Follow-up Assignment:

Building a gRPC Server for Encoded Circle Data

Part 01

December 9, 2025

1. Create a new project with Gradle as the build system.
   - Open the file `build.gradle.kts` and replace its contents with the contents of the `build.gradle.kts` file attached to this assignment.
   - While still in `build.gradle.kts`, run "Sync Gradle Changes" (*Ctrl+Shift+O*).
   - Within the project structure, go to the directory `src->main->java` and create a subdirectory `proto`.
   - Copy into the `proto` directory the file attached to the assignment named `circles.proto`.
   - In the directory `src->main->java` create the package `pj`, then copy into it the file `GrpcServer.java`.
   - In the directory `src->test->java` create the package `pj`, then copy into it the file `GrpcServerTest.java`.
   - Prepare an H2 database with exactly the same parameters as in exercise 10-01 and load into it the circles from the previously created `circles.bin` file.
2. Place the file `GrpcServer.java` in the directory `src->main->java` in the package `pj`, and then complete its *TODO* sections.
   - In the `main` method, create a gRPC server by initializing an object of type `Server` using the class `ServerBuilder`.
     - Use the method `ServerBuilder.forPort(PORT)` to specify the port number on which the server will listen.
     - Add the gRPC service implementation by calling `addService(new CircleServiceImpl())`.
     - Finish the configuration by calling `build()` to create a ready server instance.
   - Print a message informing that the server has started, including the port number.
     - Use `System.out.println` to inform the user that listening has started.
   - Start the server by calling the method `server.start()` to begin handling connections.
     - This step activates listening on the given port and enables handling gRPC client requests.
   - Block the main program thread while waiting for the server to shut down.

  - Call the method `server.awaitTermination()` to prevent the program from exiting before the server is stopped.
- Review the class `CircleServiceImpl`, in particular the `getCircles` method.

3. Go to the file `circles.proto` and complete the body of the service with the method header that needs to be handled.
   - In the `service` section, after the *TODO* marker, insert the statement `rpc GetCircles (Empty) returns (CircleResponse);`.
   - Next, open the `Gradle` tool window on the left side of the IntelliJ window, expand the *Tasks* menu, then *build*, and select the *build* option.
4. Then run the unit tests verifying the correctness of the server, contained in the file `GrpcServerTest.java`, and present the positive results to the instructor.