



Topic 3:

**Đóng gói,
phát hành phần mềm**

Giảng viên: Phạm Thị Thương – Bộ môn CNPM – Khoa CNTT

Email: ptthuong@ictu.edu.vn



Mục đích

- Tìm hiểu, cài đặt, áp dụng một số công cụ, kỹ thuật hỗ trợ các hình thức đóng gói phần mềm cơ bản.



Nội dung chính

- 1. Đóng gói, phát hành p.mềm**
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub



1. Đóng gói, phát hành p.mềm

➤ Works:

- Tạo *binary distribution* (executable installer), or package dạng nén (với tên file phần mở rộng: *.jar, *.war, *.ear, *.zip, *.apk, ...) và có thể downloaded từ internet hoặc có thể vận hành một cách trực tiếp .



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. **Đóng gói định dạng *.Jar (Java Archive)**
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub



2. Đóng gói dạng file *.Jar

- a) **What/Why is JAR – Java Archive**
- b) Content of an Archive?
- c) How to make a Jar?
- d) How to run a Jar?



a. What/why is JAR?

- JAR (Java ARchive) – **Run-able Java File**

- Why?

- Thông thường, để phân bố một phiên bản p.mềm, ta chỉ muốn phát hành 1 file đơn đến người dùng chứ không muốn phát hành cả cấu trúc thư mục dự án được lấp đầy với các file class đến người dùng.

- ⇒ JAR files được thiết kế cho mục đích này.

a. What/why is JAR?

➤ What JAR?

- Dùng như các phiên bản phần mềm
 - Có thể chạy độc lập, trực tiếp, ko cần biên dịch lại.
- Dùng như các shared libraries
 - Tái sử dụng trong các dự án ↑ p.mềm
 - Có thể được truy cập, downloaded, invoked, or imported trực tiếp vào ứng dụng.
 - Tăng tính an toàn, an ninh, khả năng mở rộng và tính linh động của ứng dụng
 - Ví dụ: cập nhật phiên bản cũ bằng cách thay thế file jar cũ bằng file jar mới.



2. Đóng gói dạng file *.Jar

- a) What/Why is JAR – Java Archive
- b) Content of an Archive?**
- c) How to make a Jar?
- d) How to run a Jar?



b. Content of an Archive?

- A JAR file contains:
 - Class files;
 - Other file types;
 - Sound, image, library, ... files which may be included in the project.
 - The manifest file.
 - Mỗi gói jar có 1 file manifest (~ file cấu hình)
 - Mô tả các đặc trưng/thuộc tính của Archive.
 - Ví dụ: See [link](#)



2. Đóng gói dạng file *.Jar

- a) What/Why is JAR – Java Archive
- b) Content of an Archive?
- c) How to make a Jar?**
- d) How to run a Jar?



c. How to make a JAR file?

- i. Tạo gói jar sử dụng trực tiếp lệnh jar
- ii. Tạo gói Jar trong IDE (ví dụ: Netbeans IDE, or Eclipse IDE)

i. Tạo jar dùng command line

➤ Các bước:

- ❖ Bước 1: Tạo file Java (ví dụ tạo trong notepad)
 - Ví dụ: file VD.java trong notepad
- ❖ Bước 2: Biên dịch file java vừa tạo
 - Cd đến thư mục chứa VD.java
 - Type: `javac VD.java`
- ❖ Bước 3: Tạo file manifest.txt trong thư mục hiện thời
- ❖ Bước 4: Dùng lệnh jar in JDK để đóng gói
 - Type: `jar -cvfm VD.jar manifest.txt *.class`

i. Tạo jar dùng command line

➤ Lưu ý:

Jar -cvfm <tên gói.jar> <file manifest> <*.class> ↵

➤ Trong đó:

- c: tạo một archive mới
- v: hiện output kết quả đóng gói trên console
- f: đặc tả tên archive với tên như chỉ định
- m: sử dụng file manifest chỉ định

Lưu ý: Gõ: jar ? để xem các tùy chọn của lệnh jar.

=> Kết quả:

```
E:\Namhoc_2019\_KyI_2019_2020\1_BaotriPM\Code_Experiments\Test_DemoDonggoi\VD_Java_Notepad>jar -cvfm VD.jar manifest.txt *.class
added manifest
adding: VD.class(in = 403) (out= 279)(deflated 30%)
```



c. How to make a JAR file?

- i. Make jar file, use the jar command in command line (DOS)
- ii. **Make Jar files in IDE (ví dụ: Netbeans)**

ii. Make Jar in Netbeans IDE

➤ Các bước

➤ Tương tự trên:

1. Create a java project,
2. Compile dự án
3. Click: Run → Clean and Build Project.

➤ Ví dụ:

- Thực hiện trên **dự án TestGUI** với ứng dụng GUI Calculator (đóng gói dạng .JAR)



ii. Make Jar in Netbeans IDE

➤ Lưu ý:

- File manifest.MF tự động sinh ra trong thư mục META-INF của Archive
- File Jar xuất hiện trong thư mục dist của project (NetBeans), or trong thư mục đã chỉ định (Eclipse)



2. Đóng gói dạng file *.Jar

- a) What/Why is JAR – Java Archive
- b) Content of an Archive?
- c) How to make a Jar?
- d) How to run a Jar?**

d. How to run a JAR file?

- Sử dụng lệnh:

Java -jar <tên gói.jar> ↵

- ▶ Ví dụ:

java -jar VD.jar ↵

2. Đóng gói dạng file *.Jar

➤ Java Build Tools

➤ 3 công cụ build java nổi tiếng:

➤ **Apache Ant; Maven; & Gradle.**

= > help: compile, build, packagem test & deploy
package



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. **Đóng gói dạng exe installer**
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub



3. Đóng gói dạng exe installer

➤ Why:

➤ Gói định dạng .jar có thể bị dịch ngược

➤ Có thể không an toàn:

➤ Ví dụ: Import jar vào project, giải nén ta thấy hết cấu trúc dự án.

➤ => Dùng đóng gói dạng installer.



3. Đóng gói dạng exe installer

➤ 3 main steps:

1. Create jar file
2. Install “exe4j” or launch4j to convert jar file to exe
3. Install “inno setup Compiler” to create installer



1. Create jar file

- Tạo file *.jar cho dự án
 - Như mục 2 (ở trên)
 - Ví dụ: **Calculator.jar**



2. Install “exe4j” or launch4j to convert jar file to exe

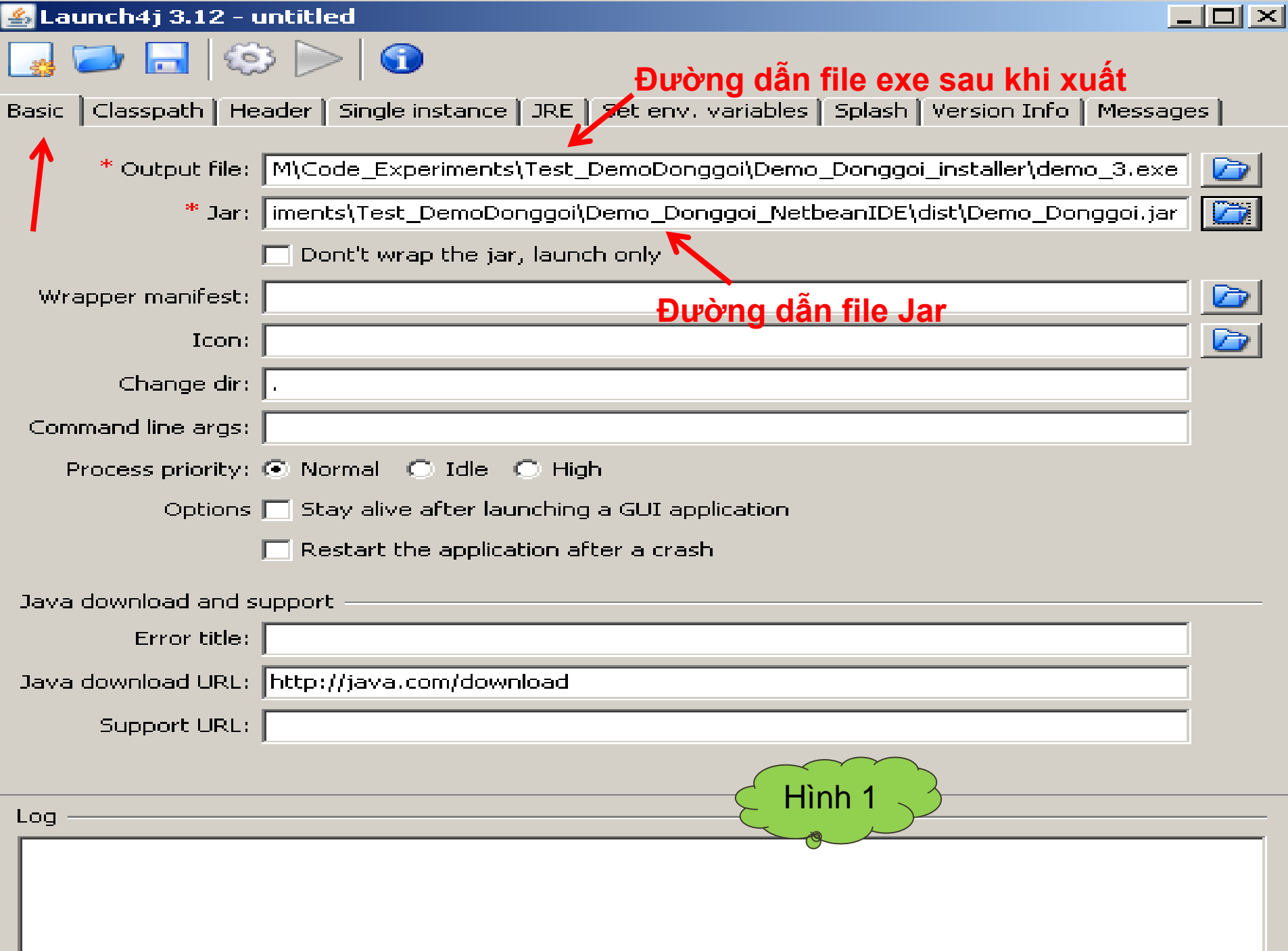
- Sử dụng Launch4j:

- Link download:

- <https://sourceforge.net/projects/launch4j/files/launch4j-3/3.8/launch4j-3.8-win32.exe/download>

2. Install launch4j

- Cài đặt và thiết lập launch4j
 - Chạy file để cài đặt
 - Thiết lập thông tin gói:
 - Màn hình chính
 - Tại **tab basic**: điền thông tin như hình 1)
 - **Tab JRE**: điền như hình 2)
- => kết quả: hình 3 (below)



Launch4j 3.12 - untitled

Basic Classpath Header Single instance JRE Set env. variables Splash Version Info Messages

Bundled JRE path: ☐ 64-bit ☐ Fallback option

Search options

Min JRE version:

Max JRE version:

Options

Initial heap size: MB % of available memory

Max heap size: MB % of available memory

JVM options:

Variables / registry:

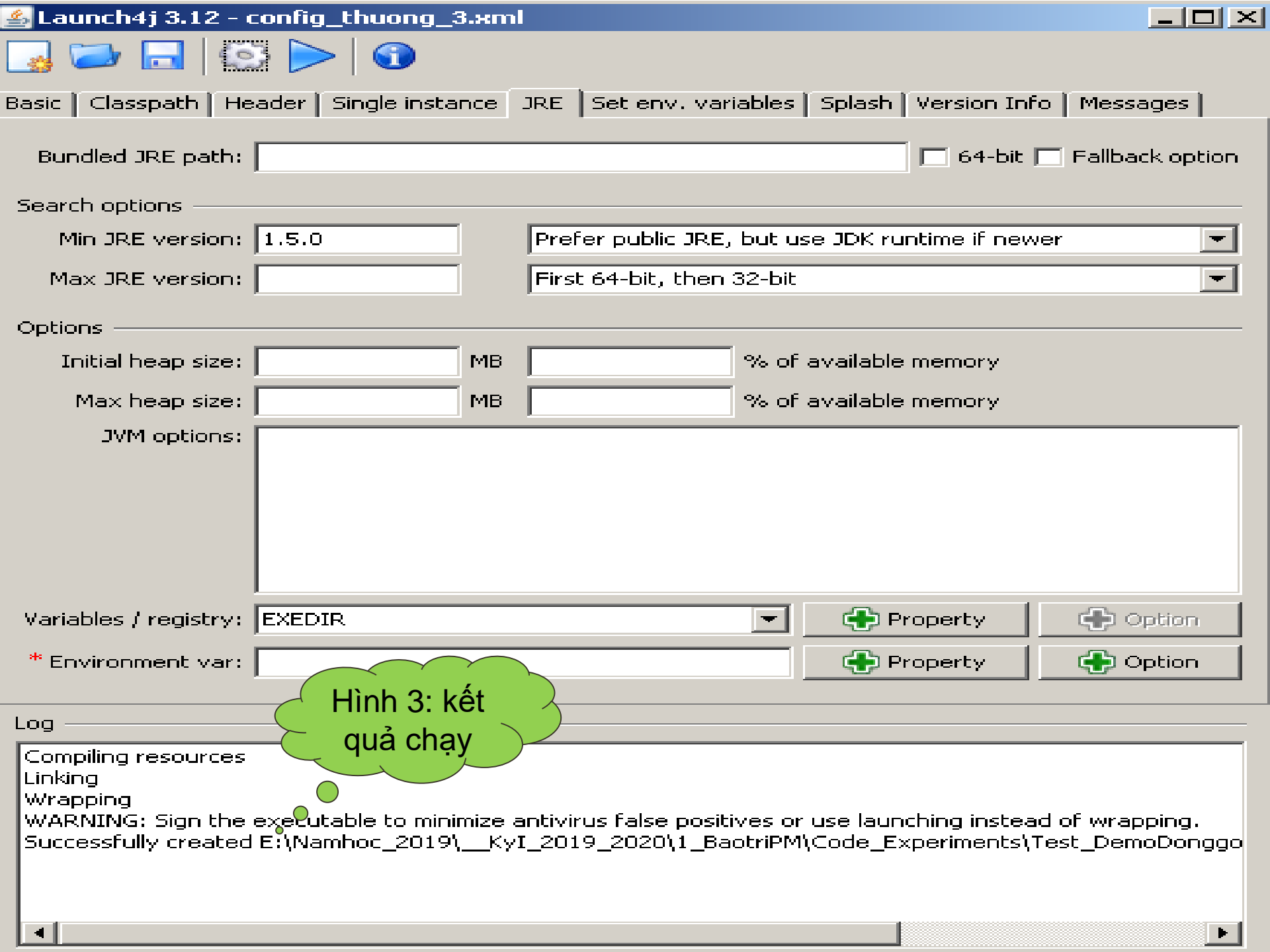
* Environment var:

Đường dẫn jre (để trống)

Run

Phiên bản nhỏ nhất để run chương trình
có dạng x.x.x

Hình 2





2. Install launch4j

- Ví dụ:
 - Demo

3. Install “inno setup Compiler”

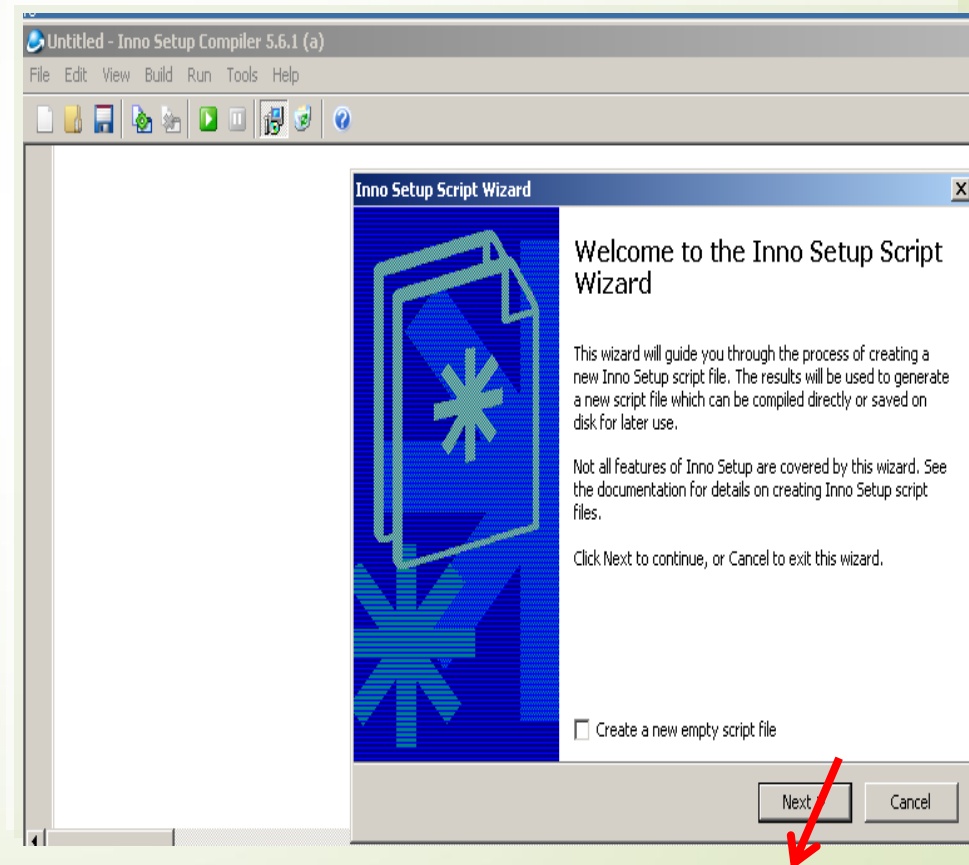
- Sử dụng tool “inno setup Compiler”: để tạo installer
- Các bước đóng gói:
 - Chạy file cài đặt
 - **Các bước tạo bộ setup?**

Các bước tạo bộ setup?

➤ Bước 1:

➤ File → New:

=> Click Next



Các bước tạo bộ setup?

- Bước 2:
 - Nhập các thông tin phiên bản như hình
 - Click Next

Inno Setup Script Wizard

Application Information
Please specify some basic information about your application.

Application name:
My Program

Application version:
1.0

Application publisher:
ICTU - TNU

Application website:
http://www.example.com/

bold = required

< Back **Next >** Cancel

Các bước tạo bộ setup?

- Bước 2:
 - Check: Allow user to change the application folder
 - Click Next:

Inno Setup Script Wizard

Application Folder
Please specify folder information about your application.

Application destination base folder:
Program Files folder

Application folder name:
My Program

☒ Allow user to change the application folder

Other:

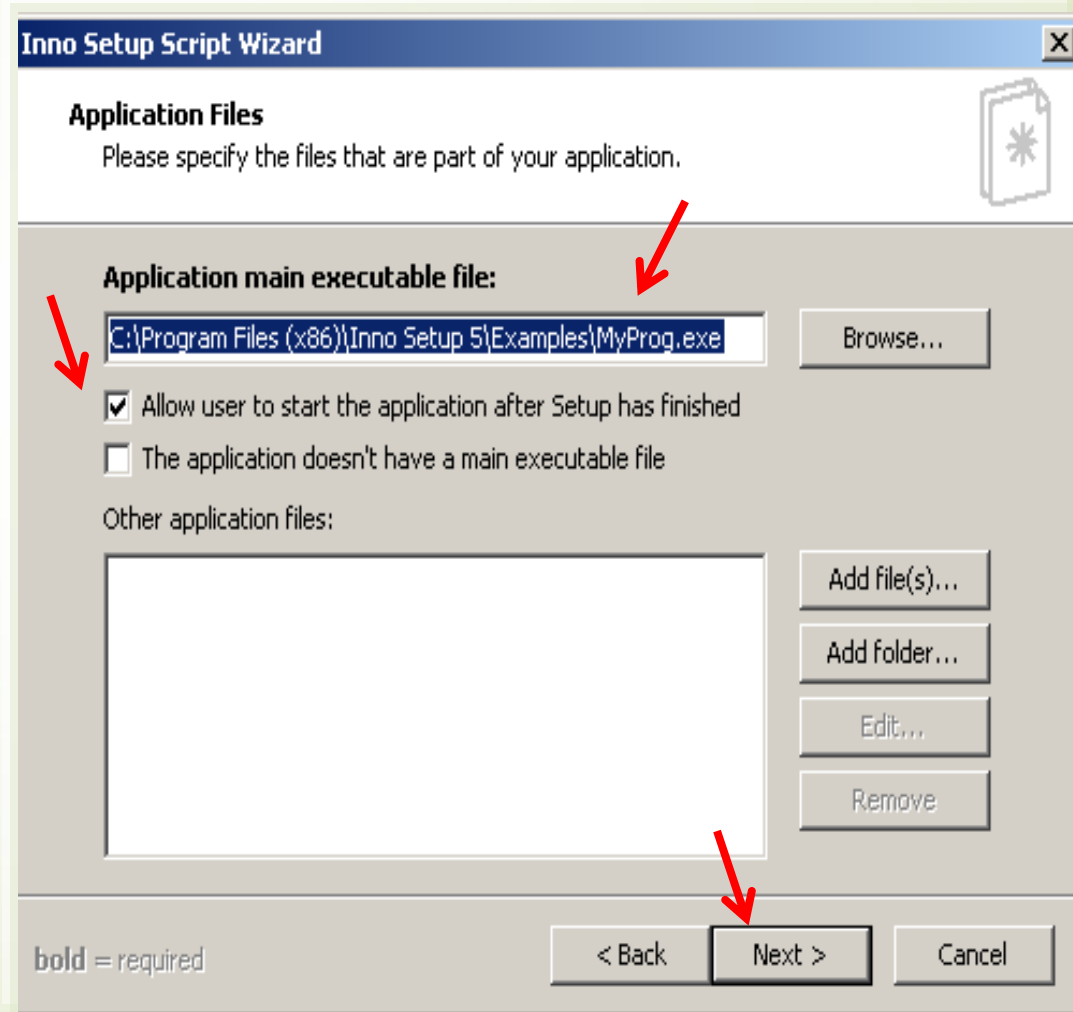
☐ The application doesn't need a folder

bold = required

< Back **Next >** Cancel

Các bước tạo bộ setup?

- Bước 3:
 - Nhập path lưu file thực thi
 - Check vào các tùy chọn như hình
 - Click Next:



Các bước tạo bộ setup?

- Bước 4:
 - Check vào các tùy chọn (như hình)
 - Click: Next

Inno Setup Script Wizard

Application Shortcuts
Please specify which shortcuts should be created for your application.

☒ Create a shortcut to the main executable in the common Start Menu Programs folder
Application Start Menu folder name:
Demo_package

☒ Allow user to change the Start Menu folder name
☐ Allow user to disable Start Menu folder creation
☐ Create an Internet shortcut in the Start Menu folder
☐ Create an Uninstall shortcut in the Start Menu folder

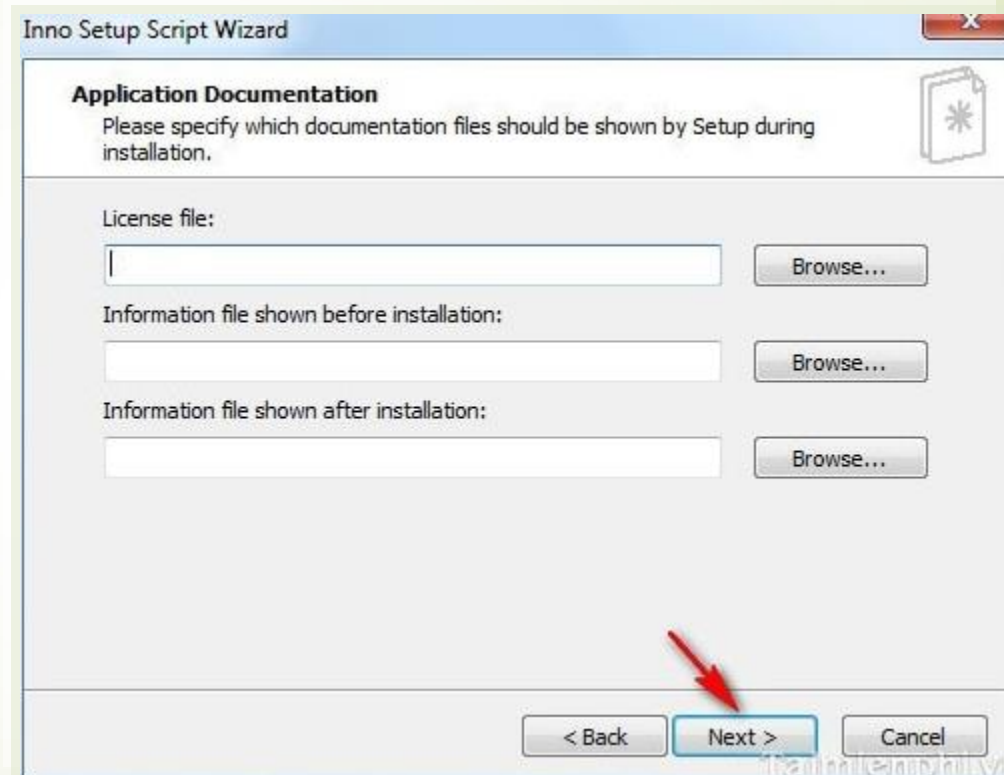
Other shortcuts to the main executable:
☒ Allow user to create a desktop shortcut
☐ Allow user to create a Quick Launch shortcut on older versions of Windows

bold = required

< Back **Next >** Cancel

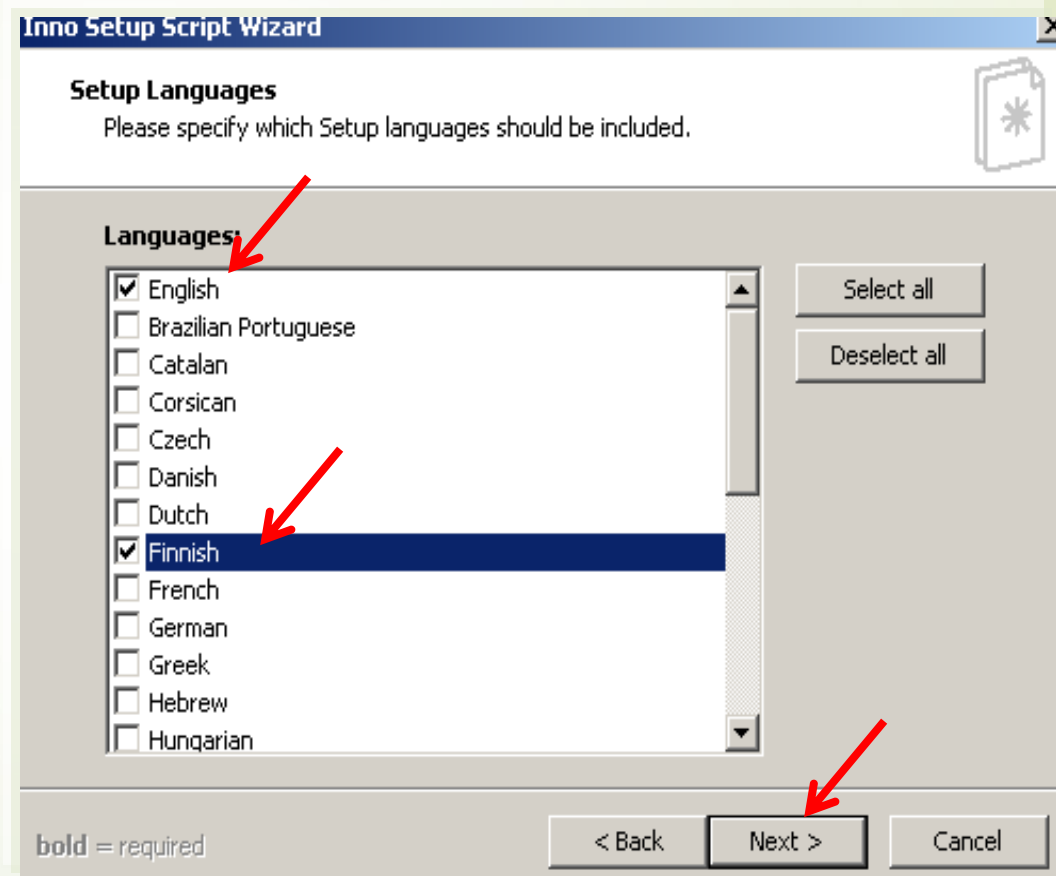
Các bước tạo bộ setup?

- Bước 5:
 - Nhập các thông tin cấp phép (nếu có)
 - Click Next:



Các bước tạo bộ setup?

- Bước 6:
 - Chọn gói ngôn ngữ
 - như hình
 - Click Next:



Các bước tạo bộ setup?

- Bước 7:
 - Nhập các thông tin bộ installer
 - Như hình
 - Click Next → Finish.

Inno Setup Script Wizard

Compiler Settings
Please specify some basic compiler settings.

Custom compiler output folder:
E:\Namhoc_2019_KyI_2019_2020\1_BaotriPM\Code_Experim Browse...

Compiler output base file name:
setup1

Custom Setup icon file:
E:\Namhoc_2019_KyI_2019_2020\1_BaotriPM\Code_Experim Browse...

Setup password:

< Back Next > Cancel

Các bước tạo bộ setup?

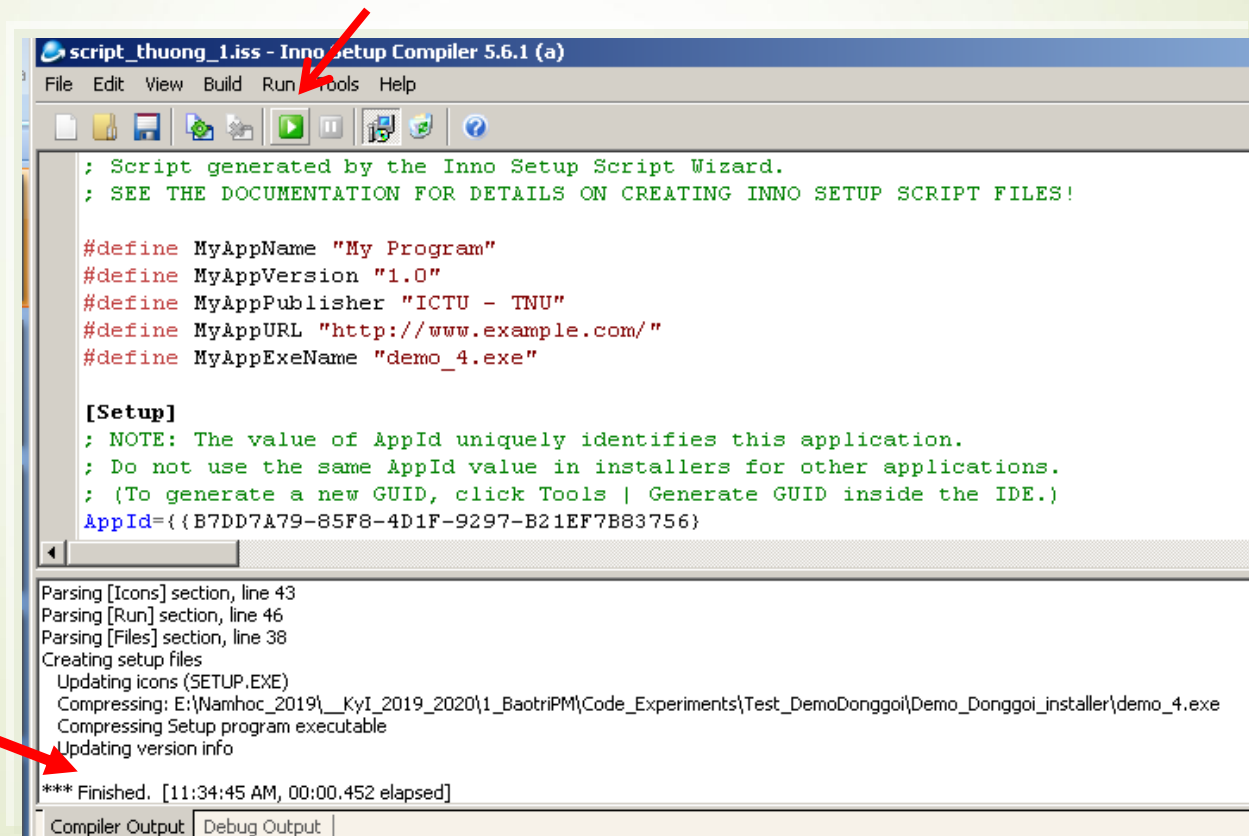
- Kết quả:
 - Ứng dụng được tạo thành công
 - Biểu tượng (hình)
 - Cài đặt và chạy chương trình như ứng dụng thông thường.



setupFree.exe

3. Install “inno setup Compiler”

- ➡ Click Run để cài đặt và chạy thử:



```
script_thuong_1.iss - Inno Setup Compiler 5.6.1 (a)
File Edit View Build Run Tools Help

; Script generated by the Inno Setup Script Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

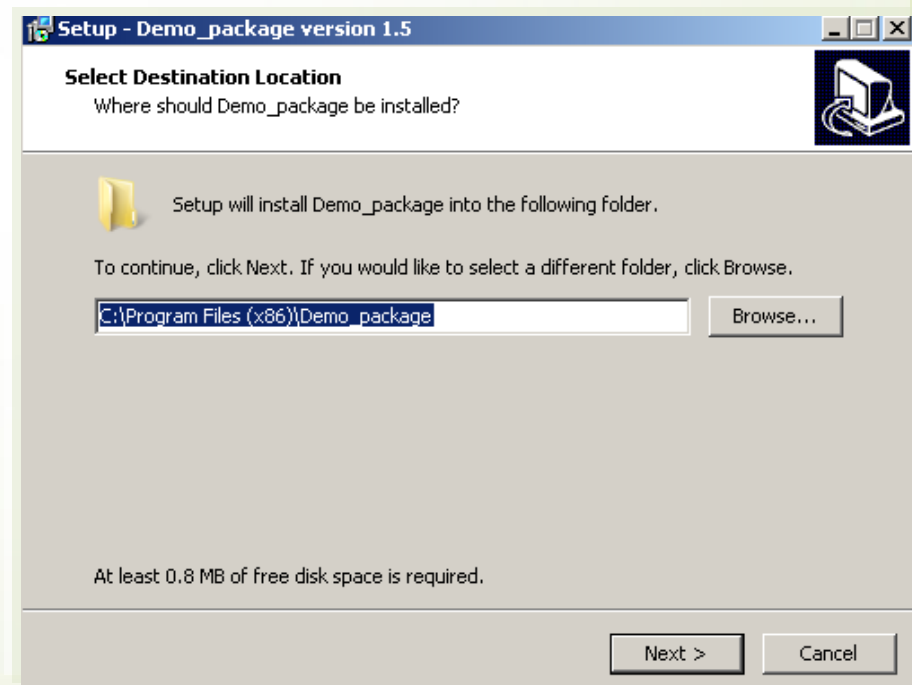
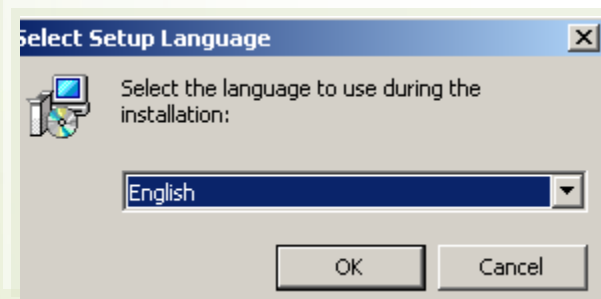
#define MyAppName "My Program"
#define MyAppVersion "1.0"
#define MyAppPublisher "ICTU - TNU"
#define MyAppURL "http://www.example.com/"
#define MyAppExeName "demo_4.exe"

[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={{ B7DD7A79-85F8-4D1F-9297-B21EF7B83756}

Parsing [Icons] section, line 43
Parsing [Run] section, line 46
Parsing [Files] section, line 38
Creating setup files
Updating icons (SETUP.EXE)
Compressing: E:\Namhoc_2019\__KyI_2019_2020\1_BaotriPM\Code_Experiments\Test_DemoDonggoi\Demo_Donggoi_installer\demo_4.exe
Compressing Setup program executable
Updating version info
*** Finished. [11:34:45 AM, 00:00.452 elapsed]
```

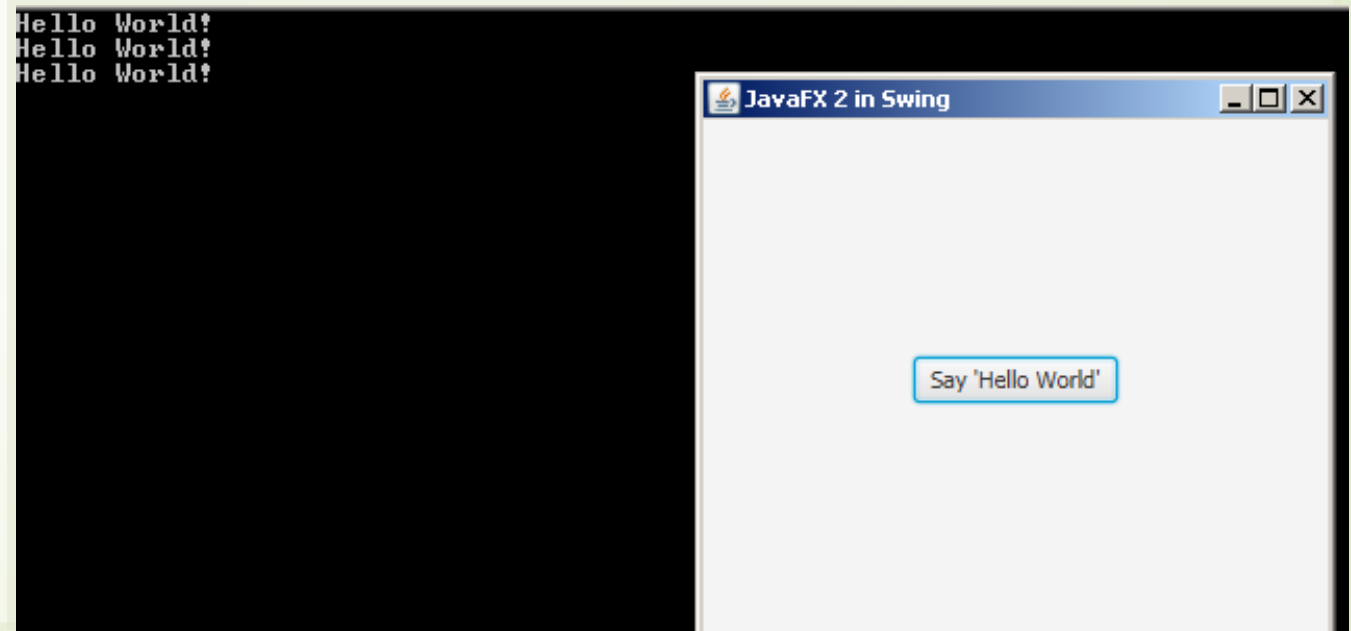
3. Install “inno setup Compiler”

➡ Hình ảnh khi cài đặt.



3. Install “inno setup Compiler”

- Kết quả cài đặt và chạy:
 - Hình





Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. **Đóng gói p.mềm dạng portable**
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub

4. Đóng gói p.mềm dạng portable

➤ Phần mềm Portable là gì?

- Portable là phần mềm được đóng gói để sử dụng ngay mà không cần phải cài đặt
 - ~ Sau khi tải về, chỉ cần giải nén và chạy, quá trình chạy không khác so với bản cài đặt





4. Đóng gói p.mềm dạng portable

- Ưu điểm của phần mềm dạng Portable

- Không cần cài đặt lên máy tính.

- => nâng cao hiệu suất sử dụng của máy tính, tránh tạo rác khi máy tính phải cài quá nhiều phần mềm.

- Không tốn tài nguyên hệ thống.

- Tiện lợi khi sử dụng


- Có thể copy vào USB và sử dụng trên các máy tính.



4. Đóng gói p.mềm dạng portable

➤ **Nhược điểm**

- Khó Update tự động
 - Mỗi lần có phiên bản mới, cần tạo lại bản portable khác
- Khó quản lý nếu người dùng sử dụng quá nhiều phần mềm Portable.
- Một số phần mềm sẽ không tự khởi động cùng hệ thống



4. Đóng gói p.mềm dạng portable

- Cách đóng gói p.mềm dạng portable
 - Tải Multi Packer tại địa chỉ:
 - <https://down.shockingsoft.com/MultiPacker.zip>.



4. Đóng gói p.mềm dạng portable

- Cài đặt Multi Packer:

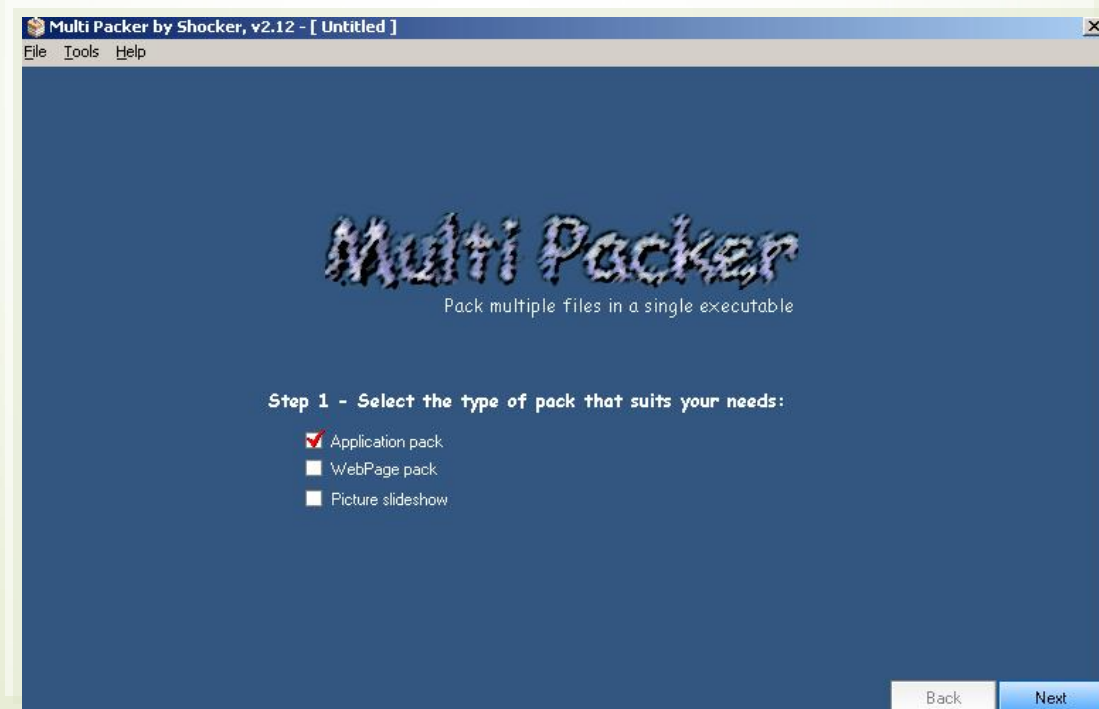
- Có 3 tùy chọn ~ 3 chức năng chính của c.trình:

- Application pack (tạo phần mềm portable),
 - WebPage pack (đóng gói trang web thành file *.exe)
 - Picture slideshow (tạo slideshow hình ảnh dưới dạng file *.exe).

4. Đóng gói p.mềm dạng portable

➡ Bước 1:

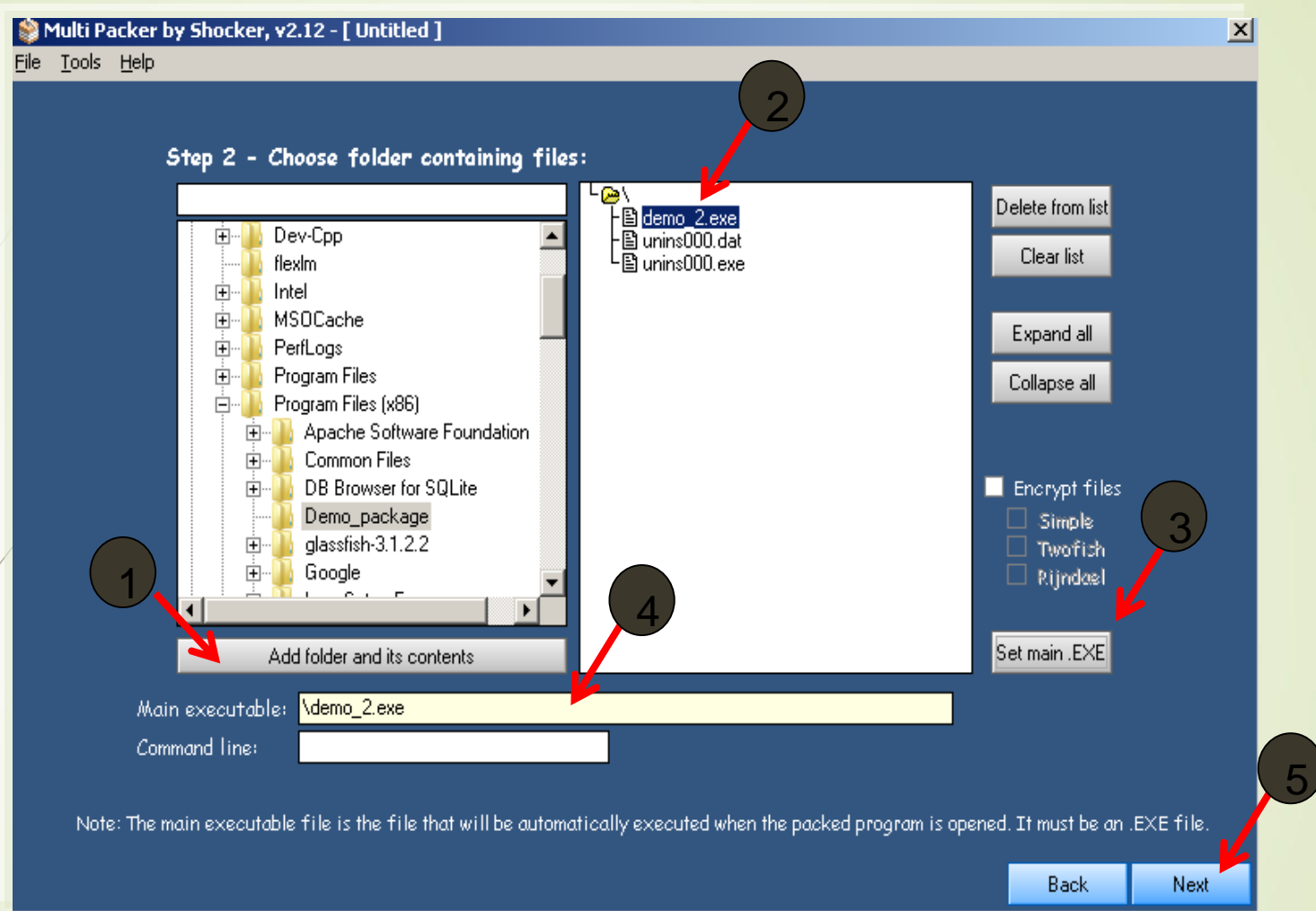
=> Chọn Application pack → Next để bắt đầu.




4. Đóng gói p.mềm dạng portable

➤ Bước 2:

- Chọn thư mục **chứa các file đã cài đặt** của phần mềm cần tạo portable
 - Ví dụ: C:\Program Files (x86)\Demo_package,
- Nhấn **nút Add folder and its contents:**
 - => Đưa toàn bộ nội dung thư mục này sang khung kế bên
- Chọn file.exe chính của p.mềm, click Set main .EXE
File này sẽ xuất hiện trong hộp Main executable.
Click Next để chuyển sang bước cuối cùng.



Lưu ý: Đối với những phần mềm được tích hợp nhiều tính năng và khó xác định được file thực thi chính, ta right click vào shortcut của nó trên desktop, chọn Properties và tìm đường dẫn tới file thực thi ở mục Target.



4. Đóng gói p.mềm dạng portable

➤ Bước 3:

➤ Nhập các thông tin thiết lập quyền truy cập ứng dụng

➤ Như hình (dưới)

gõ pass (ict...)

gõ pass lặp lại

Step 3 - Choose a password (and reenter it) to protect your pack [leave empty for none]:

Note: Choosing an encryption will require a password

☒ Use compression☒ Use exclusion list

Final step - Create pack

Pack creating started...

Checking files:

- OK : C:\Program Files (x86)\Demo_package\demo_2.exe
- OK : C:\Program Files (x86)\Demo_package\unins000.dat

Adding files:

- Adding: C:\Program Files (x86)\Demo_package\demo_2.exe
- Adding: C:\Program Files (x86)\Demo_package\unins000.dat
- Adding: C:\Program Files (x86)\Demo_package\unins000.exe

Done: "E:\Namhoc_2019_Kyl_2019_2020\1_BaotriPM\Code_Experiments\Test_DemoDonggoi\Demo_Donggoi"

Giữ nguyên 2
tùy chọn

E:\Namhoc_2019_Kyl_2019_2020\1_BaotriPM\Code_Experiments\Test_DemoDonggoi\Demo_Donggoi

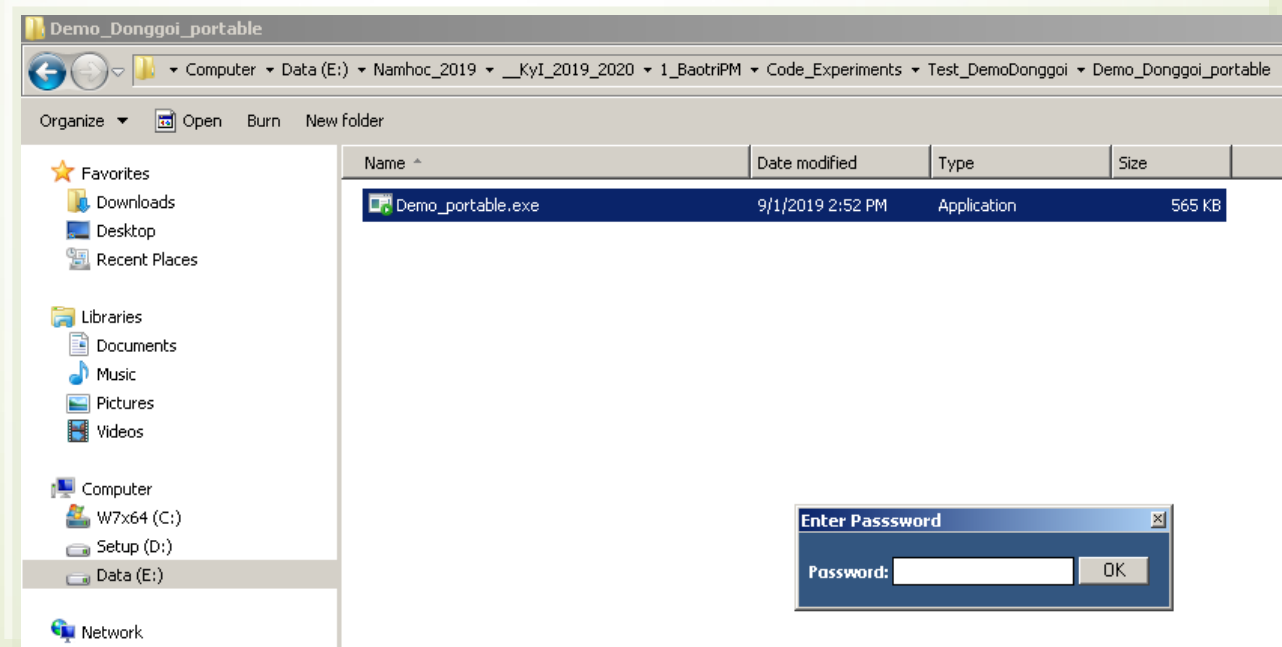
Browse

Back

Close

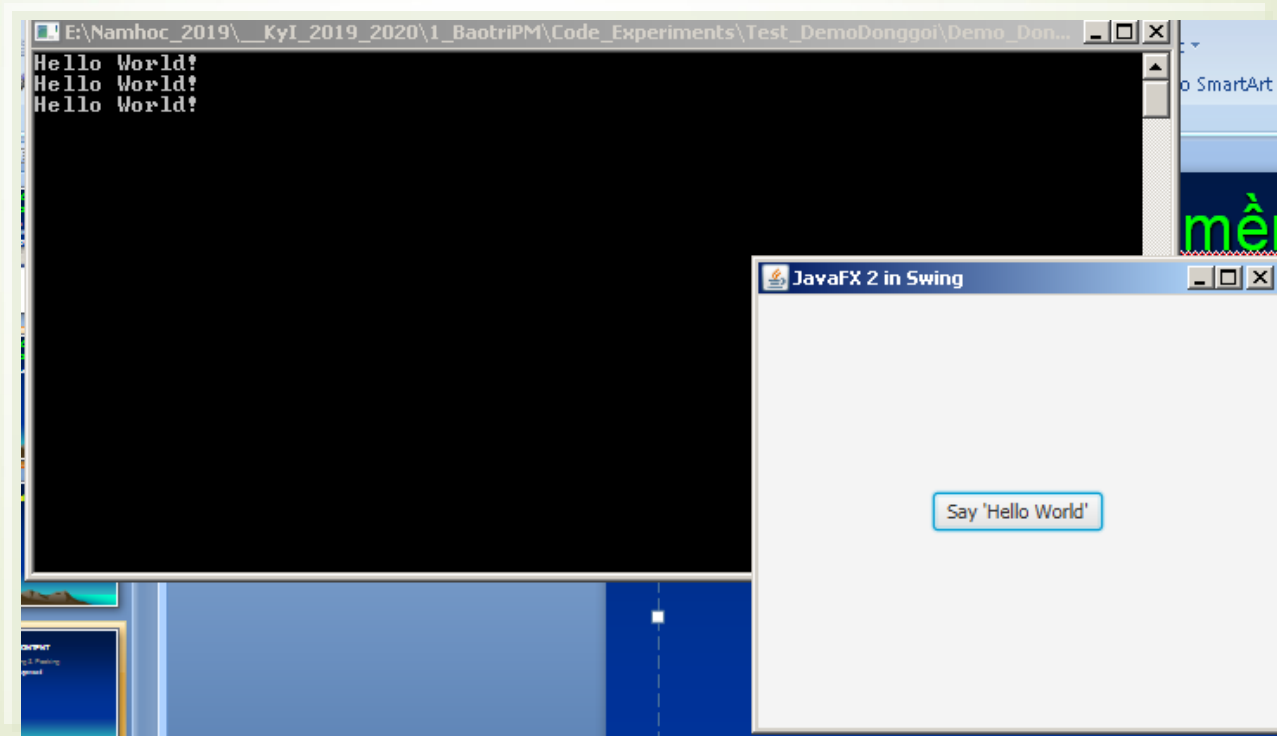
4. Đóng gói p.mềm dạng portable

- Portable:
 - Tiện dụng, nhỏ gọn.



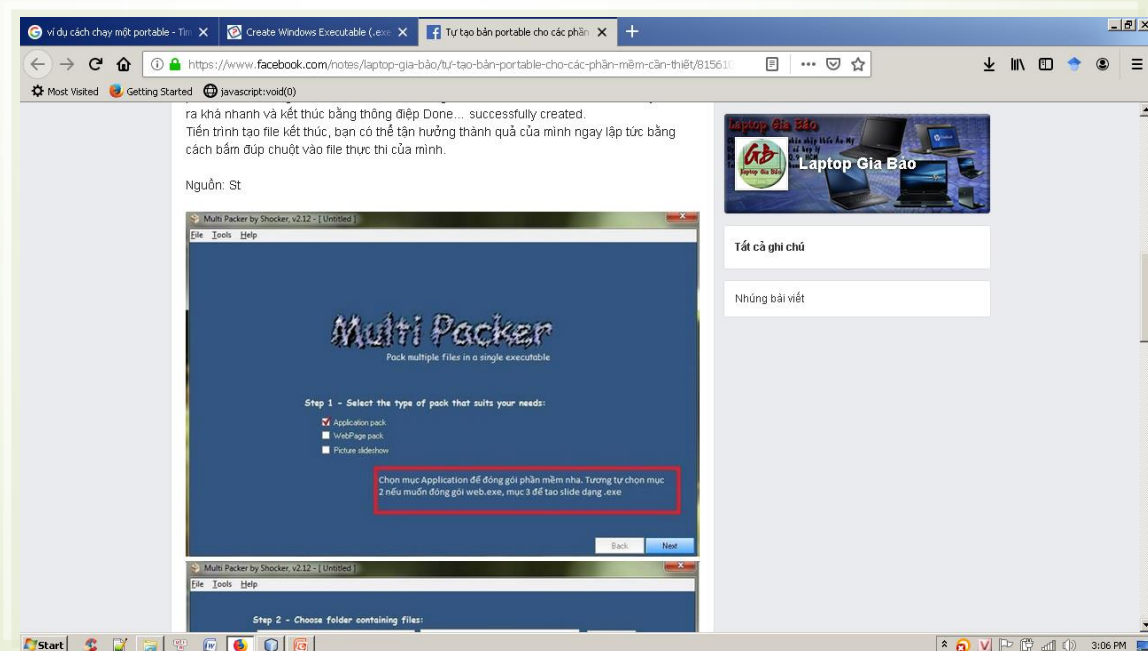
4. Đóng gói p.mềm dạng portable

➡ Nhập pass, kết quả chạy:



4. Đóng gói p.mềm dạng portable

- Tương tự:
 - Chọn mục 2: nếu muốn đóng gói web.exe
 - Chọn mục 3: tạo slide dạng .exe





Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
- 5. Tạo và đóng gói dạng plugin**
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub

5. Đóng gói định dạng plugin

- ▶ Plugin (or add-in, add-on, or extend ~ trình cắm, phần bổ trợ)
 - ▶ Mục đích:
 - ▶ Thêm tính năng, tiện ích, hoặc tùy biến các chức năng của ứng dụng chính (host apps)
 - ▶ Plugin phụ thuộc vào các dịch vụ của ứng dụng chính và không tự hoạt động nếu ứng dụng chính không chạy. Ngược lại các ứng dụng chính hoạt động độc lập với plugin
 - => người có thể thêm&update plugin mà không ảnh hưởng đến ứng dụng chính



5. Đóng gói định dạng plugin

- Các p.mềm hỗ trợ plugin vì nhiều lý do:
 - Cho phép các nhà phát triển thứ 3 tạo ra các tính năng để mở rộng phần mềm đó
 - Để hỗ trợ một cách dễ dàng cho việc bổ sung thêm các tính năng mới cho phần mềm
 - Để giảm kích thước của ứng dụng
 - Để tách mã nguồn từ ứng dụng vì giấy phép phần mềm không tương thích
 - ...



5. Đóng gói định dạng plugin

- Plugins – Example:

- FindBugs™ Eclipse plugin

- Aim: look for bugs in Java programs based the concept of bug patterns

- Is a very popular plugin for source code analysis

- ~ It uses static analysis to look for bugs in Java code based on bug patterns

- ~ Free software

5. Đóng gói định dạng plugin

- Plugins – FindBugs™ Eclipse plugin

- Bugs ~ coding mistakes

- Có hơn 400 bug patterns can be found in link:

- <https://spotbugs.readthedocs.io/en/latest/bugDescriptions.html>

- For example:

- Find bugs of JDK 1.7.0-b12

- Result: See link:

- <https://findbugs.cs.umd.edu/demo/jdk7/index.html>

5. Đóng gói định dạng plugin

► Bug Categories:

1. **Correctness:** lỗi code xuất hiện → mã nguồn không như ý developers
2. **Bad practice:** vi phạm các khuyến cáo viết mã/practice tồi
3. **Performance:** các vấn đề về lỗi thực thi, hiệu năng
4. **Multithreaded correctness:** xử lý bất đồng bộ & các vấn đề trong môi trường đa luồng.
5. **Internationalization:** các vấn đề liên quan đến áp dụng chuẩn quốc tế hóa (internationalization).
6. **Malicious code vulnerability:** Lỗi hỏng mã độc hại
7. **Security:** các lỗi bảo mật liên quan đến giao thức (protocols), SQL Injection, ...
8. **Dodgy (phong cách):** mã khó hiểu, bất thường, mập mờ hoặc được viết theo cách tự dẫn đến lỗi
9. ...

5. Đóng gói định dạng plugin

► Bug Categories – Examples:

1. **Correctness: some bugs:**

- Collections should not contain themselves
- equals() method used to compare array and not array
- Equals() method used to compare incompatible arrays
- Equals() method always return true/false
- A collection is added by itself
- ...

5. Đóng gói định dạng plugin

► Bug Categories – Examples:

2. Bad practice: some bugs:

- Equals () method does not check for null argument
- Method () might drop exception
- Store of non serializable object into HttpSession
- ...

3. Dodgy: some bugs:

- Class is final but declares protected field
- Redundant comparison of non null value and null
- Exception is caught when exception is not throws
- ...

5. Đóng gói định dạng plugin

➤ Bug Categories – Examples:

4. Malicious code vulnerability – Lỗi hỏng mã độc hại: some bugs:

- Classloaders should be only created inside doPrivileged block (khởi đặc quyền)
- Field isn't final and can't be protected from malicious code
- May explore internal representation by returning reference to mutable (có thể thay đổi) object
- ...

5. Đóng gói định dạng plugin

► Bug Categories – Examples:

5. Performance: some bugs:

- Method invokes ineffecticiant new String() constructor
- Method invokes toString() method on String
- ...

6. Multithreaded correctness - Dụng độ đa luồng: some bugs

- Sequence of calls to concurrent abstraction may not be atomic
- Using notify() rather than notifyAll()
- Notify(): thông báo
- ...

5. Đóng gói định dạng plugin

➤ Bug Categories – Examples:

7. Security: some bugs:

- Empty Database password
- Http cookie formed from untrusted input
- ...

5. Đóng gói định dạng plugin

➤ Ex.,: FindBugs™ Eclipse plugin

➤ Download at:

<http://findbugs.sourceforge.net/downloads.html/>

➤ Copy URL:

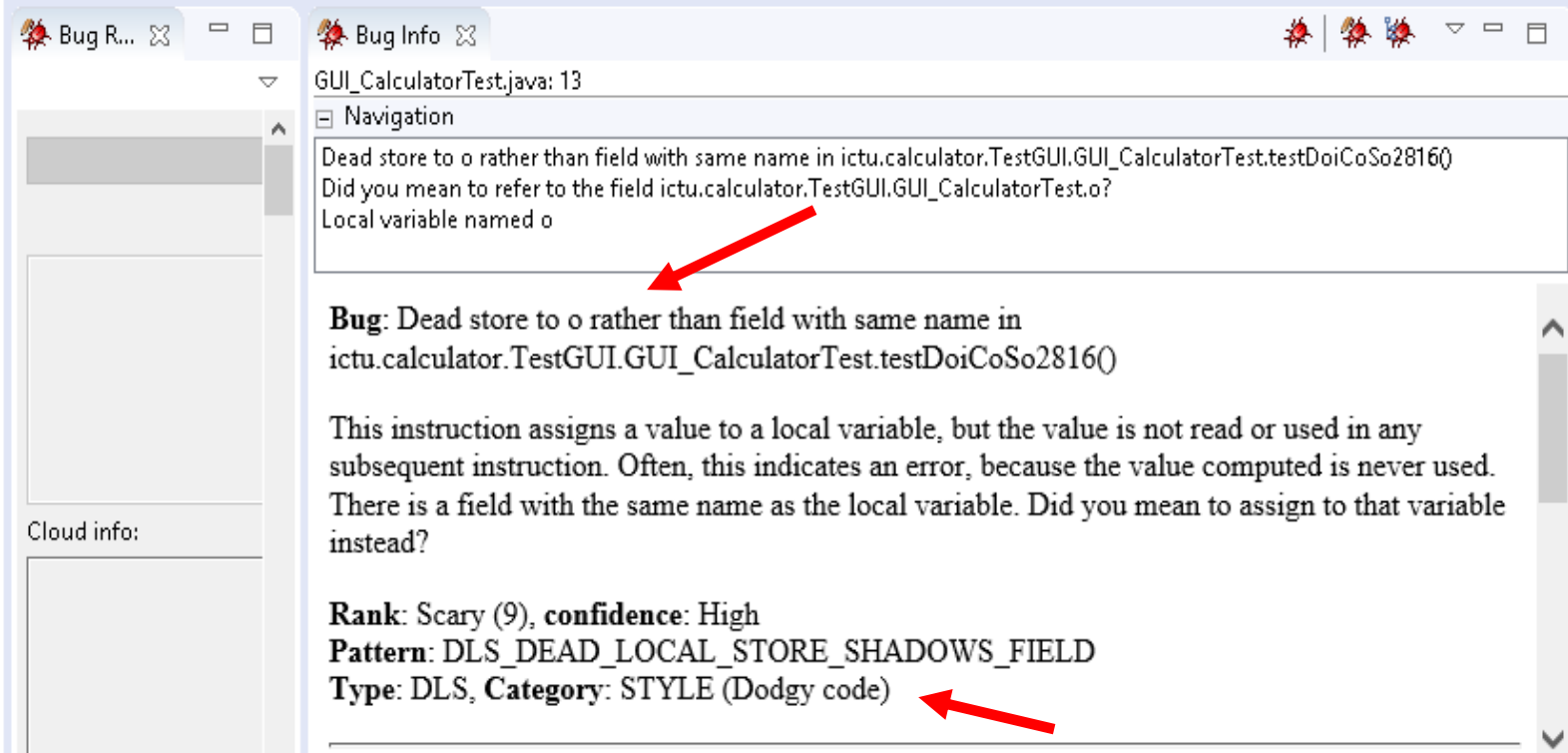
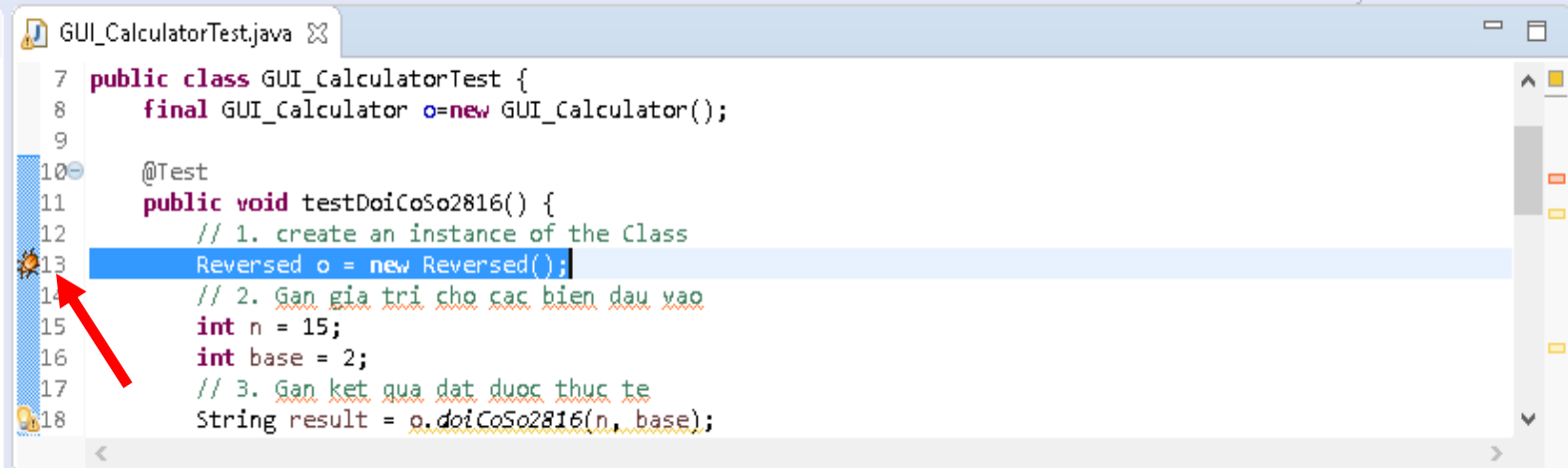
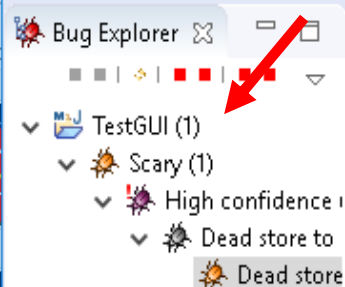
➤ <http://findbugs.cs.umd.edu/eclipse>

➤ Open Eclipse IDE

➤ Click on Help → Install New Software → Add → type: Name: FindBugsEclipse; URL: paste above URL (copied) → ...

➤ Restart Eclipse IDE, right click on project → FindBugs → FindBugs → ... and see results...

➤ Ví dụ: find bugs in **TestGUI project** (xem hình dưới)



5. Đóng gói định dạng plugin

➤ Ex.,: FindBugs™ Eclipse plugin

➤ Note:

➤ Can apply this plugin to find bugs on existing Java Archives (*.jar, *.ear, *.zip, *.war files, ...) by:

➤ Creating an empty Java Project and attach archives to the project class path

5. Đóng gói định dạng plugin

➤ Tạo plugin trong Eclipse

- Ví dụ: sau khi plugin into Eclipse, click on it, it will appearing a dialog with message: **“Welcome to ... the Software Operation & maintenance Subject – Year 2020”**

- File → New → Project → Plugin Project → Next → Type project name: **StringCheckPlugin** → Next → ... → Choose template: Hello, World Command → Next, edit message: → Finish → Open perspective (to plugin to Eclipse IDE)

➤ Chạy plugin:

- Right click on **StringCheckPlugin** → Run As → Eclipse Application → loading workbench → Eclipse along with plugin on menu bar → click icon to run plugging:

=> Kết quả (hình dưới)

File Edit Navigate Search Project Sample Menu Run Window Help



Say hello world (Ctrl+6)

Project Explorer

Biểu tượng
của plugin

Kết quả chạy
như hình dưới



Project Explorer

Icons for file operations and project navigation.

Outline Task List

An outline is not available

StringCheckPlugin

Welcome to ... the Software Operation and Maintenance Course - Year 2020

OK

5. Đóng gói định dạng plugin

- Thông tin chung của plugin
 - Use to declare dependences in the POM.xml file of Maven Project.
 - Xem hình (dưới)

General Information
This section describes general information about this plug-in.

ID:	<input type="text" value="StringCheckPlugin"/>	
Version:	<input type="text" value="1.0.0.qualifier"/>	
Name:	<input type="text" value="StringCheckPlugin"/>	
Vendor:	<input type="text"/>	
Platform Filter:	<input type="text"/>	
<u>Activator:</u>	<input type="text"/>	<input type="button" value="Browse..."/>

☐ Activate this plug-in when one of its classes is loaded

☒ This plug-in is a singleton



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
- 6. Tạo và phát hành web services**
7. Đóng gói ứng dụng mobile
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub

6. Deploying a web service

- Requirements:
 - Eclipse IDE for EE apps
 - Tomcat Apache Server
- How to create & deploy a web service onto server (ex., tomcat)?

=> 4 steps:

Directory
structure outline
of Tomcat and
Axis2 containers



6. Deploying a web service

► 4 steps:

1. Tạo một dynamic web project: **TestService**

► In this project, create java class file
(Ex., **Calculator.java**) & methods in it (ex., add(int, int) method)

⇒ All methods in this class will be a part of web service

⇒ This class will be web service

2. Make sure the Tomcat was started in Eclipse IDE

3. Make the **add(int, int)** method as a web service

4. Check the generated visual document in wsdl in WebContent of the project

=> URL of our web service:

<http://localhost:8080/TestServiceClient/sampleCalculatorProxy/TestClient.jsp>

6. Deploying a web service

- Demo in Eclipse IDE
 - ???
 - More details: see [this document](#)
- More details:
 - See link to practice
 - <https://www.youtube.com/watch?v=611A3SAf5-k>
 - More detail: see [this document](#)



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. **Đóng gói ứng dụng mobile**
8. Publish website lên web hosting
9. Deploy phần mềm sử dụng GitHub

7. Đóng gói ứng dụng mobile

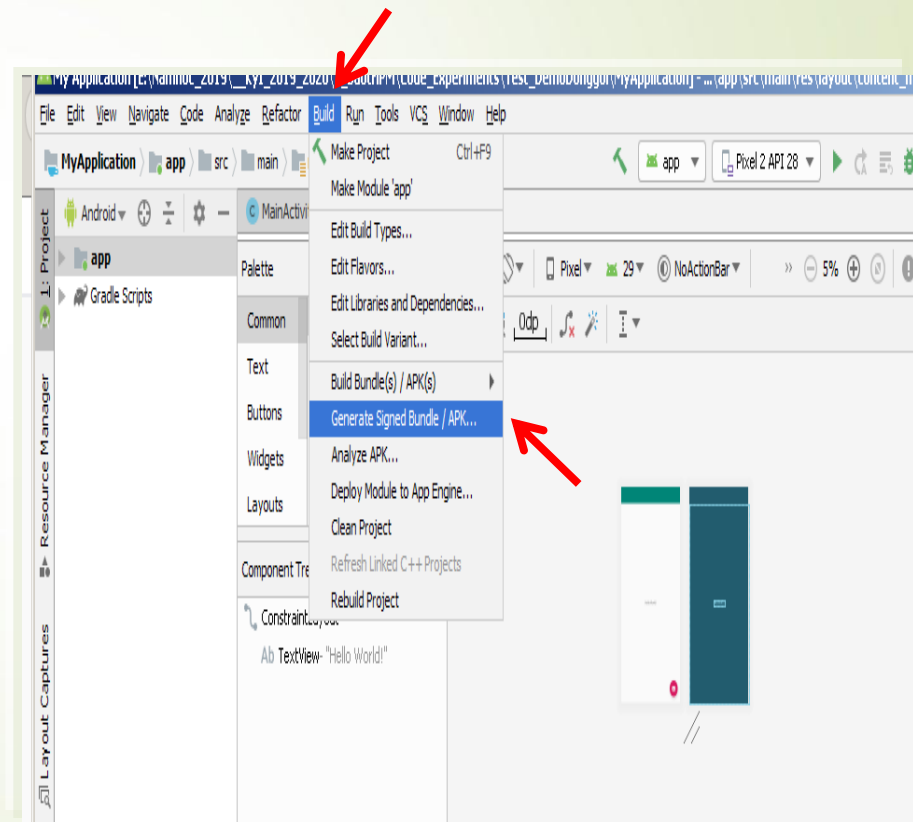
- Định dạng gói (phụ thuộc platform cụ thể):
 - *.ipa for iOS,
 - *.apk for Android
 - ~ Export ứng dụng thành một APK (Android Package) để publish ứng dụng lên chợ Google Play, App Stores, Website, or Amazon.

7. Đóng gói ứng dụng mobile

➡ Ví dụ:

➡ Export ứng dụng với **Android Studio**:

➡ **Bước 1**: Mở Project, chọn Build → Generate Signed APK:



7. Đóng gói ứng dụng mobile

➤ Bước 2:

- Nhập thông tin như màn hình
- Click Next

The screenshot shows the 'Generate Signed Bundle or APK' dialog box. It contains the following fields and controls:

- Module:** A dropdown menu showing 'app'.
- Key store path:** A text field containing 'eriments\Test_DemoDonggoi\Demo_Donggoi_mobile\Keystore.jks'. A red arrow points to this field with the text 'Nhập path của key store, nếu chưa có, nhấn tạo mới'.
- Key store password:** A password field with masked characters. A red arrow points to the '...' button next to it with the text 'ctu'.
- Key alias:** A text field containing 'key0'. A red arrow points to the '...' button next to it with the text 'K14'.
- Key password:** A password field with masked characters.
- Remember passwords:** An unchecked checkbox.
- Buttons:** 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Help'.

Red arrows and text annotations are used to guide the user through the process of entering the key store information.

7. Đóng gói ứng dụng mobile

➤ Bước 3:

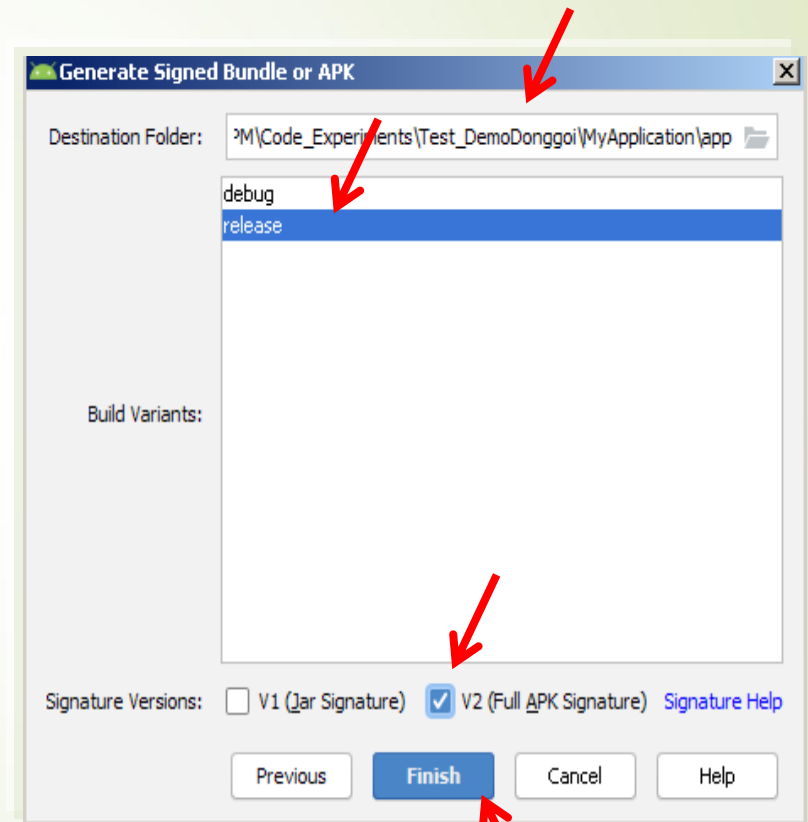
➤ Chọn:

- Release (đóng gói để phát hành),
- Debug (đóng gói để test và debug)

➤ Check:

- V2(Full APK Signature)

➤ Click Finish.



7. Đóng gói ứng dụng mobile

➤ Lưu ý:

- Build ứng dụng iOS độc lập => cần 1 tài khoản Apple Developer
 - Tuy nhiên không cần với Android app.
- Submit app lên app store
 - Cần tài khoản developer trên kho đó.
 - Upload lên Google Play cần trả phí



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
- 8. Đóng gói và publish website lên web hosting**
9. Deploy website sử dụng GitHub



8. Publish website lên web hosting

- Bước 1: Chọn nhà cung cấp hosting đáng tin cậy
- Bước 2. Chọn phương pháp upload website
- Bước 3. Upload website (source code) lên hosting
- Bước 4: Kiểm tra xem file đã ở trong thư mục public html chưa
- Bước 5: Upload database lên web hosting
- Bước 6: Kiểm tra xem website hoạt động bình thường không

8. Publish website lên web hosting

- Một số dịch vụ web hosting (free):
 - Neocities, Blogspot, và Wordpress.
 - GitHub, Dropbox, ...
 - GitHub có một tính năng rất hữu dụng gọi là GitHub pages, cho phép ta có một website trên mạng với URL: **<https://username.github.io>**.
 - Link: <https://pages.github.com/>
 - Dropbox: Nếu upload trang web, nó sẽ được hiển thị dưới dạng trang web một cách tự động.
 - More details: see [Host websites With Dropbox](#)

8. Publish website lên web hosting

► SV thực hành:

1. Đóng gói ứng dụng Web (ví dụ: *.war)
 - ???
2. Deploy the website (after packaging - *.war file onto the server (ex., Tomcat Apache)
 - By putting the *.war inside in the WebApps folder of Tomcat.
3. Test website after deploying:
 - Mở Tomcat on Web browser & test:
 - URL: <http://localhost:8080/<ten gói>/<ten trang.jsp>>
 - Ví dụ: http://localhost:8080/TestWebsite_2020/Hello.js

8. Publish website lên web hosting

➤ Ví dụ:

- Thực hành trên dự án: TestWebsite_2020 project
- URL truy cập website after deploying to Tomcat server:
 - http://localhost:8080/TestWebsite_2020/Hello.jsp



Nội dung chính

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Đóng gói và publish website lên web hosting
9. **Deploy website sử dụng GitHub**



9. Publish website to web hosting

- Ví dụ:

- **Xuất bản qua GitHub**

- Sử dụng tính năng GitHub pages.

Xuất bản Web qua GitHub

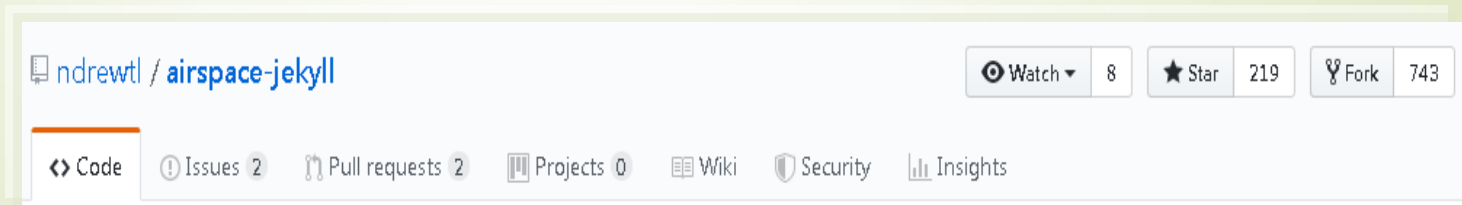
- Ví dụ:

- Dùng mẫu website [Airspace Jekyll template](#) của Andrew Lee.
- Một số link themes website khác:
 - <http://themes.jekyllrc.org/>
 - <http://jekyllthemes.org/>
 - <http://drjekyllthemes.github.io/>
 - Neocities, Blogspot, và Wordpress.
 - <https://colorlib.com/wp/free-blogger-templates/>
- Các bước publish?

Xuất bản Web qua GitHub

➤ Bước 1:

- Truy cập vào tài khoản GitHub
 - Tạo kho GitHub và đưa app lên đó, hoặc
 - Truy cập vào link của dự mẫu trên:
<http://themes.jekyllrc.org/airspace/>
 - Fork kho chứa template code bằng cách:
 - Nhấn “Homepage” để có repo, sau đó click vào nút “Fork” (on the top right-hand corner):
- => Kết quả: copy toàn bộ kho này sang kho GitHub của ta



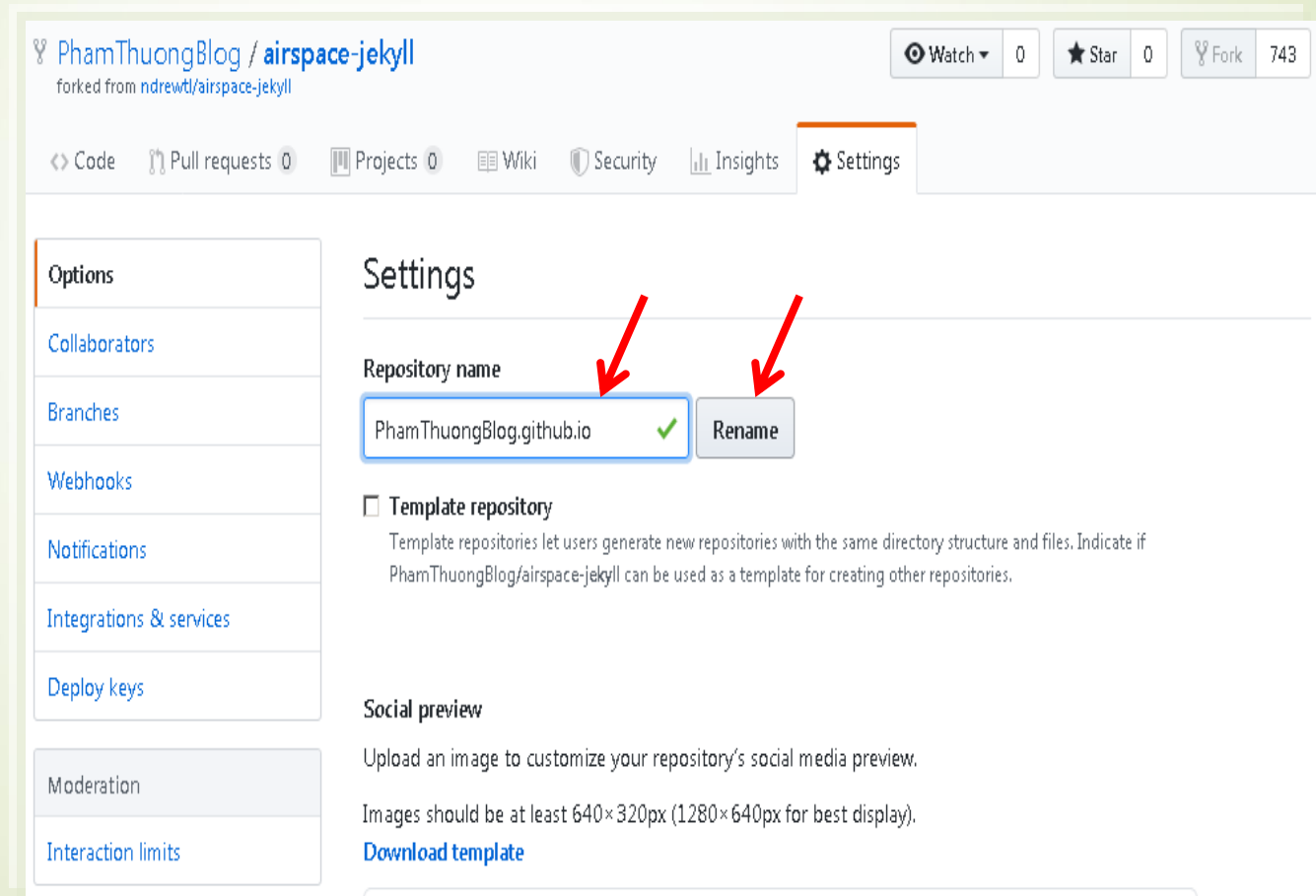
Xuất bản Web qua GitHub

=> Sau 1 vài giây, github sẽ full “copy” of the designated website đến tài khoản GitHub.

➤ Bước 2:

- Chọn tab: Settings tab, đặt tên kho là: *username.github.io*
 - Username là tên tài khoản GitHub:
 - ví dụ: **PhamThuongBlog**
- Click Rename:

Xuất bản Web qua GitHub



The screenshot shows the GitHub repository settings page for 'PhamThuongBlog / airspace-jekyll'. The repository is a fork of 'ndrewtl/airspace-jekyll'. The 'Settings' tab is selected, and the 'Repository name' field is highlighted with a blue border and a green checkmark. Two red arrows point to the 'Repository name' field and the 'Rename' button. The 'Template repository' checkbox is unchecked. The 'Social preview' section is visible at the bottom.

PhamThuongBlog / **airspace-jekyll**
forked from ndrewtl/airspace-jekyll

Watch 0 Star 0 Fork 743

Code Pull requests 0 Projects 0 Wiki Security Insights Settings

Options

- Collaborators
- Branches
- Webhooks
- Notifications
- Integrations & services
- Deploy keys

Settings

Repository name
PhamThuongBlog.github.io ✓ [Rename](#)

☐ **Template repository**
Template repositories let users generate new repositories with the same directory structure and files. Indicate if PhamThuongBlog/airspace-jekyll can be used as a template for creating other repositories.

Social preview
Upload an image to customize your repository's social media preview.
Images should be at least 640×320px (1280×640px for best display).
[Download template](#)

Xuất bản Web qua GitHub

➡ Bước 3:

➡ Cấu hình template cho Github như sau:

➡ Sửa file: `_config.yml`.

➡ Mỗi theme yêu cầu tùy biến các biến khác nhau (một số fields cần điền để khởi động website) .

=> Mẫu Airspace: sửa 2 biến bắt buộc là `url` và `baseUrl`:

```
16 # Site settings
17 title: Airspace
18 subtitle: "A free HTML5 agency template"
19 url: "http://PhamThuongBlog.github.io" # the base hostname & protocol for your site
20 # baseUrl: "/the/subpath/of/your/site" # Set this value to "" if you want your site to be root
21 baseUrl: "/"
22 cover: "/assets/cover.png"
23 logo: "/assets/logo.png"
24
```

Xuất bản Web qua GitHub

➤ Bước 3:

➤ Cuộn trang xuống và click: “Commit changes”.

➤ Kết quả, URL của website:

<https://PhamThuongBlog.github.io/>

⇒ Mở URL trong trình duyệt để run website.

Màn hình kết quả (dưới)



[Home](#) [Work](#) [Blog](#) [Service](#) [Contact](#)

A DIGITAL MARKETING & DESIGN AGENCY

We love the Web and the work we do. We work closely with our clients to deliver the best possible solutions for their needs



Tóm lại

1. Đóng gói, phát hành p.mềm
2. Đóng gói định dạng *.Jar (Java Archive)
3. Đóng gói dạng exe installer
4. Đóng gói p.mềm dạng portable
5. Tạo và đóng gói dạng plugin
6. Tạo và phát hành a web service
7. Đóng gói ứng dụng mobile
8. Đóng gói và publish website lên web hosting
9. Deploy website sử dụng GitHub



Tài liệu tham khảo

- Github pages:

- <https://medium.com/@vovakuzmenkov/get-a-website-with-github-pages-9151ed636446>

Discussion





Bài kiểm tra số 1



Next Topic:

Quản lý mã nguồn
với Git&GitHub