**ЛАБОРАТОРНАЯ РАБОТА №10**

**РАЗРАБОТКА СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ**

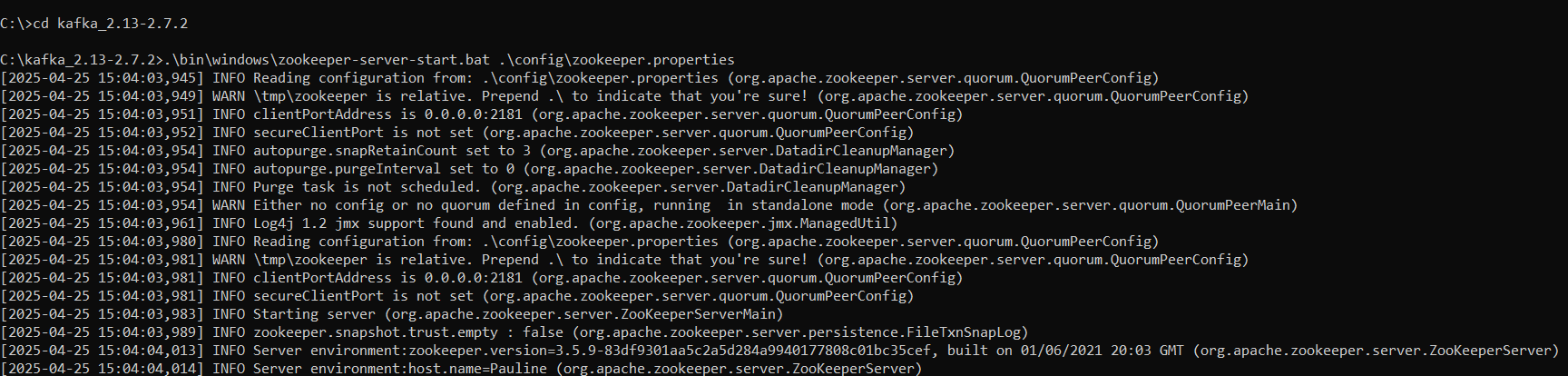
**С ИСПОЛЬЗОВАНИЕМ APACHE KAFKA**

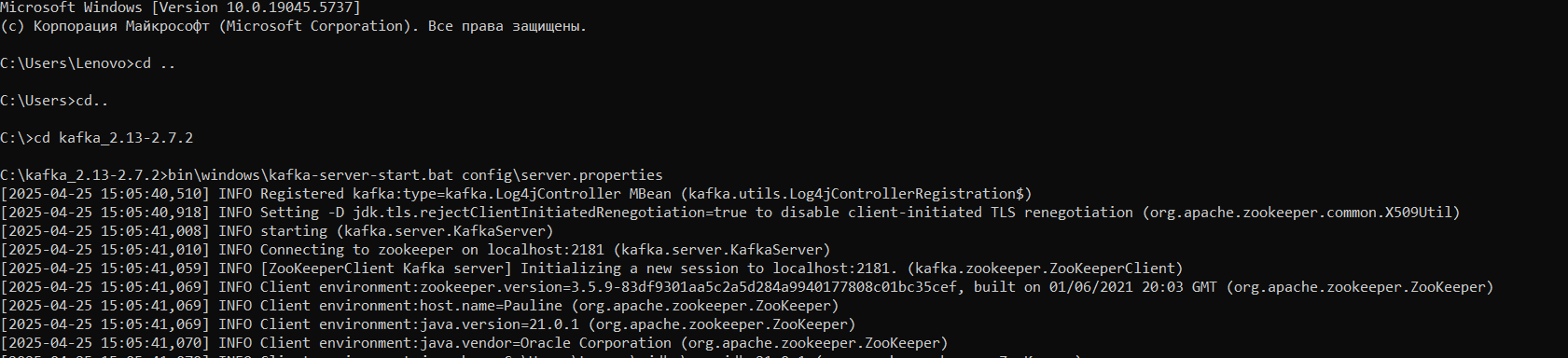
**Цель работы:** Разработка системы асинхронной обработки сообщений с использованием Apache Kafka, включая создание продюсеров, консьюмеров и настройку их взаимодействия.

**ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

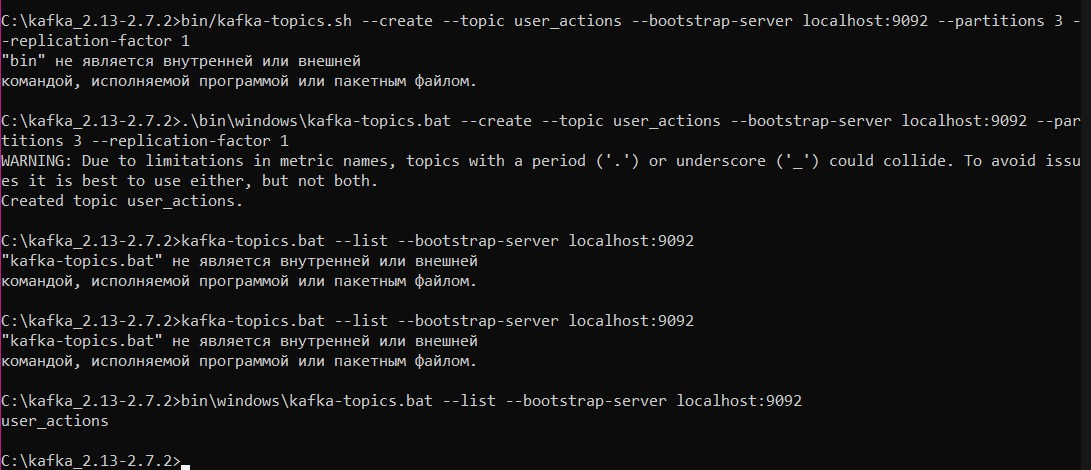
1. Настройка и запуск Kafka:

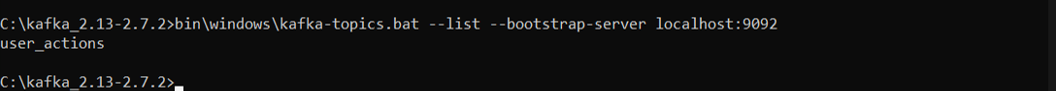
• Инициализируйте ZooKeeper и запустите Kafka-брокер, используя локальную среду или контейнер Docker.





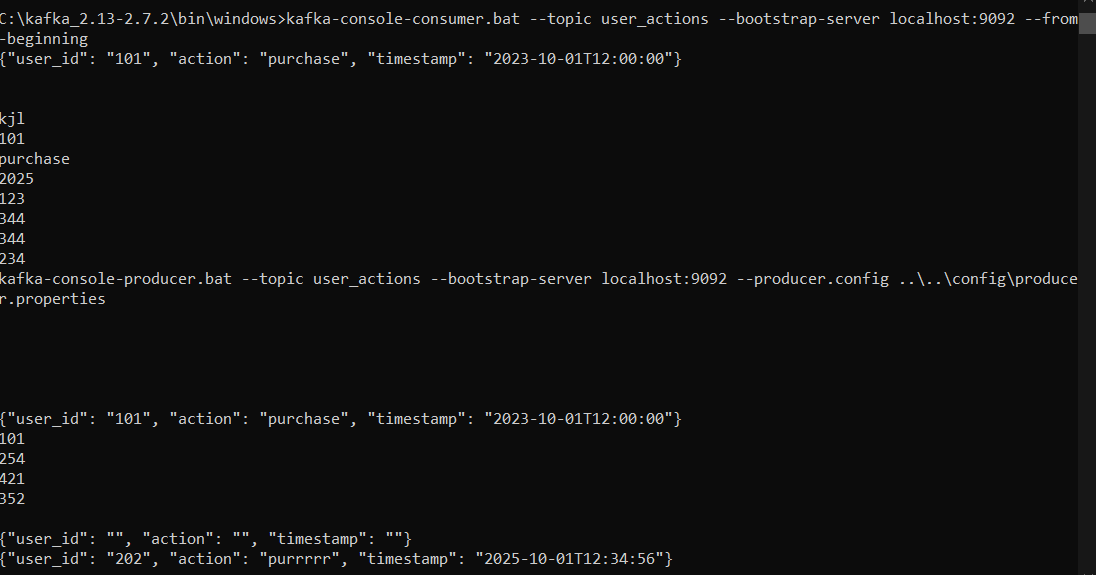
• Инициализируйте топик user\_actions, разделив его на три партиции.



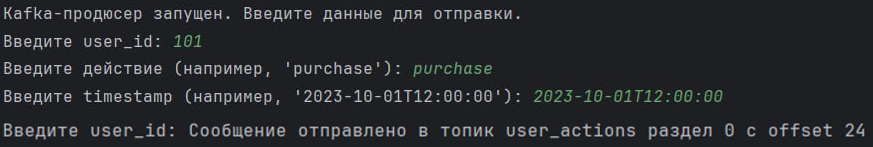


2. Создание продюсера:

• Разработайте приложение на Java или Python, которое публикует JSON-сообщения в заданный топик. Пример структуры сообщения: {"user\_id": 101, "action": "purchase", "timestamp": "2023-10-01T12:00:00"}.

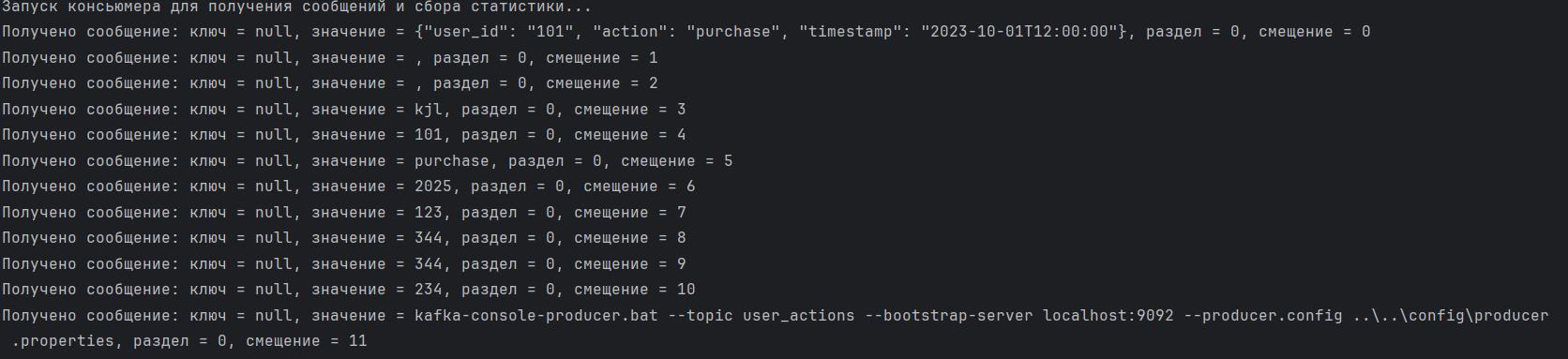


• Обеспечьте ввод сообщений через консоль или загрузку из файла.

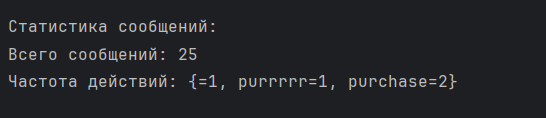


3. Разработка консьюмера

* Реализуйте обработчик сообщений из топика с возможностью фильтрации по типу действия, например, "purchase".



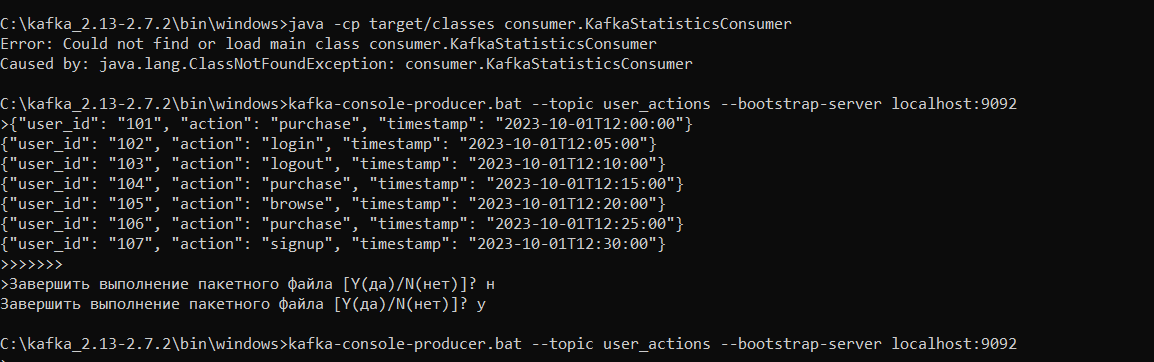
* Отображение статистики в консоли, включая количество обработанных сообщений и наиболее распространенные действия.



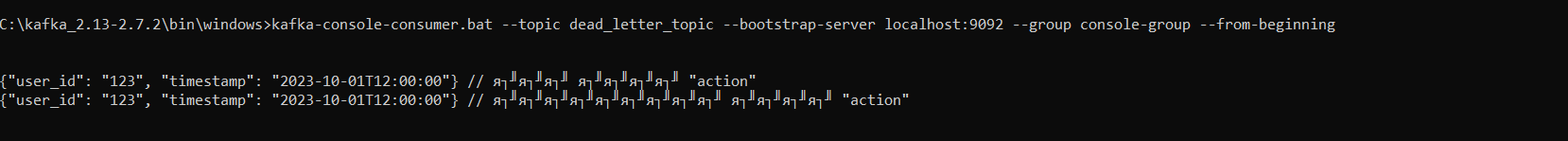
4. Управление ошибками и масштабирование:

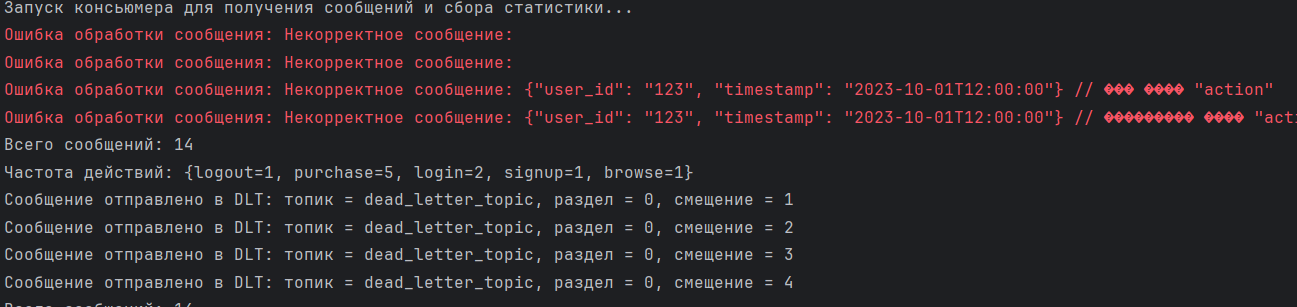
• Настройте Dead Letter Topic (DLT) для обработки некорректных сообщений.

• Запустите два консьюмера в одной группе и продемонстрируйте, как сообщения распределяются между ними.



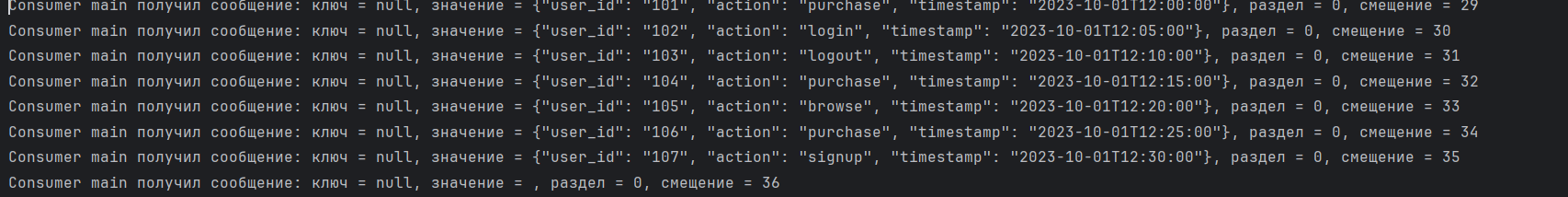
• Запустите два консьюмера в одной группе и покажите, как происходит распределение сообщений между ними.

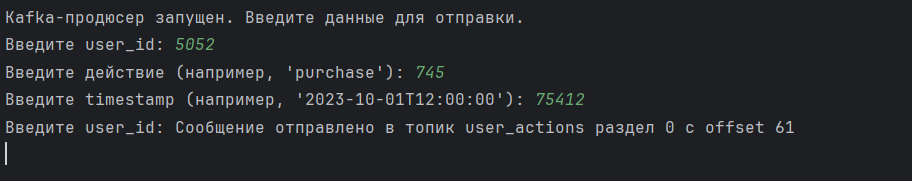


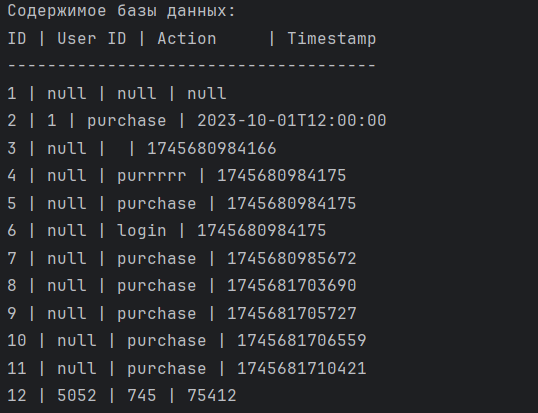


5. Подключение к внешней системе

• Сохраняйте данные в базе данных PostgreSQL или MySQL.







**Преимущества Kafka**

1. Масштабируемость – высокая производительность, позволяющая обрабатывать миллионы сообщений в секунду благодаря разделению на партиции.
2. Надёжность – сообщения сохраняются на диск, предотвращая их потерю при сбоях.
3. Многопоточность – поддержка нескольких продюсеров и потребителей для равномерного распределения нагрузки.
4. Гибкость – универсальность в применении: сбор логов, потоковая аналитика, обработка данных в реальном времени, взаимодействие микросервисов.
5. Простая интеграция – совместимость с различными языками программирования и системами, такими как Spark, Hadoop, базы данных.

**Возникшие сложности**

1. Настройка инфраструктуры – требует понимания архитектуры Kafka, включая ZooKeeper, брокеров и разделы.
2. Управление группами потребителей – требует грамотной настройки распределения сообщений и контроля смещений (offset).
3. Обработка ошибок – сложность конфигурации Dead Letter Topic (DLT) при некорректной настройке продюсера или механизма обработки ошибок.
4. Интеграция с внешними сервисами – например, подключение SQLite требует дополнительных зависимостей и продуманной логики взаимодействия.
5. Ребалансировка групп – частые изменения состава группы могут приводить к задержкам и снижению производительности.
6. Парсинг данных – сложности при обработке JSON-сообщений, если данные не стандартизированы.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. **Зачем Kafka использует ZooKeeper?**  ZooKeeper управляет состоянием кластера Kafka, координирует распределение партиций между брокерами и поддерживает согласованность и отказоустойчивость.

2. **Как обеспечить порядок сообщений в партиции?**  Используйте единого продюсера с постоянным ключом для сообщений. Это гарантирует, что все сообщения с этим ключом попадут в одну партицию и будут обработаны строго в порядке их отправки.

3. **В чём разница между auto.offset.reset=earliest и latest?** earliest позволяет начинать чтение с самого раннего доступного сообщения, а latest – с самых новых данных. Если смещение отсутствует, earliest предоставляет доступ к первым сообщениям в партиции, тогда как latest начинает обработку только тех данных, которые поступают после подключения потребителя.

4. **Как Kafka обеспечивает отказоустойчивость?** Система использует механизм репликации партиций между брокерами, что обеспечивает сохранность данных. В случае сбоя одного брокера другой, имеющий копию партиции, продолжает обработку сообщений.