

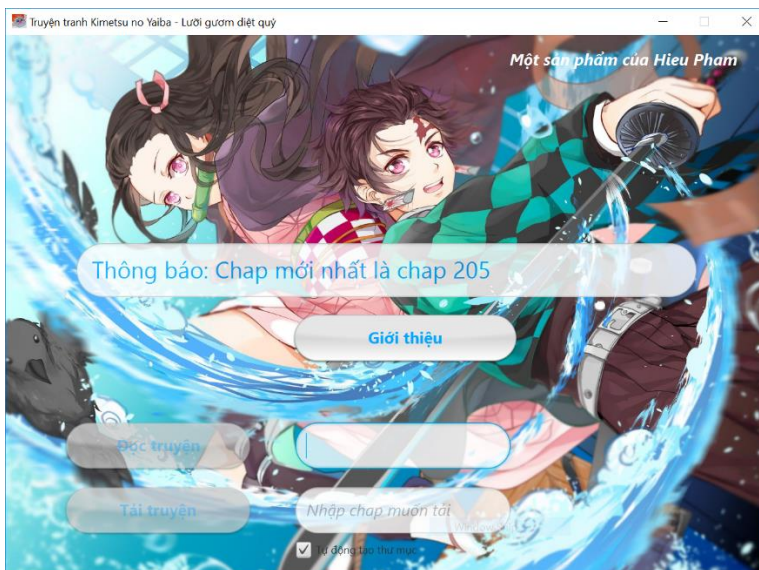
## Project 2: Crawl truyện tranh Kimetsu no Yaiba

### 1. Giới thiệu:

- Điểm cốt lõi: sử dụng Jsoup để crawl dữ liệu
- Phát triển: mở rộng các chức năng (đọc, tải và lưu về máy), sử dụng javafx để tạo giao diện, xử lý các sự kiện

#### 1.1. Giao diện và chức năng:

##### 1.1.1. Giao diện chính:



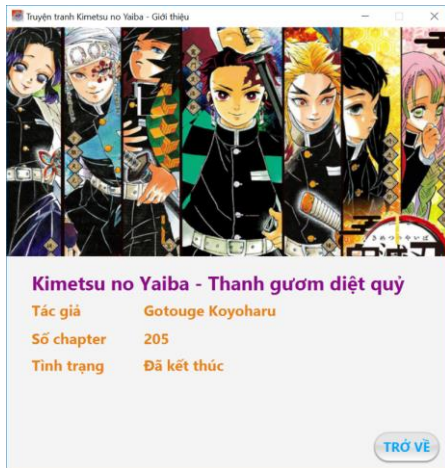
Chú ý là: Icon được set (cả bên ngoài và bên trong – tức trên title)

#### Chức năng:

- Ô hiển thị thông báo về chap mới nhất (để người dùng biết giới hạn khi nhập chap muốn tải/đọc).
- Button giới thiệu => click để chuyển sang cửa sổ Giới thiệu
- Button Đọc truyện và ô Nhập chap muốn đọc: value nhập vào phải thỏa mãn điều kiện ( $1 \leq \text{value} \leq 205$ ) thì nút Đọc truyện mới hiện ra. Khi đó click vào sẽ chuyển sang cửa sổ Đọc truyện với chap tương ứng.
- Button Tải truyện, ô Nhập chap muốn tải và checkbox Tự động tạo thư mục: value nhập vào phải thỏa mãn điều kiện ( $1 \leq \text{value} \leq 205$ ) thì nút Tải

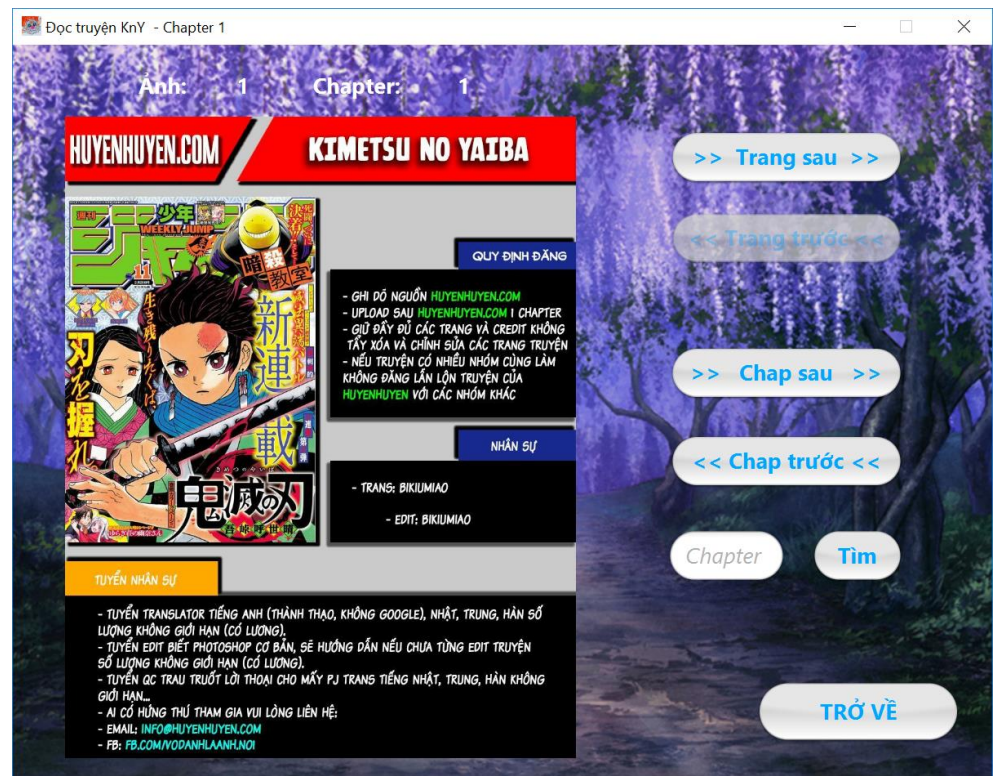
truyện mới hiện ra. Khi đó click vào nút Tải truyện thì lập tức chap sẽ được lưu về máy. Mặc định check box đã được tick, nghĩa là toàn bộ ảnh thuộc chap truyện tải về sẽ được gói trong 1 thư mục có tên tương ứng. Nếu bỏ tick, thì sẽ không tạo thư mục gói các ảnh.

### 1.1.2. Giao diện Giới thiệu:



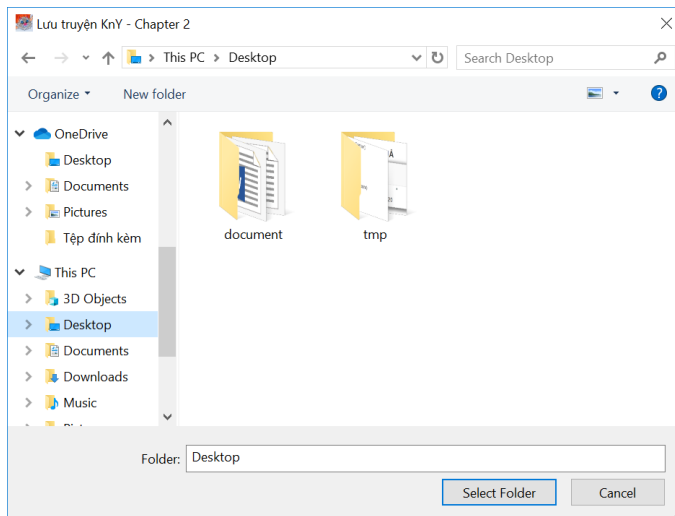
Chỉ để cung cấp 1 số thông tin. Có button trở về để quay lại giao diện chính.

### 1.1.3. Giao diện đọc truyện:



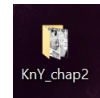
VD ở đây là đọc chap 1. Một chap sẽ gồm nhiều ảnh. Dùng các button bên phải để chuyển ảnh và chuyển chap. Button Tìm để chuyển đến chap muốn đọc nhanh. Việc hiển thị ảnh được lập trình để tăng trải nghiệm (lazy-load ??)

#### 1.1.4. Tải truyện:



Cho phép chọn vị trí thư mục lưu truyện. Do chưa được tối ưu nên việc lưu truyện sẽ hơi lâu và trong lúc đó app sẽ not responding. Khi tải về hoàn tất sẽ có thông báo.

Với việc tick check box Tự động tạo thư mục thì kết quả ta thu được là



1.2.Link web: <https://truyentranh24.com/kimetsu-no-yaiba>

Lý do chọn:

- Có thể phân tích được element HTML chứa thông tin cần lấy một cách dễ dàng
- Link ảnh có thể truy cập/save về mà không bị chặn (lỗi 403). Hầu hết các web khác đều gặp lỗi này.

1.3.Link bài viết hướng dẫn: <https://codelearn.io/sharing/dung-jsoup-java-de-crawl-truyen-onepiece>

Tuy nhiên, link truyện trong bài viết không thể crawl được (mình gặp khó khăn khi crawl: không thể phân tích chính xác element HTML chứa thông tin cần lấy).

#### 1.4.Maven:

Jsoup, Javafx

## 2. Chi tiết:

Code nằm ở src/main/java, gồm 3 package: application (chứa class Main để chạy), gui (giao diện và controller) và handle (chứa các class phục vụ việc crawl)

### 2.1.Package handle:

**2.1.1. Class Chap:** chứa thuộc tính url (đường dẫn tới chap) và chapNumber (là số chỉ chap, vd chap 1, 2 ...). chapNumber sẽ được lấy từ url (ta sẽ phân tích đặc điểm của url) thông qua method findChapNumber.

#### 2.1.2. Class Crawler:

Đoạn code cần chú ý khi crawl (chú ý sử dụng linh hoạt tùy thuộc vào cấu trúc trang web)

VD:

```
Document document = Jsoup.connect(url).get();
```

```
Elements elements = document.getElementsByClass("chapter-item")
```

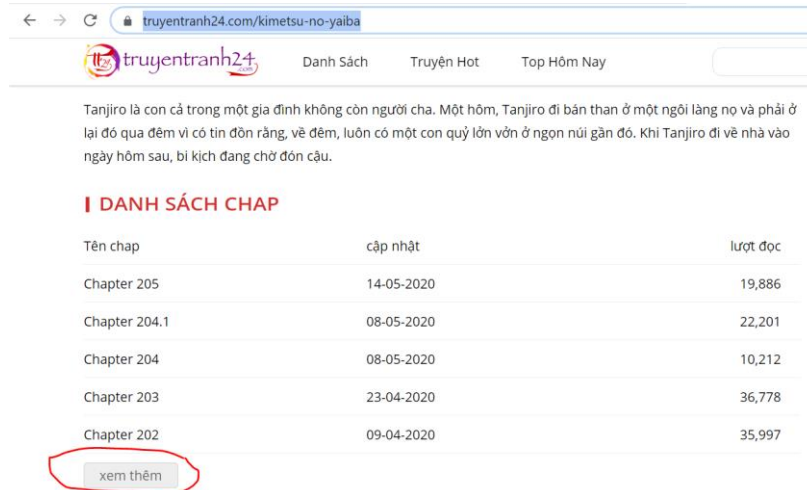
```
for (int i = 0; i < elements.size(); i++) {
```

```
    Elements eTag = elements.get(i).getElementsByTag("a");
```

```
    for (int j = 0; j < eTag.size(); j++) {
```

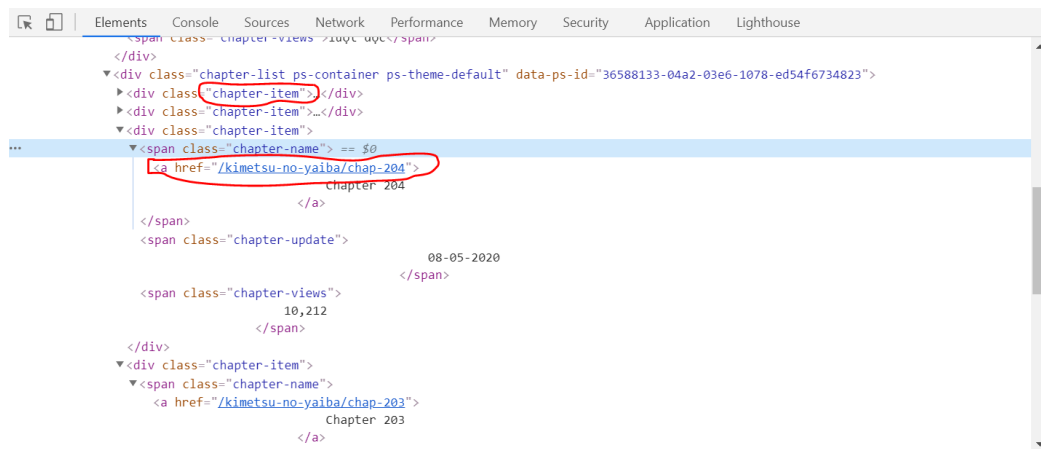
```
        String linkChap = eTag.first().absUrl("href");
```

- **Method getAllChaps**: từ link url trang chính của truyện, ta sẽ lấy ra link của toàn bộ chap truyện. Có thể trả về List hoặc HashMap. List sẽ phù hợp cho những việc cần thứ tự, HashMap thì phù hợp hơn khi cần search (List cũng có thể search nhanh, nhưng cần có quan hệ tương ứng giữa index trong list và số chỉ của chap)



The screenshot shows the website [truyen tranh 24.com/kimetsu-no-yaiba](http://truyen tranh 24.com/kimetsu-no-yaiba). It features a header with the site logo and navigation links. Below the header is a short synopsis of the story. The main content is a table titled "DANH SÁCH CHAP" (Chapter List) with three columns: "Tên chap" (Chapter Name), "cập nhật" (Updated), and "lượt đọc" (Views). The table lists chapters from 202 to 205. A button labeled "xem thêm" (View more) is located below the table.

Tên chap	cập nhật	lượt đọc
Chapter 205	14-05-2020	19,886
Chapter 204.1	08-05-2020	22,201
Chapter 204	08-05-2020	10,212
Chapter 203	23-04-2020	36,778
Chapter 202	09-04-2020	35,997



**Vấn đề phát sinh**: có nút "Xem thêm" => không thể lấy toàn bộ link chap (vì link không hiện trực tiếp trên web) => giải pháp khắc phục tạm thời: link chap thường có cấu trúc giống nhau, chỉ khác nhau bởi số thứ tự của chap.

Vd:

<https://truyen tranh 24.com/kimetsu-no-yaiba/chap-204> và

<https://truyen tranh 24.com/kimetsu-no-yaiba/chap-205>

nên ta chỉ cần replace string là ok. Lấy link của chap newest (bởi ta sẽ luôn lấy được link này), dùng vòng for cho số thứ tự chap chạy từ 1 => newest, replace số thứ tự trong link => ta có toàn bộ link của các chap.

- **Method findLastestChap**: tìm chap mới nhất, phụ thuộc vào cấu trúc trình bày của trang web mà ta có cách tìm khác nhau.

- **Method getAllImagesOnPage**: 1 chap sẽ gồm nhiều ảnh, vì vậy ta sẽ lấy toàn bộ ảnh của 1 chap (khi biết link của chap), và cho vào List(vì rõ ràng các ảnh này có thứ tự, không cần truy xuất trực tiếp).

- **Method saveImage**: tham số truyền vào là link ảnh, tên ảnh và đường dẫn vị trí lưu. Ta sẽ không dùng trực tiếp mà dùng trong saveChapter.

Chú ý cách save này có thể vận dụng sau này.

```
public static void saveImage(String src, String name, String dir) {
    try {
        URL url = new URL(src);
        InputStream in = url.openStream();

        OutputStream out = new BufferedOutputStream(new FileOutputStream(dir + "\\\" + name));
        for (int i; (i = in.read()) != -1;) {
            out.write(i);
        }

        out.close();
        in.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- **Method saveChapter**: ta sẽ lấy list ảnh của chap, sau đó duyệt list và lưu từng ảnh. Chú ý chỗ khoanh đỏ:

```
ArrayList<String> listImgs = getAllImagesOnPage(srcChap);
for (int i = 0; i < listImgs.size(); i++) {
    saveImage(listImgs.get(i), (i+1) + ".jpg", dir);
}
```

Chú ý: cách save chưa tối ưu (trải nghiệm chờ tải về hoàn tất khá tệ, có thể lâu và chương trình not responding)

Mở rộng: saveManga – tức save tất cả chap



```

public static void saveManga(String url, String dir) throws IOException {
    HashMap<String, Chap> chaps = Crawler.getAllChaps(url);
    (new File(dir + "\\Kimetsu_no_Yaiba")).mkdir(); // tạo thư mục
    for(String k: chaps.keySet()) {
        String name = "KnY_chap" + k;
        (new File(dir + "\\Kimetsu_no_Yaiba\\" + name)).mkdir(); // tạo thư mục
        saveChapter(chaps.get(k).getUrl(), dir + "\\Kimetsu_no_Yaiba\\" + name);
    }
}

```

Song khá nguy hiểm, vì chạy rất lâu, máy có thể bị treo. Ngoài ra crawl quá nhanh và nhiều thì có thể bị lỗi. Và do method saveChapter có thông báo sau khi hoàn tất việc save 1 chap, nên khi saveManga sẽ bị quá tải thông báo (vì số thông báo = số chap).

**Chú ý:** Ở đây ta học được cách tạo thư mục:

`(new File(đường dẫn + name)).mkdir();`

**Vấn đề:** Lỗi khi save ảnh – lỗi forbidden 403 => phải tìm web khác để crawl



## 2.2.Package gui

**2.2.1. App:** set giao diện ban đầu, sử dụng Gui.fxml và Gui.css.

```

Parent root = FXMLLoader.load(getClass().getResource("Gui.fxml"));
Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("Gui.css").toExternalForm());

```

**Cú pháp để set icon cho app** (hiện khi chạy, vd trên title...)

```

// để icon ở gui thì không lỗi, để chỗ khác và dẫn địa chỉ tuyệt đối lại lỗi???
primaryStage.getIcons().add(new Image("/gui/icon.jpg"));

```

**Chú ý:** Không thể chạy trực tiếp class này, sẽ gặp lỗi

**Error: JavaFX runtime components are missing, and are required to run this application**

Ta khắc phục bằng cách chạy gián tiếp thông qua class Main trong package application.

**2.2.2. AllScene:** set các scene tương ứng (được sử dụng trong các controller để xử lý sự kiện)

Ta có: setReadScene (khi nhấn nút đọc truyện), setIntroScene (khi nhấn nút Giới thiệu), setMainScene (khi nhấn nút Trở về)

Tương ứng ta sử dụng GuiRead.fxml, GuiRead.css và GuiIntro.fxml, GuiIntro.css.

**Chú ý:** Ta lấy ra stage từ event truyền vào, vd:

```
public void setReadScene(ActionEvent event, String chapNum) {  
    try {  
        Stage stage = (Stage)((Node)event.getSource()).getScene().getWindow();
```

**Vấn đề:** việc set full màn hình chưa làm được như ý muốn.

**2.2.3. Controller:** controller của Gui.fxml, xử lý các sự kiện của Main Scene.

- **Nhắc lại về Controller:** implements Initializable khi cần phải khởi tạo ban đầu....???? Để xử lý sự kiện, ta tạo thuộc tính @FXML tương ứng và set fx:id cho phần tử trong file fxml; rồi tạo method xử lý sự kiện (tham số truyền vào là ActionEvent event – nhớ là import cho đúng javafx.event.ActionEvent)

- Ta cần set và control các button (giới thiệu, đọc truyện, tải truyện), text field (thông báo, nhập chap muốn đọc, muốn tải), checkbox (tự động tạo thư mục khi tải về)

- **Override method initialize:**

- + Hiện thông báo chap mới nhất trên text field thông báo (dùng method findLatestChap của Crawler)



+ **set Disable** cho các button Đọc truyện và Tải truyện. Bởi vì ta chỉ cho phép người dùng click các nút đó khi value nhập vào thỏa mãn (không được để trống, phải nằm trong khoảng giá trị cho phép).

Mặc định ban đầu set true (tức ẩn button).

Sau đó:

```
numberDownload.textProperty().addListener((obs, oldValue, newValue) -> {  
    try {  
        // có thể phát sinh lỗi ở đây, nếu newValue trống  
        double value = Double.parseDouble(newValue);  
        double max = Double.parseDouble(number);  
        if(value >= 1 && value <= max ) {  
            download.setDisable(false);  
        } else {  
            // phải có else  
            download.setDisable(true);  
        }  
    } catch (Exception e) {  
        download.setDisable(true);  
    }  
});
```

Ta không thể set thông thường, mà phải addListener và dùng bộ giá trị (obs, oldValue, newValue) (ở đây ta có dùng biểu thức lambda -> {} ???). Chú ý phải có else. Lưu ý có rất nhiều trường hợp xảy ra: vd người dùng nhập 1 (set false), nhưng nếu họ tiếp tục nhập đến 1111 thì mình phải set true, tức việc set disable phải thay đổi với từng hành động. Nếu không có else thì sẽ không xử lý được trường hợp trên.

- **Method clickIntro**: set scene thành Intro Scene
- **Method clickRead**: set scene thành Read Scene.

**Chú ý**: ta có 1 biến static **messageToReadScene** để transfer message giữa Controller và ControllerRead (đây chỉ là giải pháp tình thế). Lý do là việc set ReadScene độc lập với ControllerRead, nên khi đó ControllerRead không biết được ReadScene là của chap nào, dẫn tới không thể xử lý các hành động như set image của chap, next chap, ... messageToReadScene chính là biến dùng để lưu trữ số chỉ chap được set cho Read Scene.

Vấn đề: ta phải dùng giải pháp trên vì:

```
// truyền dữ liệu giữa 2 controller nhưng không hiểu sao bị lỗi null pointer exception
// public String transferMessageToReadScene() {
//     return numberRead.getText();
// }
}
```

#### - Method clickDownload:

Ta sẽ mở Directory Chooser cho người dùng chọn thư mục lưu trữ. Ta phải set owner cho chooser.showDialog(...) để tạo sự phụ thuộc giữa hộp thoại chooser và cửa sổ hiện tại. Lấy stage từ event:

```
Stage stage = (Stage)((Node)event.getSource()).getScene().getWindow();
```

Và:

```
File selectedDirectory = chooser.showDialog(stage);
```

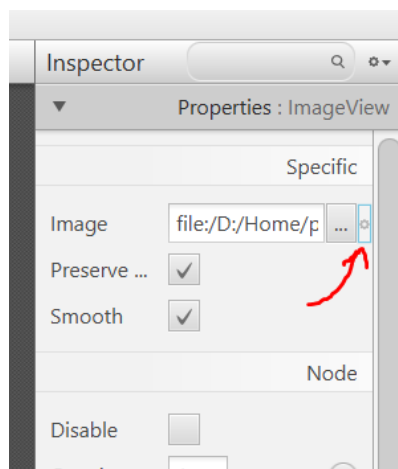
Ta sẽ lấy ra chap cần download từ HashMap (có được từ việc getAllChaps), đồng thời ta lấy đường dẫn tuyệt đối từ selectedDirectory.

```
Chap chap = Crawler.getAllChaps(LINK_WEB).get(chapNum);
String path = selectedDirectory.getAbsolutePath();
```

Kiểm tra checkbox “Tự động tạo thư mục”. Nếu nó được selected thì ta sẽ tạo 1 thư mục trước khi thực hiện save chap.

#### 2.2.4. Gui.fxml: có 1 số lưu ý ta cần quan tâm

Về việc set background (thông qua chọn Image cho Image View trong Scene Builder).



Để ảnh hiện lên, ta phải click nút nhỏ bên cạnh, chọn Switch to absolute path

Với text field, cần chú ý 1 số thông số: prompt text, editable, focus tran... (quên mất để làm gì rồi ☹ )

#### 2.2.5. *Gui.css*: có 1 số điều cần quan tâm

Viết .button thì sẽ set cho toàn bộ button trong file fxmll, tương tự với .text-field

2.2.6. *ControllerIntro*: Chỉ đơn giản xử lý sự kiện nhấn nút Trở về => setMainScene

#### 2.2.7. *ControllerRead*:

Ta sử dụng các biến chaps (lưu toàn bộ chap), chap (lưu chap hiện tại), listImgs (lưu list ảnh của chap hiện tại). Ngoài ra có các biến indexImg và indexChap dùng cho các thao tác next, pre ảnh và chap.

##### - **Override method initialize:**

+ Nhận transfer message từ Controller (value số chỉ chap được set) => set indexChap, còn indexImg = 0.

+ Khởi tạo chaps => chap => listImgs

+ Khi đã có listImgs => set Image View()

**Xử lý set Disable cho các button** (rất nhiều vấn đề ở đây, nên giải pháp đưa ra chỉ là giải pháp tình thế)

+ Button Find: tương tự như set cho nút Đọc truyện/ Tải truyện

+ Button nextImg và preImg:

+ Button nextChap và preChap:

2 dấu cộng sau khá giống nhau, nhưng không hiểu sao xảy ra lỗi logic với button nextImg (khi xem hết ảnh của 1 chap => nextImg được set disable true => lúc sau nếu setNewChap => button newImg vẫn disable true => phải dùng giải pháp tình thế ở setNewChap). Tức setNewChap không tác động đến addListener của 2 nút preImg và nextImg??? (nút preImg có thể chỉ may mắn đúng ???)

- **Method clickReturnMainScene**: set về Main Scene
- **Các method**: **setTitle** (nhận tham số event, dùng khi setNewChap), **setLabelNumberChap**, **setLabelNumberImg** (thay đổi label cho phù hợp tình trạng hiện tại của chap/img)
- **Method setImageView()**: set image cho image view dựa trên indexImg hiện tại

```
public void setImageView() {  
    Image img = new Image(listImgs.get(indexImg), true); // có true và không có true sẽ có sự khác biệt khi load ảnh  
    setLabelNumberImg();  
    imageView.setImage(img);  
}
```

- **Method setNewChap(ActionEvent event)**: set chap  
+ numberChap => chap tương ứng => listImgs tương ứng  
+ setTitle(event) (cần truyền event để lấy được stage)  
+ indexImg = 0 (vì chuyển sang chap mới thì mặc định đọc từ đầu) => setImageView()  
+ set disable(false) cho button nextImg (đây là giải pháp tình thế)
- **Method điều khiển button thay đổi Img (next, pre)**: Kiểm tra điều kiện indexImg => thay đổi indexImg => setImageView
- **Method điều khiển button thay đổi chap (next, pre)**: Kiểm tra điều kiện indexChap => thay đổi indexChap => setNewChap

- **Methof clickFind(event):** lấy value từ ô tìm kiếm rồi set indexChap => setNewChap

### 2.3.Package application:

Chỉ chứa class Main (để thực thi chương trình), giải pháp khắc phục lỗi (khi chạy trực tiếp class App).

## 3. Fix lỗi:

3.1.Lỗi: “Error: JavaFX runtime components are missing, and are required to run this application” khi chạy class App

<https://stackoverflow.com/questions/56894627/how-to-fix-error-java-fx-runtime-components-are-missing-and-are-required-to-run>

⇒ Viết thêm class Main

3.2.Tìm hiểu thêm: Đường dẫn tương đối và tuyệt đối

<https://vuthai-vision.blogspot.com/2020/07/uong-dan-tuong-oi-va-uong-dan-tuyet-oi.html>

3.3.Tìm hiểu thêm: Biểu thức Lambda

<https://viblo.asia/p/cung-tim-hieu-ve-bieu-thuc-lambda-QpmleAEDlrd>

3.4.Lỗi khi set path ảnh trong javafx css

<https://stackoverflow.com/questions/16630487/correct-way-to-set-the-path-of-a-backgroundimage-in-javafx-css>

Đọc thêm: Cách set background cho scene

<https://stackoverflow.com/questions/9738146/javafx-how-to-set-scene-background-image>

3.5.Lỗi không hiện ảnh trên ImageView trong Scene Builder

<https://stackoverflow.com/questions/29397367/javafx-scenebuilder-imageview-not-working>

⇒ Switch to absolute path

3.6. Lỗi prompt text không hiện ở text field???

<https://stackoverflow.com/questions/23039898/setprompttext-function-not-working-initially-with-textfield>

Chú ý đọc kỹ cả comment vì có tác dụng phụ

3.7. Set full màn hình:

<https://stackoverflow.com/questions/31426912/how-to-make-window-fullscreen-maximized-in-scene-builder>

và

<https://stackoverflow.com/questions/36197764/javafx-stage-resizable-false-and-maximized-true-hides-the-taskbar>

3.8. Directory Chooser

<https://stackoverflow.com/questions/9375938/javafx-filechooser>

3.9. java.io.IOException: Server returned HTTP response code: 403 for URL

<https://stackoverflow.com/questions/4797593/java-io-ioexception-server-returned-http-response-code-403-for-url>

và

<https://stackoverflow.com/questions/16181159/imageio-read-returns-403-error>

Nhưng vẫn không fix được, có vẻ do trang web chặn crawl => phải tìm web khác



3.10. Tìm hiểu thêm: difference of artifactId and name in maven POM

<https://stackoverflow.com/questions/38594036/difference-of-artifactid-and-name-in-maven-pom>

3.11. Giao tiếp giữa các Controller

<https://www.genuinecoder.com/javafx-communication-between-controllers/>

và

<https://stackoverflow.com/questions/14187963/passing-parameters-javafx-fxml>

Tuy nhiên trong khi làm theo thì bị lỗi:

<https://stackoverflow.com/questions/23461148/fxmlloader-getcontroller-returns-null>

Cuối cùng vẫn không fix được nên đã dùng giải pháp tình thế là biến static messageToReadScene

3.12. Tìm hiểu: Lazy load images in javafx

<https://stackoverflow.com/questions/56249776/how-to-lazy-load-images-in-javafx>

Dùng flag “true”

```
Image myImage= new Image("/icons/test.png",true);
```

3.13. Thay đổi icon xuất hiện trên title và taskbar (launch4j không làm được việc này)

<https://stackoverflow.com/questions/55026550/app-icon-wrapped-with-launch4j-not-changing-in-taskbar>