Compiling

We write programs in a high-level programming language like Scala, Java, C++, or C.

A compiler translates the source code to object code (machine code).

For C and C++, it is customary to compile to native machine code. It can be executed directly by the processor.

Native machine code is different for different processors, operating systems, and can depend on library versions.

Java and Scala are normally translated to object code for the JVM (Java virtual machine). A Java runtime environment is needed on the computer to execute the program. The exact same object code works on any system. JVM is heavily used on servers.

Compiling a class

The Scala compiler compiles each class to an object file.

$$\begin{tikzcd}
\text{class Point} \arrow[dd] & \text{in file } \texttt{point.scala} \arrow[l] \\
 & \texttt{scalac point.scala} \arrow[l] \\
\texttt{Point.class} & \text{object code} \arrow[l]
\end{tikzcd}$$

Tip: Use `fsc` (fast Scala compiler) instead of `scalac`.

```
> scala
Welcome to Scala version 2.9.1.final
scala> val p = new Point(1, 2)
p: Point = [1,2]
```

JVM finds `Point.class`

Compiling a program

The JVM is purely object-oriented:

- No global variables;
- Every function is a method of some class.

We cannot compile Scala scripts:

```
> scalac hello.scala
error: expected class or object definition
println("Hello World")
^
```

A Scala program has to be packaged inside a class.

Chicken and egg problem: How to make an object of this class type? We can only use `new` inside a method!

Singleton objects

A singleton is a class type for which only one single object can exist.

Define a singleton using `object`:

```
scala> object Kermit {
     |   println("Kermit is born")
     |   def kermit() {println("I am green")}
     | }
defined module Kermit
```

This defines `Kermit`, it doesn't create an object yet.

```
scala> Kermit.kermit()
Kermit is born
I am green
scala> Kermit.kermit()
I am green
```

Singleton objects

The name of the singleton is like a `val`. It refers to the object on the heap:

```
scala> Kermit.kermit()
I am green
scala> Kermit
res1: Kermit.type = Kermit$@1eae15f
```

The singleton object is created the first time that it is used.

```
scala> Kermit.kermit()
Kermit is born
I am green
scala> Kermit.kermit()
I am green
```

If you never use a singleton, it is never created.

Compiled Scala programs

1. Define a singleton object `Kermit` that contains a method `main(args: Array[String]): Unit`.

2. Compile the source file, resulting an a class file `Kermit.class` (and several other files containing $ signs):

```
> scalac kermit.scala
```

3. Run the program:

```
> scala Kermit arguments
```

Name of the object

Actually, to run the compiled program, we only need the JVM and the Scala standard library:

```
> java -cp .:scala-library.jar Kermit
```