Note\*: việc tạo một form là một hoạt động thường xuyên khi mà phát triển ứng dụng. Trong nội dung này sẽ dạy bạn một số thứ cơ bản về screen layout (bố trí màn hình), làm cách nào thểm controls vào một layout pane (khung bố trí) và tạo một sự kiện input.

(Trong nôi dung này chúng ta sẽ làm rõ làm sao để xây dựng chương trình này bằng JavaFX).

1. **Create a GridPane Layout.**

Cho một cái form login, sử dụng một GridPane layout bởi vì điều này cho phép chúng ta tạo ra một flexible grid với các rows và columns trong đó để bố trí các controls ben ngoài. Chúng ta có thể đặtcác controls trong bất kì các cell nào của grid và có thể điều khiển mở rộng các ô nếu cần.

* Sau đây là đoạn code tạo gridpane với gap và padding.

|  |
| --- |
| GridPane grid = new GridPane();  grid.setAlignment(Pos.CENTER);  grid.setHgap(10);  grid.setVgap(10);  grid.setPadding(new Insets(25, 25, 25, 25));  Scene scene = new Scene(grid, 300, 275);  primaryStage.setScene(scene); |

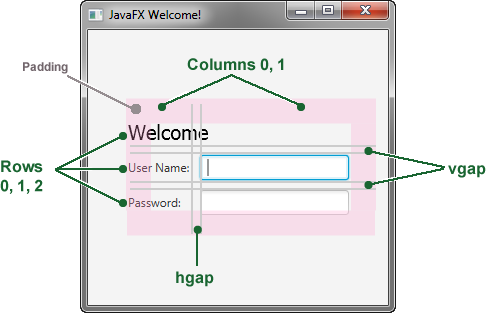
Trong đoạn code trên:

* + Tạo một đối tượng Gridpane và đặt tên cho nó là grid.
  + Thuộc tính căn chỉnh vị trí mặc định là từ đỉnh trên cùng bên trái tới vị trí ở giữa.
  + Quản lí khoảng cách giữa các ô theo chiều dọc và chiều ngang thì là giống nhau với 10px.
  + Thuộc tính Padding quản lý khoảng cách xung quanh các cạnh của grid pane.
  + Inset trong padding được sắp xếp theo thứ tự top, right, bottom, và left. (trong ví dụ thì là 25px cho mỗi side).
  + Scene được tạo ra với grid là node gốc, thứ mà được làm việc thường xuyên với các layout container. Một khi window bị điều chỉnh kích thước, các nodes cùng với grid pane cũng sẽ được điều chỉnh kích thước theo cái ràng buộc layout của chúng. Trong ví dụ này grid được duy trì ở giữa khi chúng ta tăng hoặc thu nhỏ kích thước của window. The padding properties chắc chắn rằng luôn có một cái padding xung quanh cái grid khi chúng ta làm cho window nhỏ hơn.
  + Code trên set kích thước width và height là 300 và 275. Nếu chúng ta không set giá trị cho scene thì kích thước mặc định của nó sẽ là nhỏ nhất sao cho phù hợp với nội dung bên trong.

1. **Thêm text, labels và textfield.**

|  |
| --- |
| Text scenetitle = new Text("Welcome");  scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));  grid.add(scenetitle, 0, 0, 2, 1);  Label userName = new Label("User Name:");  grid.add(userName, 0, 1);  TextField userTextField = new TextField();  grid.add(userTextField, 1, 1);  Label pw = new Label("Password:");  grid.add(pw, 0, 2);  PasswordField pwBox = new PasswordField();  grid.add(pwBox, 1, 2); |

* + Dòng đầy tiên tạo ra một đối tượng Text thứ mà không thể chỉnh sửa, set giá trị cho nó là “WELCOME”, và gán tên là scenetitle.
  + Dòng tiếp theo sử dụng phương thức setfont() để set Font-Family, weight, và size cho biến scenetitle.
  + Phương thức add() của đối tượng gridpane dùng để thêm scenetitle vào trong grid. Đối số thứ 2 và thứ 3 là cột và hàng bắt đầu của scenetitle. Đối số thứ 4 và thứ 5 là cột mở rộng ra thêm một khoảng là và hàng mở rộng ra.



1. **Thêm button và Text.**

|  |
| --- |
| Button btn = new Button("Sign in");  HBox hbBtn = new HBox(10);  hbBtn.setAlignment(Pos.BOTTOM\_RIGHT);  hbBtn.getChildren().add(btn);  grid.add(hbBtn, 1, 4); |

* Dòng đầu tiên tạo ra một đối tượng button có tên text là “Sign in”
* Dòng thứ hai tạo ra một Hbox layout pane có tên là hbBtn cùng với spacing là 10px. Spacing ở đây là khoảng cách chiều ngang giữa các con trong Hbox. Hbox là một layout và bố trị các control theo chiều ngang.

1. **Thêm code để xử lí sự kiện.**

Tạo một lớp để xử lí sự kiện.

|  |
| --- |
| class a implements EventHandler<ActionEvent> // têm của sự kiện cần xử lí  {  @Override  public void handle(ActionEvent event) {  System.out.println("nguyentandung"); // xử lí sự kiện.  }    } |

Thêm sự kiện vào bên trong btn.

|  |
| --- |
| btn.setOnAction(obj); |