

Lập trình hướng đối tượng

Giảng viên: Nguyễn Thị Thu Trang, trangntt@soict.hust.edu.vn

Phòng thí nghiệm 01: Thiết lập môi trường và kiến thức cơ bản về Java

Trong phòng thí nghiệm này, bạn sẽ thực hành với:

- Thiết lập môi trường phát triển
- Viết một số mã của Java và biên dịch nó bằng Java JDK trên Eclipse IDE

Từ khóa: JDK, JRE, cài đặt Java, trình soạn thảo văn bản lập trình, IDE, git, github

0 Nộp bài tập

Đối với lớp học phòng thí nghiệm này, bạn sẽ phải nộp bài làm của mình hai lần, cụ thể:

- **Ngay sau lớp học lab:** đối với thời hạn này, bạn nên đưa bất kỳ công việc nào bạn đã làm trong thời gian lớp học lab vào github.
- **10 giờ tối ngày sau lớp học:** đối với thời hạn này, bạn nên bao gồm **tất cả** hai chương trình (2.2.5 & 2.2.6) và sáu ứng dụng trong phần bài tập (6.1, 6.2, 6.3, 6.4, 6.5 & 6.6) của phòng thí nghiệm này, vào một thư mục có tên là "**Lab01**" và đẩy nó đến nhánh chính của kho lưu trữ hợp lệ.

Mỗi học sinh phải nộp bài làm của mình và không cho hoặc nhận viện trợ không được phép. Nếu không, chúng tôi sẽ áp dụng các phương pháp đo lường cực đoan để ngăn chặn gian lận. Vui lòng viết câu trả lời cho tất cả các câu hỏi vào một tệp văn bản có tên "answers.txt" và gửi nó trong kho lưu trữ của bạn.

1 Bắt đầu

1.1 Bộ phát triển Java

Java Platform, Standard Edition Development Kit (JDK) là một môi trường phát triển để xây dựng các ứng dụng, applet và thành phần bằng ngôn ngữ lập trình Java. Có nhiều bản phát hành của nền tảng (mới nhất là JDK 15) có sẵn để tải xuống. Trong số đó, JDK 8 là phiên bản Java được sử dụng rộng rãi nhất và nó cũng là bản phát hành hỗ trợ dài hạn (LTS) cuối cùng có chứa JavaFX (mà chúng tôi sẽ làm việc rộng rãi sau này trong khóa học này). Vì những lý do trên, **JDK 8 bắt buộc phải được cài đặt cho tất cả các phòng thí nghiệm trong khóa học này**. Tuy nhiên, nếu bạn đã cài đặt một phiên bản mới hơn, bạn vẫn có thể cài đặt JavaFX riêng biệt (sẽ có một hướng dẫn cài đặt trong phòng thí nghiệm JavaFX trong trường hợp này).

Các bước cài đặt cho JDK 8 được minh họa như sau:

Bước 1: Kiểm tra xem JDK đã được cài đặt sẵn chưa

1. Mở Command Prompt (trên Windows. Nhấn Windows + R để mở hộp "Chạy". Nhập "cmd" và sau đó nhấp vào "OK" để mở Command Prompt thông thường) hoặc Terminal (trên Linux hoặc macOS).
2. Đưa ra lệnh sau.

```
$ javac -version
```
3. Trong trường hợp số phiên bản JDK được trả về (ví dụ: JDK x.x.x), thì JDK đã được cài đặt. Khi phiên bản JDK trước 1.8, thông báo "*Không tìm thấy lệnh 'javac'*" hoặc thông báo "*'javac' không được nhận*".

dạng là lệnh nội bộ hoặc bên ngoài, chương trình có thể hoạt động hoặc tệp hàng loạt.", hãy tiến hành bước 2 để cài đặt Oracle JDK 8. Nếu không, hãy chuyển sang 1.2.

Lưu ý: Linux thường chọn OpenJDK làm JDK mặc định vì OpenJDK là mã nguồn mở. Tuy nhiên, Oracle JDK không hoàn toàn tương thích với OpenJDK và bạn nên sử dụng Oracle JDK.

Bước 2: Tải xuống Oracle JDK 8

1. Truy cập trang web tải xuống Java SE Development Kit (JDK) 8 tại liên kết sau.
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
2. Tải xuống tệp cài đặt, chọn tệp thích hợp cho hệ điều hành của bạn, trong phần "*Bộ công cụ phát triển Java SE 8u241*". Tệp được đề xuất cho Linux là tệp Lưu trữ nén. Chúng tôi có thể cần Tài khoản Oracle để tải xuống cho Giấy phép Oracle JDK đã thay đổi cho các bản phát hành kể từ ngày 16 tháng 4 năm 2019.

Bước 3: Cài đặt và cấu hình

- Windows:

1. Cài đặt Oracle JDK 8. Chạy trình cài đặt đã tải xuống và làm theo hướng dẫn.
2. Cấu hình. Khởi chạy Bảng điều khiển → Hệ thống và Hệ thống → Bảo mật → Cài đặt hệ thống nâng cao → Biến môi trường trong tab Nâng cao. Trong danh sách dưới "Biến hệ thống", bạn cần tìm hai biến:
 - Cái đầu tiên là JAVA_HOME, hãy kiểm tra xem nó đã tồn tại chưa, nếu chưa thì thêm mới bằng cách chọn "Mới..." và đặt tên biến là JAVA_HOME. Sau đó, bạn cần đặt giá trị của nó làm đường dẫn đến nơi đặt cài đặt JDK, đó là thư mục "jdk-x" trong thư mục "Java", ví dụ: C:\Program Files\Java\jdk1.8.0_241.
 - Biến thứ hai là Path, bạn cần sửa đổi nó bằng cách thêm mục sau vào đó: %JAVA_HOME%\bin

- Linux:

1. Tạo thư mục cài đặt. Chúng ta sẽ cài đặt Oracle JDK 8 trong thư mục `"/usr/local/java"`.

```
$ cd /usr/cục bộ
$ sudo mkdir java
```

2. Giải nén gói đã tải xuống (ví dụ: `jdk-8u241-linux-x64.tar.gz`) vào thư mục cài đặt.

```
$ cd /usr/cục bộ/java
$ sudo tar XJVF ~/download/jdk-8u241-linux-x64.his.gj
x: trích xuất, z: để giải nén gz, v: verbose, f: tên tệp
```

3. Thông báo cho Linux sử dụng JDK/JRE này

Thiết lập vị trí của java, javac và javaws

```
$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/java/jdk1.8.0_241/bin/java" 1
--cài đặt mức độ ưu tiên đường dẫn tên liên kết tượng trưng
$ sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/local/java/jdk1.8.0_241/bin/javac" 1
$ sudo update-alternatives --install "/usr/bin/javaws" "javaws"
"/usr/local/java/jdk1.8.0_241/bin/javaws" 1
```

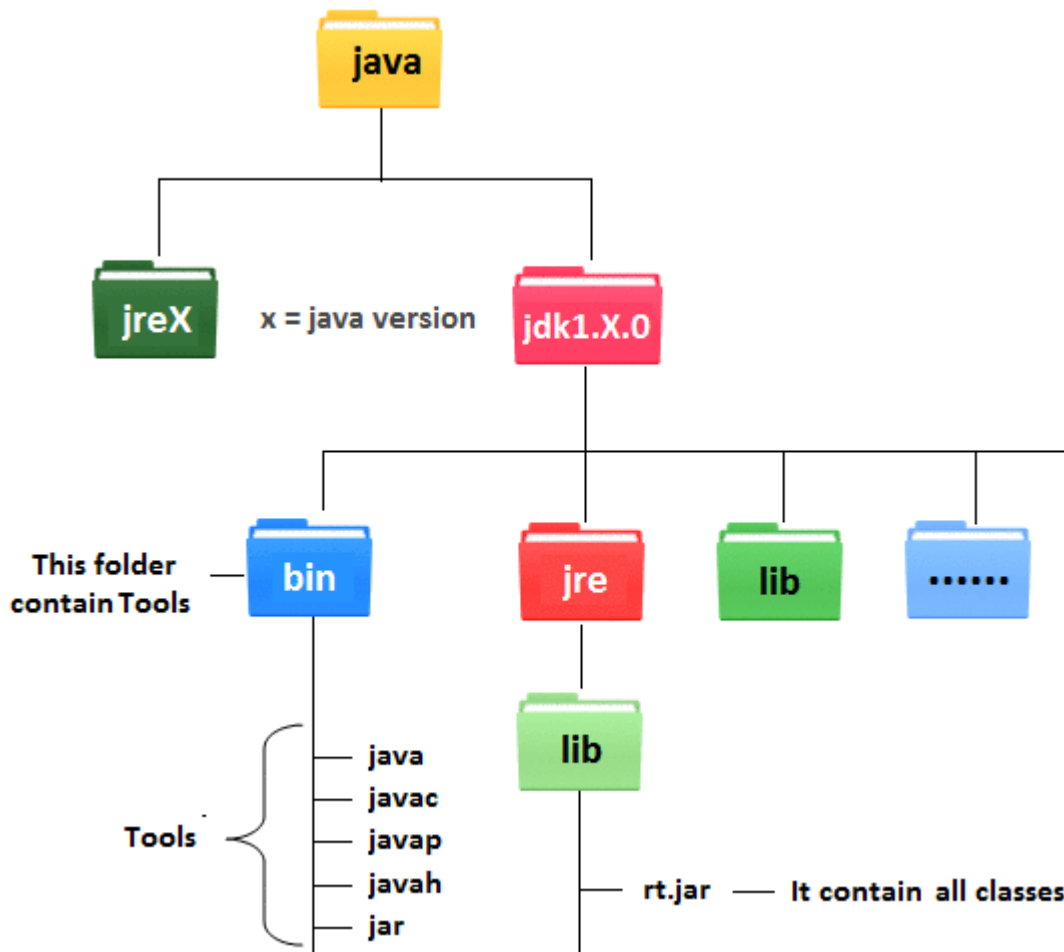
Sử dụng Oracle JDK / JRE này làm mặc định

```
$ sudo update-alternatives --set java /usr/local/java/jdk1.8.0_241/bin
/ java
--đặt đường dẫn tên
$ sudo update-alternatives --set javac /usr/local/java/jdk1.8.0_241/bin/javac
$ sudo update-alternatives --set javaws /usr/local/java/jdk1.8.0_241/bin/javaws
```

- **MacOS:** Nhấp đúp vào tệp DMG và làm theo hướng dẫn.

Bước 4: Xác minh cài đặt JDK. Đưa ra lệnh sau.

```
$ javac -version
```



Hình 1. Môi trường phát triển.

1.2 Trình soạn thảo văn bản lập trình

Đối với phần tiếp theo (phần 2), bạn có thể sử dụng bất kỳ trình soạn thảo văn bản nào để viết mã nguồn java của mình. Ở đây, trình soạn thảo văn bản Notepad được sử dụng như một hình minh họa.

2 Các chương trình đầu tiên

2.1 Các bước lập trình Java

Các bước viết một chương trình Java được minh họa trong các bước sau và trong Hình 2.

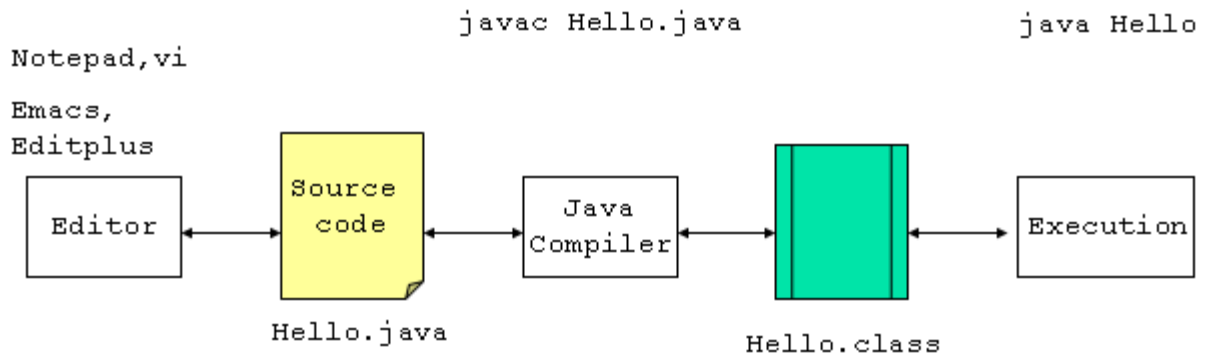
Bước 1: Viết mã nguồn như mã trong Hình 3 và lưu vào, ví dụ: tệp "HelloWorld.java".

Bước 2: Biên dịch mã nguồn thành mã byte di động Java (hoặc mã máy) bằng trình biên dịch Java của JDK bằng cách đưa ra lệnh sau.

```
$javac HelloWorld.java
```

Bước 3: Chạy bytecode đã biên dịch bằng cách sử dụng Java Runtime của JDK bằng cách đưa ra lệnh sau.

```
$java HelloWorld
```



Hình 2. Biên dịch ứng dụng Java bằng dòng lệnh

```

HelloWorld.java:
    this is a class
    class name
    start of the class
public class HelloWorld {
    method name
    public static void main(String[] args) {
        a statement
        it says print to
        standard output
        System.out.println("Hello, world");
    }
}
    end of the class
    public, so that
    everyone can access
```

Hình 3. Ứng dụng Java đầu tiên

Kết quả được thể hiện trong Hình 4.

```
%> javac HelloWorld.java
%> java HelloWorld
Hello, world
```

Hình 4. Kết quả của ứng dụng Java đầu tiên

Để minh họa rõ hơn, chúng ta có thể xem các video demo sau.

<https://www.youtube.com/watch?v=G1ubVOI9IBw>

https://www.youtube.com/watch?v=2Xa3Y4xz8_s

2.2 Các chương trình Java đầu tiên

2.2.1 Viết, biên dịch ứng dụng Java đầu tiên:

- **Bước 1: Tạo một tệp mới.** Từ giao diện Notepad, chọn Tệp → Tệp mới.
- **Bước 2: Lưu tệp.** Từ giao diện Notepad, chọn Tệp tin → Lưu. Duyệt qua thư mục mong muốn, thay đổi file tên thành "*HelloWorld.java*" và nhấn nút "*Lưu*".
- **Bước 3: Viết mã nguồn.** Mã nguồn được thể hiện trong Hình 5.

```
1 //Example 1: HelloWorld.java
2 //Text-printing program
3 public class HelloWorld {
4
5     public static void main(String args[]){
6         System.out.println("Xin chao \n cac ban!");
7         System.out.println("Hello \t world!");
8
9     } // end of method main
10 }
```

Hình 5. Ứng dụng Java đầu tiên

- **Bước 4: Biên dịch.** Trên Command Prompt hoặc Terminal, hãy thay đổi thư mục làm việc hiện tại¹ vào thư mục nơi chúng ta đã lưu mã nguồn. Sau đó đưa ra các lệnh sau.

```
$javac HelloWorld.java
$java HelloWorld
```

- **Bước 5:** Sau khi hoàn thành nhiệm vụ này, bạn nên thêm và cam kết tất cả công việc của mình vào nhánh master. Làm theo hai bước sau để đạt được kết quả mong muốn:
 - **Bước 1:** Thêm tất cả các thay đổi trong thư mục hiện tại và các thư mục con của nó vào khu vực dàn dựng
- ```
(master) $ git add .
```
- **Bước 2:** Cam kết thay đổi trong kho lưu trữ cục bộ.
- ```
(master) $ git commit -m "Hoàn thành bài tập 2.2.1"
```

¹ Trong các hệ điều hành khác nhau, *lệnh cd <tên thư mục mong muốn>* (*cd* là viết tắt của thư mục thay đổi) cho phép chúng ta thay đổi thư mục làm việc hiện tại thành thư mục mong muốn. Bên cạnh đó, trong Windows 10, để truy cập một ổ đĩa khác, chúng ta nhập chữ cái của ổ đĩa, theo sau là ":". Ví dụ: để thay đổi ổ đĩa làm việc hiện tại thành ổ đĩa D, chúng tôi đưa ra lệnh "*d:*"

2.2.2 Viết, biên dịch chương trình Java hộp thoại đầu tiên

- **Bước 1: Tạo một tệp mới.** Từ giao diện Notepad, chọn Tệp → Tệp mới.
- **Bước 2: Lưu tệp.** Từ giao diện Notepad, chọn Tệp tin → Lưu. Duyệt qua thư mục mong muốn, thay đổi file tên thành " *FirstDialog.java*", và nhấp vào nút "*Lưu*".
- **Bước 3: Viết mã nguồn.** Mã nguồn được thể hiện trong Hình 6

```
1 // Example 2: FirstDialog.java
2 import javax.swing.JOptionPane;
3 public class FirstDialog{
4     public static void main(String[] args){
5         JOptionPane.showMessageDialog(null,"Hello world! How are you?");
6         System.exit(0);
7     }
8 }
```

Hình 6. Ứng dụng Java hộp thoại đầu tiên

- **Bước 4: Biên dịch.** Trên Command Prompt hoặc Terminal, hãy thay đổi thư mục làm việc hiện tại vào thư mục nơi chúng ta đã lưu mã nguồn. Đưa ra các lệnh sau.

```
$ javac FirstDialog.java
$ java Hộp thoại đầu tiên
```

- Bước 5:** Như bước 5 trong mục 2.2.1, thêm và commit công việc của bạn bằng cách sử dụng 'git add' và 'git commit -m <message>'.

2.2.3 Viết, biên dịch hộp thoại đầu vào ứng dụng Java

- **Bước 1: Tạo một tệp mới.** Từ giao diện Notepad, chọn Tệp → Tệp mới.
- **Bước 2: Lưu tệp.** Từ giao diện Notepad, chọn Tệp tin → Lưu. Duyệt qua thư mục mong muốn, thay đổi file tên thành " *HelloNameDialog.java*" và nhấp vào nút "*Lưu*".
- **Bước 3: Viết mã nguồn.** Mã nguồn được hiển thị trong Hình 7

```

1 // Example 3: HelloNameDialog.java
2 import javax.swing.JOptionPane;
3 public class HelloNameDialog{
4     public static void main(String[] args){
5         String result;
6         result = JOptionPane.showInputDialog("Please enter your name:");
7         JOptionPane.showMessageDialog(null, "Hi " + result + "!");
8         System.exit(0);
9     }
10 }

```

Hình 7. Hộp thoại đầu vào đầu tiên Ứng dụng Java

- **Bước 4: Biên dịch.** Trên Command Prompt hoặc Terminal, hãy thay đổi thư mục làm việc hiện tại vào thư mục nơi chúng ta đã lưu mã nguồn. Đưa ra các lệnh sau.

```

$ javac HelloNameDialog.java
$ java HelloNameDialog

```

- **Bước 5:** Như bước 5 trong mục 2.2.1, thêm và commit công việc của bạn bằng cách sử dụng 'git add' và 'git commit -m <message>'.

2.2.4 Viết, biên dịch và chạy ví dụ sau:

- **Bước 1: Tạo một tệp mới.** Từ giao diện Notepad, chọn Tệp → Tệp mới.
- **Bước 2: Lưu tệp.** Từ giao diện Notepad, chọn Tệp → Lưu. Duyệt qua thư mục mong muốn, thay đổi file tên thành "ShowTwoNumbers.java" và nhấp vào nút "Lưu".
- **Bước 3: Viết mã nguồn.** Mã nguồn được thể hiện trong Hình 8

```

1 // Example 5: ShowTwoNumbers.java
2 import javax.swing.JOptionPane;
3 public class ShowTwoNumbers {
4     public static void main(String[] args){
5         String strNum1, strNum2;
6         String strNotification = "You've just entered: ";
7
8         strNum1 = JOptionPane.showInputDialog(null,
9             "Please input the first number: ", "Input the first number",
10             JOptionPane.INFORMATION_MESSAGE);
11         strNotification += strNum1 + " and ";
12
13         strNum2 = JOptionPane.showInputDialog(null,
14             "Please input the second number: ", "Input the second number",
15             JOptionPane.INFORMATION_MESSAGE);
16         strNotification += strNum2;
17
18         JOptionPane.showMessageDialog(null, strNotification,
19             "Show two numbers", JOptionPane.INFORMATION_MESSAGE);
20         System.exit(0);
21     }
22 }

```

Hình 8. Ứng dụng Java hiển thị hai số đã nhập và tổng của chúng

- **Bước 4: Biên dịch.** Trên Command Prompt hoặc Terminal, hãy thay đổi thư mục làm việc hiện tại vào thư mục nơi chúng ta đã lưu mã nguồn. Đưa ra các lệnh sau.

```

$ javac ShowTwoNumbers.java
$ java ShowTwoNumbers

```

- **Bước 5:** Như bước 5 trong mục 2.2.1, thêm và commit công việc của bạn bằng cách sử dụng 'git add' và 'git commit -m <message>'.

2.2.5 Viết một chương trình để tính tổng, chênh lệch, tích và thương của 2 số đôi do người dùng nhập.

Ghi chú

- Để chuyển đổi từ String sang double, bạn có thể sử dụng
`double num1 = Double.parseDouble(strNum1)`
- Kiểm tra số chia của phép chia
- Đừng quên thêm và cam kết công việc của bạn bằng lệnh 'git add.' và 'git commit -m <message>'

2.2.6 Viết một chương trình để giải quyết:

Để đơn giản, chúng tôi chỉ xem xét gốc thực của các phương trình trong nhiệm vụ này.

- Phương trình bậc một (phương trình tuyến tính) với một biến

Lưu ý: Phương trình bậc một với một biến có thể có một dạng như $ax + b = 0$ ($a \neq 0$)

Bạn nên xử lý trường hợp người dùng nhập giá trị 0 cho a.

Đừng quên thêm và cam kết công việc của bạn bằng lệnh '**git add.**' và '**git commit -m <message>**'

- Hệ phương trình bậc một (hệ tuyến tính) với hai biến

Lưu ý: Một hệ phương trình bậc một với hai biến và có thể được viết như sau. $x_1 x_2$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

Bạn nên xử lý trường hợp các giá trị của các hệ số tạo ra vô số nghiệm và trường hợp chúng không tạo ra lời giải.

Gợi ý:

Sử dụng các yếu tố quyết định sau:

$$\begin{aligned} D &= \begin{vmatrix} a_{11} & a_{12} & a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12} \\ D_1 &= \begin{vmatrix} b_1 & a_{12} & b_2 & a_{22} \end{vmatrix} = b_1a_{22} - b_2a_{12} \\ D_2 &= \begin{vmatrix} a_{11} & b_1 & a_{21} & b_2 \end{vmatrix} \\ &= a_{11}b_2 - a_{21}b_1 \end{aligned}$$

- Phương trình bậc hai với một biến

Lưu ý: Một phương trình bậc hai với một biến (tức là phương trình bậc hai) có thể có một dạng như , trong đó x là biến và a, b và c là hệ số (). $ax^2 + bx + c = 0$ ($a \neq 0$)

Bạn nên xử lý trường hợp các giá trị của các hệ số tạo ra một căn kép và trường hợp chúng không tạo ra gốc. Bạn cũng nên xử lý trường hợp người dùng nhập giá trị 0 cho a.

Gợi ý:

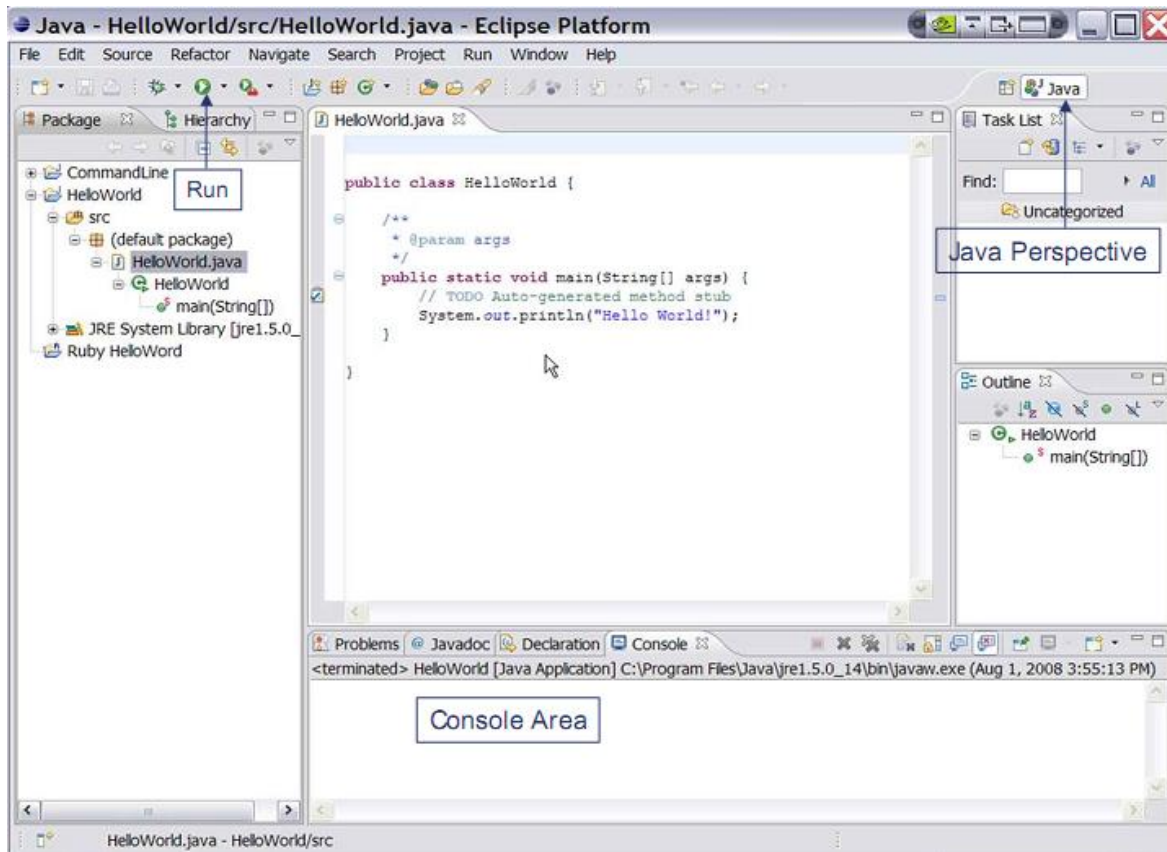
Sử dụng chất phân biệt $\Delta = b^2 - 4ac$

Sau khi hoàn thành mã trong phần 2, bạn nên đẩy tất cả các thay đổi của mình, bao gồm cả nhiệm vụ 2.2.1 đến 2.2.6 vào nhánh **master** của kho lưu trữ hợp lệ mà bạn đã tạo.

Gợi ý: Bạn nên sử dụng "git push origin <tên của nhánh mà bạn muốn đẩy vào kho lưu trữ từ xa>", trong trường hợp này là "**master**", để đẩy tất cả các công việc của bạn lên **nhánh master**.

3 Giới thiệu về Eclipse / Netbean

Trong phần trước, chúng tôi đã viết các ứng dụng Java đầu tiên của mình trong trình soạn thảo văn bản lập trình như Notepad. Từ phòng thí nghiệm này trở đi, chúng tôi sử dụng một môi trường phát triển tích hợp, được gọi là IDE, giống như một trình soạn thảo văn bản, nhưng cung cấp nhiều tính năng khác nhau như sửa đổi, biên dịch và gỡ lỗi phần mềm. Một số IDE phổ biến nhất cho Java là JetBrains IntelliJ, NetBeans và Eclipse. Trong khóa học này, chúng tôi sử dụng Eclipse cho các trình diễn của chúng tôi.



Hình 9. IDE Nhật thực

Hướng dẫn cài đặt:

Lưu ý: Bạn nên cài đặt Java 8 hoặc phiên bản mới hơn trước khi cài đặt IDE.

Trong hướng dẫn này, chúng tôi không cần trình cài đặt; chúng tôi chỉ cần tải xuống tệp ZIP và giải nén chúng.

- Netbeans: Tải xuống tệp nhị phân tại liên kết sau. Đọc README.html để biết thêm chi tiết. Ứng dụng nằm bên trong **thư mục bin**.

<https://www.apache.org/dyn/closer.cgi/netbeans/netbeans/11.2/netbeans-11.2-bin.zip>

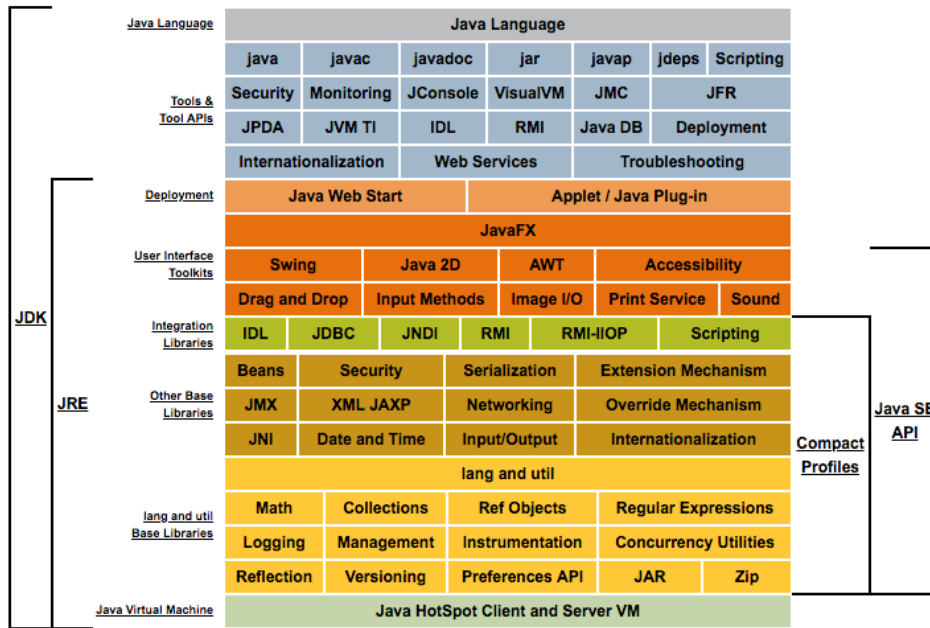
Nếu bạn muốn sử dụng các phiên bản trước Apache Netbeans, bạn có thể xem chúng ở [đây](#) (điều này có thể không tương thích với các phiên bản Java sau này).

- Eclipse: Chúng tôi khuyên bạn nên sử dụng **Eclipse IDE cho các nhà phát triển Java doanh nghiệp**. Tải xuống tệp nhị phân phù hợp tại liên kết sau. <https://www.eclipse.org/downloads/packages/>

4 Trợ giúp Javadocs:

- Mở index.html trong thư mục tài liệu (tải xuống từ <https://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html>)

Description of Java Conceptual Diagram



Hình 10. Sơ đồ khái niệm Java

- Nhấp vào liên kết [Java SE API](#)

Java™ Platform
Standard Ed. 8

All Classes All Profiles

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im
java.awt.im.spi

All Classes

AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractAnnotationValueVisitor8
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractChronology
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractElementVisitor8
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry
AbstractMap.SimpleImmutableEntry
AbstractMarshallerImpl
AbstractMethodError
AbstractObservableSynchronizer
AbstractPreferences
AbstractProcessor
AbstractQueue
AbstractQueueLongSynchronizer
AbstractQueueSynchronizer
AbstractRegionPainter
AbstractRegionPainter.PaintContext

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Java™ Platform, Standard Edition 8
API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

Profiles

- compact1
- compact2
- compact3

Packages

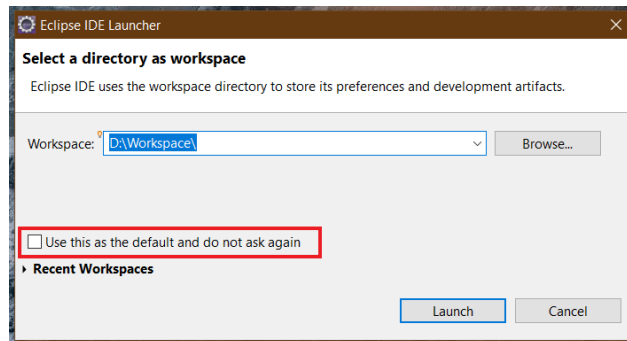
Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing beans -- components based on the JavaBeans™ architecture.

Hình 11. Java SE API

- Khung trên cùng bên trái: tất cả các gói trong API Java
- Khung dưới cùng bên trái: các lớp tương ứng trong gói đã chọn ở trên
- Khung bên phải: Thông tin chi tiết
- Nhấp vào một khung và tìm thông tin cần thiết (Ctrl + F)

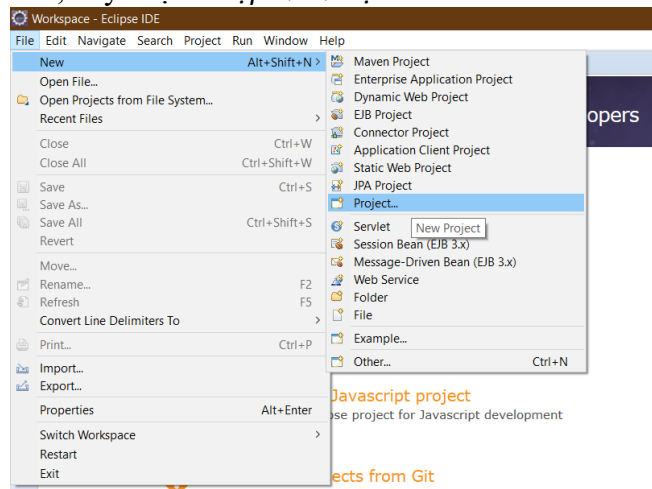
5 Dự án Java đầu tiên của bạn

1. Từ thư mục cài đặt Eclipse, chạy Eclipse IDE.
2. Trong cửa sổ Eclipse IDE Launcher, chọn thư mục không gian làm việc của bạn nơi bạn muốn lưu (các) dự án. Nếu bạn muốn sử dụng thư mục đã chọn làm mặc định, hãy chọn hộp. Sau đó, nhấp vào Khởi động cái nút.



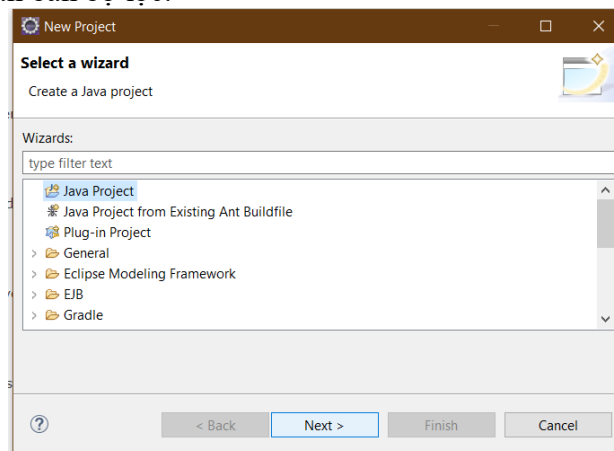
Hình 12. Cửa sổ Eclipse Launcher

3. Để tạo một dự án Java mới, hãy chọn **Tệp** → **Dự án mới...**



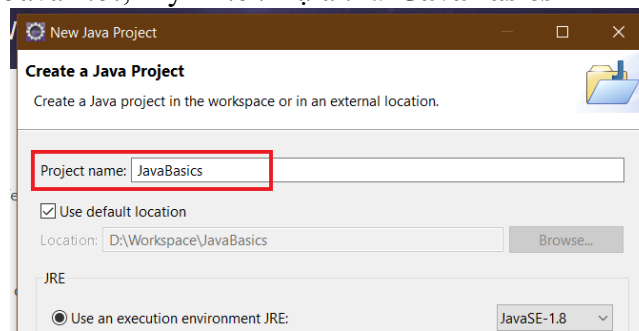
Hình 13. Tạo dự án Java mới

4. Trên cửa sổ bật lên, chọn **Dự án Java**, sau đó nhấp vào **nút Tiếp theo >**. Nếu bạn không thể tìm thấy nó, hãy nhập văn bản bộ lọc.



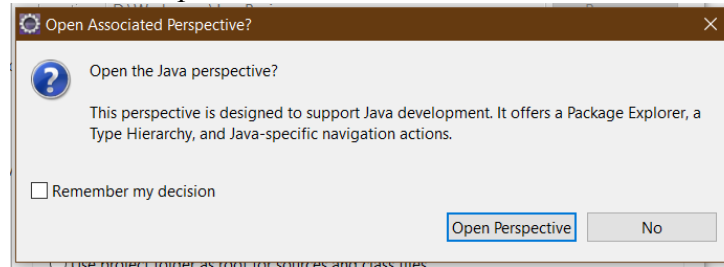
Hình 14. Cửa sổ dự án mới

5. Trên cửa sổ Dự án Java mới, hãy để tên Dự án là **"JavaBasics"**. Sau đó, nhấp vào Kết thúc cái nút.



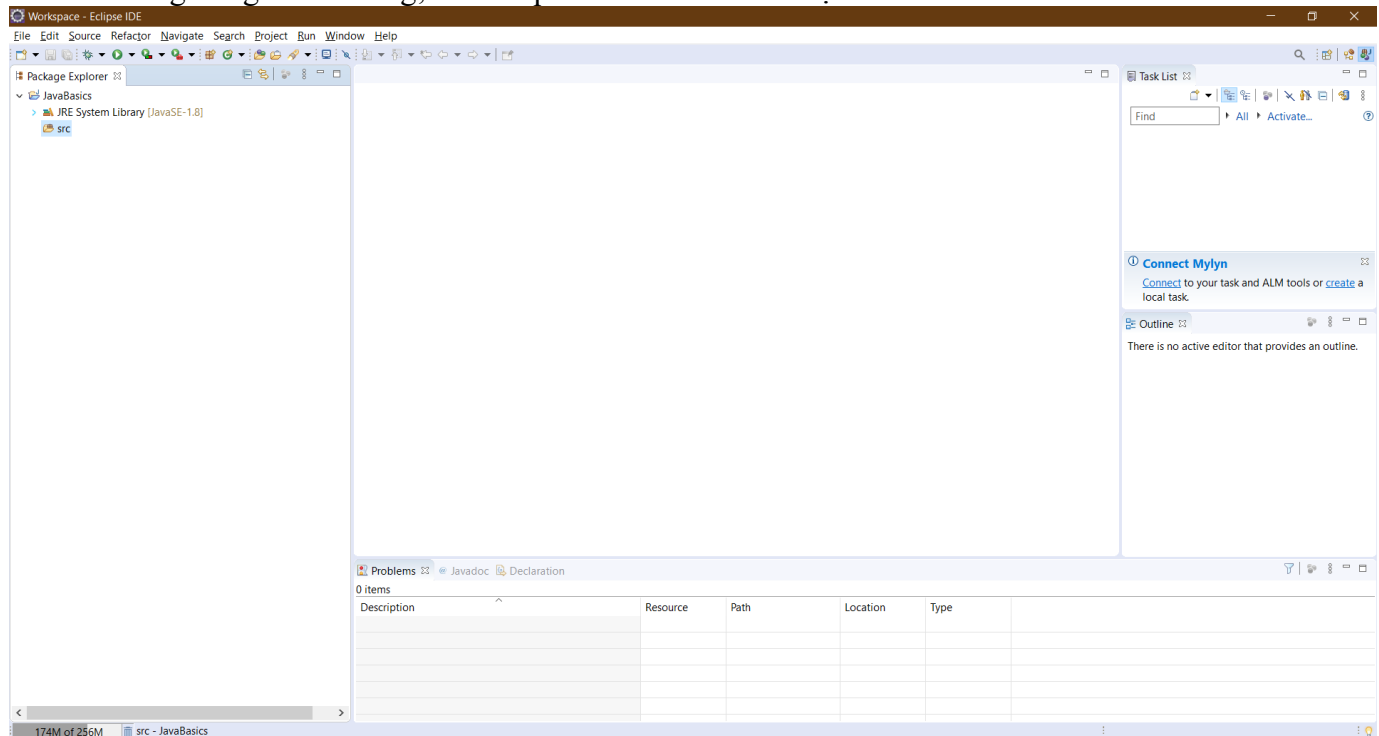
Hình 15. Cửa sổ dự án Java mới

6. Trên cửa sổ bật lên, chọn *Mở phối cảnh*.



Hình 16. Mở cửa sổ phối cảnh liên kết

7. Đóng trang Chào mừng; sau đó quan điểm Java xuất hiện.



Hình 17. Quan điểm Java

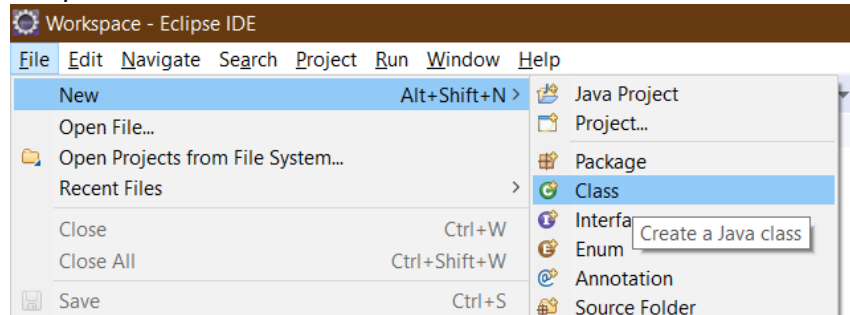
6 Bài tập

6.1 Viết, biên dịch và chạy chương trình ChoosingOption :

Lưu ý: Chúng tôi sử dụng dự án JavaBasics cho bài tập này.

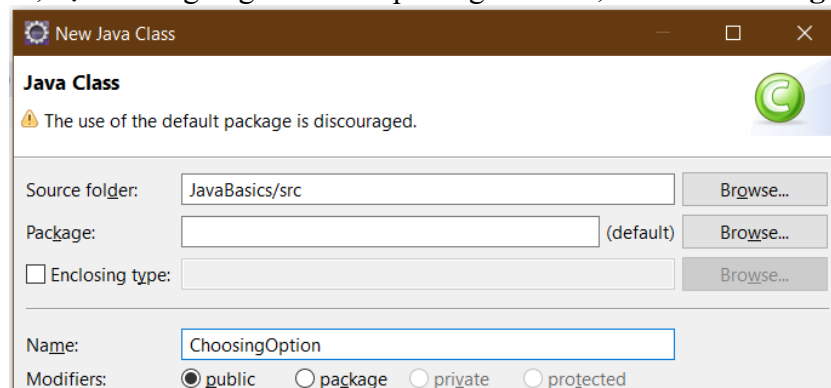
Bước 1: Tạo lớp.

- Chọn *Tệp* → *lớp* → *mới*



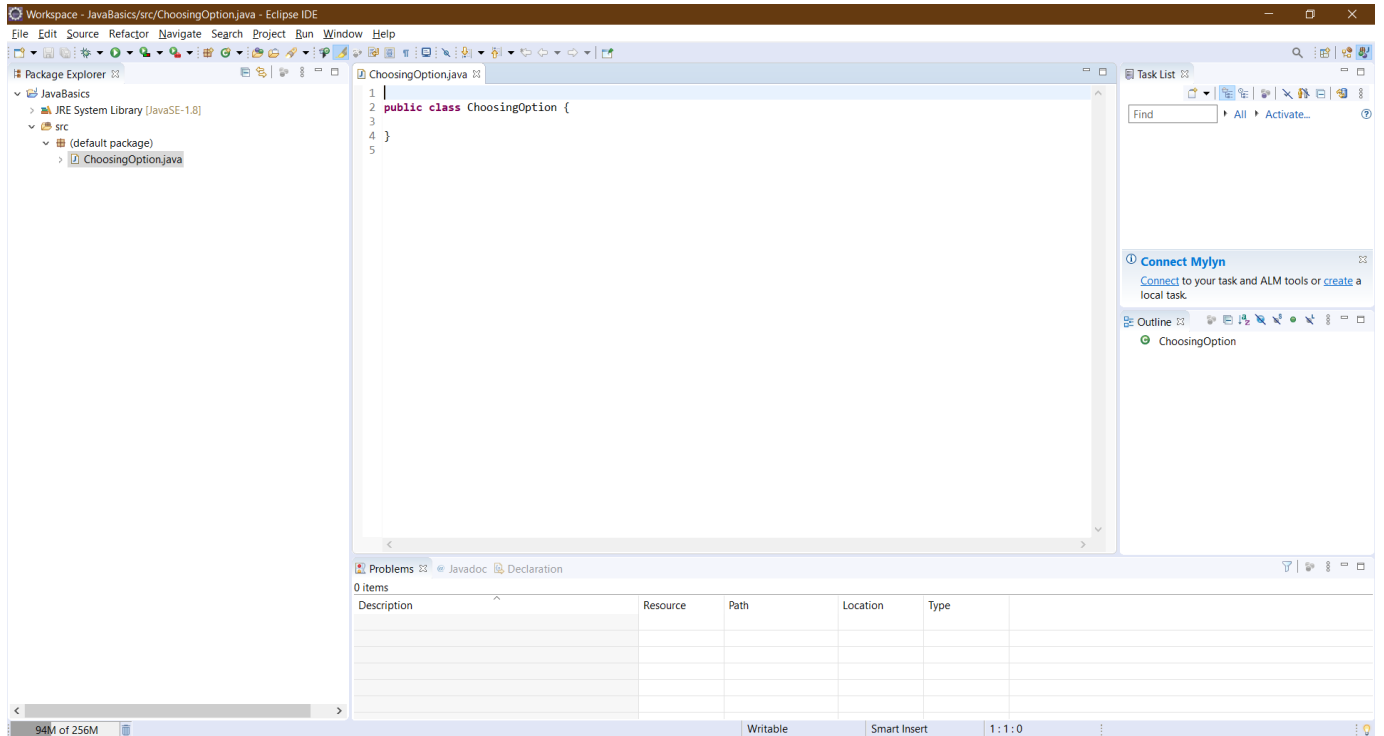
Hình 25. Tạo lớp

- Trên cửa sổ bật lên, đặt *Name* giống như tên lớp trong Hình 26, đó là "**ChoosingOption**"



Hình 26. Cửa sổ lớp Java mới

Chúng tôi có một lớp học mới cụ thể là *Chọn Tùy chọn* được tạo ra như trong Hình 27.



Hình 27. Một lớp học mới được tạo ra

Bước 2: Viết chương trình. Mã nguồn được minh họa trong Hình 28.

```

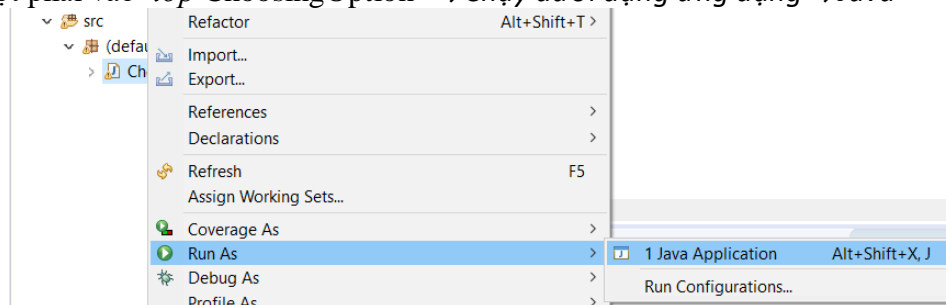
1 import javax.swing.JOptionPane;
2 public class ChoosingOption{
3     public static void main(String[] args){
4         int option = JOptionPane.showConfirmDialog(null,
5             "Do you want to change to the first class ticket?");
6
7         JOptionPane.showMessageDialog(null,"You've chosen: "
8             + (option==JOptionPane.YES_OPTION?"Yes":"No"));
9         System.exit(0);
10    }
11 }

```

Hình 28. Chọn tùy chọn ứng dụng

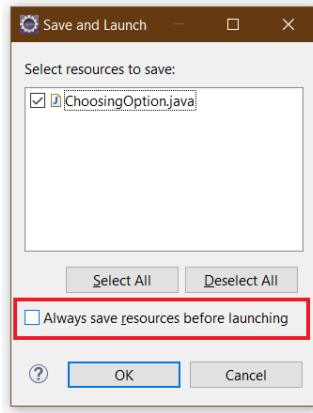
Bước 3: Lưu và khởi chạy.

- Nhấp chuột phải vào *lớp* ChoosingOption → Chạy dưới dạng ứng dụng → Java



Hình 29. Chạy ứng dụng (1)

- Chọn Luôn lưu tài nguyên trước khi khởi chạy, sau đó nhấp vào nút OK



Hình 30. Lưu và khởi chạy

Bước 4: Thêm, cam kết và đẩy tất cả công việc của bạn lên nhánh "'master'" .

Câu hỏi:

- Điều gì xảy ra nếu người dùng chọn "Hủy"?
- Cách tùy chỉnh các tùy chọn cho người dùng, ví dụ chỉ có hai tùy chọn: "Có" và "Không", HOẶC "Tôi làm" và "Tôi không" (Gợi ý: Sử dụng Javadocs hoặc sử dụng trợ giúp của Eclipse / Netbean IDE).

6.2 Viết một chương trình cho đầu vào / đầu ra từ bàn phím

Lưu ý: Chúng tôi sử dụng dự án JavaBasics cho bài tập này.

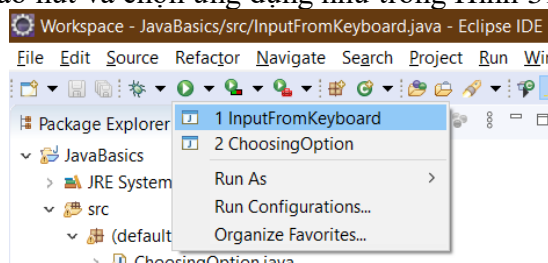
Bước 1: Tạo lớp.

- Chọn *Tệp* → *lớp* → *mới*
- Trên cửa sổ bật lên, đặt *Tên* là "**InputFromKeyboard**"

Bước 2: Viết chương trình. Mã nguồn được minh họa trong Hình 32.

Bước 3: Lưu và khởi chạy.

- Phương pháp 1: Nhấp chuột phải vào *lớp* InputFromKeyboard → *Run As* → *Java Application*.
- Phương pháp 2: Nhấn vào nút và chọn ứng dụng như trong Hình 31



Hình 31. Chạy ứng dụng (2)


```
1 import java.util.Scanner;
2 public class InputFromKeyboard{
3     public static void main(String args[]){
4         Scanner keyboard = new Scanner(System.in);
5
6         System.out.println("What's your name?");
7         String strName = keyboard.nextLine();
8         System.out.println("How old are you?");
9         int iAge = keyboard.nextInt();
10        System.out.println("How tall are you (m)?");
11        double dHeight = keyboard.nextDouble();
12
13        //similar to other data types
14        //nextByte(), nextShort(), nextLong()
15        //nextFloat(), nextBoolean()
16
17        System.out.println("Mrs/Ms. " + strName + ", " + iAge + " years old. "
18                           + "Your height is " + dHeight + ".");
19    }
20 }
21 }
```

Markers Properties Servers Data Source Explorer Snippets Problems Console Search

```
<terminated> InputFromKeyboard [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_171.jdk/Contents/Home/bin/
What's your name?
Trang
How old are you?
35
How tall are you (m)?
1.65
Mrs/Ms. Trang, 35 years old. Your height is 1.65.
```

Hình 32. Ứng dụng InputFromKeyboard

6.3 Viết một chương trình hiển thị một hình tam giác có chiều cao n sao (*), n do người dùng nhập.

Ví dụ: n = 5:

```
 *
**
***
****
*****
*****
*****
```

Lưu ý: Bạn phải tạo một dự án Java mới cho bài tập này.

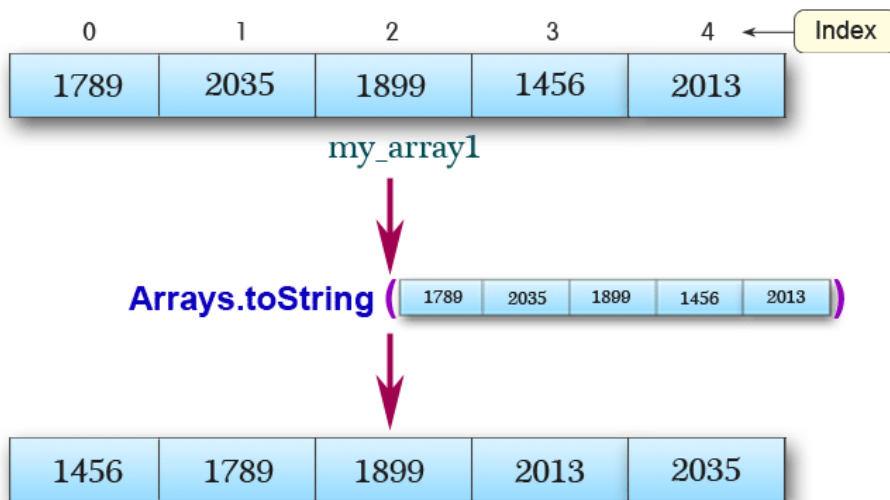
6.4 Viết một chương trình để hiển thị số ngày trong tháng, do người dùng nhập (cả tháng và năm). Nếu đó là tháng/năm không hợp lệ, hãy yêu cầu người dùng nhập lại.

Lưu ý: Bạn phải tạo một dự án Java mới cho bài tập này.

- Người dùng có thể nhập tháng bằng tên đầy đủ, chữ viết tắt, bằng 3 chữ cái hoặc bằng số. Để minh họa, đầu vào hợp lệ của *tháng Giêng* là Tháng Giêng, Tháng Giêng, Tháng Giêng và 1.
- Người dùng phải nhập một năm vào một số không âm và nhập tất cả các chữ số. Ví dụ, đầu vào hợp lệ của năm 1999 chỉ là 1999, chứ không phải 99, "một nghìn chín trăm chín mươi chín", hoặc bất cứ điều gì khác.
- Một năm là một năm thường là 365 ngày hoặc một năm nhuận là 366 ngày. Mỗi năm chia hết cho 4 là một năm nhuận, ngoại trừ những năm chia hết cho 100, nhưng không chia hết cho 400. Ví dụ, năm 1800 không phải là năm nhuận, nhưng năm 2000 là năm nhuận. Trong một năm, có mười hai tháng, được liệt kê theo thứ tự như sau.

Tháng	Tháng 1	Tháng hai	Tháng 3	Tháng 4	Tháng 5	Tháng 6	Tháng 7	Tháng 8	Tháng 9	Tháng 10	Tháng 11	Tháng 12
Sự rút ngắn	Jan.	Tháng Hai.	Mar.	Apr.	Tháng 5	Tháng 6	Tháng 7	Tháng tám.	Tháng chín.	Tháng Mười.	Ngày.	Tháng mười hai.
Trong 3 chữ cái	Jan	Tháng Hai	Mar	Apr	Tháng 5	Jun	Tháng bảy	Tháng tám	Tháng chín	Tháng Mười	Ngày	Tháng mười hai
Trong số	1	2	3	4	5	6	7	8	9	10	11	12
Ngày trong tháng trong năm thường	31	28	31	30	31	30	31	31	30	31	30	31
Ngày trong tháng trong năm nhuận	31	29	31	30	31	30	31	31	30	31	30	31

6.5 Viết một chương trình Java để sắp xếp một mảng số và tính toán tổng và giá trị trung bình của các phần tử mảng.



Lưu ý: Bạn phải tạo một dự án Java mới cho bài tập này.

- Mảng có thể được nhập bởi người dùng hoặc một hằng số.

6.6 Viết một chương trình Java để thêm hai ma trận có cùng kích thước.

Lưu ý: Bạn phải tạo một dự án Java mới cho bài tập này.

- Ma trận có thể được nhập bởi người dùng hoặc hằng số.

Sau khi hoàn thành tất cả các nhiệm vụ của mình, Bạn nên đẩy tất cả công việc bạn đã thực hiện trong phòng thí nghiệm này vào một nhánh "**master**".

7 Tham khảo

Hock-Chuan, C. (2020, tháng Giêng). *Cách cài đặt JDK 13 (trên Windows, macOS & Ubuntu) và bắt đầu lập trình Java*. Lấy từ Đại học Công nghệ Nanyang: https://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html