

Bài: Cài đặt môi trường Java

Xem bài học trên website để ủng hộ Kteam: [Cài đặt môi trường Java](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài trước Kteam đã giới thiệu đến bạn [LỊCH SỬ PHÁT TRIỂN & TÍNH NĂNG CỦA JAVA](#). Trong bài viết này Kteam sẽ hướng dẫn bạn **Cách cài đặt Java và đặt biến môi trường Java**. Việc cài đặt **Java Runtime Environment** để giúp ta có thể lập trình Java.

Nội dung

Bài này sẽ giới thiệu bao gồm các nội dung sau:

- Giải thích về JVM, JRE và JDK
- Cài đặt môi trường Java
- Cấu hình Java

Giải thích về JVM, JRE và JDK

JVM (Java Virtual Machine) là môi trường dùng để chạy ứng dụng được viết bằng ngôn ngữ lập trình Java. **JVM** giúp Java có thể chạy nhiều nền tảng (Platform) khác nhau, mỗi nền tảng sẽ có Platform tương ứng như **JVM** giành cho Linux, Mac, Window... nhưng cơ chế hoạt động đều như nhau nên khi ta viết ứng dụng Java trên Window vẫn có thể chạy được trên Linux, Mac và ngược lại.

JRE (Java Runtime Enviroment) bao gồm các thư viện, JVM và những thành phần bổ sung để chạy những ứng dụng được viết bằng Java. Những máy tính của người dùng bình thường chỉ cần cài đặt JRE là đủ chạy ứng dụng Java Desktop (được viết từ những thư viện AWT, Swing hay JavaFX)

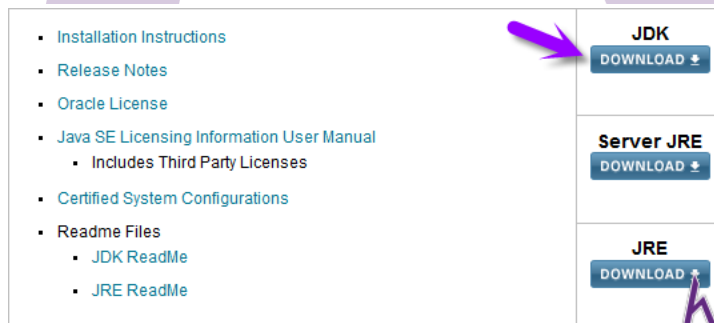
JDK (Java Development Kit) giành cho các lập trình viên Java là tập hợp JRE và những công cụ cần thiết để phát triển ứng dụng bằng Java. Ví dụ như trình biên dịch javac dùng để biên dịch java thành file **bytecode .class**.

Cài đặt môi trường Java

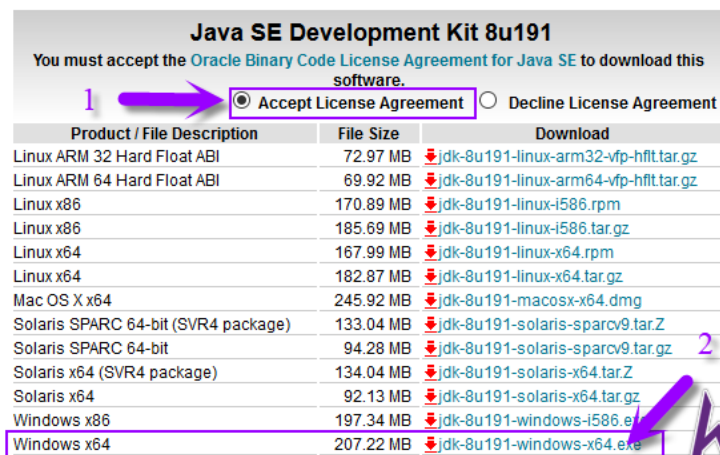
Bây giờ ta sẽ tải **JDK** để cài đặt môi trường Java. Ta sẽ tải tại trang oracle ở trang bên dưới:

www.oracle.com/technetwork/java/javase/downloads/index.html

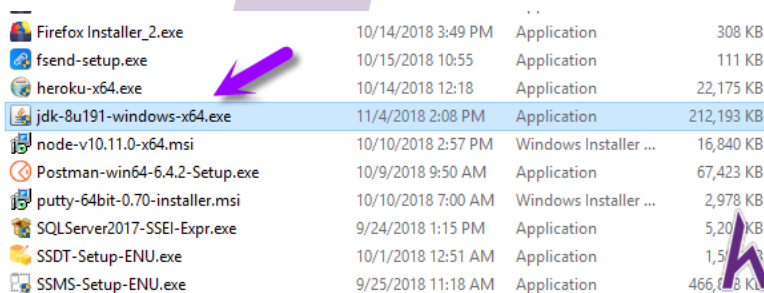
Kteam sẽ cài đặt **Java 8** là phiên bản ổn định ở thời điểm hiện tại (Nếu bạn cài đặt Java phiên bản mới hơn cũng không sao, các bài viết vẫn tập trung những kiến thức căn bản mà phiên bản Java nào cũng có)



Tiếp theo ta nhấn **Accept License Agreement** và **phiên bản jdk** dành cho hệ điều hành của mình (ở đây Kteam dùng **Windows x64**)



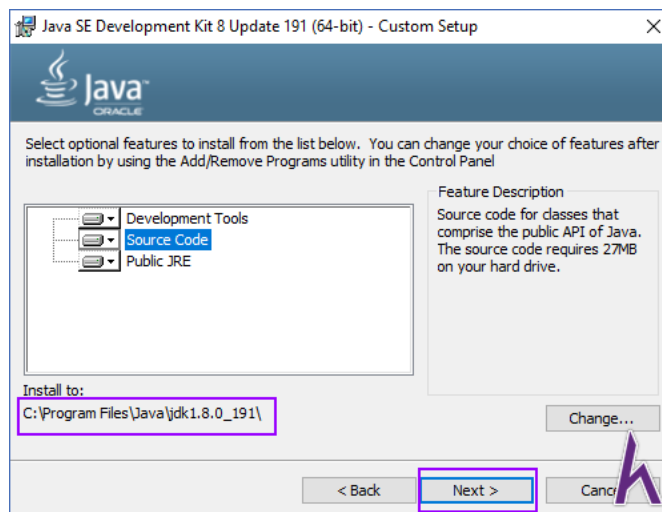
Bây giờ ta sẽ mở file cài đặt **jdk** lên



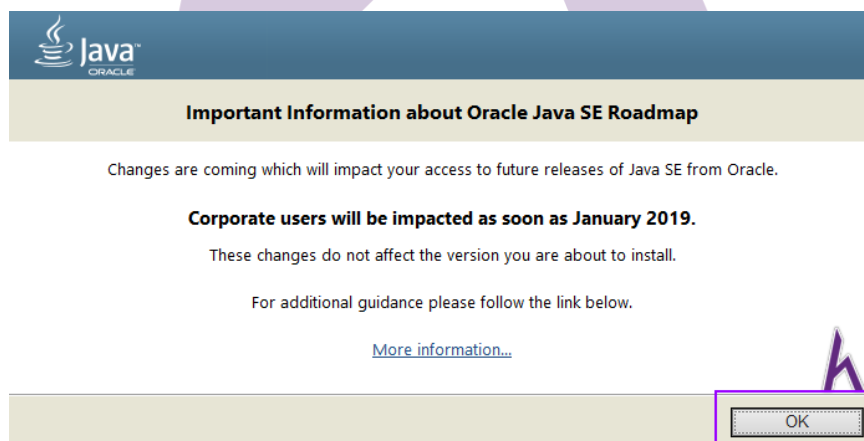
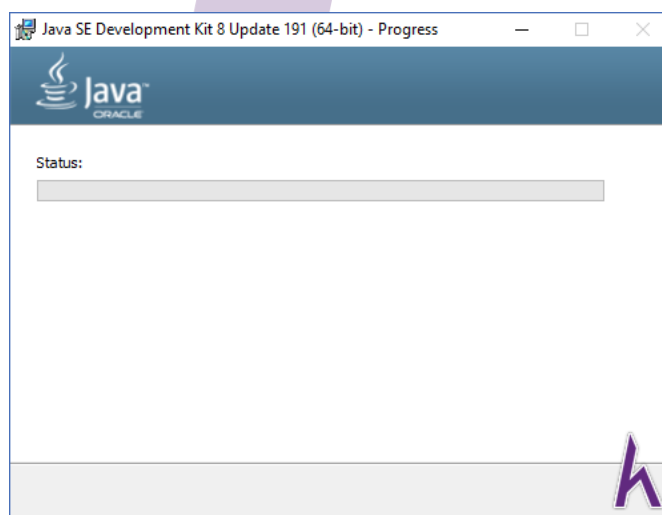
Ta sẽ nhấn theo những **ô vuông tím** mà Kteam vẽ ra



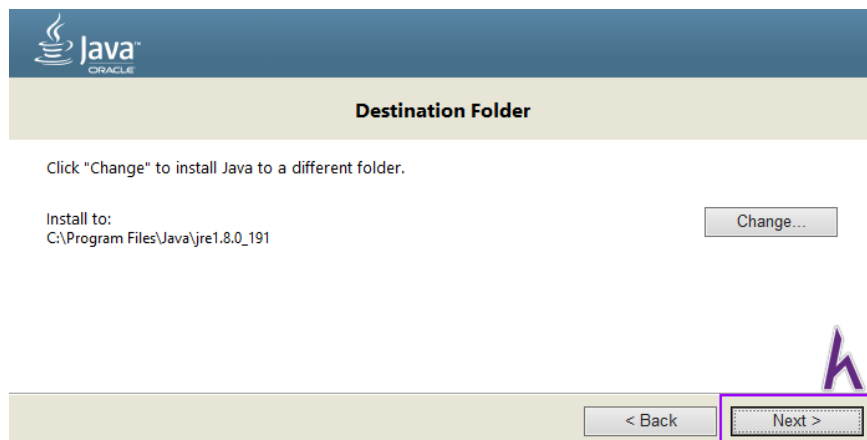
Trước khi nhấn **Next** hãy nhớ đường dẫn cài đặt **jdk** (như trong hình dưới là ở **C:\Program Files\Java\jdk1.8.0_191**) để cấu hình java.



Ta đợi chương trình cài đặt



Tiếp theo ta sẽ cài **JRE**



Ta đợi chương trình cài đặt **JRE**

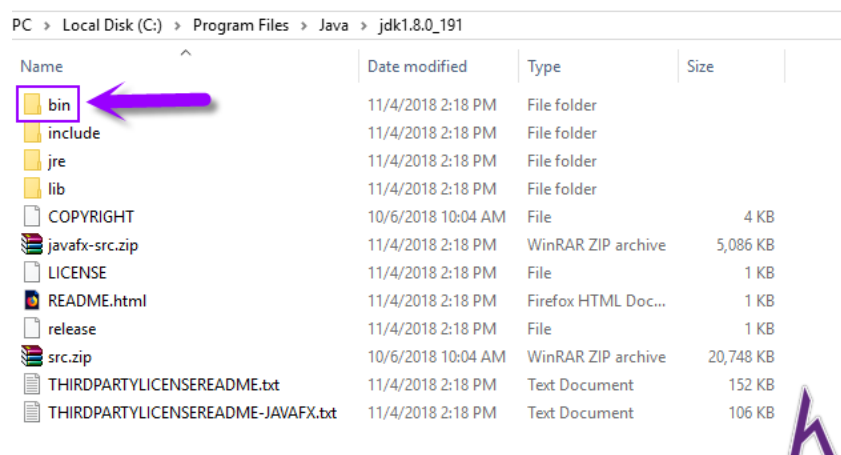


Nhấn **Close** để kết thúc

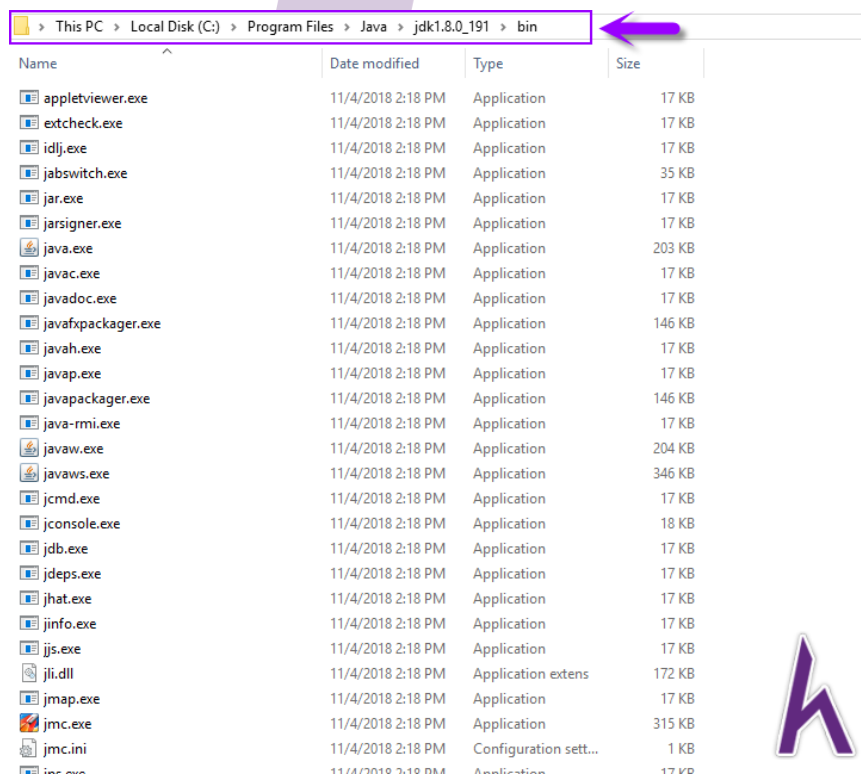


Cấu hình Java

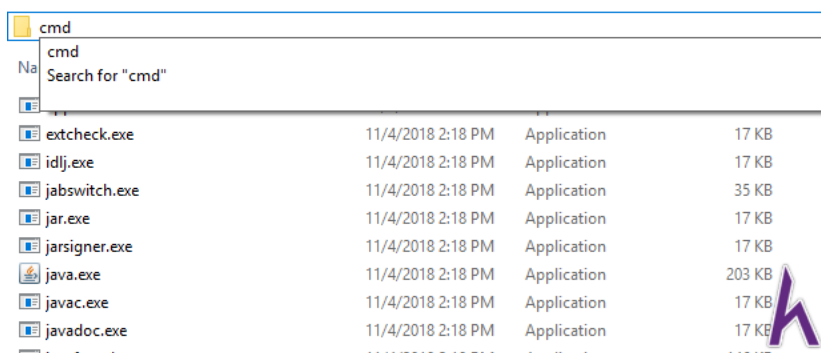
Sau khi cài đặt Java thành công ta hãy vào mục lưu **jdk**, như Kteam đã viết thì mục jdk của máy Kteam được lưu ở **C:\Program Files\Java\jdk1.8.0_191**. Sau khi vào, ta hãy mở folder **bin**



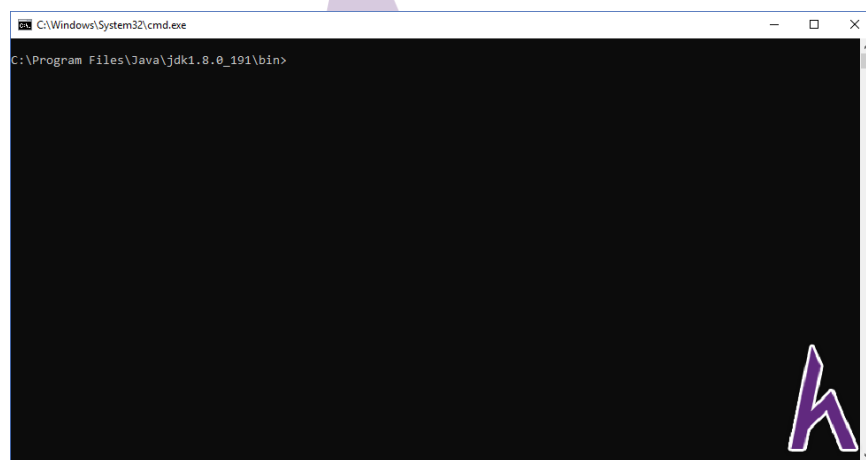
Trong mục này có nhiều file **.exe** để chạy chương trình java như **java.exe**, **javac.exe**,... Để chạy thử Kteam sẽ mở **Command Line** tại folder **bin** này. Ta chọn thanh địa chỉ phía trên



Ta xóa đi và viết **cmd** rồi **Enter**



Như vậy **Command Line** sẽ hiển thị ra.



Bây giờ ta thử gõ **java** hoặc **javac** kiểm tra xem

```

C:\Windows\System32\cmd.exe

C:\Program Files\Java\jdk1.8.0_191\bin>java
Usage: java [-options] class [args...]
           (to execute a class)
or java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                  The default VM is server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose:[class|gc|jni]
                  enable verbose output
    -version       print product version and exit
    -version:<value>
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  require the specified version to run
    -showversion   print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  include/exclude user private JREs in the version search
    -? -help       print this help message
    -X             print help on non-standard options
    -ea[:<packagename>...][:<classname>]
    -enableassertions[:<packagename>...][:<classname>]
                  enable assertions with specified granularity
    -da[:<packagename>...][:<classname>]
    -disableassertions[:<packagename>...][:<classname>]
                  disable assertions with specified granularity
    -esa | -enablesystemassertions
                  enable system assertions
    -dsa | -disablesystemassertions
                  disable system assertions
    -agentlib:<libname>[=<options>]
                  load native agent library <libname>, e.g. -agentlib:hprof
                  see also, -agentlib:jdwp=help and -agentlib:hprof=help
    -agentpath:<pathname>[=<options>]
                  load native agent library by full pathname
    -javaagent:<jarpath>[=<options>]
                  load Java programming language agent, see java.lang.instrument
    -splash:<imagepath>
                  show splash screen with specified image
See http://www.oracle.com/technetwork/java/javase/documentation/index.html for more details.

```

```

C:\Program Files\Java\jdk1.8.0_191\bin>javac
Usage: javac <options> <source files>
where possible options include:
    -g             Generate all debugging info
    -g:none        Generate no debugging info
    -g{lines,vars,source}
                  Generate only some debugging info
    -nowarn        Generate no warnings
    -verbose       Output messages about what the compiler is doing
    -deprecation   Output source locations where deprecated APIs are used
    -classpath <path>
                  Specify where to find user class files and annotation processors
    -cp <path>     Specify where to find user class files and annotation processors
    -sourcepath <path>
                  Specify where to find input source files
    -bootclasspath <path>
                  Override location of bootstrap class files
    -extdirs <dirs>
                  Override location of installed extensions
    -endorseddirs <dirs>
                  Override location of endorsed standards path
    -proc:{none,only}
                  Control whether annotation processing and/or compilation is done.
    -processor <class1>[,<class2>,<class3>...]
                  Names of the annotation processors to run; bypasses default discovery process
    -processorpath <path>
                  Specify where to find annotation processors
    -parameters    Generate metadata for reflection on method parameters
    -d <directory>
                  Specify where to place generated class files
    -s <directory>
                  Specify where to place generated source files
    -h <directory>
                  Specify where to place generated native header files
    -implicit:{none,class}
                  Specify whether or not to generate class files for implicitly referenced files
    -encoding <encoding>
                  Specify character encoding used by source files
    -source <release>
                  Provide source compatibility with specified release
    -target <release>
                  Generate class files for specific VM version
    -profile <profile>
                  Check that API used is available in the specified profile
    -version       Version information
    -help          Print a synopsis of standard options
    -Akey[=value]  Options to pass to annotation processors
    -X            Print a synopsis of nonstandard options
    -J<flag>       Pass <flag> directly to the runtime system
    -Werror        Terminate compilation if warnings occur
    @<filename>    Read options and filenames from file

```

Ở mục **bin** này ta có thể mở được chương trình java, tuy nhiên nếu ta chạy những lệnh này ở khu vực khác thì **Command Line** sẽ không nhận ra được. Ví dụ như Kteam mở một **CMD** bình thường ở Desktop.

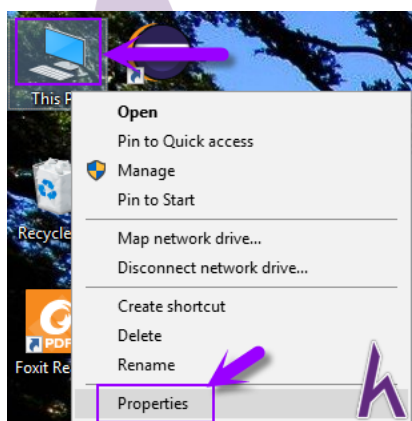
```
Command Prompt
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Windows 10>java
'java' is not recognized as an internal or external command,
operable program or batch file.

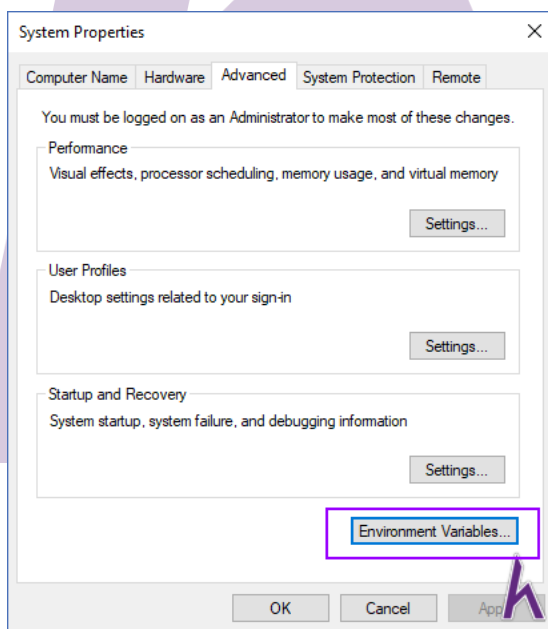
C:\Users\Windows 10>
```

Ta thấy **CMD** này không nhận biết được Java. Vì vậy ta cần phải cấu hình môi trường Java để những câu lệnh Java có thể chạy bất cứ đâu trong hệ điều hành của mình.

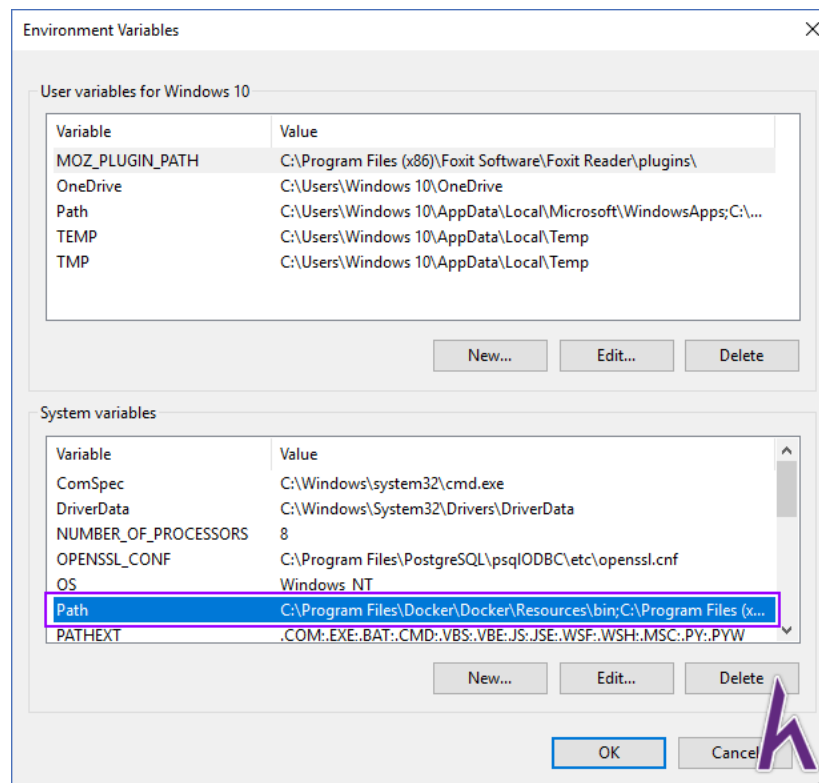
Đầu tiên nhấn chuột phải vào **This PC** chọn **Properties**



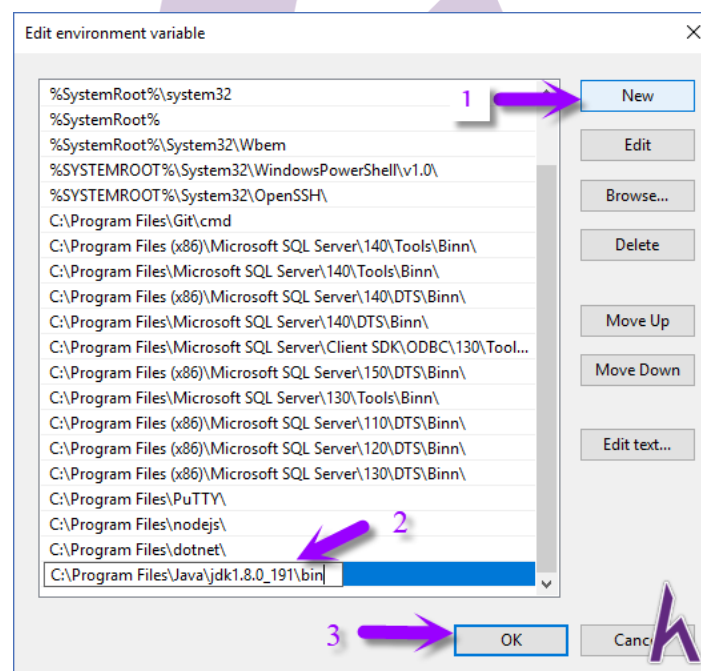
Ta chọn **Environment Variables...**



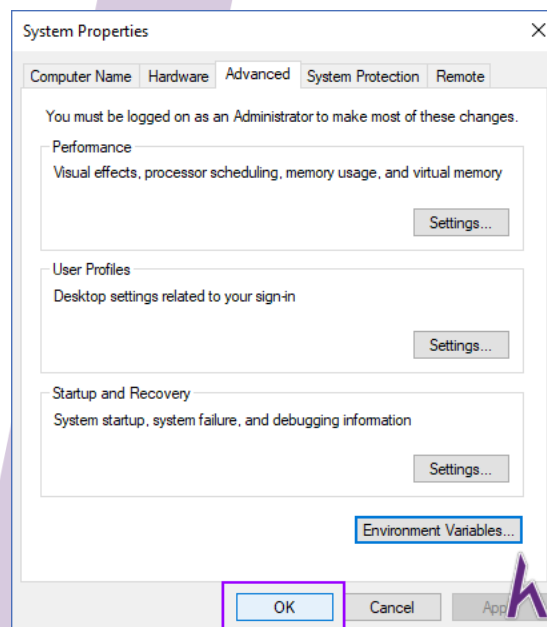
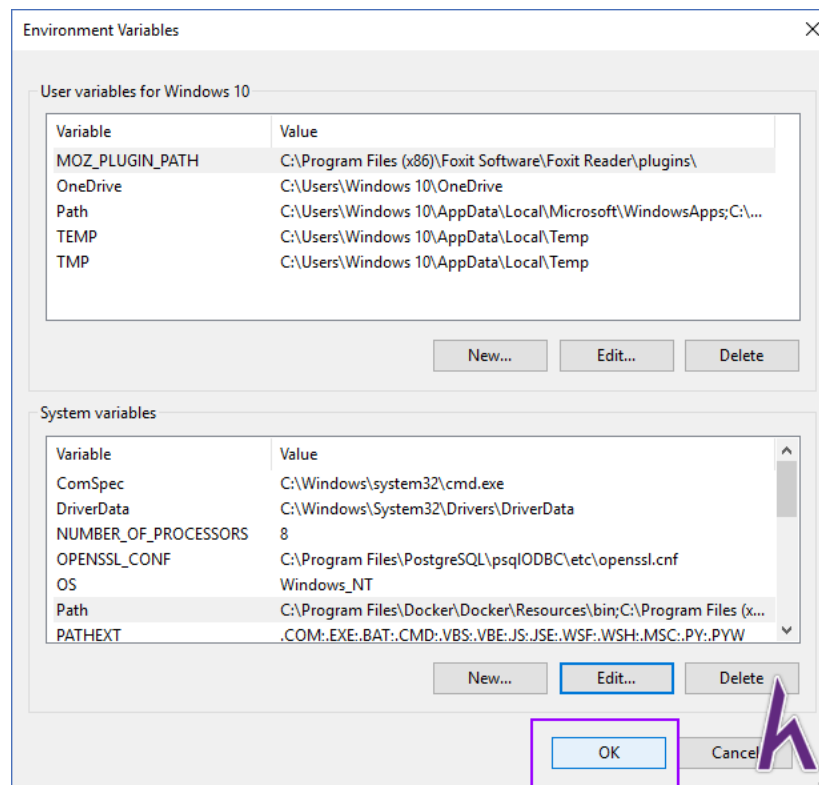
Ở phần **System variables** ta chọn mục **Path**



Ta copy đường dẫn folder **bin** của **jdk** đưa vào đây



Rồi ta nhấn hết **OK** để lưu lại



Giờ ta thử một **Command Line** bất kì và thử nhấn **java** hay **javac** để kiểm tra

```

C:\Users\Windows 10>java
Usage: java [-options] class [args...]
       (to execute a class)
    or  java [-options] -jar jarfile [args...]
       (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                   The default VM is server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                   A ; separated list of directories, JAR archives,
                   and ZIP archives to search for class files.
    -D<name>=<value>
                   set a system property
    -verbose:[class|gc|jni]
                   enable verbose output
    -version       print product version and exit
    -version:<value>
                   Warning: this feature is deprecated and will be removed
                   in a future release.
                   require the specified version to run
    -showversion   print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                   Warning: this feature is deprecated and will be removed
                   in a future release.
                   include/exclude user private JREs in the version search
    -? -help       print this help message
    -X             print help on non-standard options
    -ea[:<packagename>...[:<classname>]]
    -enableassertions[:<packagename>...[:<classname>]]
                   enable assertions with specified granularity
    -da[:<packagename>...[:<classname>]]
    -disableassertions[:<packagename>...[:<classname>]]
                   disable assertions with specified granularity
    -esa | -enablesystemassertions
                   enable system assertions
    -dsa | -disablesystemassertions
                   disable system assertions
    -agentlib:<libname>[=<options>]
                   load native agent library <libname>, e.g. -agentlib:hprof
                   see also, -agentlib:jdwp=help and -agentlib:hprof=help
    -agentpath:<pathname>[=<options>]
                   load native agent library by full pathname
    -javaagent:<jarpath>[=<options>]
                   load Java programming language agent, see java.lang.instrument
    -splash:<imagepath>
                   show splash screen with specified image
See http://www.oracle.com/technetwork/java/javase/documentation/index.html for more details.

```

Như vậy ta đã cấu hình xong **Java**

Kết luận

Như vậy chúng ta đã tìm hiểu cách cài đặt và cấu hình môi trường Java

Ở bài sau, Kteam sẽ giới thiệu đến bạn về [VIẾT CHƯƠNG TRÌNH JAVA ĐẦU TIÊN](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.