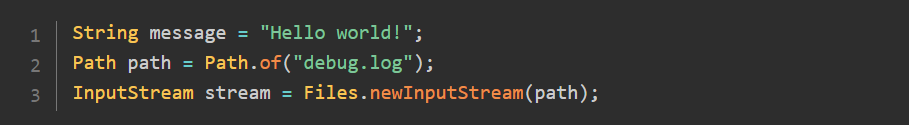
**Using the Var Type Identifier:**

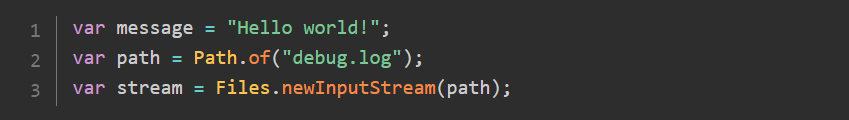
**The Var Keyword:** Starting with Java SE 10, you can use the var type identifier to declare a local variable. In doing so, you let the compiler decide what is the real type of the variable you create. Once created, this type cannot be changed.

**Consider the following example:**



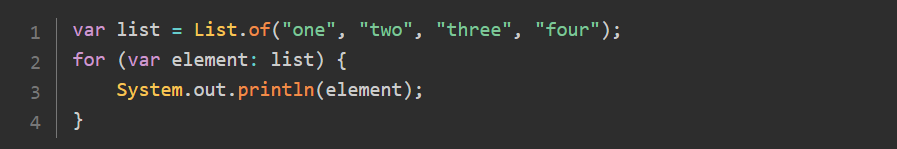
In that case, having to declare the explicit types of the three variables **message**, **path** and **stream** is redundant.

With the **var** type identifier the previous code can be written as follow:

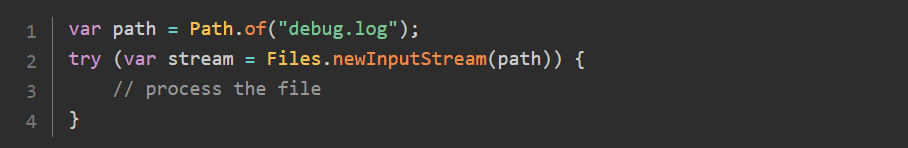


Examples with Var

The following example shows you how you can use the **var** type identifier to make your code simpler to read. Here the strings variable is given the type **List<String>** and the **element** variable the type **String.**



On this example, the **path** variable is of type **Path**, and the stream variable is of type **InputStream**.



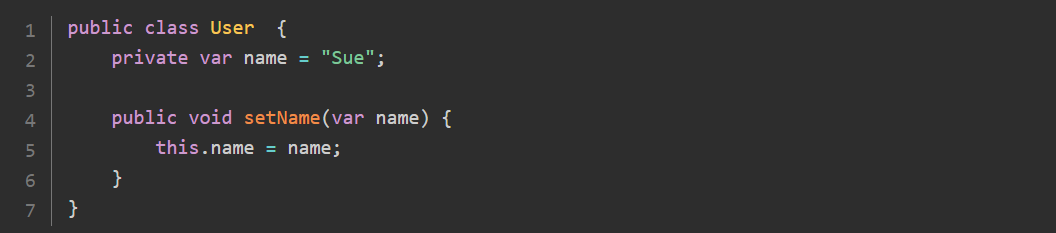
Two previous examples, you have used **var** to declare a variable in a for statement and in a try-with-resources statement.

**Restrictions on Using Var**

There are restrictions on the use of the **var** type identifier.

1. You can only use it for local variables declared in methods, constructors and initializer blocks.
2. **var** cannot be used for fields, not for method or constructor parameters.
3. The compiler must be able to choose a type when the variable is declared. Since **null** has no type, the variable must have an initializer.

Following the these restriction, the following class does not compile, because using **var** as a type identifier is not possible for a field or a method parameter.



The same goes for the following code.

In that case, the compiler cannot guess the real type of **message** because is lacks an initializer.

