Follow-up Assignment:

Consuming Circle Data from a gRPC Service in a Swing Client

Part 02

December 9, 2025

1. Place the files `MyFrame.java` and `PositionAndColor.java`, attached to this assignment, in the package `pj` (*src->main->java*).

2. In the file `MyFrame.java`, complete the method `loadFromGrpc()` in the place marked with the `TODO` comment.

   - Create a gRPC communication channel, configuring it to connect to a locally running server.
     - Use `ManagedChannelBuilder.forAddress("localhost", 50051)` to specify the address and port of the server.
     - Enable unencrypted communication mode by calling `usePlaintext()`.
     - Finish the configuration by calling `build()`, which will create an instance of the channel `channel`.
   - Create a blocking stub for the `CircleService` service, allowing synchronous RPC calls.
     - Use `CircleServiceGrpc.newBlockingStub(channel)` to build a client bound to the created channel.
   - Invoke the remote method `getCircles` on the stub, passing an empty input message.
     - Create an `Empty` object using `Empty.newBuilder().build()` and pass it to the stub method.
     - Receive the result in an object `CircleResponse response`.
   - Prepare a temporary list for the encoded circle values that will be reconstructed from the data received from the server.
     - Create a `java.util.ArrayList<Integer>` and name it `tempList`.
   - Iterate over all `CircleDTO` objects received in the response and transform each one into encoded form.
     - Read the values of the fields `x`, `y` and the color components `r`, `g`, `b`.
     - Create a `Color` object from the RGB components.
     - Pass the coordinates and color to the method `PositionAndColor.encode` to obtain a single numeric value representing a circle.

- Add the result to `tempList`.

- After completing the iteration, convert the list of values to an `int[]` array intended for further processing.
  - Use a stream from `tempList` and the method `mapToInt(i -> i).toArray()` to obtain the array `finalData`.

3. Run the program and test its behavior. Present the obtained results to the instructor.