Python-to-Java Converter

Your Name

December 31, 2024

1 Introduction

This project demonstrates how to read a Python script, tokenize (lex) its contents, convert Python constructs into equivalent Java code, and then dynamically compile and run the generated Java code—all in one workflow. The core classes involved are:

- **Interpreter**: Reads the Python file and tokenizes each line into a list of tokens, tracking indentation for code block structure.
- **PythonToJavaConverter**: Takes the tokenized Python code and converts it into Java code, handling `if/elif/else`, `for`, `while`, and common statements like `print`, variable assignments, `break`, etc.
- **Main**: Orchestrates the entire process:
  1. Reads the Python file path.
  2. Invokes the `Interpreter` to tokenize the file.
  3. Uses `PythonToJavaConverter` to produce Java source code.
  4. Writes the generated code to a `.java` file.
  5. Dynamically compiles and executes the Java file.

2 Features

- **Tokenization of Python**: The `Interpreter` class splits a Python file into tokens, preserving indentation levels.
- **Syntax Conversion**: The `PythonToJavaConverter` class handles typical Python constructs:
  - `if/elif/else` → Java `if/else if/else`
  - `for ... in range(...)` → Java `for` loops
  - `while` → Java `while`
  - `print(...)` → `System.out.println(...)`
  - Variable assignments with type inference
- **Dynamic Java Compilation**: The `Main` class calls the system Java compiler (via `ToolProvider`) to compile the generated code.
- **Runtime Invocation**: After compiling, `Main` loads the class and invokes its `main` method via reflection.

3 Prerequisites

- **Java JDK** (not just a JRE). Required for the `javax.tools.JavaCompiler` API.
- Python file to convert (e.g.):

```python
x = 5
if x > 0:
    print("Positive")

for i in range(3):
    print(i, x)
```

4 Getting Started

1. **Clone or Download** this repository.
2. **Open** the project in an IDE or ensure you can compile and run Java files from the command line.

5 How to Use

1. **Specify the Python File**

   In `Main.java`, Tester of this program should change path with their own (on line 17), because this path is for my computer and may not work on different machine.

```java
String filePath = "C:\\Users\\user\\Desktop\\FOP\\python_code_test.txt";
```

   It is desired to copy txt.file, which is uploaded on GitHub, to your txt.file

   The program will:

   - Read and tokenize the Python file.
   - Generate corresponding Java code as a string.
   - Save the Java code to `TranslatedJavaCode.java`.
   - Compile `TranslatedJavaCode.java`.
   - Run the compiled class, printing output to the console.

2. **Check the Output**

   You should see the generated Java code (optionally printed in the console) and its execution result.

6 Project Structure

```
.
 Main.java                  # Orchestrates reading/conversion/compilation/execution
 PythonToJavaConverter.java # Converts tokenized Python into Java code
 Interpreter.java           # Reads Python file and tokenizes it
```

7 Customization

- Modify `PythonToJavaConverter.java` to handle or transform specific Python features differently.
- Enhance `Interpreter.java` to parse more advanced Python syntactical constructs (multi-line strings, functions, classes, etc.).
- Adjust type inference or error handling as needed.

8 Known Limitations

- Currently handles a subset of Python syntax:
  - `if/elif/else`, `for ... in range(...)`, `while`
  - Basic assignments and prints
  - Minimal type inference (int, double, boolean)
- Indentation-based block parsing only, ignoring more advanced Python features such as functions or class definitions.