Java Technologies - Lab 7

[valid 2025-2026]

Messaging
We continue our project.

Compulsory (1p)

- Install a messaging broker such as Kafka, ActiveMQ, or RabbitMQ.
- Create another Spring Boot project (QuickGrade) dedicated to student grades.
- The QuickGrade application must publish events to a topic, containing (student code, course code, grade).
- PrefSchedule will consume the messages and print them to the console.

Homework (2p)

- Create another table in the PrefSchedule application containing student grades.
- The grades received as events from QuickGrade must be stored in the database - only those related to compulsoy courses.
- Create REST endpoints for getting the grades and loading them from a CSV file.
- Implement a Dead-Letter Queue (DLQ) handler for failed messages and test retry semantics.

Advanced (2p)

- Implement a pipeline of message processing: QuickGrade publishes basic grade events, a second component adds student name and year, and a third component adds course name, and semester, before PrefSchedule consumes it.
- Use advanced feature of the message broker in order to improve scalability. For example, use Kafka partitions and consumer groups or experiment with "exactly-once" semantics, and measure processing throughput using Kafka metrics.