Episode 15 (Characteristics of the Java Language)

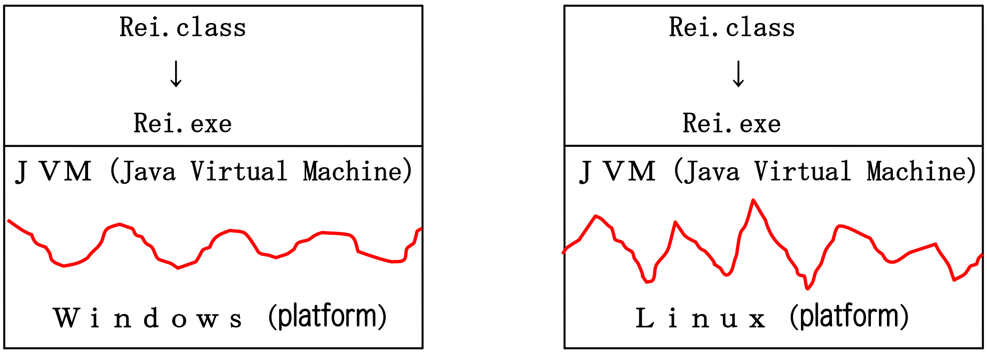
Kitsune! I've heard that Java programs can run on Windows, Linux and Mac, what does that mean?

Windows, Linux, or Mac refers to an operating system. It's also called a platform.

The Java language can indeed run on different platforms. This is the primary feature of the Java language.

If you use different platforms, you usually have to re−create a program for each platform. However, programs written in the Java language do not need to be rewritten.

Here is an illustration of why.



The above diagram shows how the JVM (Java Virtual Machine) smoothes out the differences (bumpiness) between platforms such as Windows and Linux, so that the same Java program can be run on any platform.

Rei.class is an intermediate language that is compiled from Rei.java (source file). The JVM turns it into Rei.exe and executes it.

Kitsune, I got the synopsis, but how do you make the JVM usable on each platform? I don't know how that works.

Tanuki, you said before that virtual machines are software, and of course the Java Virtual Machine is software.

First, you need to understand the difference between JVM (Java Virtual Machine), JRE (Java Runtime Environment) and JDK (Java Development Kit).

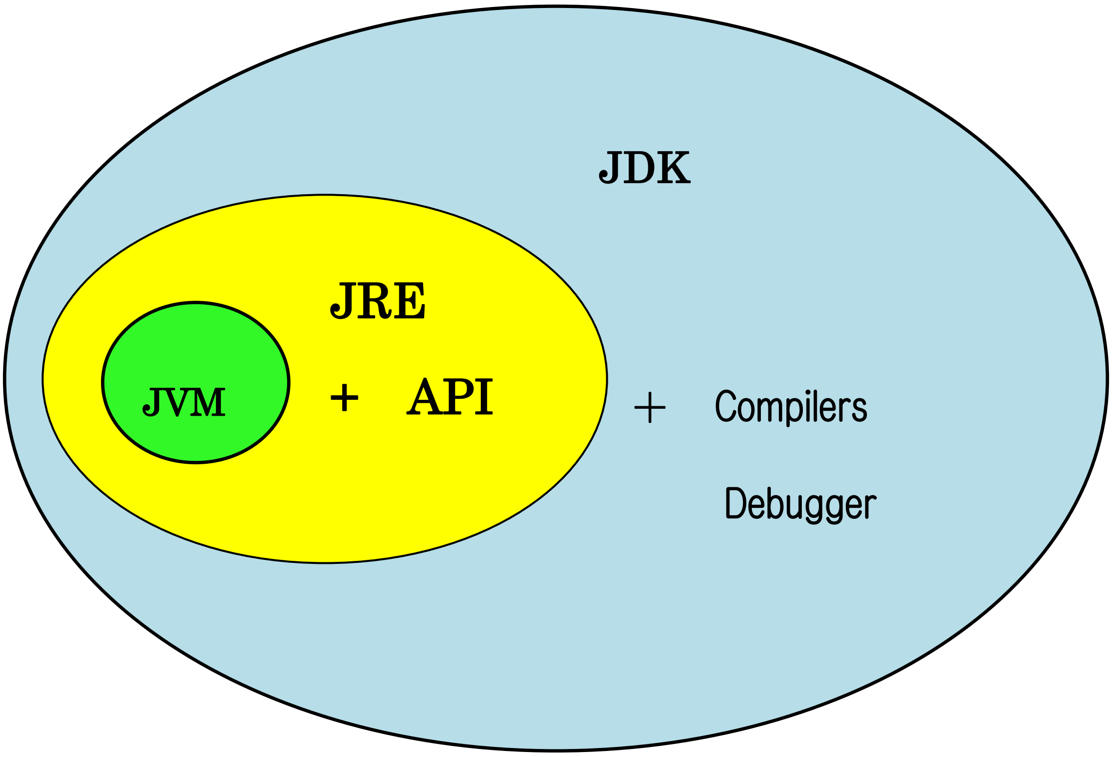
Jre creates a java runtime environment. In other words, it is software made of JVM + API.

The API can be imported and used with class libraries. It is also available as a package.

JDK is a software consisting of Jre + compiler (Javac) + debugger, which is also called software for Java development.

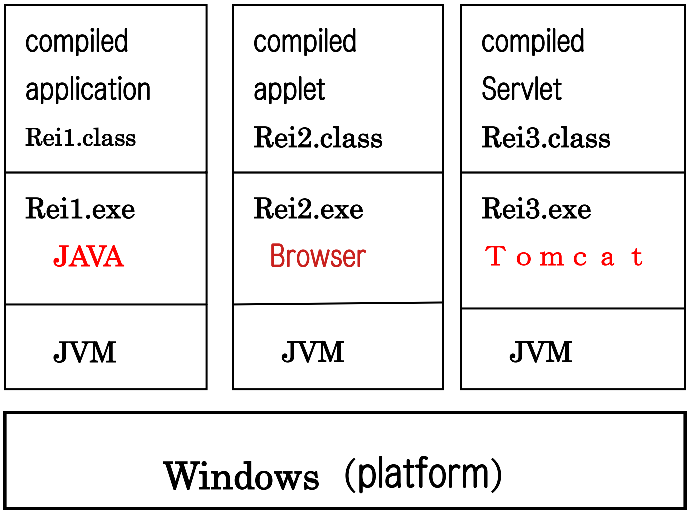
This means that once you install the JDK, you have everything you need, including the JVM.

However, the JDK you need to install is different for each platform.



I see…I just need to install the JDK.

Java Features, Part 2.
I can make three kinds of programs. One is an applet that runs in a browser. The second is a servlet (JSP, etc.) executed by Tomcat, and the third is a regular application. All of them have one thing in common, though: they are all based on the JVM.
In particular, Tomcat is a web container that runs on the JVM, interprets and executes servlets, and bridges the web server and database. The three types of mechanisms are illustrated in the following diagram.



compiled application Rei1.class	compiled applet Rei2.class	compiled Servlet Rei3.class
Rei1.exe JAVA	Rei2.exe Browser	Rei3.exe Tomcat
JVM	JVM	JVM
Windows (platform)		

Fox! How do you distinguish between these three types of programs?

Tanuki, you’ve got a good point.
An applet has its functionality in the [java.applet] package, which contains class libraries to import, and a servlet has its functionality in the [javax.servlet] package, which contains class libraries.

First, let's cover a simple example program for a Java application and how to make it compile & run.

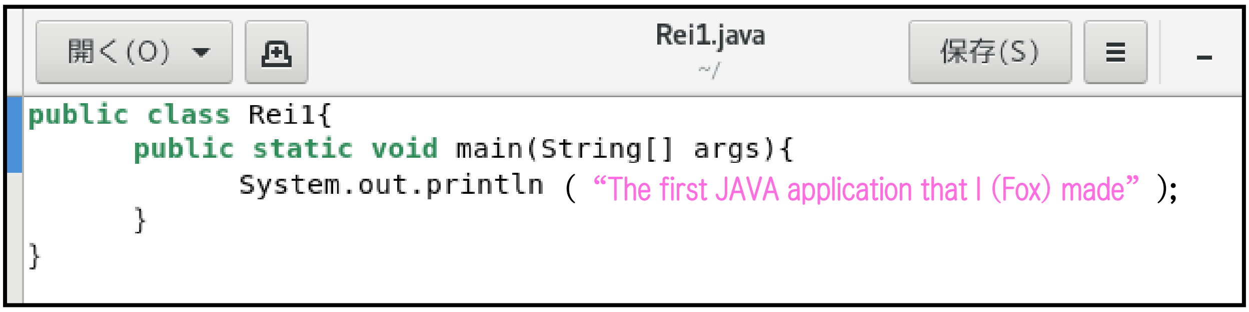
Tanuki, you need to prepare yourself by looking up how to install the JDK on the net, because it has to be installed according to your platform and you need to pass it through. Some users may use an integrated development environment such as Eclipse, and I'll leave that to you since everyone has their own way of using Java.

I will give you the most basic explanation, which is to use notepad or gedit as an editor, and to run the program on a terminal. This is the best way to learn.

By the way, if you have a CentOS7 virtual machine, which was covered in the virtual environment in episodes 8 to 11, JDK1.8 will be automatically installed and ready to use. If the virtual machine is in that state, you can operate and execute as described below. Also, you can practice without worrying because it does not affect the host machine Windows in any way.

Well, it is not so easy to understand that a Java virtual machine (JVM) has been created on a CentOS7 virtual machine.

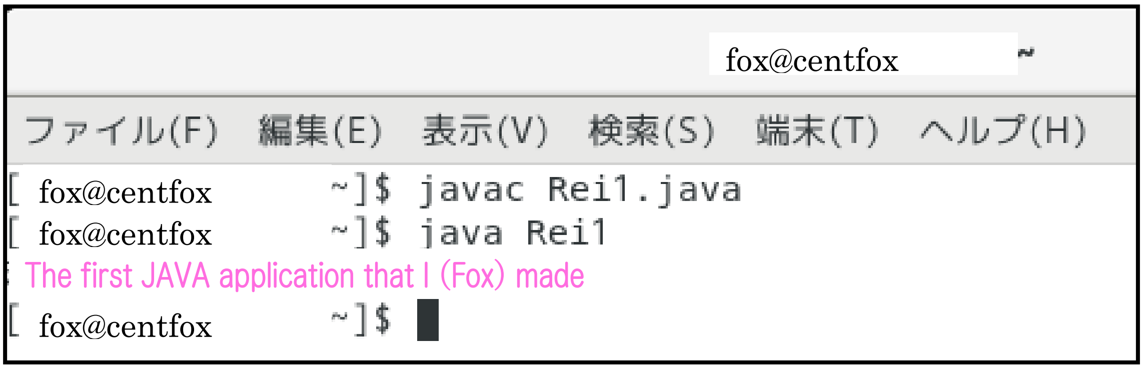
Source file of Java application (Rei1.java) created by gedit on virtual machine (CentOS7)



```
public class Rei1{
        public static void main(String[] args){
                System.out.println ( “The first JAVA application that I (Fox) made” );
        }
}
```

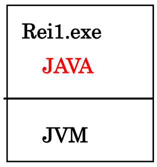
I see, so I can start the virtual machine (CentOS7), call gedit installed in it, start it, type as shown in the above figure, and save it with the file name Reil.java. What's next?

Start terminal, compile and run



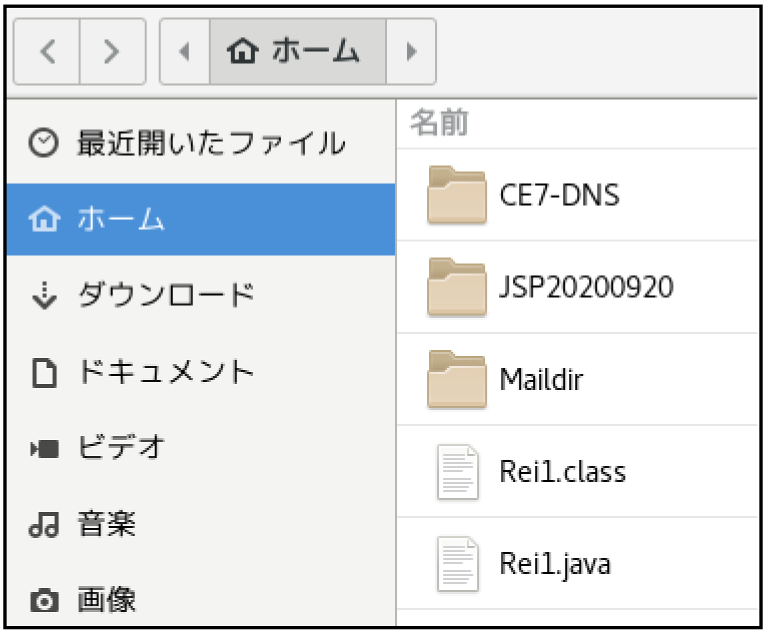
```
[ fox@centfox       ~]$ javac Rei1.java
[ fox@centfox       ~]$ java Rei1
 The first JAVA application that I (Fox) made
[ fox@centfox       ~]$ █
```

Tanuki, no rush!
Programming is all about input and output. Let's take our time.
First of all, the Reil.java file must be saved in my (/home/fox) home directory, or else you will get an error message saying that such a file does not exist. After confirming this and executing the first line, a compiled intermediate file named Reil.class will be created in lla's directory. Next, type "java Reil" as in the second line, and press the return key to execute it. At this time, you should not specify the intermediate file as [java Reil.class]. You are not understanding what I have just explained: Java converts the intermediate file Reil.class into an executable file [Reil.exe], and executes it.

Kitsune, so now we know what this diagram ()mean

Kitsune, may I take a peek at your home directory?

Contents of ／home／fox



ホーム

最近開いたファイル
ホーム
ダウンロード
ドキュメント
ビデオ
音楽
画像

名前
CE7-DNS
JSP20200920
Maildir
Rei1.class
Rei1.java

No problem! There’s an extra directory in there, though.

You can see that Reil.java and the compiled Reil.class have been created. Also note that the executable Reil.exe does not exist. This is important because this is why it works on many platforms.

Next is an example of running a Java applet!

Enter and save the Java applet source (Rei2.java) with gedit

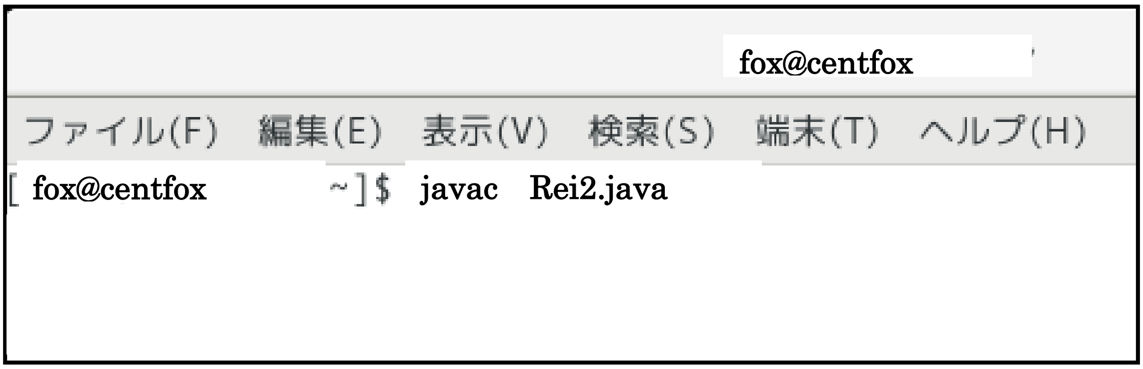
```
import java.applet.Applet;
import java.awt.Graphics;

        public class rei3 extends Applet {
                public void paint(Graphics g){
                        int x,y;
                        for (x=0;x<=5;x++){
                                g.drawString("fox",x*10, x*10);
                        }
                        for (y=1;y<=4;y++){
                                g.drawString("fox",50-y*10,50+y*10);
                        }
                }
        }
```

Enter and save the HTML file (Rei2.html) that will be the base of applet execution

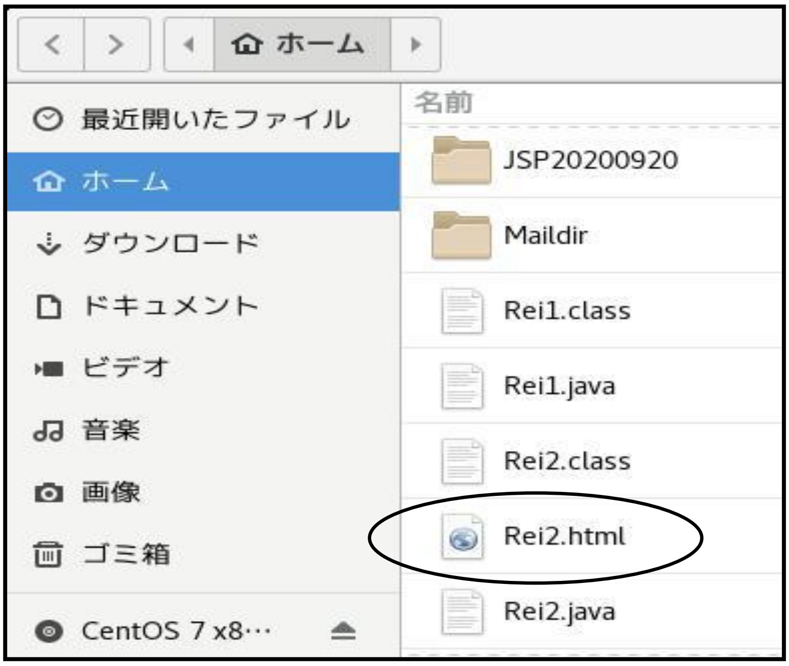
```
<html>
<head>
<title>Rei2</title>
</head>
<body>
<hr>
<applet   code = "Rei2.class"    width=420    height=420></applet>
<hr>
</body>
</html>
```

Start the terminal and compile the applet (Rei2.java)

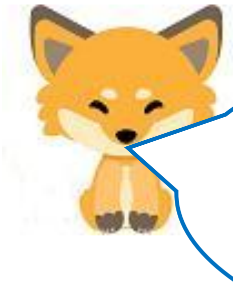


```
[ fox@centfox        ~]$  javac   Rei2.java
```

Contents of ／home／fox

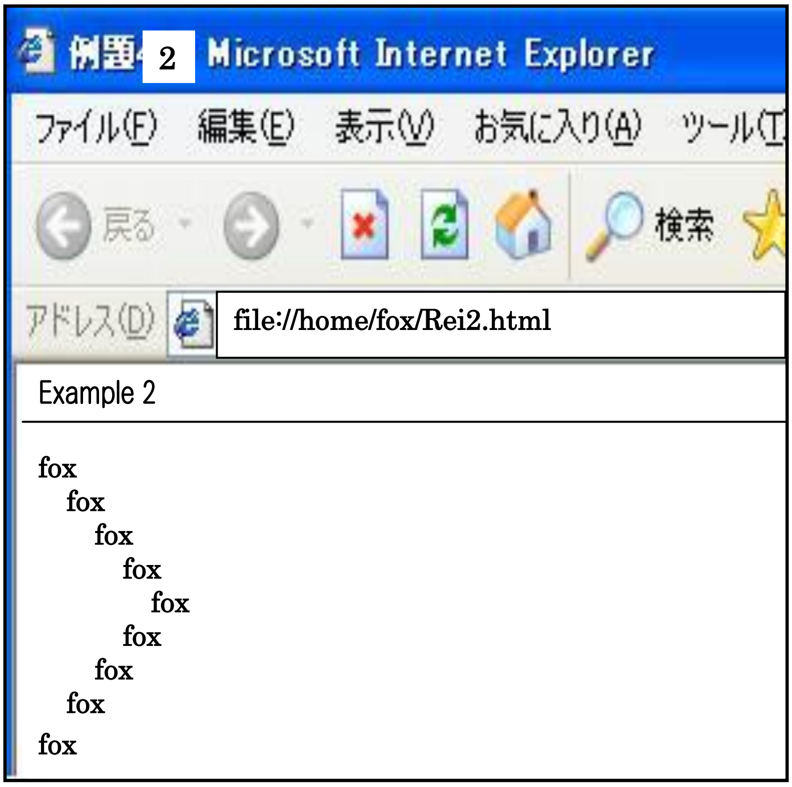


JSP20200920
Maildir
Rei1.class
Rei1.java
Rei2.class
Rei2.html
Rei2.java



Rei2.java and Rei2.html are saved in my directory (／home／fox).
If a web server has been built, place the files in (／ver／www／html), start a browser, and enter ”http:／／www.fox.com／Rei1.html” in the URL to start the browser, but this time, simply double－click the [Rei2.html] file shown above with the mouse. The browser will convert Rei2.class to Rei2.exe, which will be displayed as shown below.

Result of applet execution


例題 2 Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T
戻る
検索
アドレス(D)

file://home/fox/Rei2.html

Example 2

```
fox
  fox
    fox
      fox
        fox
      fox
    fox
  fox
fox
```

Fox! Your browser is up and running, and you've got "Example 2" and the top and bottom lines, but you don't see the 9 "foxes". Apparently the HTML is running, but the applet Rei2.class is not. In addition, some browsers give me a message saying, "Do not use this plugin because it is dangerous. They are trying to trick me again!

Sorry, sorry! I didn't mean to be a dummus...before 2017, Java applets were usable on browsers, but starting in 2018, Java applets have security issues and are no longer supported by all browsers. Furthermore, they are moving away from using applets. I'm sorry, but I'm going to stop using them.

The explanation so far is just to give you an idea of what Java applets are like.

Roger! Sometimes they are no longer supported.

I guess the last thing I want to talk about is Java servlets!

No, I'm not going to talk about Java servlets (especially JSP).

It's important, but you have to prepare a lot of things.

I can make it on a CentOS7 virtual machine, but I need to install and prepare a web server, DNS server, Tomcat, etc. there before I can practice servlets.

I'll talk about it again when those preparations are ready.

I should mention, however, that the JSP code has changed considerably with the passage of time.

Next up: Episode 16.

Translated at DeepL