## Учреждение образования

## Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

Тема

«Особенности технологий *Java EE*»

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Выполнил студент

ФИТ 4 курс 7 группа Бобрович Г.С.

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:

[Java EE](https://javarush.com/groups/posts/2637-vvedenie-v-java-ee) — это платформа, построенная на основе Java SE, которая предоставляет API и среду времени выполнения для разработки и запуска крупномасштабных, многоуровневых, масштабируемых, надежных и безопасных сетевых приложений.

* Какие версии *Java EE* существуют?
* J2EE 1.2 (Декабрь 1999)
* J2EE 1.3 (Сентябрь 2001)
* J2EE 1.4 (Ноябрь 2003)
* Java EE 5 (Май 2006)
* Java EE 6 (Декабрь 2009)
* Java EE 7 (Май)
* Java EE 8 (Август 2017)
* Jakarta EE 8 (Сентябрь 2019)

В 2017 году произошла новая веха в развитии платформы: Oracle передал контроль над развитием Java EE организации Eclipse Foundation. А в апреле 2018 года Java EE переименовали в Jakarta EE, которая полностью совместима с Java EE 8.

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

[Контейнеры](https://javarush.com/groups/posts/2637-vvedenie-v-java-ee) — это интерфейс между размещенными на них компонентами и низкоуровневыми платформо-независимыми функциональными возможностями, поддерживающими компонент.

Контейнеры предоставляют размещенным на них компонентам определенные службы. Например, управление жизненным циклом разработки, внедрение зависимости, параллельный доступ и т. д. Контейнеры скрывают техническую сложность и повышают мобильность.

[Java EE сервер](https://javarush.com/groups/posts/2637-vvedenie-v-java-ee) — это серверное приложение, которое реализует API-интерфейсы платформы Java EE и предоставляет стандартные службы Java EE. Серверы Java EE иногда называют серверами приложений. Данные сервера могут содержать в себе компоненты приложения, каждый из которых соответствует своему уровню в многоуровневой иерархии. Сервер Java EE предоставляет этим компонентам различные сервисы в форме контейнера.

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

|  |  |
| --- | --- |
| Технология | Назначение |
| Servlets | Java-классы, которые динамически обрабатывают клиентские запросы и формируют ответы (обычно HTML страницы). |
| Java Server Faces (JSF) | Фреймворк для построения веб приложений с пользовательским интерфейсом. Позволяет включать на страницу компоненты пользовательского интерфейса (например, поля и кнопки), преобразовывать и валидировать данные компоненты, а также сохранять эти данные в хранилищах на стороне сервера |
| Java Server Faces Facelets technology | Представляет из себя подтип приложения JSF, в котором вместо JSP страниц используются XHTML страницы |
| Java Server Pages (JSP) | Текстовые документы, которые компилируются в сервлеты. Позволяет добавлять динамический контент на статические страницы (например, HTML-страницы) |
| Java Server Pages Standard Tag Library (JSTL) | Библиотека тегов, в которой инкапсулирована основная функциональность в контексте JSP страниц. |
| Expression Language | Набор стандартных тегов, которые используются в JSP и Facelets страницах для доступа к Java EE компонентам. |
| Contexts and Dependency Injection for Java EE (CDI) | Представляет собой набор сервисов, предоставляемых Java EE контейнерами, для управления жизненным циклом компонентов, а также внедрения компонентов в клиентские объекты безопасным способом. |
| Java Beans Components | Объекты, которые выступают в роли временного хранилища данных для страниц приложения. |
| Enterprise Java Beans (enterprise bean) components | EJB — это управляемые компоненты, в которых заключена основная функциональность приложения. |
| JAX-RS RESTful web services | Представляет из себя API для разработки веб-сервисов, соответствующих архитектурному стилю REST. |
| JAX-WS web service endpoints | API для создания и использования веб-сервисов SOAP. |
| Java Persistence API (JPA) entities | API для доступа к данным в хранилищах данных и преобразования этих данных в объекты языка программирования Java и наоборот. |
| Java EE managed beans | Управляемые компоненты, которые предоставляют бизнес-логику приложения, но не требуют транзакционных функций или функций безопасности EJB. |
| Java Message Service | API службы сообщений Java (JMS) — это стандарт обмена сообщениями, который позволяет компонентам приложения Java EE создавать, отправлять, получать и читать сообщения. Что обеспечивает распределенную, надежную и асинхронную связь между компонентами. |
| The Java Database Connectivity API (JDBC) | Низкоуровневое API для доступа и получения данных из хранилищ данных. Типичное использование JDBC — написание SQL запросов к конкретной базе данных. |
| The Java Persistence API | API для доступа к данным в хранилищах данных и преобразования этих данных в объекты языка программирования Java и наоборот. Гораздо более высокоуровневое API по сравнению с JDBC. Скрывает всю сложность JDBC от разработчика под капотом. |
| The Java EE Connector Architecture | API для подключения других корпоративных ресурсов, таких как:  ERP (англ. Enterprise Resource Planning, система планирования ресурсов предприятия),  CRM (англ. Customer Relationship Management, система управления взаимоотношениями с клиентами). |
| The Java Transaction API (JTA) | API для определения и управления транзакциями, включая распределенные транзакции, а также транзакции, затрагивающие множество хранилищ данных. |

[API](https://skillbox.ru/media/code/chto_takoe_api/) (англ. Application Programming Interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

Все эти взаимодействия происходят с помощью функций, классов, методов, структур, а иногда констант одной программы, к которой обращаются другие. Это основной принцип работы API.

* Объясните понятия *JCP* и *JSR.*

[Java Community Process (JCP)](https://ru.wikipedia.org/wiki/Java_Community_Process) — сформированный в 1998 году формальный процесс, который позволяет заинтересованным лицам участвовать в формировании будущих версий спецификаций платформ языка Java.

Основа JCP — так называемые JSR (Java Specification Request/Запрос на Спецификацию Java), формальные документы, описывающие спецификации и технологии, которые предлагается добавить к Java платформе.

Разработка JSR включает следующие шаги:

* Предложение (Initiation). Спецификация предлагается членами сообщества, и, путём голосования, принимается Исполнительным Комитетом (JCP Executive Committee). В это время JSR, присваивается номер.
* Черновик для сообщества (Community Draft). После принятия JSR, формируется экспертная группа, которая разрабатывает первый черновик спецификации. Этот черновик выносится на рассмотрение Исполнительного Комитета и членов сообщества. По итогам обсуждения выносится решение о том, переходить к следующей стадии, или необходим новый доработанный Community Draft.
* Открытый черновик (Public Draft). Черновик, дошедший до этой стадии, выкладывается в свободный доступ, и любой человек, имеющий доступ к сети Интернет, может рассмотреть его и высказать своё мнение. Экспертная группа использует эти отзывы для дальнейшего совершенствования спецификации. После этого доработанная спецификация выносится на голосование Исполнительного Комитета для окончательного утверждения. К этому времени становятся доступными Эталонная реализация (reference implementation) и набор автоматических тестов Technology Compatibility Kit (TCK).
* Поддержка (Maintenance). Окончательная версия спецификации, эталонной реализации и TCK, обновляется по мере поступления запросов на доработку, уточнение и расширение. Некоторые запросы могут потребовать пересмотра спецификации экспертной группой и выпуска новой версии, или даже формирования нового JSR.

Приведите краткие выводы по содержанию работы.

Вывод: были изучены особенности технологий Java EE, ее история версий и виды технологий.