This thesis aimed at creating a software library enabling interoperability between C# and Java programming languages by enabling the use of a subset of Java features from C# code. Requirements that the library should meet were determined via analysis of potential use case scenarios and similar existing implementations. The emphasis was put on providing a user-friendly and type-safe API.

The implemented solution enables the invocation of static and instance Java methods from C#. It also allows creating Java object instances from C# code or obtaining them as return values of invoked Java methods. C# represents Java object instances as proxy classes that emulate the API of corresponding Java classes, possibly including an inheritance hierarchy between them and the interfaces they implement. Implementation of C# proxy types (classes and interfaces) is generated at compile time via the incremental source generator.

Generated proxy types are based on API provided by the implemented interoperability library. This library is mainly based on a combination of Java Native Interface (JNI) and .NET Platform Invoke (P/Invoke). However, the solution also experiments with optimizing certain invocation kinds using Foreign Function API provided by the recent Java Project Panama.