

frgaal.org compiler

This is a compiler for a Java-like language:

```
import java.util.List;
import java.util.Arrays;

public class Main {
    private static List<Integer> useVar() {
        var list = Arrays.asList(6, 1, 3, 5);
        useTextBlock(list);
        return list;
    }

    private static void useTextBlock(List<Integer> list) {
        String text = ""
            initial list content
            is..."";
        System.err.println(text + list);
    }

    private static String useInstanceOf(List<?> list) {
        final Object element = list.get(1);
        if (element instanceof Integer number) {
            return useSwitchExpr(number);
        }
        return "not a number!";
    }

    private static String useSwitchExpr(int number) {
        return switch (number) {
            case 3 -> "ok";
            default -> "bad";
        };
    }

    public static void main(String... args) {
        List<Integer> list = useVar();
        list.sort(null);
        System.err.println("after sorting: " + list);
        String switchTest = useInstanceOf(list);
        System.err.println(switchTest);
    }
}
```

The features of the language are similar to features of the Java language, but unlike in Java, many of them can be compiled to run on Java 8.

Supported Features

When targetting Java 8 or later (i.e. using `-target 10` or later), all Java 8 language features are supported. In addition to this the following features are supported:

- "var" local variables, introduced in Java 10. Must use `-source 10` to enable.
- switch expressions, introduced in Java 14. Must use `-source 14` to enable.
- text blocks, introduced as preview in Java 13. Must use `-source 14 --enable-safe-preview` to enable, classfiles will be usable on the target version specified by `-target <version>`.
- pattern matching in instanceof, introduced as preview in Java 14. Must use `-source 14 --enable-safe-preview` to enable, classfiles will be usable on the target version specified by

```
-target <version>.
```

Usage with Maven

To use this compiler, specify following in your pom.xml file build section:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <dependencies>
        <dependency>
          <groupId>org.frgaal</groupId>
          <artifactId>compiler-maven-plugin</artifactId>
          <version>14.0.1</version>
        </dependency>
      </dependencies>
      <configuration>
        <compilerId>frgaal</compilerId>
        <source>14</source>
        <target>1.8</target>
        <compilerArgs>
          <arg>-Xlint:deprecation</arg>
          <arg>--enable-safe-preview</arg>
        </compilerArgs>
      </configuration>
    </plugin>
  </plugins>
</build>
```

With such a change the compiler of your project no longer depends on the used JDK. All the compiler code is downloaded from Maven central and it can run on anything from JDK8 up. If you want to update your compiler to get a bugfix or to use latest language feature, you can change to something newer. However, until you do that, no matter what breaking changes appear in the JDK, your project is still going to compile into exactly the same .class files.

Usage on command line

To use the frgaal compiler, run it as follows:

```
$ java -jar compiler.jar
```

When running through reflection, run the org.frgaal.Main class.

Preview Features

Some features are preview features in Java. These generally require command line parameters `--enable-preview -source <version>`, where `<version>` is the version of the JDK that is compiling the sources. The resulting classfiles are marked as preview, and can only be used in conjunction with `--enable-preview` and a matching JDK version. Preview features may change incompatibly between JDK versions.

This mode is supported by the frgaal compiler, but there is also a safe preview mode provided. Not all preview features have this safe preview mode enabled, though. The safe preview mode is enabled by:

Supported Features

`--enable-safe-preview -source <version>`, where `<version>` must match the major version of the frgaal compiler. The classfiles produced by frgaal will not be marked as preview, and will be usable on the selected target Java version, or newer Java versions. The source code may not compile on newer versions of the frgaal compiler, and may need adjustments.

System Paths

By default, the frgaal compiler uses system classes that correspond to the specified target platform version. That means that regardless of the platform on which the frgaal compiler runs, only APIs from the target platform can be used. To disable this behavior use:

- `-bootclasspath`, `-Xbootclasspath` or `-system` and specify the desired target platform API
- `-XDignore.symbol.file` to disable this behavior and use runtime platform's APIs (which may or may not match the target version)

Caveats

The current caveats include:

- `module-info.java` cannot be compiled with target 8.
- the records preview feature, introduced in Java 14 cannot be used in the safe preview mode.

Building

Run the "build.sh" script. The compiler will be in the `dist/compiler-*.jar` file.

License

The license is GPLv2+CPE.