**Вот десять идей проектов для портфолио Java-разработчика:**

1 Простое веб-приложение CRUD (создание, чтение, обновление, удаление) с использованием популярной веб-инфраструктуры Java, такой как Spring или JavaServer Faces (JSF).

2 Простая веб-служба RESTful, которая позволяет клиентам создавать, извлекать, обновлять и удалять ресурсы.

3 Небольшое приложение электронной коммерции, которое позволяет пользователям просматривать продукты, добавлять товары в корзину и размещать заказы.

4 Простое приложение для чата, использующее WebSockets и платформу обмена сообщениями Java, такую как RabbitMQ или ActiveMQ.

5 Простая игра, такая как «Крестики-нолики» или «Виселица», демонстрирующая понимание игровой логики и разработку пользовательского интерфейса.

6 Простой инструмент визуализации данных, который позволяет пользователям исследовать и анализировать данные с помощью такой библиотеки, как D3.js или Chart.js.

7 Простой парсер, извлекающий информацию с веб-сайта и сохраняющий ее в базе данных или файле.

8 Простое приложение машинного обучения, которое демонстрирует понимание популярной библиотеки машинного обучения Java, такой как Weka или MLlib.

9 Простое настольное приложение, демонстрирующее понимание разработки настольных приложений с использованием JavaFX или Swing.

10 Простое мобильное приложение, демонстрирующее понимание разработки мобильных приложений с использованием Java для Android.

Вот некоторые из проектов, которые вы можете включить в свое портфолио, чтобы продемонстрировать свои навыки Java-разработчика среднего уровня. Конечно, конкретные проекты, которые вы выберете, будут зависеть от ваших интересов и навыков, которые вы хотите выделить, но они должны стать хорошей отправной точкой.

**1. Простое веб-приложение CRUD:**

а. Регистрация пользователей и вход в систему.

б. Возможность создавать, читать, обновлять и удалять записи в базе данных.

в. Реализуйте разбиение на страницы для отображения больших наборов данных.

д. Реализовать функции фильтрации и поиска.

е. Реализуйте адаптивный и удобный дизайн с помощью CSS-фреймворка.

Одной из идей веб-приложения CRUD может быть простое приложение диспетчера задач. Приложение позволит пользователям создавать, читать, обновлять и удалять задачи.

Основные функции приложения будут включать в себя:

а. Регистрация пользователя и вход в систему: пользователи смогут создать учетную запись и войти в приложение.

б. Возможность создавать, читать, обновлять и удалять записи в базе данных: пользователи смогут создавать новые задачи, просматривать существующие задачи, обновлять задачи и удалять задачи.

в. Реализуйте разбивку на страницы для отображения больших наборов данных: приложение сможет обрабатывать и отображать большое количество задач с помощью разбивки на страницы, что позволит пользователям перемещаться по задачам по страницам.

д. Реализовать функции фильтрации и поиска: приложение позволит пользователям фильтровать и искать задачи по различным критериям, таким как статус задачи, срок выполнения или ключевое слово.

е. Реализуйте адаптивный и удобный дизайн с помощью CSS-фреймворка: интерфейс приложения должен быть удобным и адаптивным, чтобы оно хорошо работало как на настольных компьютерах, так и на мобильных устройствах.

Это простое, но полезное приложение, которое будет охватывать основные функции, необходимые для веб-приложения CRUD, оно продемонстрирует ваши навыки создания интерактивного, удобного веб-приложения, взаимодействующего с базой данных. А также станет хорошим дополнением к любому портфолио.

**2. Простой веб-сервис RESTful:**

а. Реализуйте конечные точки для создания, извлечения, обновления и удаления ресурсов.

б. Используйте соответствующие методы HTTP (например, GET, POST, PUT, DELETE) для каждой конечной точки.

в. Реализовать проверку и обработку ошибок для входящих запросов.

д. Реализуйте правильное использование кодов состояния HTTP для указания результатов запросов.

е. Используйте популярную библиотеку Java, такую как Jersey или Spring Web, для создания службы.

**3. Небольшое приложение для электронной коммерции:**

а. Внедрите каталог продуктов, в котором пользователи могут просматривать доступные продукты.

б. Реализуйте функцию корзины, которая позволяет пользователям добавлять и удалять товары.

в. Реализуйте процесс заказа, который позволит пользователям приобретать товары.

д. Внедрите безопасный платежный шлюз для обработки транзакций.

е. Внедрите панель администратора для управления продуктами, заказами и информацией о клиентах.

**4. Простое приложение для чата:**

а. Внедрите WebSockets для связи между пользователями в режиме реального времени.

б. Используйте платформу обмена сообщениями Java, такую как RabbitMQ или ActiveMQ, для обработки очередей сообщений.

в. Реализовать аутентификацию и авторизацию пользователей.

д. Реализовать возможность создавать и присоединяться к различным чатам.

е. Реализуйте отзывчивый и удобный интерфейс с помощью библиотеки JavaScript, такой как React или Angular.

**5. Простая игра:**

а. Реализуйте игровую логику и правила.

б. Реализуйте пользовательский интерфейс для игры с помощью JavaFX или Swing.

в. Внедрите систему подсчета очков и возможность отслеживать высокие баллы.

д. Реализовать меню настроек для настройки параметров игры.

е. Реализуйте простой ИИ-плеер для одиночного режима.

**6. Простой инструмент визуализации данных:**

а. Реализуйте способ импорта данных из разных источников, таких как файлы CSV, Excel или JSON.

б. Используйте библиотеку JavaScript, например D3.js или Chart.js, для создания различных типов визуализации, таких как линейная, гистограмма, круговая диаграмма и точечная диаграмма.

в. Реализуйте способ взаимодействия с визуализацией, например масштабирование, панорамирование или наведение.

д. Реализовать адаптивный дизайн и удобный интерфейс.

е. Реализуйте способ экспорта визуализации в формат изображения или PDF.

**7. Простой парсер:**

а. Реализуйте способ извлечения информации с определенного веб-сайта с помощью библиотеки очистки Java, такой как jsoup или Selenium.

б. Реализуйте способ хранения информации в базе данных или файле.

в. Реализуйте способ планирования очистки через определенные промежутки времени.

д. Реализовать способ обработки нумерации страниц и бесконечной прокрутки страниц.

е. Реализуйте способ обработки ошибок и обработки различных сценариев, таких как изменение структуры веб-сайта.

**8. Простое приложение машинного обучения:**

а. Реализуйте способ загрузки набора данных из файла или URL-адреса.

б. Реализуйте способ выбора алгоритма классификации или регрессии из популярной библиотеки машинного обучения Java, такой как Weka или MLlib.

в. Реализуйте способ обучения и оценки модели.

д. Реализуйте способ делать прогнозы, используя обученную модель.

е. Реализовать способ визуализации результатов в удобной для пользователя форме.

**9. Простое настольное приложение:**

а. Реализуйте удобный графический пользовательский интерфейс с помощью настольной библиотеки Java, такой как JavaFX или Swing.

б. Реализуйте способ взаимодействия с базой данных или внешним API для извлечения данных или управления ими.

в. Реализуйте способ обработки пользовательского ввода и событий.

д. Реализуйте способ создания, чтения, обновления и удаления записей из базы данных или файла.

е. Реализуйте способ экспорта или печати данных или результатов.

**10. Простое мобильное приложение:**

а. Реализуйте удобный графический пользовательский интерфейс с помощью Java для Android.

б. Реализуйте способ взаимодействия с удаленной базой данных или API для извлечения данных или управления ими.

в. Реализуйте способ обработки пользовательского ввода и событий.

д. Реализуйте способ доступа к функциям устройства, таким как камера, GPS и датчики.

е. Реализовать способ обработки автономного режима и синхронизации данных.