Exercise 8:
OpenAPI Specification of Spring REST Controller

Submission in AULIS by 15.01.26, 9:45am

Assignment 1: OpenAPI Specification of Spring REST Controller

Prospective clients of your Order REST controller need information on possible ways to use your service! Your task is to create / generate an OpenAPI specification that is as complete as possible and to derive a *Swagger UI* documentation out of it. Think of

- General API information (title, description, version, contact information, server(s))
- Model / schema information (*Order* and *LineItem* classes (or matching DTOs) with attributes, description if not self-explaining, and information on required attributes)
- For each route of your controller:
  - URI + HTTP method
  - Summary + description
  - Input parameters (name, data type, required?, description, type (path or body), references to schema)
  - Return values (description, reference to schema)
  - Media types consumed (→ input parameters) or produced (→ return values)
  - Responses (status code, description, reference to schema)

Example (excerpt, JSON):

```
"/products/{id}": {
      ...,
      "put": {
        "summary": "Update product",
        "description": "Update existing product with given id.",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "description": "id",
            "required": true,
            "type": "integer",
            "format": "int32"
```

```
      },
      {
        "in": "body",
        "name": "updatedProduct",
        "description": "updated product",
        "required": true,
        "schema": {
          "$ref": "#/definitions/Product"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "The updated or newly added product.",
        "schema": {
          "$ref": "#/definitions/Product"
        }
      },
      "404": {
        "description": "Product with requested id not found."
      }
    }
  },
  ...
```

You may choose one of the following approaches:

1.) Generation of OpenAPI specification using annotations

This approach assumes that you have already implemented a REST controller which is to be documented afterwards by adding OpenAPI annotations to your REST controller and additionally providing general information on your API (servers, version, contact etc.) in a new configuration class.

2.) Generation of REST controller API and Swagger UI based on OpenAPI specification

- Write an OpenAPI specification (e.g. in YAML notation) for your REST service.
- Generate a Java interface (your API) from that specification. This interface will not only contain annotated Java methods, but also required model classes. These might be your DTOs and should be named properly.
- Modify your (existing) REST controller such that it implements the generated API.

For this approach, you will need to get acquainted with the *openapi-generator-maven-plugin* (https://github.com/OpenAPITools/openapi-generator/tree/master/modules/openapi-generator-maven-plugin).

Both approaches required to have the *SpringDoc* framework (https://springdoc.org/) generate both the OpenAPI specification and the Swagger UI.

Submission:

- Submit your REST controller with OpenAPI annotations as *one* upload to AULIS.
- Please include
    - a Postman collection of REST requests (same or improved version of exercise sheet 6)
    - a README file with information on
        - how to start your REST controller (it will probably require to start RabbitMQ via docker?)
        - and on the URLs for accessing the OpenAPI specification (JSON or YAML) and a derived visual *Swagger UI* documentation.