.
6. Опишіть ролеву модель безпеки, яка використовується Java EE сервером застосувань.

6.7. Лабораторна робота №7

6.7.1. Тема

Засвоєння технологій веб-сервісів JAX-WS/JAX-RS.

6.7.2. Теоретичні відомості

Теоретичні відомості щодо технологій веб-сервісів JAX-WS/JAX-RS наведено в дидактичних матеріалах до дисципліни (лекція «Технології JAX-WS та JAX-RS для розробки веб-сервісів») та в [14].

6.7.3. Вимоги до виконання роботи

Для здачі роботи необхідно підготувати звіт.

Вимоги до звіту надані в розділі 3.

6.7.4. Завдання

Завдання виконується на базі лабораторної №5.

У веб-застосуванні, яке було розроблене при виконанні лабораторної роботи №5, виконати наступне:

- 1) Реалізувати точку монтування веб-сервісу в вигляді EJB компоненту на базі технології JAX-WS. Здійснити виклик розробленого веб-сервісу із веб-застосування.
- 2) Реалізувати веб-сервіс на базі технології JAX-RS.

6.7.5. Контрольні питання

1. Опишіть відмінність SOAP-веб-сервісів від RESTful-веб-сервісів.
2. Яка Java EE-технологія забезпечує роботу із SOAP-веб-сервісами.
3. Яка Java EE-технологія забезпечує роботу із RESTful-веб-сервісами.
4. Які Java EE-компоненти можуть бути точками монтування SOAP-веб-сервісів.

ЛІТЕРАТУРА

1. Java Code Convention. - <http://www.oracle.com/technetwork/java/codeconv-138413.html>.
2. Java EE & Java Web Learning Trail. - <http://netbeans.org/kb/trails/java-ee.html>
3. О.П. Сирота. Дидактичні матеріали до дисципліни «Мова програмування Java та технології J2EE», модуль «Технології Java EE».
4. Кей С. Хорстманн, Гари Корнелл. Java 2. Библиотека профессионала, том 2. Тонкости программирования. – Вильямс, 2002. – 1120 с.
5. Bruce W. Perry. Java Servlet & JSP Cookbook. - O'Reilly. - 2004. - 746p.
6. Mike Keith, Merrick Schincariol. Pro JPA 2: Mastering the Java™ Persistence API. – APress: 2009. – 538 p.
7. Bob McCune. Exploring the Java Persistence API. - <http://www.intertech.com/resource/usergroup/Exploring%20the%20JPA.pdf>
8. JSF 2.0 Tutorials. - <http://www.coreservlets.com/JSF-Tutorial/jsf2/>
9. JSF for nonbelievers: The JSF application lifecycle. - <http://www.ibm.com/developerworks/library/j-jsf2>
10. JSF 2.0 Tutorials. Integrated Ajax Support in JSF 2.0 - <http://www.coreservlets.com/JSF-Tutorial/jsf2/#Ajax>
11. JSF 2.0 Tutorials. Page templating with facelets. - <http://www.coreservlets.com/JSF-Tutorial/jsf2/#Templating>
12. Deby Panda, Reza Rahman, Derek Lane. EJB 3 in Action. – Manning: 2007.-677p.
13. J2EE BluePrints. Designing Enterprise Applications with the J2EE Platform.. http://www.adobe.com/support/jrun/documentation/pdf/j2ee_ent_app_design

[.pdf](#)

14. The Java EE 6 Tutorial. Part III Web Services. -

<http://docs.oracle.com/javaee/6/tutorial/doc/bnayk.html>

ДОДАТОК 1
ШАБЛОН ЗВІТУ

Національний Технічний Університет України
“Київський Політехнічний Інститут”

Факультет інформатики та обчислювальної
техніки Кафедра технічної кібернетики

Звіт до лабораторної роботи №1 з
модуля
“Технології розробки програмного забезпечення”

Прийняв

Асистент кафедри ОТ
Подрубайло
О.О.

“...” 20xx р.

Виконав

Студент групи ІІІ-ХХ

Іванов Х.Х.

Київ 20xx

Лабораторна робота №1

Тема:

Завдання:

UML діаграми

Текст програми

Висновки

Програма вирішує поставлене завдання. Програма працює правильно.