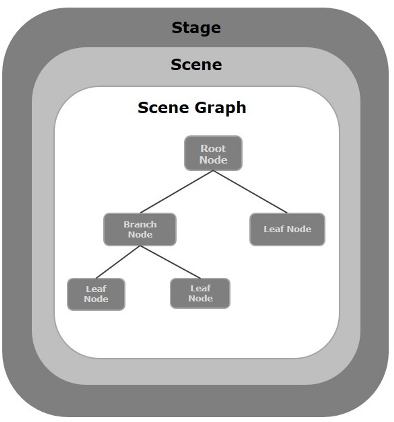
**CẤU TRÚC ỨNG DỤNG CỦA JAVAFX.**

1. **Tổng quan các thành phần của JavaFX.**

Nói chung một ứng dụng JAVAFX sẽ có ba thành phần chính là Stage, Scene và Nodes như trong sơ đồ sau.



* **Stage:** (sân khấu)

Giai đoạn (một cửa sổ) chứa tất cả các đối tượng của ứng dụng JavaFX. Nó được đại diện bởi lớp Stage của gói javafx.stage. Stage chính được tạo bởi chính nền tảng của nó. Đối tượng stage đã tạo sẽ được truyền như một đối số cho phương thức start() của lớp Application (được giải thích trong phần tiếp theo).

Mỗi sân khấu có hai tham số xác định vị trí của nó là width và height. Nó được chia thành Content Area (khu vực nội dung) và Decoration (Trang trí) (thanh tiêu đề và đường viền).

Có 5 loại sân khấu có sẵn:

* + Trang trí.
  + Chưa trang trí.
  + Trong suốt.
  + Thống nhất
  + Tiện ích.

Bạn phải gọi phương thức show() để hiển thị nội dung của một vùng.

* **Scence(Bối cảnh).**

Một bối cảnh tượng trưng cho nội dung vật lý của một ứng dụng JavaFX. Nó chứa tất cả nội dụng của một biểu đồ cảnh. Lớp Scence của gói javafx.scence đại diện cho đối tượng cảnh. Tại một thể hiện, đối tượng cảnh chỉ được thêm vào một giai đoạn.

Bạn có thể tạo một cảnh bằng cách khởi tạo lớp Scence. Bạn có thể chọn kích thước của cảnh bằng cách truyền kích thước của nó (chiều cao và chiều rộng) cùng với nút gốc tới hàm của nó.

* **Scence Graph và Node.**

Một Scence Graph là một cây giống như cấu trúc dữ liệu thể hiện nội dung của một Scence. Ngược lại, một Node là một đối tượng trực quan/ đồ họa của biểu đồ cảnh.

Một Node bao gồm:

* + Các đối tượng hình học (đồ họa) (2D, 3D) như hình tròn, hình chữ nhật, đa giác,…
  + Các điều khiển giao diện người dùng như – Nút, hộp kiểm, hộp lựa chọn, vùng văn bản,…
  + Các vùng chứa (Ngăn bố cục) chẳng hạn như Border Pane, Grid Pane, Flow Pane,…
  + Các phần tử media như đối tượng âm thanh, video và hình ảnh.

Lớp Node của gói javafx.scence đại diện cho một nút trong JavaFX, lớp này là siêu lớp của tất cả các node.

* + Root Node – Scence Graph đầu tiên được gọi là Root Node.
  + Branch Node/ Parent Node: Node có các node con được gọi là branch node/ parent node. Abstract class có tên là Parent của gói javafx.scence là lớp cơ sở của tất cả các parent node và các parent node đó sẽ thuộc các loại sau:
    - Group – Group node là một node tập hợp có chứa danh sách các nút con. Bất cư khi nào group node được hiển thị, tất cả các nút con của nó được hiển thị theo thứ tự. Mọi biến đổi, trạng thái hiệu ứng được áp dụng trên nhóm sẽ được áp dụng cho tất cả các nút con.
    - Region – đây là lớp cơ sở của tất cả các điều khiển giao diện người dùng dựa trên Node JavaFX. Chẳng hạn như biểu đồ, Ngăn và điều khiển.
    - Webview – node này quản lý công cụ web và hiển thị nội dung của nó.
  + Leaf Node – Node không có nút con được gọi là nút lá.

Bắt buộc phải chuyển nút gốc vào độ thị cảnh. Nếu nhóm được chuyển làm gốc, tất cả các nút sẽ được cắt bớt vào cảnh và bất kỳ thay đổi nào về kích thước của cảnh sẽ không ảnh hưởng đến bố cục của cảnh.

* **Tạo một ứng dụng JavaFX.**

Để tạo một ứng dụng JavaFX, bạn cần khởi tạo lớp Application và triển khai phương thức trừu tượng start() của nó. Trong phương thức này, chuáng ta sẽ viết mã cho ứng dụng JavaFX.

* **Lớp Application.**

Lớp Application của gói javafx.application là điểm đầu vào của ứng dụng trong JavaFX. Để tạo một ứng dụng JavaFX, bạn cần kế thừa lớp này và triển khai phương thức trừu tượng start() của nó. Trong phương thức này, bạn cần viết toàn bộ mã cho đồ họa JavaFX.

Trong phương thức main, bạn phải khởi chạy ứng dụng bằng phương thức launch(). Phương thức này gọi nội bộ phương thức start() của lớp Application như được hiển thị trong chương trình sau.

A screenshot of a computer

Description automatically generated

* **Chuẩn bị scence graph.**

Theo ứng dụng của bạn, bạn cần chuẩn bị một biểu đồ cảnh với các nút bắt buộc. Vì nút gốc là nút đầu tiên, bạn cần tạo một nút gốc. Là một nút gốc, bạn có thể chọn từ Nhóm, Vùng hoặc webview.

Group: một nút group được đại diện bởi lớp có tên nhóm thuộc gói javafx.scence, bạn có thể tạo nút nhóm bằng cách khởi tạo lớp như hình dưới dây.

A white background with black text

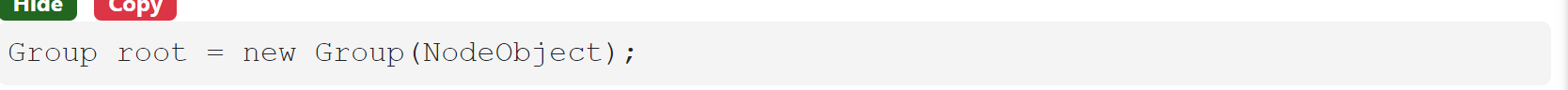
Description automatically generated

* + Phương thức getchildren() của lớp Group cung cấp cho bạn một đối tượng của lớp Observablelist chứa các nút. Chúng ta có thể lấy đối tượng này và thêm các nút vào nó như hình dưới đây.

A screenshot of a computer

Description automatically generated

* + Chúng ta cũng có thể thêm các đối tượng Node vào nhóm, chỉ bằng cách chuyển chúng đến lớp Nhóm và phương thức khởi tạo của nó tại thời điểm khởi tạo, như được hiển thị ở hình bên dưới.



* + Vùng – nó là lớp cơ sở của tất cả các điều khiển giao diện người dùng dựa trên Node JavaFX, chẳng hạn như:
    - Biểu đồ - lớp này là lớp cơ sở của tất cả các biểu đồ và nó thuộc về gói javafx.sence.chart.
    - Lớp này có hai lớp con, đó là – Piechart và XYChart. Hai lớp nafyu lần lượt có các lớp con như AreaChart, Barchart, bubbleChart, được sử dụng để vẽ các loại biểu đồ XY-PLANE khác nhau trong JavaFX.
    - Bạn có thể sử dụng các lớp này để nhúng biểu đồ vào ứng dụng của mình.
    - Pane – Pane là lớp cơ sở của tất cả các ngăn bố cục như AnchorPane, BorderPane, DialogPane, v.v Lớp này thuộc về một gói được gọi là javafx.scence.layout.
    - Bạn có thể sử dụng các lớp này để chèn các bố cục được xác định trước đó trong ứng dụng của mình.
    - Control – đây là lớp cơ sở của các điều khiển giao diện người dùng như Accordion, ButtonBar, ChoiceBõ, ComboBoxBase, HTMLEDITOR,.. c ác lớp này thuộc về gói javafx.scene.control.

Trong một nhóm bạn có thể khởi tạo bất kỳ lớp nào trong số cacacs lớp được đề cập ở tên và sử dụng chúng làm nút gốc, như được hiển thị trong chương trình sau.

A group of people in a white background

Description automatically generated with medium confidence

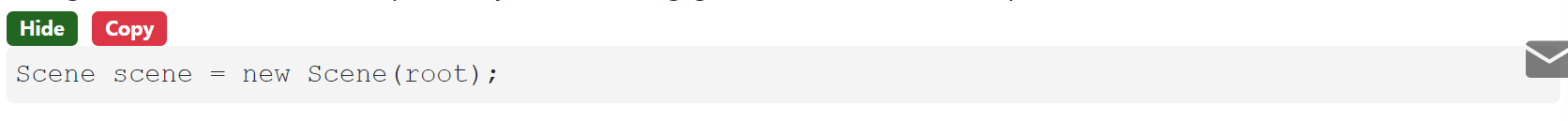
* + Sau đây là sơ đồ đại diện cho hệ thống phân cấp lớp nút của JavaFX.

A diagram of a company

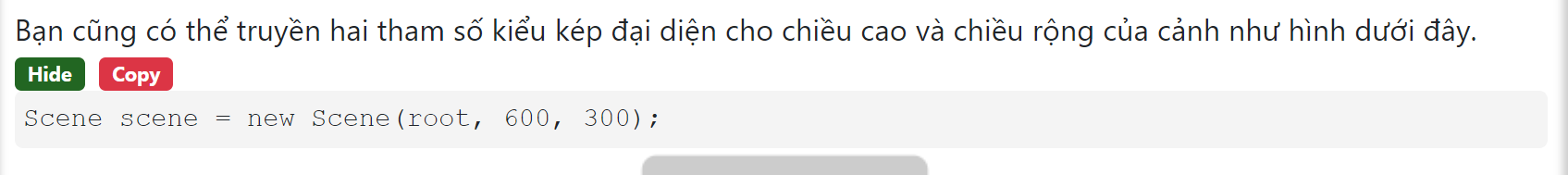
Description automatically generated

* **Chuẩn bị scene.**

Sử dụng gói javafx.scene để tạo đối tượng, khi khởi tạo đối tượng bắt buộc phải truyền nút gốc cho hàm khởi tạo scene.



Chúng ta cũng có thể kết hợp thêm cả kích thước cho scene.



* **Chuẩn bị stage.**

Đây là vùng chứa của bất kỳ ứng dụng JavaFX nào và nó cung cấp một cửa sổ cho ứng dụng. Nó được đại diện bởi lớp Stage của gói javafx.stage. Một đối tượng của lớp này được truyền dưới dạng tham số của phương thức start() của lớp Application.

Sử dụng đối tượng này, chúng ta có thể thực hiện các thao tác khác nhau trên sân khấu. Trước hết, bạn có thể thực hiuenej những điều sau:

* + Đặt tiêu đề cho vùng hiển thị bằng phương thức setTitle().
  + Đính kèm đối tượng cảnh vào vùng hiển thị bằng phương thức setScene().
  + Hiển thị nội dung của cảnh bằng phương thức show() như hình dưới đây.

A close-up of a computer screen

Description automatically generated

* **Vòng đời của ứng dụng JavaFX.**

Lớp Application JavaFX có ba phương thức vòng đời đó là:

* + Start() – phương thức điểm nhập nơi mã đồ họa JavaFX sẽ được viết.
  + Stop() – một phương thức trống có thể bị ghi đè, ở đây bạn có thể viết logic để dùng ứng dụng.
  + Init() – một phương thức trống có thể được ghi đè, nhưng bạn không thể tạo sân khấu hoặc cảnh trong phương thức này.

Ngoài ra, nó còn cung cấp một phương thức static có thể là launcher() để khởi chạy ứng dụng JavaFX.

Vì phương thức launch() là tĩnh nên chúng ta cần phải gọi nó từ một ngữ cảnh tĩnh (nói chung là trong hàm main()). Bất cứ khi nào một ứng dụng JavaFX được chạy, cách hành động sau sẽ thực hiện theo một thứ tự.

* + Một thể hiện của lớp application sẽ được tạo.
  + Phương thức init() được gọi.
  + Phương thức start() được gọi.
  + Trình khởi chạy đợi ứng dụng hoàn tất và gọi phương thức stop().
* **Chấm dứt ứng dụng JavaFX.**

Khi cửa sổ cuối cùng của ứng dụng bị đóng, ứng dụng JavaFX sẽ bị kết thúc hoàn toàn. Bạn có thể tắt hành vi này bằng cách chuyển giá trị Boolean “False” cho phương thức tĩnh setImplicitExit() (Nên được gọi từ ngữ cảnh tĩnh).

Bạn có thể kết thúc ứng dụng JavaFX một cách rõ ràng bằng các phương thức Platform.exit() hoặc System.exit(int).

1. **Ví dụ 1 – tạo một cửa số trống.**

Phần này hướng dẫn bạn cách tạo một ứng dụng mẫu JavaFX hiển thị một cửa sổ trống. Sau đây là các bước.

* Bước 1: tạo một lớp java và kế thwuaf lớp Appliacation của gói javafx.application và triển khai phương thức start() của lớp này như sau.

**A white background with colorful text

Description automatically generated**

* Bước 2: tạo đối tượng của group.

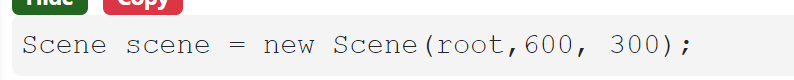
Trong phương thức start() tạo mối đối tượng nhóm bằng cách khởi tạo lớp có tên group, thuioocj về gói javafx.scene.

Group root = new Group();

* Bước 3: tạo đối tượng cảnh.

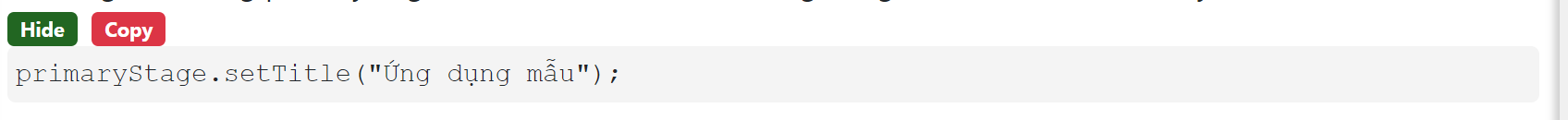
Tạo một cảnh bằng cahcs khởi tạo lớp có tên là Scên thuộc về gói javafx.scene. Đối với lớp này, hãy chuyển đối tượng Group(gốc) được tạo ở bước trước.

Ngoài đối tượng gốc bạn cũng có thể truyền hai tham số kép đại diện cho chiều cao và chiều rộng của màn hình cùng với đối tượng ủa lớp Group như sau:



* Bước 4: đặt tiêu đề của sân khấu.

Bạn cosd thể đặt tiêu đề cho vùng hiển thị bằng phương thức setTitle() của lớp Stage. PrimaryStage là đối tượng của lớp Stage được truyền cho phương thức khởi đầu của lớp cảnh, như một tham số.



* Bước 5: thêm scene vào trong stage.

Bạn có thể thêm một đối tượng scene vào sân khấu bằng cách sử dụng phương thức setScên() của lớp có tên stage. Thêm đối tượng scene đã chuẩn bị ở các bước trước bằng cách sử dụng phương pháp này.

A close-up of a logo

Description automatically generated

* Bước 6: hiển thị nội dụng của stage.

Hiển thị nội dụng của scene b ằng phương thức show() của lớp stage.

A close up of a website

Description automatically generated

* Bước 7: khởi chạy ứng dụng.

Khởi chạy ứng dụng javaFX bằng cách gọi phương thức tĩnh launch() của lớp Application từ phương thức main như sau.

A close-up of a computer screen

Description automatically generated

* **Ví dụ về một chương trình JavaFX đơn giản.**

A screenshot of a computer program

Description automatically generated

* **Ví dụ về chương trình tạo đường thằng trong javaFX.**

|  |
| --- |
| import javafx.application.Application;  import javafx.scene.Group;  import javafx.scene.Scene;  import javafx.scene.shape.Line;  import javafx.stage.Stage;  public class DrawingLine extends Application{  @Override  public void start(Stage stage) {  //Creating a line object  Line line = new Line();    //Setting the properties to a line  line.setStartX(100.0);  line.setStartY(150.0);  line.setEndX(500.0);  line.setEndY(150.0);    //Creating a Group  Group root = new Group(line);    //Creating a Scene  Scene scene = new Scene(root, 600, 300);    //Setting title to the scene  stage.setTitle("Sample application");    //Adding the scene to the stage  stage.setScene(scene);    //Displaying the contents of a scene  stage.show();  }  public static void main(String args[]){  launch(args);  }  } |

* **Ví dụ 3: hiển thị văn bản.**