Concepts of Programming
Day 1: Sep 2022

Introduction to Java

Kiran Waghmare

CDAC Mumbai

Kiran Waghmare : CDAC Mumbai

Outlines

- Introduction
- Why Java?
- Where it is used?
- Characteristics of Java
- Java Platform Editions
- Evolution of Java
- Understanding JDK, JRE and JVM
- How Java is platform-independent
- Evaluating Java libraries, middle-ware, and database options

Java

- **General purpose Programming language.**

- **Supports OOP**

- **Developed by Sun Microsystems of USA in 1991**

- **Developed by Sun Microsystems (James Gosling).**

- **Called as OAK by James Gosling**

- **Discovered to overcome the limitations of C and C++ in terms of both reliability & portability**

- **Thus, Java a really simple, reliable, portable & powerful language.**

Applications of Java

Desktop GUI

Enterprise
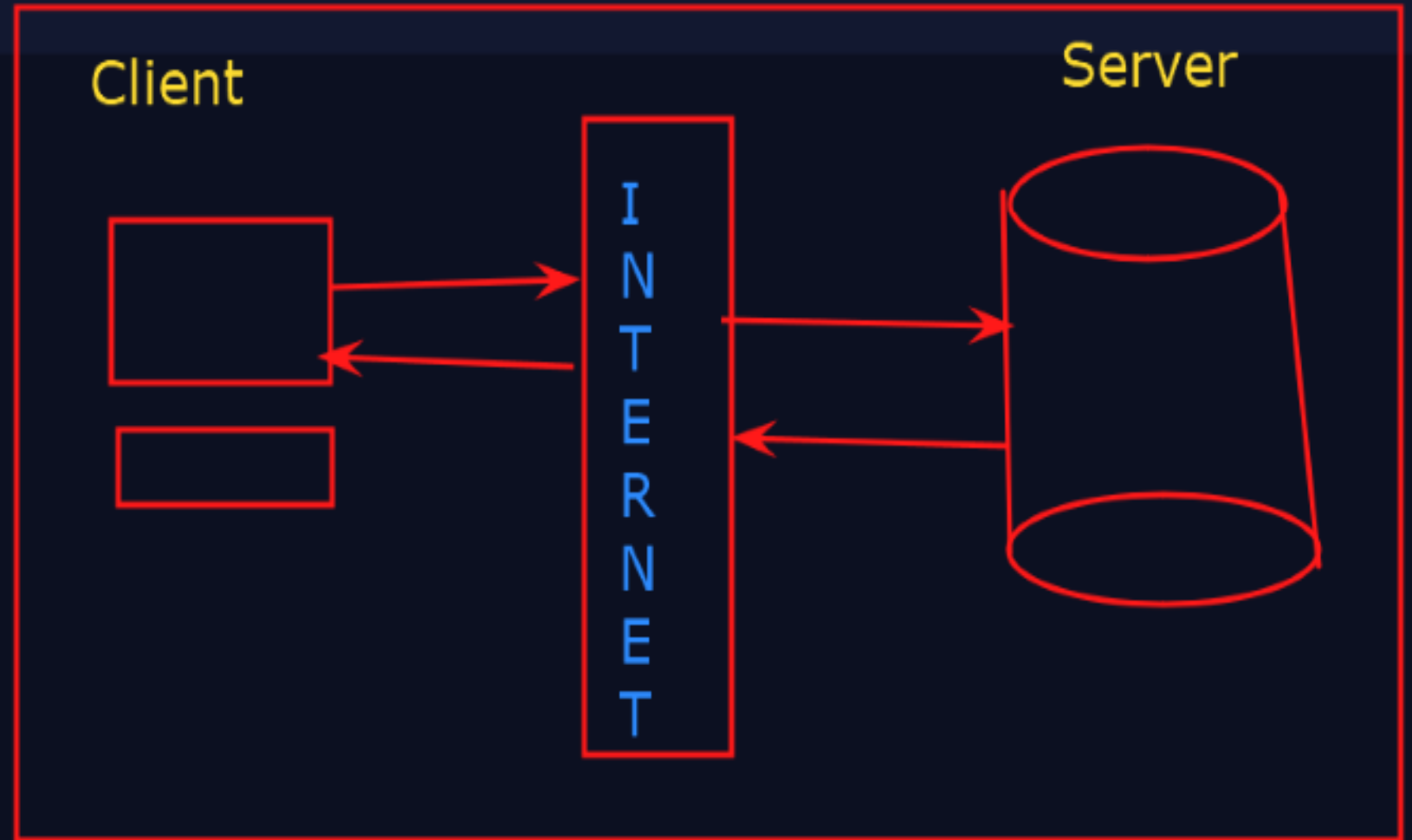
Scientific

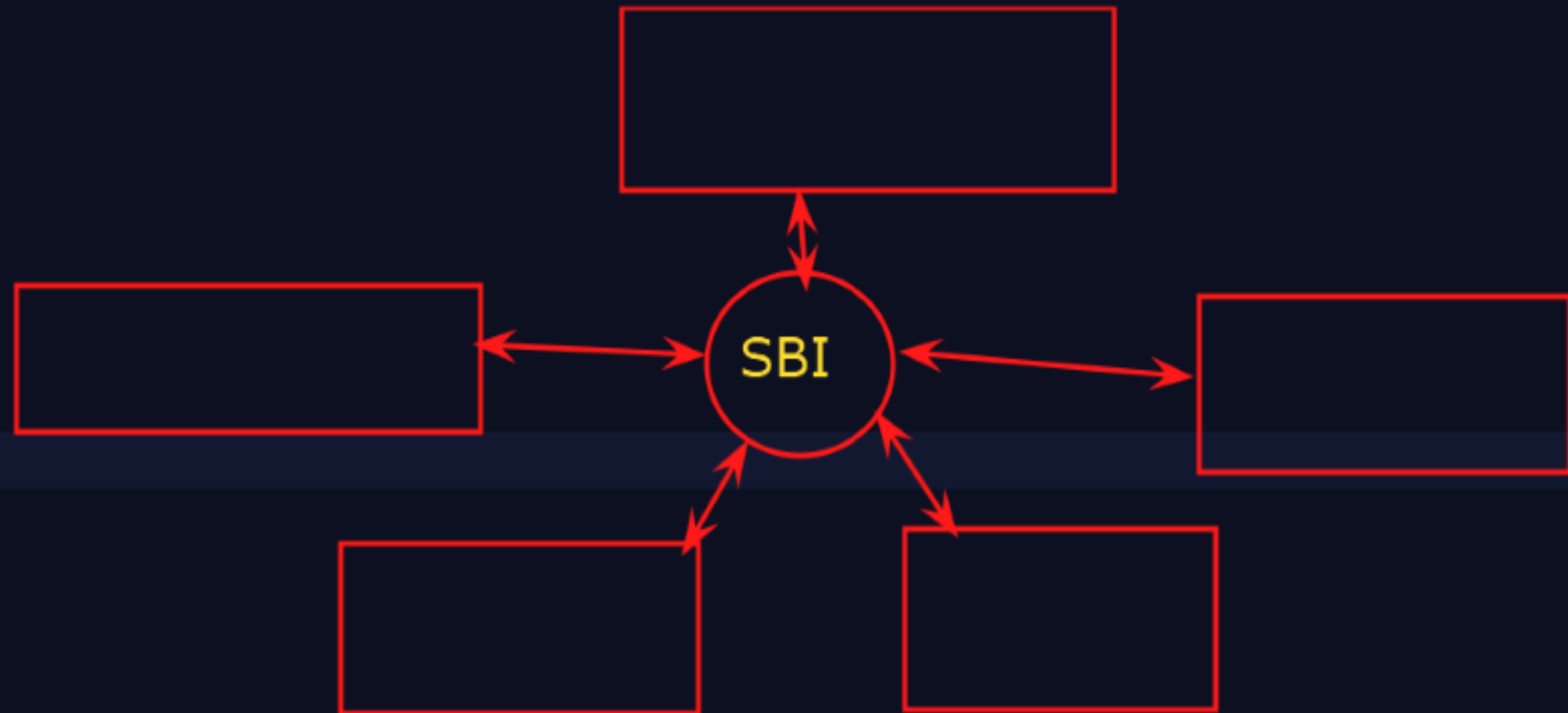**Web-based**

Cloud-based

Gaming

```
Applications of Java:
------------------------

-Desktop applications/ Standalone application:
-Web application
```


Client

Server

INTERNET

```
Applications of Java:
-------------------------

-Desktop applications/ Standalone application:
-Web application
-Enterprise application: disrtibuted applications
```
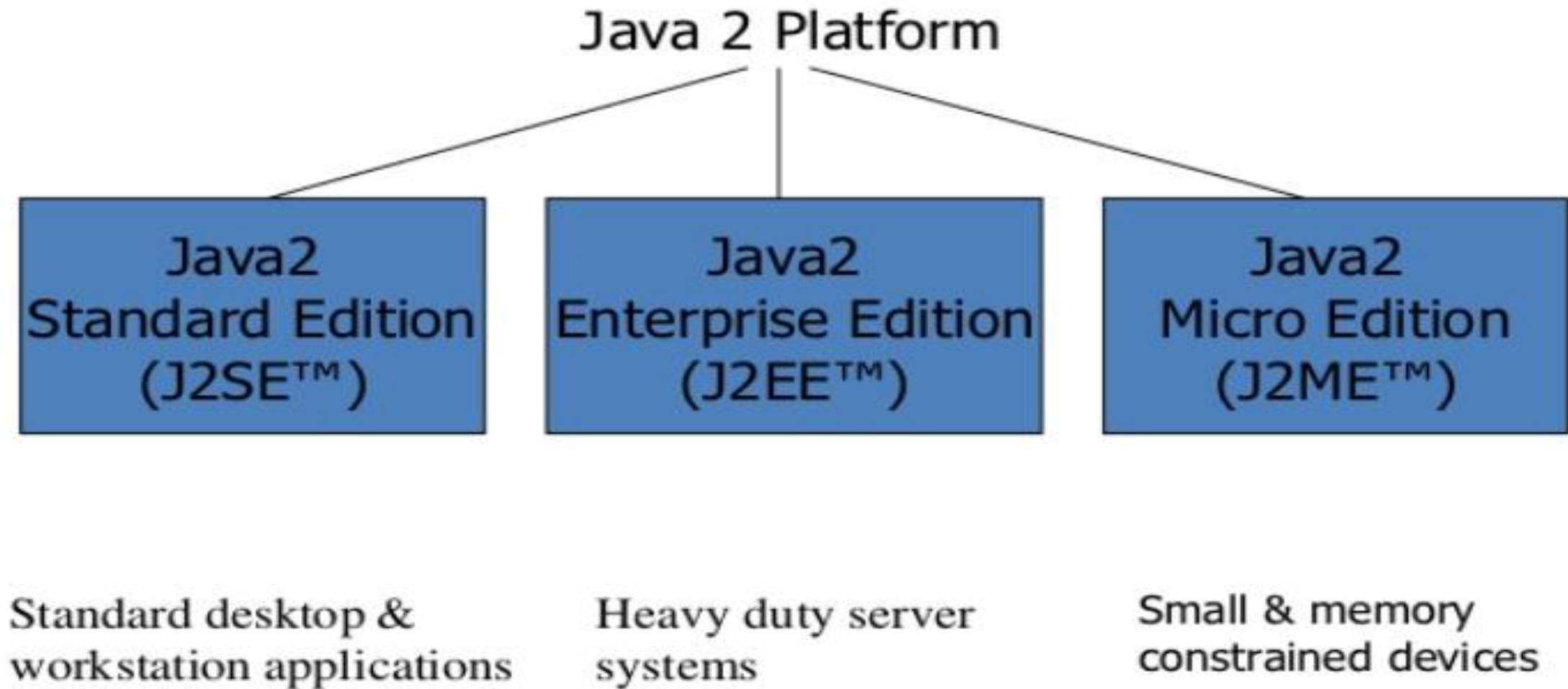
Where it is Used?

➢According to Sun, 3 billion devices run java.

- Desktop Applications such as acrobat reader, media player, antivirus etc.

- Web Applications such as irctc.co.in, javatpoint.com etc.

- Enterprise Applications such as banking applications.

- Mobile Applications

- Embedded System, Smart Card, Robotics, Games etc.

Java Platforms / Editions

Java 2 Platform

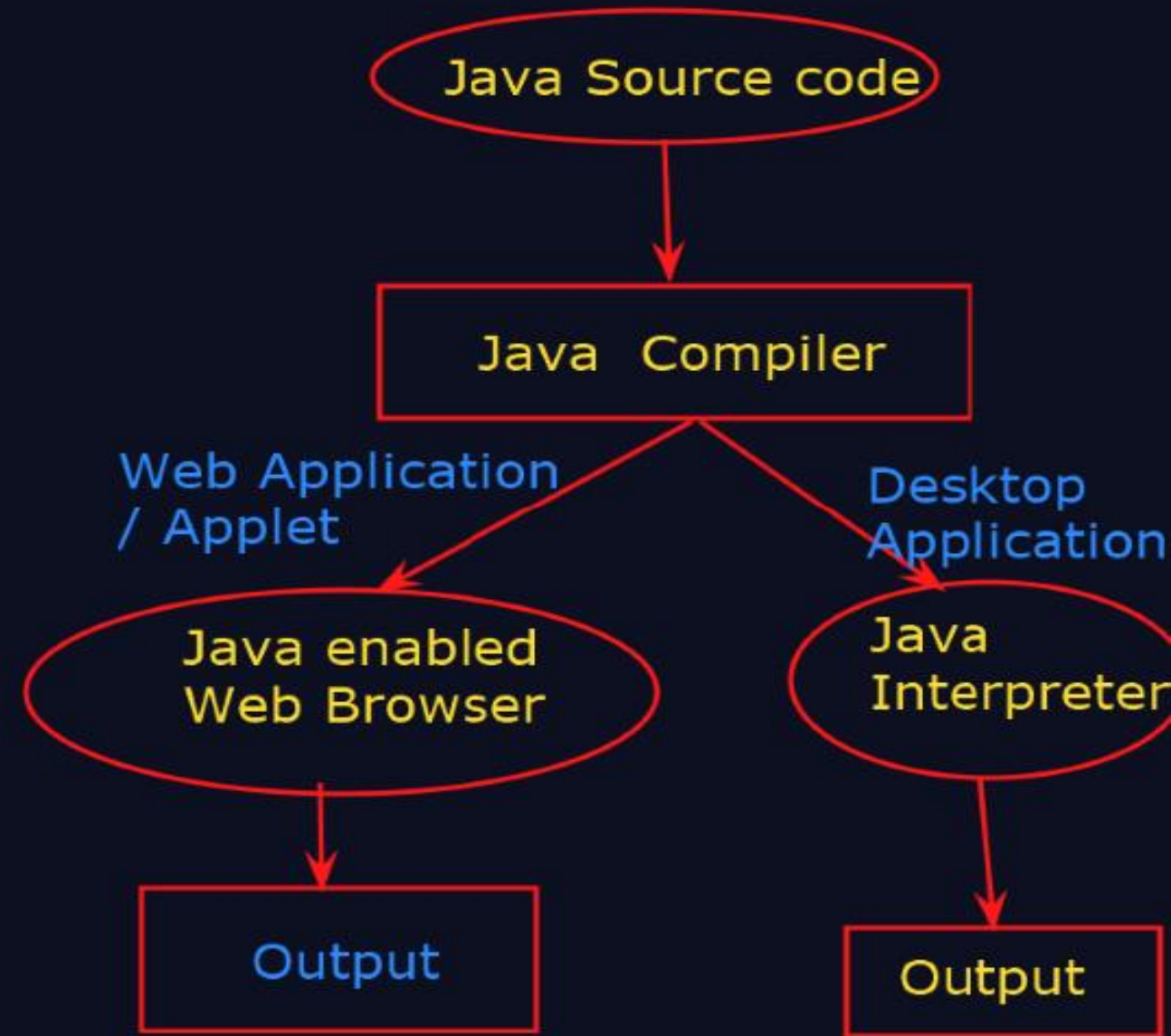| Java2 Standard Edition (J2SE™)	Java2 Enterprise Edition (J2EE™)	Java2 Micro Edition (J2ME™)
Standard desktop & workstation applications	Heavy duty server systems	Small & memory constrained devices

James Gosling

- James Gosling is generally credited as the inventor of the Java programming language

- He was the first designer of Java and implemented its original compiler and virtual machine

- He is also known as the Father of Java.
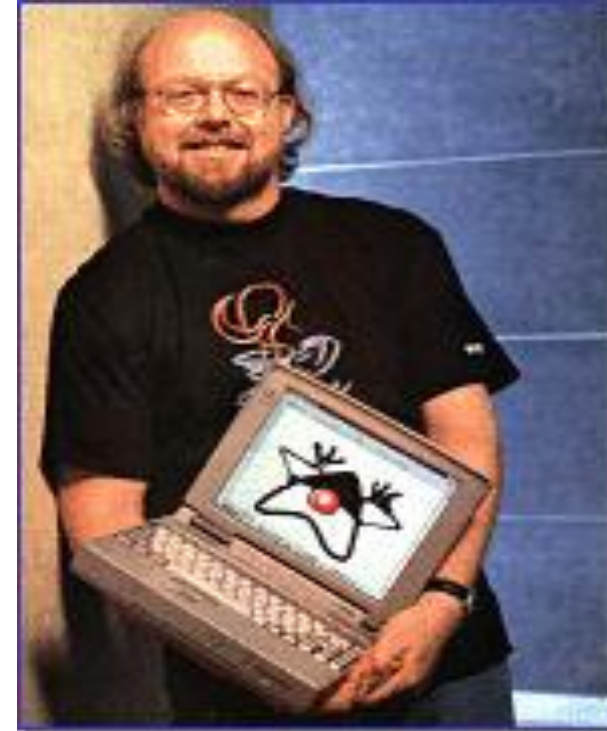
JDK Versions

- JDK Alpha and Beta (1995)
- JDK 1.0 (23rd Jan, 1996)
- JDK 1.1 (19th Feb, 1997)
- J2SE 1.2 (8th Dec, 1998)
- J2SE 1.3 (8th May, 2000)
- J2SE 1.4 (6th Feb, 2002)
- J2SE 5.0 (30th Sep, 2004)
- Java SE 6 (11th Dec, 2006)
- Java SE 7 (28th July, 2011)
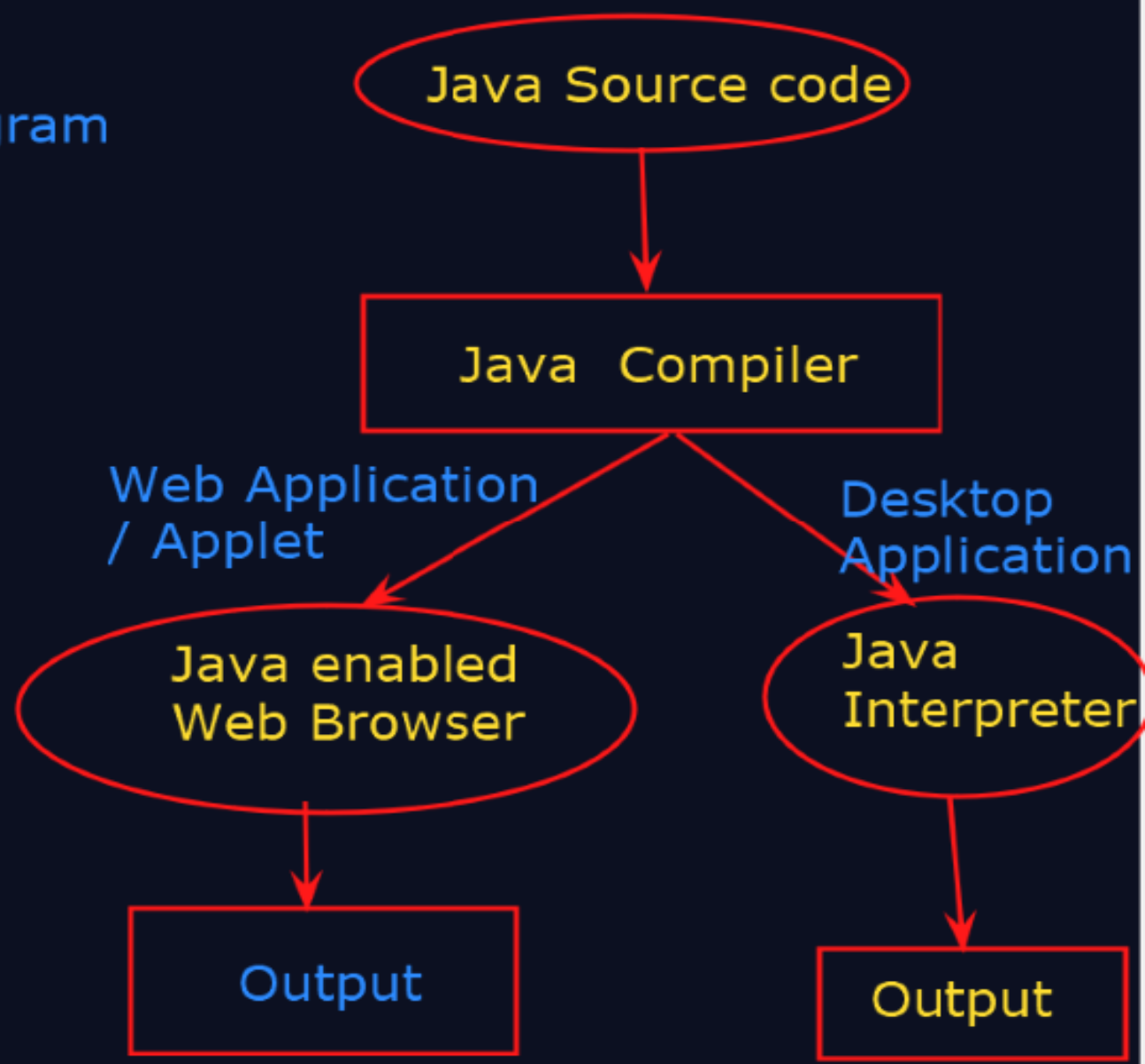- Java SE 8 (18th March, 2014)

Way of using Java programming:

Java Source code

Java Compiler

Web Application
/ Applet

Desktop
Application

Java enabled
Web Browser

Java
Interpreter

Output

Output

```
Way of using Java programming:
----------------------------------
```

main () : intial point of the java program

syntax:

public static void main(String args[])

parameter

Java Source code

↓

Java Compiler

Web Application / Applet

Desktop Application

Java enabled Web Browser

Java Interpreter

Output

Output

```
Process of building and running Java aaplication programs:
---------------------------------------------------------
```

```
Process of building and running Java application programs:
-------------------------------------------------------------
```

Process

| Text Editor |

↓

| Java Source Code | → | javadoc | → | HTML Files |

↓

| javac |

↓

| Java Class File | → | javah | → | Header Files |

↓ (bytecode)

| java | → | jdb |

↓

| Java Program Output |

1.Editor

2.Compiler
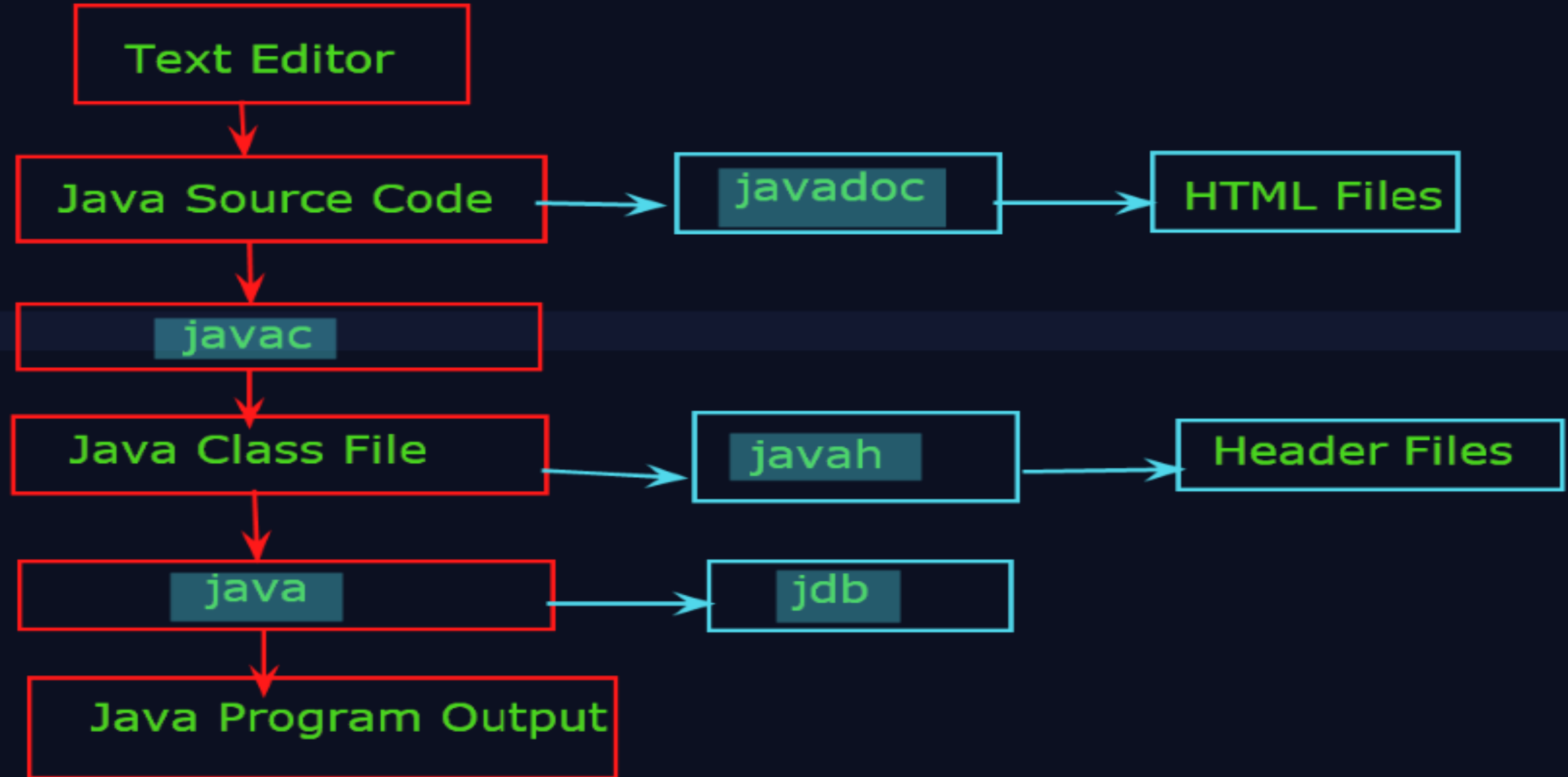
3.Interpreter

Process of building and running Java application programs:
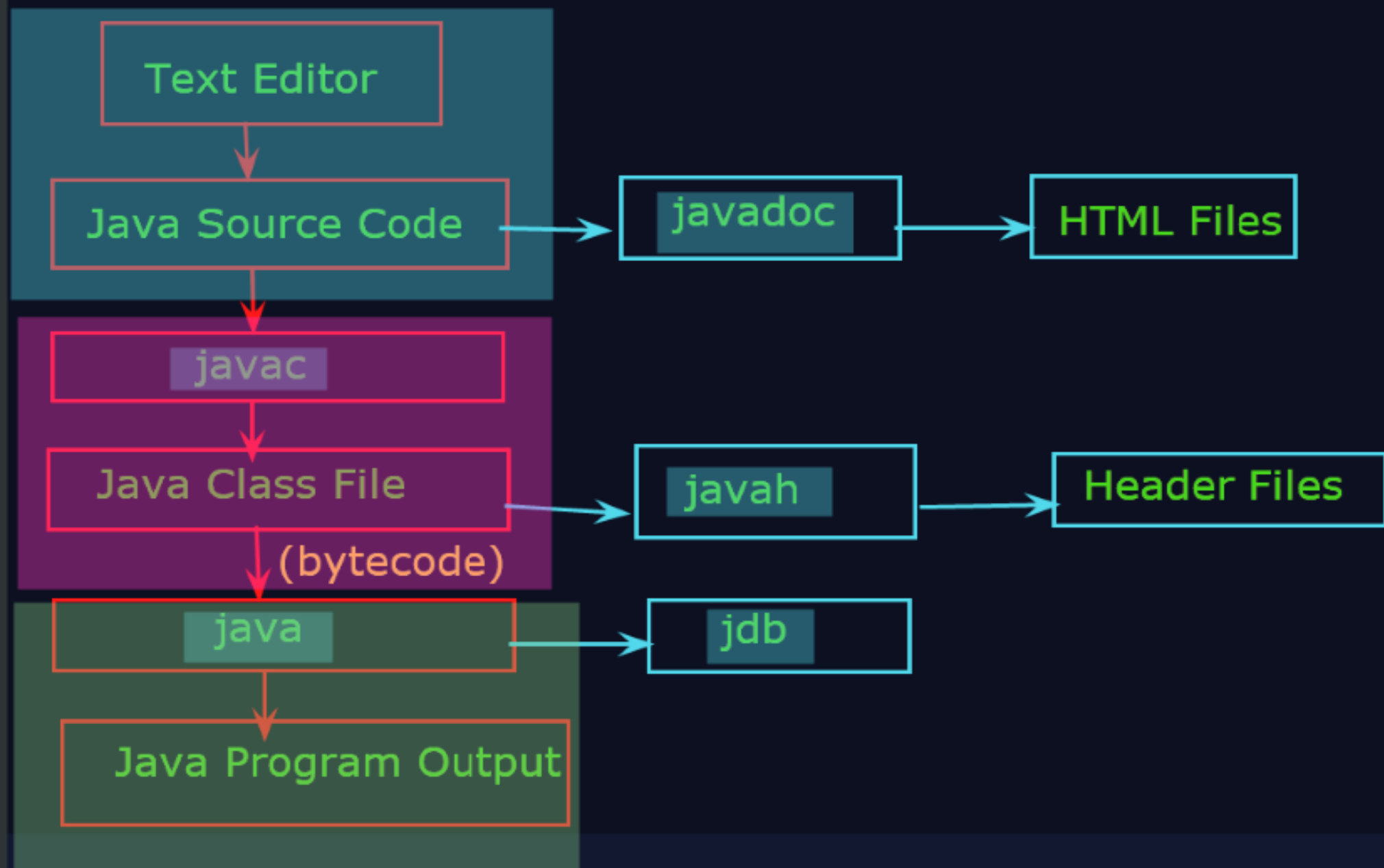
Process

1. Editor
2. Compiler
3. Interperter

JVM : Java Virtual Machine

Text Editor
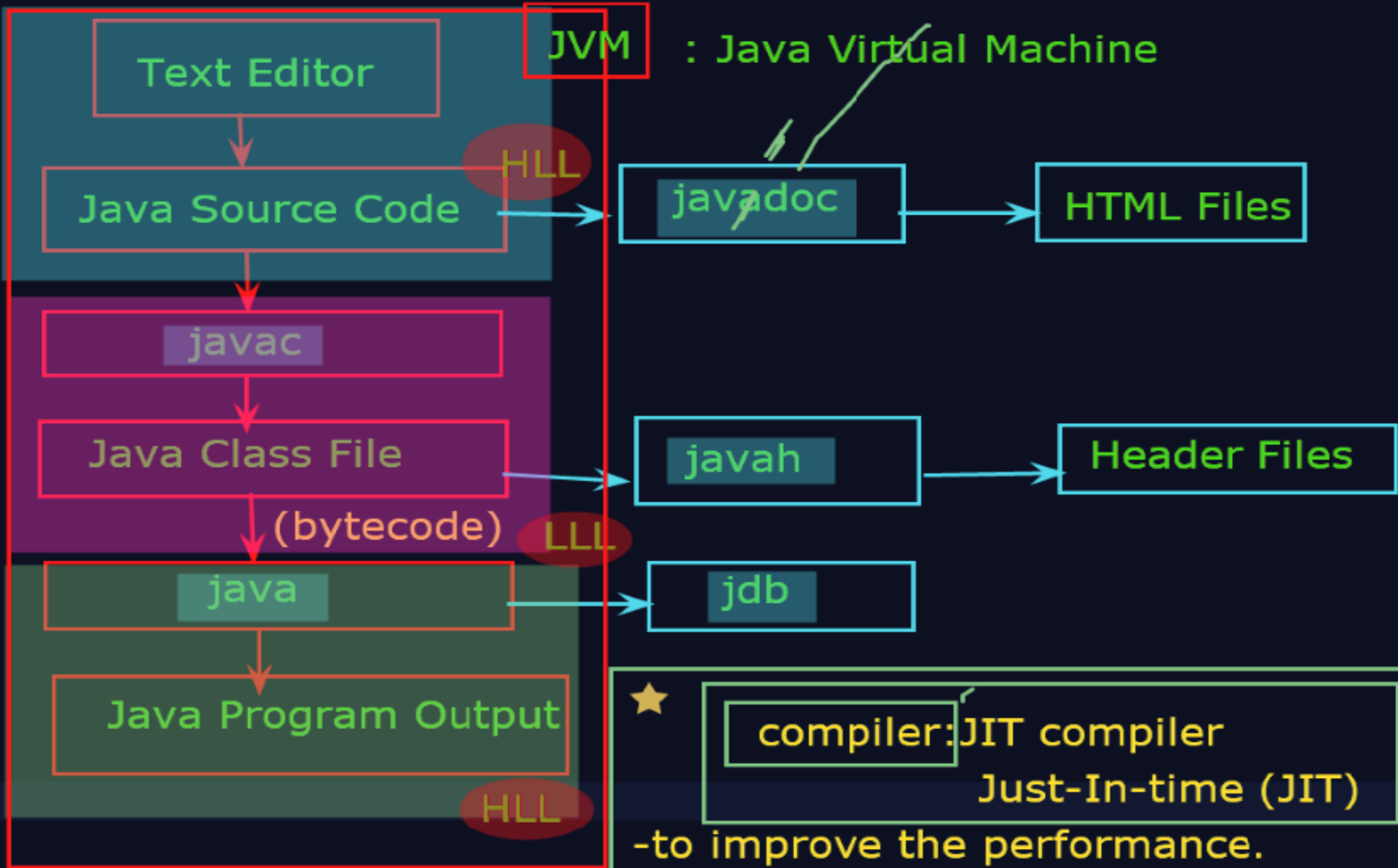
Java Source Code — HLL → javadoc → HTML Files

javac

Java Class File → javah → Header Files
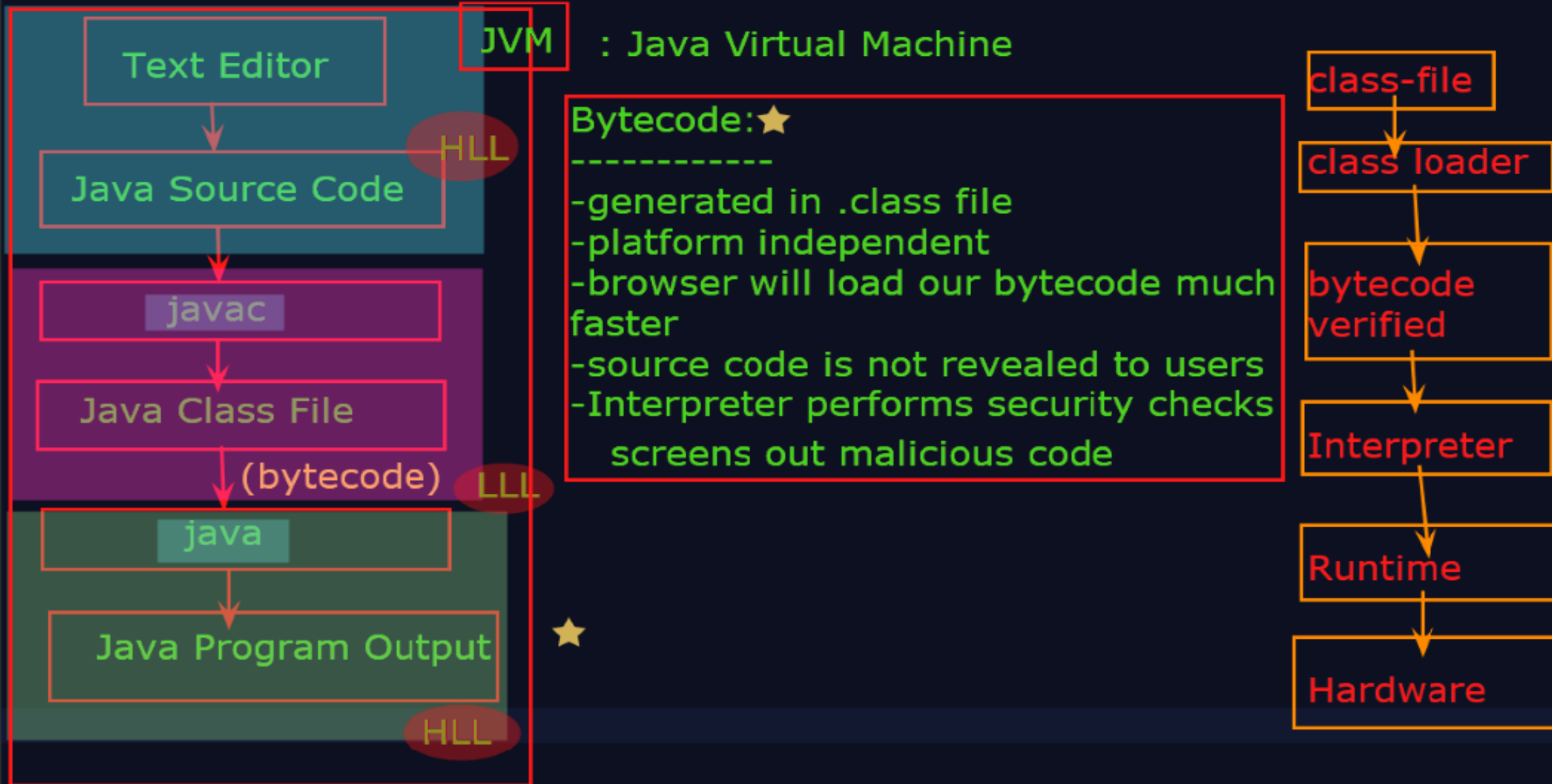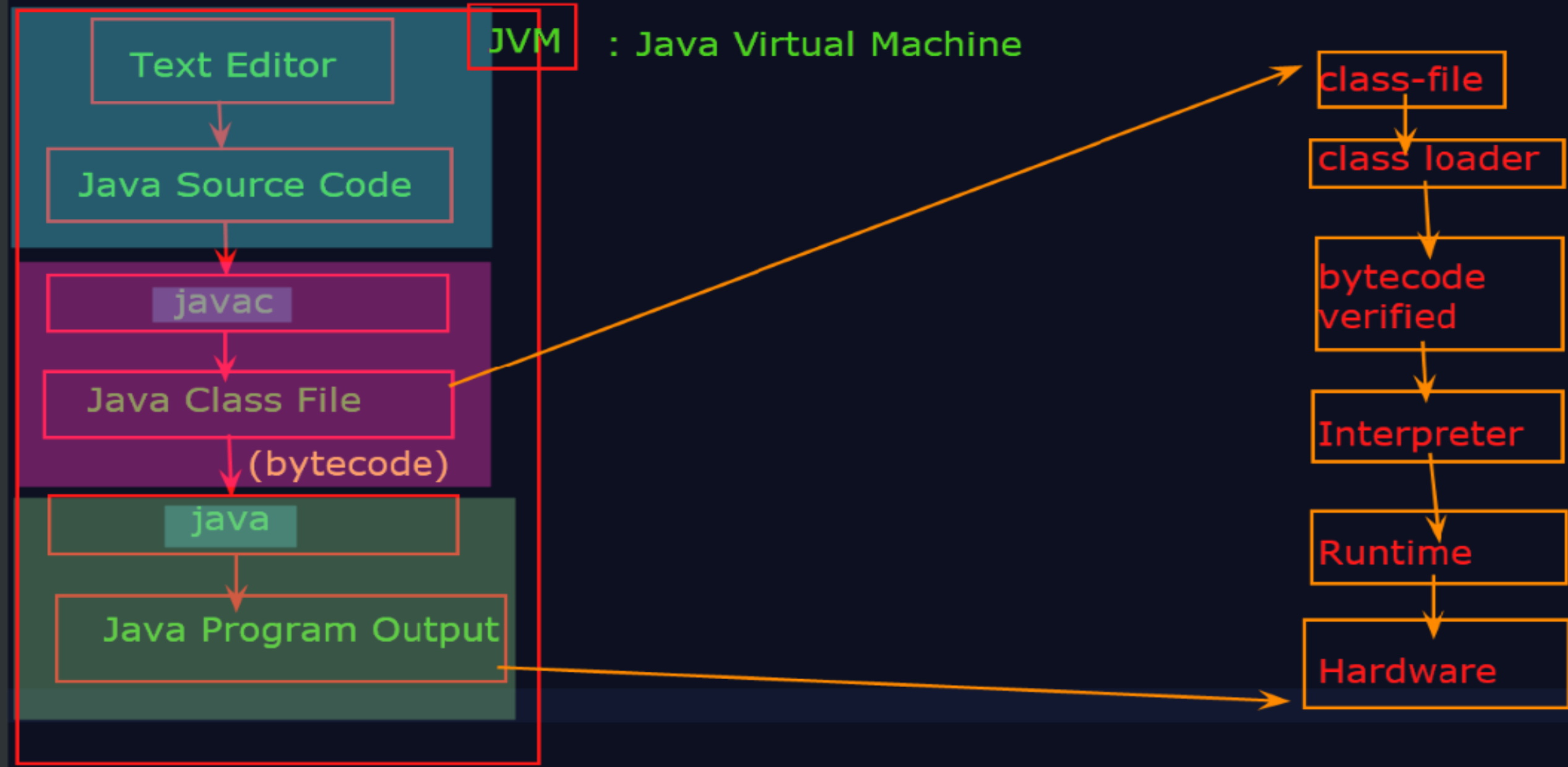
(bytecode) LLL

java → jdb

Java Program Output

HLL

★ compiler: JIT compiler
                    Just-In-time (JIT)
-to improve the performance.
-reduce the amount of time needed for compilation

Process of building and running Java application programs.

JVM : Java Virtual Machine

Text Editor

↓

Java Source Code

HLL

↓

javac

↓

Java Class File

(bytecode)

LLL

↓

java

↓

Java Program Output

HLL

Bytecode:★

-generated in .class file
-platform independent
-browser will load our bytecode much faster
-source code is not revealed to users
-Interpreter performs security checks
   screens out malicious code

★

class-file

↓

class loader

↓

bytecode verified

↓

Interpreter

↓

Runtime

↓

Hardware

Process of building and running Java application programs:

Relatioship between JDK, JRE & JVM

✓✓✓
✓Developer: D,C, E
User: E
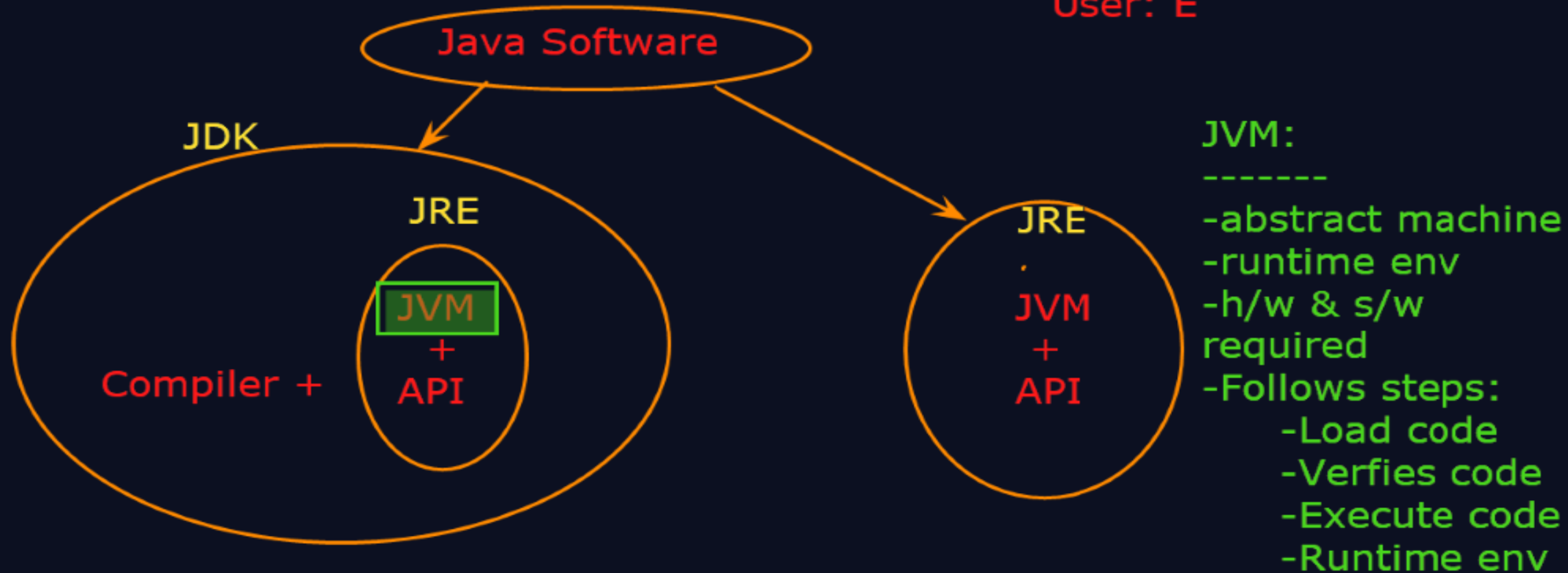
Java Software

JDK

JRE

JRE

JVM
+
API

Compiler +

JVM
+
API

JRE: Java Runtime Environment

JDK : Java DEvelopment Kit

Relatioship between JDK, JRE & JVM

Java Software

JDK

JRE

JVM
+
API

Compiler +

JRE
.
JVM
+
API

JVM:

-abstract machine
-runtime env
-h/w & s/w required
-Follows steps:
    -Load code
    -Verfies code
    -Execute code
    -Runtime env

JVM : Java Virtual Machine

JRE: Java Runtime Environment

JDK : Java DEvelopment Kit

API : Application Programming Interface
-inbuilt packages
9 packes:
----
    1. Language packages
        -classes & methods
        -java.lang.*;
    2. Utility Packages
        -java.util.*;
        -vector, Stack, List, Scanner, Date...
    3. Input/Output PAckages
        -java.io.*;
        -files input/output handling
    4. Networking Packages
        -java.net.*;
        -networking programming
        -TCP/IC

5. AWT Packages
        -java.awt.*;
        -awt:Abstract
Windowing Toolkit (GUI)
    6.Applet Package
        -java.applet.*;
    7.Sql package: java.sql.*;
        -(database
programming)