Учреждение образования

Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**Отчет по лабораторной работе**

«Особенности технологий *Java EE*»

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Выполнил студент

ФИТ ПОИТ 5-2 Валдайцев А. Д.

Проверил

кандидат технических наук

Сулим П.Е.

Отчет по лабораторной работе

защищен с отметкой баллов

Минск 2024

***Цель* *работы*:** Изучение особенностей технологий *Java EE*.

Привести в отчете развернутые ответы по следующим пунктам:

* Какие версии *Java EE* существуют?
* Объясните понятия *containers* и *application servers*.
* Какие технологии включает *Java* EE? Приведите список технологий, их назначение и соответствующие спецификации, а также эталонную реализацию для каждой.
* Объясните понятия *JCP* и *JSR.*

Приведите краткие выводы по содержанию работы.

1. **Какие версии *Java EE* существуют?**

Java EE (Java Platform, Enterprise Edition) является платформой, предназначенной для разработки и выполнения приложений в области предприятий.

*1. Java EE 1.2:*

- Выпущена в декабре 1999 года.

- Включает в себя API для разработки веб-приложений, таких как сервлеты, **JSP** (Java Server Pages), EJB (Enterprise JavaBeans) и **JDBC** (Java Database Connectivity).

- Включает технологии для управления транзакциями, безопасностью и распределенным доступом к объектам.

*2. Java EE 1.3:*

- Выпущена в сентябре 2001 года.

- Добавлены новые возможности, такие как **JMS** (Java Message Service), JAXP (Java API for XML Processing) и JAXR (Java API for XML Registries).

- Включает поддержку веб-сервисов с использованием технологий SOAP (Simple Object Access Protocol) и UDDI (Universal Description, Discovery, and Integration).

*3. Java EE 1.4:*

- Выпущена в ноябре 2003 года.

- Включает поддержку новых спецификаций, таких как JSF (JavaServer Faces) для разработки веб-интерфейсов и JPA (Java Persistence API) для работы с базами данных.

- Включает обновленные версии других технологий, таких как сервлеты, EJB и JMS.

*4. Java EE 5:*

- Выпущена в мае 2006 года.

- Включает значительные изменения и усовершенствования, такие как введение аннотаций для конфигурации приложений, упрощение разработки веб-сервисов с помощью JAX-WS (Java API for XML Web Services) и включение новых технологий, таких как Java EE Connector Architecture и Java Persistence API 2.0.

*5. Java EE 6:*

- Выпущена в декабре 2009 года.

- Включает в себя новую спецификацию веб-профиля Java EE, который предоставляет более легковесный подход к разработке приложений.

- Включает новые возможности, такие как контексты и зависимости CDI (Contexts and Dependency Injection), улучшенную поддержку RESTful веб-сервисов и EJB 3.1.

*6. Java EE 7:*

- Выпущена в мае 2013 года.

- Включает новые возможности, такие как улучшенная поддержка HTML5, WebSocket API для разработки взаимодействующих в режиме реального времени приложений и новый API для работы с JSON (Java API for JSON Processing).

- Включает обновленные версии других технологий, таких как сервлеты, JAX-RS (Java API for RESTful Web Services) и JMS 2.0.

*7. Java EE 8:*

- Выпущена в сентябре 2017 года.

- Включает в себя новые возможности, такие **как поддержка микросервисной архитектуры** с помощью Eclipse MicroProfile, улучшения в API для работы с JSON, улучшенную поддержку CDI и обновленные версии других технологий.

*8. Jakarta EE 8:*

- В 2017 году Eclipse Foundation выпустила Jakarta EE 8, который был фактическим продолжением Java EE 8.

- Jakarta EE 8 оставался совместимым с Java EE 8 и включал все его технологии и спецификации.

- Обновление подтвердило переход управления платформой от Oracle к Eclipse Foundation.

*9. Jakarta EE 9:*

- В 2020 году была выпущена Jakarta EE 9, первая версия, которая отличалась от Java EE.

- Jakarta EE 9 была совместима с Java EE 8, но включала изменения, связанные с переименованием пакетов и классов для замены термина "javax" на "jakarta".

*10. Jakarta EE 10:*

- Выпущена в 2021 году.

- Jakarta EE 10 – первая версия, полностью свободная от ограничений Java EE, и включает обновленные версии технологий и спецификаций.

1. **Объясните понятия *containers* и *application servers*:**

***Сервер приложений*** (Weblogic, JBoss) – ПО, установленное на сервере и реализующее определенную спецификацию Java EE (например, Java EE 7). Сервер приложений должен быть способен запускать Java EE приложения.

Java EE определяет 4 контейнера, так называемые ***контейнеры***:

* Контейнер для апплетов (Applet container)
* Контейнер для клиентских приложений (Application client container)
* Веб-контейнер (Web container)
* Контейнер для EJB (Enterprise JavaBeans container)

Например, JBoss и WebLogic являются ***серверами приложений***, Tomcat и Jetty – ***веб-контейнерами***. Вот почему JBoss и WebLogic могут работать с большим количеством технологий, чем веб-контейнер. Сервер приложений содержит в себе весь функционал Java EE, тогда как контейнер – только конкретный небольшой функционал.

1. **Какие технологии включает *Java* EE? Приведите список технологий, их назначение и соответствующие спецификации, а также эталонную реализацию для каждой.**

**1. Servlet API:**

- Назначение: Обработка HTTP-запросов и генерация HTTP-ответов, вся серверная бизнес-логика

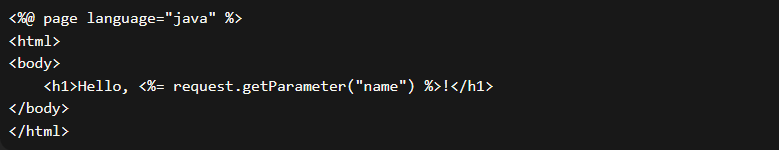
- Пример кода:



**2. JavaServer Pages (JSP):**

- Назначение: Создание динамических веб-страниц с помощью Java-кода и HTML-разметки, CSS-стилей и JS-кода.

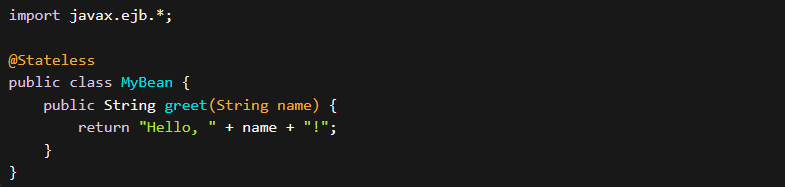
- Пример кода:



**3. Enterprise JavaBeans (EJB):**

- Назначение: Разработка компонентов, обеспечивающих бизнес-логику и управление транзакциями. Это позволяет разработчикам создавать повторно используемые и масштабируемые компоненты, которые могут быть развернуты и управляемы сервером приложений.

- Пример кода:



**4. Java Persistence API (JPA):**

- Назначение: Работа с ORM для доступа к базе данных. Он упрощает доступ и манипуляцию данными в базе данных, а также обеспечивает портативность и независимость от конкретной реализации базы данных.

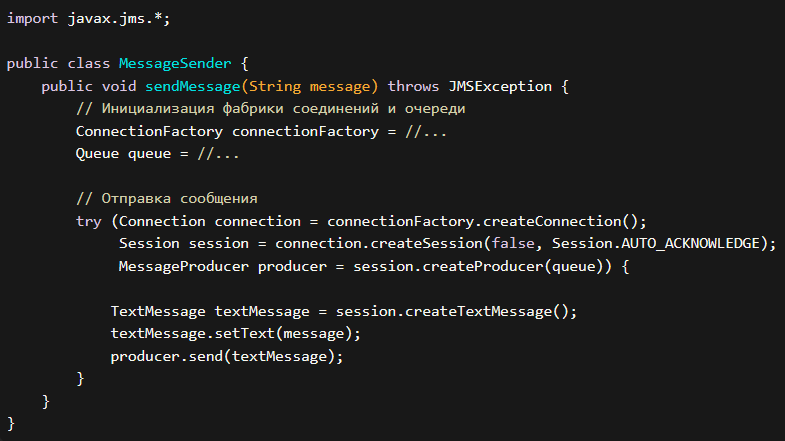
- Пример кода (сущность JPA):



**5. Java Message Service (JMS):**

- Назначение: Работа с асинхронной передачей сообщений между компонентами приложения.

- Пример кода (отправка JMS-сообщения):



1. **Объясните понятия *JCP* и *JSR:***

***JCP (Java Community Process)*** – механизм, предназначенный для разработки и стандартизации технологий Java.

Он представляет собой открытый процесс, в рамках которого члены Java-сообщества, включая компании, организации и отдельных разработчиков, могут вносить предложения и работать над разработкой новых спецификаций Java, а также обновлением и расширением существующих.

***JSR (Java Specification Request)*** – формальный запрос на создание новой спецификации Java или изменение существующей.

JSR описывает требования, цели и план разработки для конкретного проекта. Каждый JSR имеет уникальный идентификатор и назначается номер, который используется для идентификации и отслеживания процесса разработки.

1. **Источники**

[java - Difference between an application server and a servlet container? - Stack Overflow](https://stackoverflow.com/questions/5039354/difference-between-an-application-server-and-a-servlet-container)

[История версий Java EE — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9_Java_EE)

[Java Community Process — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Java_Community_Process)

[The Java Community Process(SM) Program - JSRs: Java Specification Requests - List of all JSRs (jcp.org)](https://www.jcp.org/en/jsr/all)

1. **Выводы**

В ходе выполнения лабораторной работы были рассмотрены различные версии Java EE, что позволило получить общее представление о эволюции и развитии этой платформы.

Понятия containers и application servers были деталеьно изучены, что позволило получить представление о том, как эти компоненты обеспечивают среду выполнения для разработки и развертывания Java EE приложений.

Был представлен обширный список технологий, включенных в Java EE, а также их назначение и соответствующие спецификации. Важно отметить, что каждая из этих технологий имеет пример эталонной реализации, что является ключевым фактором обеспечения совместимости и портативности приложений на платформе Java EE.

Важно отметить, что такие понятия, как JCP и JSR способствуют сотрудничеству и включению широкого сообщества разработчиков в развитие и стандартизацию Java.