

# ЛАБОРАТОРНА РОБОТА 1

**Тема:** Інструментальна платформа корпоративних додатків Java

**Мета:** Встановлення та налаштування програмного забезпечення для розробки компонент корпоративних інформаційних систем

До складових середовища розробки програмних компонент на платформі Jakarta EE (Java EE) відноситься наступне програмне забезпечення:

**1. Java SE Development Kit (JDK)** – базовий набір інструментів платформи Java (Standard Edition, стандартної редакції) для розробки та впровадження Java-додатків для десктопів та серверів [1]. Сьогодні Java SE має багато реалізацій, але слід мати на увазі, що принциповим є наявність гілки версії 1.8 та гілки, що відокремилась з версії 9. Насамперед вони відрізняються структурою дерева класів, а з точки зору платформи Enterprise Edition (EE) — підтримкою Java EE або Jakarta EE. Для стабільної роботи обраної платформи EE, або тієї, що вже використовується і не може бути замінена з певних причин на підприємстві, слід встановлювати ту версію Java SE, яка відповідає інструментам та іншому програмному забезпеченню, що ґрунтується на неї. Наприклад, якщо на підприємстві використовується платформа на основі Java SE версії 1.8, то розробку нових компонент або удосконалення існуючих доведеться виконувати з використанням саме цієї версії. Розробка компонент на нових версіях Java SE, наприклад, на 21+ не приведе до робочих та стабільних варіантів, а їх налагодження забере тільки час. Слід також зауважити, що Java SE версії 1.8 продовжує удосконалюватись у своїй реалізації, підтримує як нову архітектуру x64, так і стару x32 і має останню версію реалізації jdk-8u421 [2]. Число після літери «u» вказує на номер реалізації, тому чим воно більше, тим менше часу пройшло від її виходу.

Інша гілка Java SE, що почалась з версії 9 має регулярні релізи один раз на півроку (як правило, у березні та у вересні кожного року). Деякі релізи також мають додаткову позначку LTS (Long-Term Support). Так, зараз доступна версія Java SE 23, підтримка якої продовжиться до виходу у березні 2025 р. Java SE версії 24, але при цьому доступна версія Java SE 21 LTS, підтримка якої буде продовжуватись аж до вересня 2026 року [3, 4]. Як правило саме LTS версії використовуються при створенні платформи EE на підприємствах.

## **2. Інтегроване середовище розробки додатків**

Платформа Eclipse. Найпростіший спосіб встановити або оновити середовище розробки Eclipse це використати спеціальний інструмент — Eclipse Installer [5]. Цей інструмент призначено для ефективного вибору версії та збірки Eclipse, що буде встановлюватись на комп'ютер. На вказаній

у посиланні сторінці можна перейти до вибору необхідного пакету та встановити його самостійно без допомоги Eclipse Installer. У грудні 2024 р. вийшли нові збірки пакету Eclipse. Нова збірка Eclipse IDE, що підтримує корпоративні технології Java зветься Eclipse-JEE 2024-12-R і має інтегрований набір інструментів для розробки компонентів: EJB, JSP та JSF, JPA та Data Tools, Mylyn, Servlets, EGit тощо. У подальшому компонентний склад Eclipse можна змінити через Help->Install New Software... Наявність або відсутність компонент можна перевірити і через відповідні каталоги Eclipse - plugins та features.

### **3. Сервер додатків WildFly**

Для створення та виконання застосунків на основі корпоративних технологій Java (Jakarta EE / Java EE) необхідно мати сервер, в якому імплементовано відповідні специфікації [6]. Одним із серверів, що забезпечують підтримку усього стеку специфікацій Jakarta EE є WildFly. Актуальною версією є 35.0.1.

### **4. JBoss Tools**

JBoss Tools — це набір плагінів для Eclipse, призначений для розробки додатків на основі Jakarta EE (Java EE) та технологій Red Hat JBoss Middleware (WildFly). Він забезпечує підтримку JSF, JPA, CDI, EJB, JAX-RS, інтеграцію з WildFly та іншими серверними контейнерами, допомагає із розгортанням та налагодженням застосунків на JBoss/WildFly, надає підтримку щодо керування контейнерами Docker та хмарного розгортання на OpenShift. Також до набору входять плагіни для роботи з базами даних, інструменти для Hibernate, генерації JPA сутностей з бази даних та візуальний редактор JPQL. Інструменти підтримують системи Maven і Gradle, керування залежностями і генерацію проектів за допомогою Maven Archetypes. Вони також забезпечують автоматизацію створення Jakarta EE-компонентів та підтримують певні фронтенд-технології. Поточна версія JBoss Tools 4.29.1 [7].

### **5. СУБД Derby**

Apache Derby є повнофункціональною системою управління реляційними базами даних (СУБД), яка базується на технологіях Java та SQL [8]. Її повністю написано та імплементовано на мові програмування Java. Derby надає користувачам заснований на стандартах механізм управління базами даних, який може щільно вбудовуватись в будь-яке рішення на основі Java, а також забезпечує цілісність даних і підтримку транзакцій.

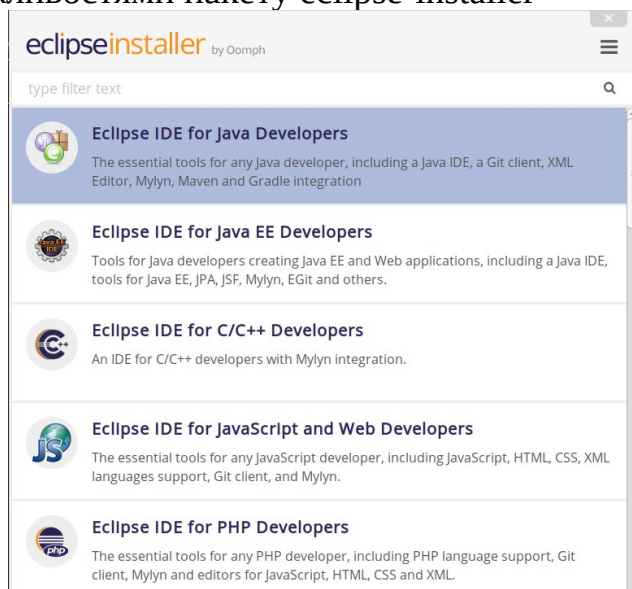
Формат бази даних на диску, який використовує Derby, є портативним та незалежним від платформи. Останнє полягає в тому, що бази даних Derby можуть переміщатись з одного комп'ютера на інший без необхідності зміни або модернізації даних. Додаток Derby може містити попередньо побудовану, заповнену базу даних і ця база даних буде працювати в будь-якій конфігурації Derby.

Останньою версією Apache Derby є 10.17.1.0, яка базується на Java SE 21 з JDBC 4.2. Ця версія не підтримує менші версії Java SE.

## ПРАКТИЧНА ЧАСТИНА

1. Встановіть JDK

2. Для інсталяції Eclipse розпакуйте відповідний файл архіву або скористайтесь можливостями пакету eclipse-installer

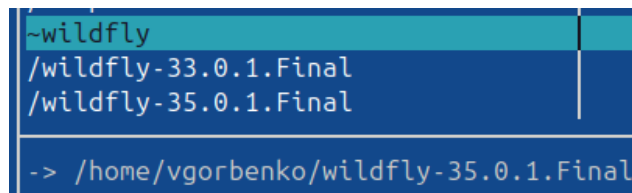


3. Інсталяція та запуск WildFly

Останньою версією WildFly є 35.0.1 і його дистрибутив знаходиться у файлі wildfly-35.0.1.Final.tar.gz. Для розпакування слід використати команду:

```
tar -xvzf wildfly-35.0.1.Final.tar.gz
```

Розпакування файлів дистрибутиву буде виконано у каталог wildfly-35.0.1.Final. Для зручності посилання на файли дистрибутиву та спрощення подальшого оновлення WildFly краще зробити символічне посилання на цей каталог за скороченим ім'ям:



Для нормальної роботи серверу необхідно встановити значення змінних середовища JBOSS\_HOME та WILDFLY\_HOME на повний шлях до каталогу з файлами WildFly. Також необхідно додати шлях до підкаталогу bin сервера у змінну середовища PATH. Це використовується для запуску

серверу WildFly поза середовища розробки. В ОС Ubuntu 24.04 для виконання цього необхідно додати певні рядки у файл «.profile»:

```
# set JBOSS_HOME if it not exists
if [ -z "$JBOSS_HOME" ] ; then
  if [ -d "$HOME/wildfly" ] ; then
    export JBOSS_HOME="$HOME/wildfly"
  fi
fi

# set WILDFLY_HOME if it not exists
if [ -z "$WILDFLY_HOME" ] ; then
  if [ -d "$HOME/wildfly" ] ; then
    export WILDFLY_HOME="$HOME/wildfly"
  fi
fi

# set PATH so it WILDFLY/bin if it exists
if [ -d "$WILDFLY_HOME/bin" ] ; then
  PATH="$WILDFLY_HOME/bin:$PATH"
fi
```

Запуск серверу WildFly в режимі standalone виконується за допомогою команди:

```
<JBOSS_HOME>/bin/standalone.sh
```

Якщо сервер запустився коректно, то за допомогою Інтернет-браузеру за адресою <http://localhost:8080> буде показано вікно подібне до



Адміністрування серверу здійснюється за допомогою Administration Console або за адресою <http://localhost:9990>. Для доступу до цієї сторінки попередньо необхідно створити користувача з правами «Management User». Для цього необхідно виконати команду

```
<JBOSS_HOME>/bin/add-user.sh
```

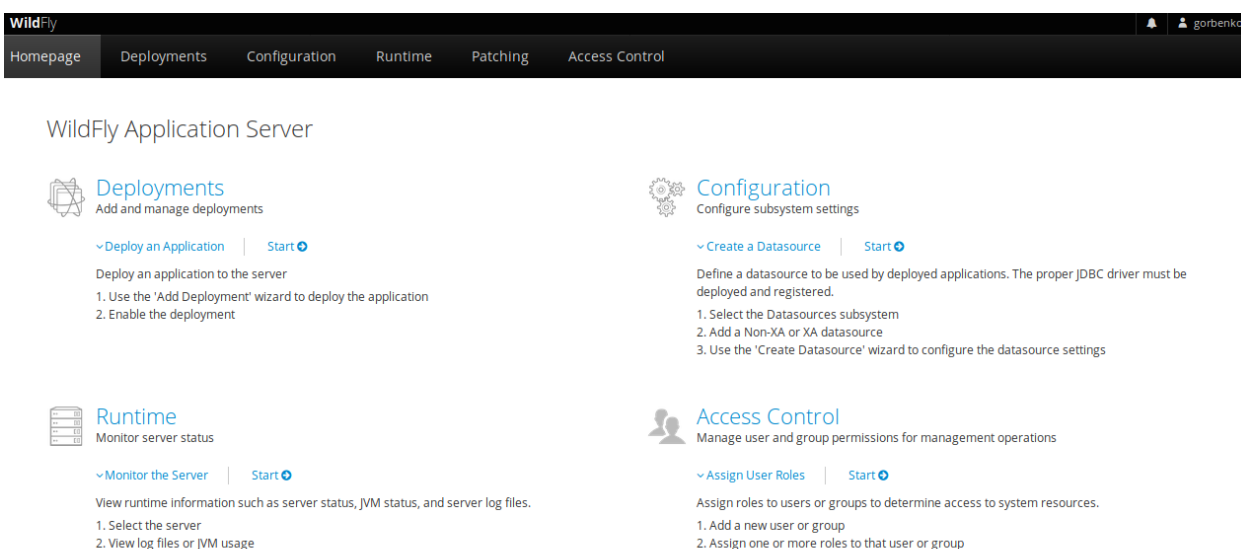
і послідовно вибрати роль користувача та задати його «Username» та «Password»:

```
vgorbenko@vgorbenko-System-Product-Name:~/wildfly/bin$ ./add-user.sh

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : █
```

Користувач додається у файл <JBOSS\_HOME>/standalone/configuration/mgmt-users.properties, а відповідна його група додається у файл <JBOSS\_HOME>/standalone/configuration/mgmt-groups.properties. Тепер доступ до Administration Console стає можливим:



Для зупинки серверу з командного рядка слід використати команду:

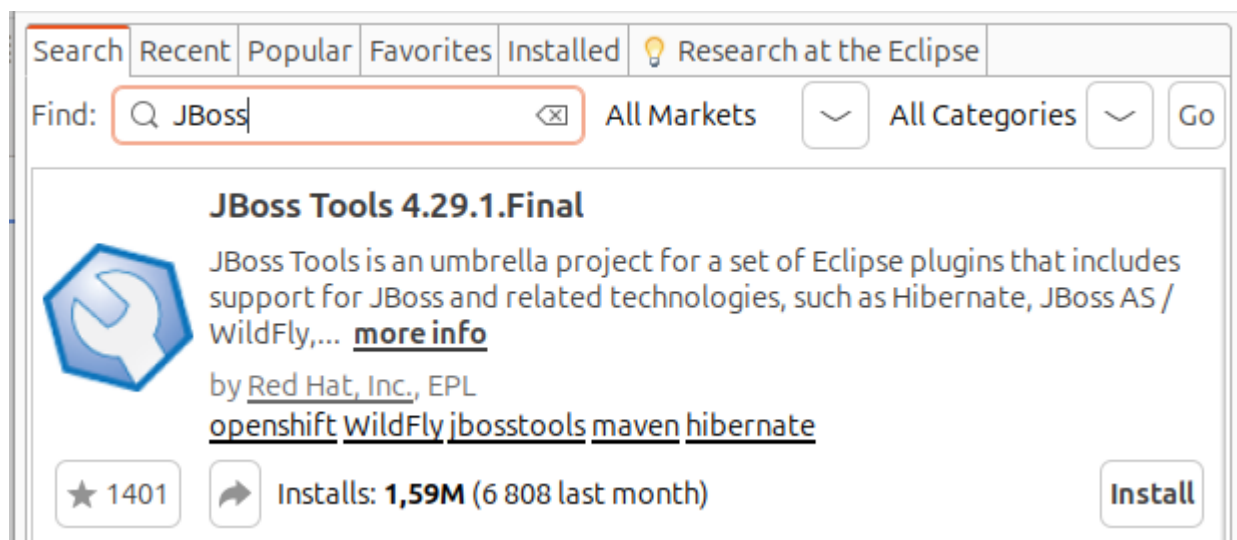
```
jboss-cli.sh - - connect - - command=: shutdown
```

Для перевірки правильності відпрацювання команди слід завантажити (перезавантажити) сторінку <http://localhost:8080> та переконатись, що сервер зупинено.

Основним засобом перегляду стану сервера WildFly і результатів виконання тих, чи інших операцій слугує перегляд та аналіз його log файлів. Основний log файл серверу WildFly має назву server.log і його розташовано у [JBOSS\_HOME]/wildfly-35.0.1.Final/standalone/log .

Сервер застосунків WildFly також може бути налаштований як сервіс — його запуск буде виконуватись одночасно із запуском операційної системи із загальними правами (wildfly:wildfly), які не будуть пов'язані з будь-яким конкретним користувачем. Також декілька серверів можуть бути об'єднані в домен із загальною точкою керування. Корисною є можливість створення завантажуваного JAR WildFly, який буде вміщувати як сервер, так і створений пакет застосунку у вигляді jar, ear або war. Усі варіанти використання WildFly добре описані в його документації.

4. Зручним засобом підключення JBoss Tools до Eclipse є Eclipse Marketplace:



Альтернативно встановлення JBoss на Eclipse можна виконати через

Help > Install New Software... >

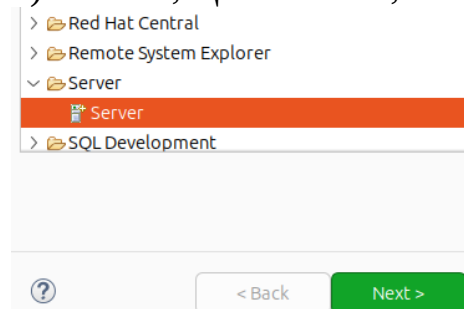
де необхідно задати шлях до пакетів

<https://download.jboss.org/jbosstools/tonic/stable/updates/>

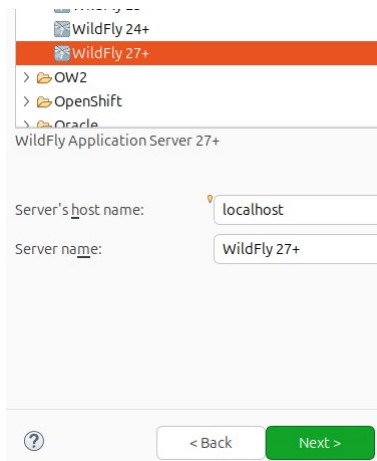
Вибір пакетів треба виконати самостійно. Пакети з позначкою «Deprecated» обирати не слід.

5. Тільки після інсталяції JBoss Tools з'являється можливість для підключення серверу додатків WildFly до IDE Eclipse. Для цього необхідно вибрати File → New → Other.

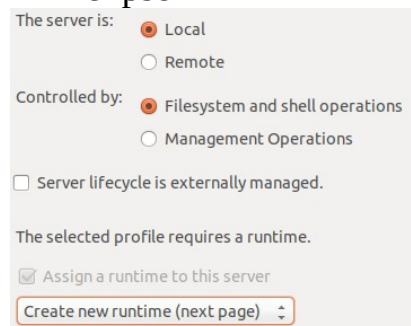
1) У вікні, що з'явиться, обираємо Server → Server та натискаємо Next.



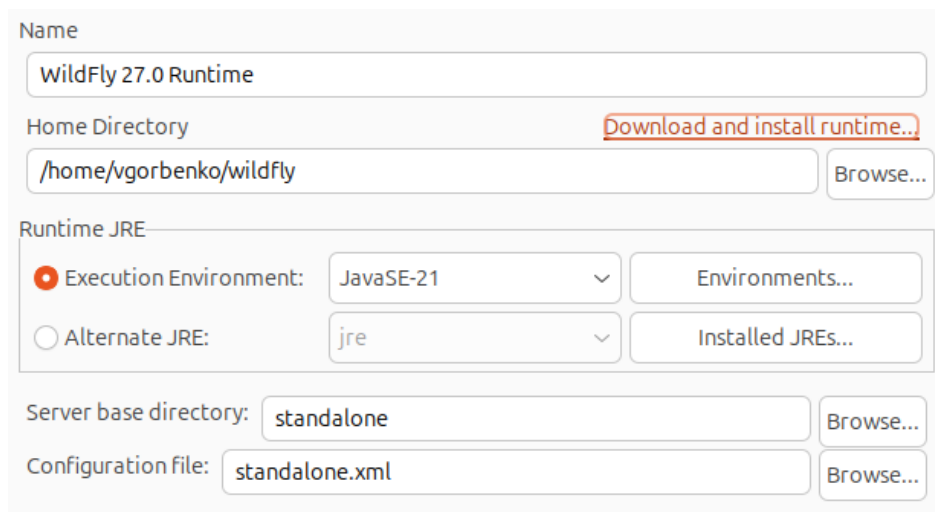
- 2) Для підключення серверу WildFly слід обрати відповідний тип із групи JBoss Community



- 3) Для вивчення та дослідження технологій Jakarta EE/Java EE сервер додатків WildFly буде використовуватись локально, тому слід залишити або встановити наступні налаштування при його підключенні до IDE Eclipse

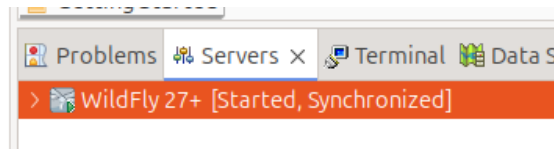


- 4) За допомогою кнопки Browse слід вказати домашній каталог сервера WildFly. Також слід налаштувати інші параметри, щоб вони відповідали наявній системі, наприклад, як показано на наступному скриншоті:





Для завершення підключення серверу додатків необхідно натиснути кнопку Finish. За шляхом Window → Show View → Servers відкриється вікно, що показує базові параметри стану серверу додатків, а також через яке можна виконати його налаштування, переглядати та видаляти встановлені додатки. Також у цьому вікні можна запускати та зупиняти сервер.



- 5) Тепер слід перевірити здатність IDE Eclipse запускати та зупиняти сервер WildFly. Це можна зробити або за допомогою відповідних засобів на панелі інструментів, або у вікні Server. При успішному початку запуску Eclipse автоматично перемикається на вікно Console, в якому будуть послідовно виводитись повідомлення запуску сервера. Як тільки вивід повідомлень зупиниться, можна перевірити працездатність серверу за допомогою браузера за посиланням <http://localhost:8080>, як це робилось раніше.
- 6) Зупинити сервер додатків WildFly можна за допомогою тих самих засобів.

## 6. Інсталяція СУБД Apache Derby

Завантажте ZIP або TAR архів дистрибутиву db-derby та розпакуйте його. Інсталяція вміщує наступні директорії:

- *demo* — вміщує демонстраційні програми;
- *bin* — вміщує скрипт-файли для виконання утиліт та налаштування середовища оточення;
- *javadoc* — вміщує документацію по арі, яку було згенеровано із коментарів файлів коду;
- *docs* — вміщує документацію по Derby;
- *lib* — вміщує файли .jar серверу Derby;
- *test* — вміщує регресійний тест для серверу Derby.

Сервер Derby має декілька способів запуску, які залежать від типу його використання та впливають на загальні налаштування. У випадку, коли необхідно запускати Derby із найменшою кількістю набору та встановлено його повний бінарний дистрибутив, тоді сервер запускається з використанням скриптів для командного рядка. Цей метод потребує встановлення змінної середовища оточення DERBY\_HOME, включення до значення змінної середовища оточення PATH розташування директорії *bin* встановленого дистрибутиву java, а також директорії DERBY\_HOME\bin.

Якщо дистрибутив Derby не має директорії *bin* або в ній не поновний набір бінарних файлів, але є усі jar-файли дистрибутиву і на додаток не має змоги (або навіть бажання) використовувати скрипт-файли, що



поставляються із Derby, тоді застосовується метод запуску серверу за допомогою файлу `derbyrun.jar`. Для забезпечення цього методу встановлюється значення змінної середовища оточення `DERBY_HOME` та додається до змінної `PATH` місце розташування бінарних файлів `java` (зазвичай це `JAVA_HOME\bin`). Файл `derbyrun.jar` не може бути переміщеним із директорії його первинного розташування у дистрибутиві Derby.

Ще один різновид запуску Derby пов'язано з наступним: відсутній `derbyrun.jar` та відповідні скрипт-файли, або є необхідність у вивченні повного синтаксису кожної команди та деталізацій параметрів змінних середовища оточення, або розробляються програмні додатки з використанням Derby. У цьому випадку для запуску відповідних інструментів використовується або повний синтаксис `java`, або Derby використовується у `java`-програмах. Для забезпечення цього необхідно встановити відповідні значення для змінної середовища оточення `DERBY_HOME`, включити до значення змінної `PATH` місце розташування файлів виконання `java` (наприклад, `JAVA_HOME\bin`), а також необхідно знати повну назву пакету для `java`-класу, що підтримує роботу серверу. Змінна середовища оточення `CLASSPATH` повинна включати необхідні JAR-файли.

Одним із найкращих варіантів є інсталяція Apache Derby до серверу додатків WildFly. У цьому випадку Apache Derby буде запускатись автоматично із сервером WildFly. Необхідними для такої інтеграції файлами є: `derby.jar`, `derbynet.jar`, `derbytools.jar`, `derbyshared.jar`. Для виконання інтеграції Apache Derby в якості вбудованого серверу до WildFly буде використовуватись `jboss-cli.sh` зі складу утиліт WildFly. Тому спочатку запускається сам сервер додатків WildFly, потім — утиліта `jboss-cli.sh`. У відповідь буде отримано командний рядок з префіксом `[disconnected /]`. Якщо сервер додатків WildFly знаходиться у робочому стані, то команда `connect` приєднає до нього і префікс рядку зміниться, наприклад, на наступний `[standalone@localhost:9990]`, після чого можна задавати команди.

Далі інсталюємо `jar`-файли серверу Derby як модулі до WildFly (у прикладах, що наведено нижче, `jar`-файли знаходяться у каталозі `/home/vgorbenko/db-derby-10.17.1.0-bin/lib` і зроблено відносне посилання `~/db-derby` до основного каталогу серверу Derby)

```
module add \  
  --name=org.apache.derby \  
  --resources=/home/vgorbenko/db-derby/lib/derby.jar,  
/home/vgorbenko/db-derby/lib/derbynet.jar,  
/home/vgorbenko/db-derby/lib/derbytools.jar,  
/home/vgorbenko/db-derby/lib/derbyshared.jar, \  
  --resource-delimiter=, \  
  --dependencies=java.api,java.transaction.api
```

Наступним реєструємо драйвер JDBC під ім'ям "derby". Для цього вводимо наступне:



```
/subsystem=datasources/jdbc-driver=derby:add(driver-name=derby, \  
driver-module-name=org.apache.derby, \  
  )
```

```
driver-class-name=org.apache.derby.jdbc.EmbeddedDriver, \  
driver-datasource-class-name=org.apache.derby.jdbc.EmbeddedDataSource, \  
driver-xa-datasource-class-name=org.apache.derby.jdbc.EmbeddedXADataSource)
```

У відповідь повинно бути отримано: { "outcome" => "success" }

```
vgorbenko@vgorbenko-System-Product-Name:~/wildfly/bin$ ./jboss-cli.sh  
You are disconnected at the moment. Type 'connect' to connect to the server or '  
help' for the list of supported commands.  
[disconnected /] connect  
[standalone@localhost:9990 /] module remove --name=org.apache.derby  
[standalone@localhost:9990 /] module add --name=org.apache.derby --resources=~/d  
b-derby/lib/derby.jar,~/db-derby/lib/derbytools.jar,~/db-derby/lib/derbynet.jar,  
~/db-derby/lib/derbyshared.jar --resource-delimiter=, --dependencies=javax.api,j  
avax.transaction.api  
  
[standalone@localhost:9990 /] /subsystem=datasources/jdbc-driver=derby:add(drive  
r-name=derby,driver-module-name=org.apache.derby,driver-class-name=org.apache.de  
rby.jdbc.EmbeddedDriver,driver-datasource-class-name=org.apache.derby.jdbc.Embed  
dedDataSource,driver-xa-datasource-class-name=org.apache.derby.jdbc.EmbeddedXAData  
taSource)  
{"outcome" => "success"}
```

Також під'єднання Apache Derby до WildFly можна буде побачити через консоль адміністратора WildFly в браузері:

<u>Datasources &amp; Drivers</u>		JDBC Driver
Datasources	>	<i>Filter by: driver name or provider type</i>
JDBC Drivers	>	 derby
		 h2

Тепер необхідно задати системні властивості, які будуть перемикаєть Derby для прослуховування порту. Для цього знову слід запустити утиліту jboss-cli.

```
/system-property=derby.drda.startNetworkServer:add(value=true)
```

```
[standalone@localhost:9990 /] /system-property=derby.drda.startNetworkServer:add(value=true)  
{"outcome" => "success"}
```

Встановлюємо специфікації для цього порту Derby

```
/system-property=derby.drda.portNumber:add(value="1527")
```

```
[standalone@localhost:9990 /] /system-property=derby.drda.portNumber:add(value="1527")  
{"outcome" => "success"}
```

## Задаємо специфікацію розташування log файлу Derby

```
/system-property=derby.stream.error.file:add(value="${jboss.server.log.dir}/derby.log")
```

```
[standalone@localhost:9990 /] /system-property=derby.stream.error.file:add(value="${jboss.server.log.dir}/derby.log")
{"outcome" => "success"}
```

## Задаємо системне значення про існування derby.log

```
/system-property=derby.infolog.append:add(value="true")
```

```
[standalone@localhost:9990 /] /system-property=derby.infolog.append:add(value="true")
{"outcome" => "success"}
```

Для визначення значення відносного шляху `${jboss.server.log.dir}` можна скористатись командою:

```
:resolve-expression(expression=${jboss.server.log.dir}
```

```
[standalone@localhost:9990 /] :resolve-expression(expression=${jboss.server.log.dir}
{
  "outcome" => "success",
  "result" => "/home/vgorbenko/wildfly/standalone/log"
}
```

Наступним етапом є призначення джерела даних DataSource. За допомогою команд `jboss-cli` можна як створити нове джерело даних, так і під'єднати вже створене. При створенні нового джерела даних необхідно впевнитись, що відповідний каталог префіксу є доступним і дозволяє в ньому створити новий каталог джерела даних (для бази даних). При під'єднанні наявної бази даних необхідно впевнитись у наявності відповідного каталогу з файлами бази даних і їх доступність.

Розглянемо варіант додавання нового джерела даних. Для позначення шляху використаємо префікс `${jboss.server.base.dir}/derbydata`, де перша частина має значення:

```
[standalone@localhost:9990 /] :resolve-expression(expression=${jboss.server.base.dir}
{
  "outcome" => "success",
  "result" => "/home/vgorbenko/wildfly/standalone"
}
```

Створюємо підкаталог `derbydata` за вказаним шляхом і встановлюємо йому властивості на читання та запис. Тепер в `jboss-cli` можна задати команди, що додасть посилання на дані в реєстрі WildFly та створить необхідний каталог та файли за вказаним шляхом.

```

data-source add \
--name=DerbyExampleDS \
--driver-name=derby \
--connection-url="jdbc:derby:${jboss.server.base.dir}/derbydata/
exampleDS;create=true" \
--jndi-name=java:jboss/datasources/DerbyExampleDS

```

```

[standalone@localhost:9990 /] data-source add --name=DerbyExampleDS --driver-name
=derby --connection-url="jdbc:derby:${jboss.server.base.dir}/derbydata/exampleDS;
create=true" --jndi-name=java:jboss/datasources/DerbyExampleDS

```

Додатковими параметрами з відповідними значеннями можуть бути:

```

--user-name=app \
--password=app

```

Тестування підключення джерела даних виконуємо за допомогою команди:

```

/subsystem=datasources/data-source=DerbyExampleDS:test-connection-in-pool

```

```

[standalone@localhost:9990 /] /subsystem=datasources/data-source=DerbyExampleDS:test-connection-in-pool
{
  "outcome" => "success",
  "result" => [true]
}

```

За вказаним шляхом в команді створення джерела даних повинний створитись новий каталог:

```

<- ~/wildfly/standalone/derbydata
.n          За назвою      За розм
/..         НАД-КАТ
/exampleDS  4096

```


Створення нового джерела даних також можна побачити за допомогою консолі адміністратора WildFly:

Datasources & Drivers		Datasource
Datasources	>	Filter by: name, xa, .../disabled, deployment <input checked="" type="checkbox"/> DerbyExampleDS <input type="button" value="View"/>
JDBC Drivers	>	<input checked="" type="checkbox"/> ExampleDS

а також її головні атрибути, які було введено в команді створення даних:

## DerbyExampleDS

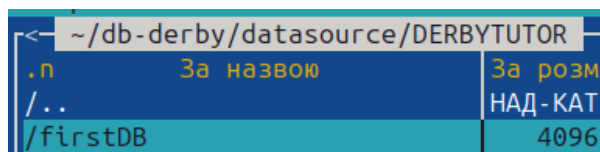
Datasource

 The datasource **DerbyExampleDS** is enabled. [Disable](#)

### Main Attributes

JNDI Name:	java:jboss/datasources/DerbyExampleDS
Driver Name:	derby
Connection URL:	jdbc:derby:\${jboss.server.base.dir}/derbydata/exampleDS;create=tr...
Enabled:	true
Statistics Enabled:	false

Наступним розглянемо варіант додавання джерела даних firstDB, яке існує в каталозі ~/db-derby/datasource/DERBYTUTOR:



<	~/db-derby/datasource/DERBYTUTOR	
.	За назвою	За розм
./		НАД-КАТ
/firstDB		4096

Для посилання на цей каталог можна використати або повне пряме посилання, наприклад:

```
/home/vgorbenko/db-derby/datasource/DERBYTUTOR
```

або використати задане значення відповідної змінної середовища:

```
${DERBY_HOME}/datasource/DERBYTUTOR
```

Оскільки база даних вже існує, то виконується команда під'єднання джерела:

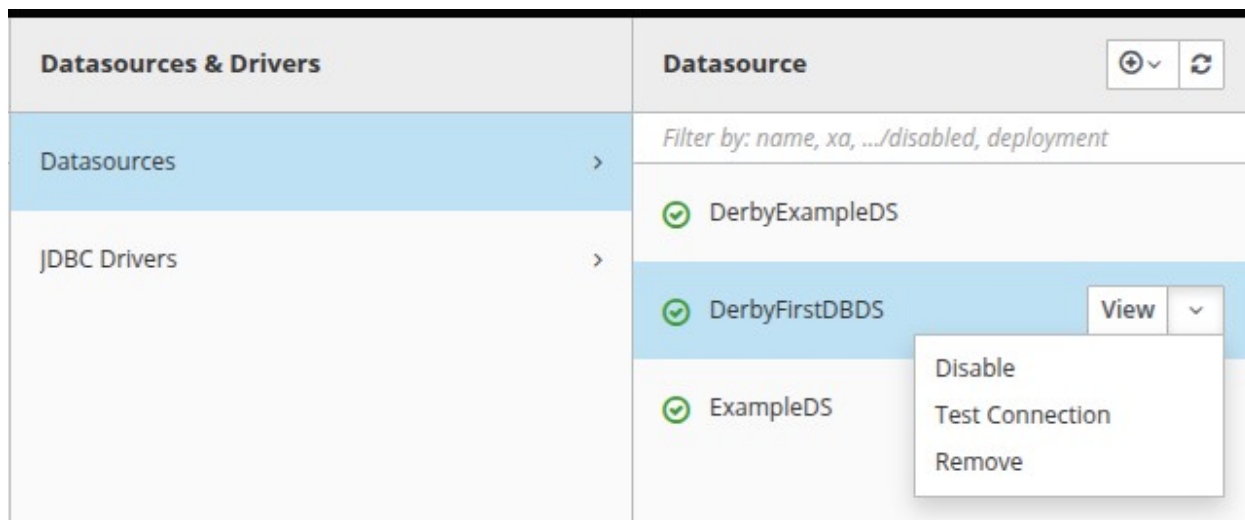
```
data-source add \  
--name=DerbyFirstDBDS \  
--driver-name=derby \  
--connection-url="jdbc:derby:/home/vgorbenko/db-derby/datasource/  
DERBYTUTOR/firstDB" \  
--jndi-name=java:jboss/datasources/DerbyFirstDBDS \  
--enabled=true
```

```
[standalone@localhost:9990 /] data-source add --name=DerbyFirstDBDS --driver-name=derby  
--connection-url="jdbc:derby:/home/vgorbenko/db-derby/datasource/DERBYTUTOR/firstDB" -  
-jndi-name=java:jboss/datasources/DerbyFirstDBDS --enabled=true
```

Тестування підключення джерела даних:

```
[standalone@localhost:9990 /] /subsystem=datasources/data-source=DerbyFirstDBDS:test-connection-in-pool  
{  
  "outcome" => "success",  
  "result" => [true]  
}
```

Також тестування підключення джерела даних можна виконати в консолі адміністратора WildFly:



7. Виконайте усі пункти практичної частини лабораторної роботи. Підготуйте звіт, в який додайте скриншоти до усіх пунктів.

### Список рекомендованої літератури

1. Jakarta EE Specifications [Електронний ресурс]. - Режим доступу: <https://jakarta.ee/specifications/>
2. Java SE Development Kit [Електронний ресурс]. - Режим доступу: <https://www.oracle.com/cis/java/technologies/javase/javase8u211-later-archive-downloads.html>
3. JDK Development Kit 21.0.5 downloads [Електронний ресурс]. - Режим доступу: <https://www.oracle.com/java/technologies/downloads/#java21>
4. JDK Project [Електронний ресурс]. - Режим доступу: <https://openjdk.org/projects/jdk/>
5. Eclipse Installer [Електронний ресурс]. - Режим доступу: <http://www.eclipse.org/downloads/packages/installer>
6. Jakarta EE Specifications. Platform and Profile Specifications [Електронний ресурс]. - Режим доступу: <https://jakarta.ee/specifications/>
7. JBoss Tools. Eclipse Plugins for JBoss Technology [Електронний ресурс]. - Режим доступу: <https://tools.jboss.org/>
8. Apache Derby: Downloads [Електронний ресурс]. - Режим доступу: [http://db.apache.org/derby/derby\\_downloads.html](http://db.apache.org/derby/derby_downloads.html)