Lab 11:

Exercise 1: Spring webflux

Write a reactive REST stock service that generates a new stock value every 5 seconds.

Write a reactive REST client that subscribes to this stock service and prints the stock value to the console.

You can schedule methods in Spring Boot with the following annotations:

```
@SpringBootApplication
@EnableScheduling
public class SpringBootSchedulingApplication {

  public static void main(String[] args) {
    SpringApplication.run(SpringBootSchedulingApplication.class, args);
  }
}
```

Enable scheduling

```
@Component
public class WelcomeTask {

  @Scheduled(fixedRate = 5000)
  public void welcome() {
    Date date = Calendar.getInstance().getTime();
    DateFormat timeFormatter = DateFormat.getTimeInstance(DateFormat.DEFAULT);
    String currenttime = timeFormatter.format(date);
    System.out.println("This task runs at " + currenttime);
  }
}
```

Run every 5 seconds

```
This task runs at 12:07:50
This task runs at 12:07:55
This task runs at 12:08:00
```

Exercise 2: CQRS.

Visit the following url: https://www.amazon.com/Domain-Driven-Design-Tackling-Complexity-Software-ebook/dp/B00794TAUG/ref=mt_kindle?_encoding=UTF8&me=&qid=1531190547

Suppose you need to design the BookCatalog service for Amazon. On the website we have about 10 buyers for every 200 viewers.

We reduce the scope of the BookCatalog to the following requirements:
For every book we show the publisher information
For every book we show the reviews
For every book we show other sellers (partner sellers)
For every book we show author information
For every book we show how many are in stock (extra requirement for this exercise)

For every book we show recommended books that customers bought when they viewed this book on the website
For every book we show recommended books that customers bought when they bought this book

The BookCatalog service has the following business rule:

- Every review for a book needs to be from a unique customer. A book cannot have multiple reviews from the same customer

You decided to apply the CQRS pattern to the BookCatalog. Draw the class diagram(s) for the Amazon BookCatalog. In the class diagram(s), show only the Service classes and their methods and the domain classes with their attributes and relationships (no methods). Apply the best practices you learned in this course.

Exercise 3: Webshop overview.

a. Suppose you have to explain the webshop that you have build so far to your colleague developers and to your boss. Draw in one overview diagram all aspects of the webshop that are important to communicate. Your boss and your fellow developers are not interested in the internal details of the webshop services. They are interested in the different services that you use, and how they work together. (Do not draw a detailed class diagram)
b. Draw a sequence diagram that shows the scenario of starting all services for the webshop. On the sequence diagram only show the services involved (not objects)
c. Draw a sequence diagram that shows the scenario of the following steps:

1. Add product to shoppingcart
2. Checkout shoppingcart
3. Add customer to order
4. Confirm the order

On the sequence diagram only show the services involved (not objects)