FAQ
AI-based language migration from COBOL to Java

1. Can a legacy software system be migrated using just one AI tool?

No. Rather than one migration tool, several tools are needed in a migration project. Common AI tools merely focus on programme code conversion. In a migration project, however, further artefacts need to be migrated as well, e. g. JCL, files, databases, middleware and screens. It requires several specialised tools to automate the migration process.

2. AI tools might provide support for the actual programme conversion, e. g. COBOL to Java, though?

COBOL, for example, includes a complex data type concept for which no 1:1 mapping exists in Java. In addition, COBOL contains 'language specialities' such as overlaying variables of different data types and others. Our experiments with various AI tools have shown that they essentially map COBOL data types to Java data types. The converted Java programmes are executable, but deliver incorrect results. In a migration project, however, semantic equivalence between the old and new systems is the most important factor. Of course, these tools can be trained accordingly. However, achieving a complete mapping from COBOL to Java requires a great deal of effort, which, according to our findings, is close to that of a new development. In our tools, these mappings are already available in the form of a Java library, which currently achieves a 95–99% coverage. To achieve this level of coverage in migration projects, years (even decades) of development were required. According to our experience, AI tools specifically trained for language conversion, e. g. COBOL to Java, do not achieve this level of coverage. This then requires manual intervention and extensions in the converted Java programmes, inevitably leading to a major manual effort and susceptibility to errors.

3. Can AI tools migrate procedural COBOL code directly into object-oriented Java code? I read something like that on the internet.

In the last years (decades), a number of scientific papers on the automatic conversion of procedural code into object-oriented code (independent of migration) has been published. To our knowledge, none of these papers has produced a result that is satisfactorily practical. If such methods do not exist, there can be no tool that automatically implements such algorithms – regardless of whether it is an AI tool or not. Of course, future developments of the migrated system can be object-oriented, so that over time, object orientation increasingly dominates, also through further refactoring of the migrated programme system in maintenance/further development. This is confirmed, for example, in projects that we are maintaining and further developing on behalf of customers after a successful migration.

pro et con
Innovative Informatikanwendungen GmbH
Reichenhainer Straße 29a
09126 Chemnitz

Telefon: +49 371 270951-0
Telefax: +49 371 270951-29

Internet: www.proetcon.de
E-Mail: info@proetcon.de

4. What support can AI tools currently provide in a migration project?

Currently, AI tools might offer a range of useful applications in migration projects, e. g. test support, code commenting, refactoring and programme documentation. However, only as assistants. As regards IBM watsonx Code Assistant for Z, this functionality is already embedded in the name.

If a customer wants to change the programming language and/or platform within a manageable project timeframe and budget, a tool-supported, compiler-based software migration currently is the method of choice. In this process, AI tools could provide valuable, assistance-based support.

pro et con
Innovative Informatikanwendungen GmbH
Reichenhainer Straße 29a
09126 Chemnitz

Telefon: +49 371 270951-0
Telefax: +49 371 270951-29

Internet: www.proetcon.de
E-Mail: info@proetcon.de