

17.4 JavaFX 컨테이너

- 레이아웃을 작성할 때 다양한 컨트롤들을 쉽게 배치할 수 있도록 해주는 역할을 한다.
- `javafx.scene.layout` 패키지에는 다양한 컨테이너 클래스들이 존재한다.
- XXXPane은 모두 컨테이너라고 보면 되고, 그 이외에 Hbox, VBox가 있다.

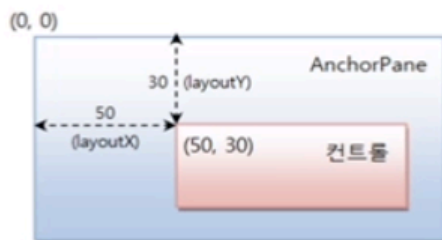
컨테이너	설명
AnchorPane	컨트롤을 좌표를 이용해서 배치하는 레이아웃
BorderPane	위, 아래, 오른쪽, 왼쪽, 중앙에 컨트롤을 배치하는 레이아웃
FlowPane	행으로 배치하되 공간이 부족하면 새로운 행에 배치하는 레이아웃
GridPane	그리드로 배치하되 셀의 크기가 고정적이지 않는 레이아웃
StackPane	컨트롤을 겹쳐 배치하는 레이아웃
TilePane	그리드로 배치하되 고정된 셀의 크기를 갖는 레이아웃
HBox	수평으로 배치하는 레이아웃
VBox	수직으로 배치하는 레이아웃

17.4.1 AnchorPane 컨테이너

// AnchorPane 컨테이너를 사용할 때는 SceneBuilder를 사용하는게 좋다

AnchorPane 컨테이너는 AnchorPane 의 좌상단(0, 0)을 기준으로 컨트롤을 좌표로 배치한다.

컨트롤 좌표는 좌상단 (layoutX, layoutY) 값을 말하는데 (0,0) 에서 떨어진 거리이다.



AnchorPane 에서 사용할 수 있는 주요 설정은 다음과 같다.

태그 및 속성	설명	적용
<code>prefWidth</code>	폭을 설정	AnchorPane
<code>prefHeight</code>	높이를 설정	AnchorPane
<code>layoutX</code>	컨트롤의 X 좌표	컨트롤
<code>layoutY</code>	컨트롤의 Y 좌표	컨트롤
<code><children></code>	컨트롤을 포함	AnchorPane

Layout

Label 아이디

TextField

Label 패스워드

PasswordField

Button 로그인

Button 취소

AppMain

아이디

패스워드

로그인

취소

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.layout.*?>
4 <?import javafx.scene.control.*?>
5
6 <AnchorPane xmlns:fx="http://javafx.com/fxml" prefHeight="150.0" prefWidth="300.0" >
7   <children>
8     <Label layoutX="42.0" layoutY="28.0" text="아이디" />
9     <Label layoutX="42.0" layoutY="66.0" text="패스워드" />
10    <TextField layoutX="120.0" layoutY="24.0" />
11    <PasswordField layoutX="120.0" layoutY="62.0" />
12    <Button layoutX="97.0" layoutY="106.0" text="로그인" />
13    <Button layoutX="164.0" layoutY="106.0" text="취소" />
14  </children>
15 </AnchorPane>

```

```

1 package sec04.exam01_anchorpane;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class AppMain extends Application {
10     @Override
11     public void start(Stage primaryStage) throws Exception {
12         Parent root = FXMLLoader.load(getClass().getResource("root.fxml"));
13         Scene scene = new Scene(root);
14
15         primaryStage.setTitle("AppMain");
16         primaryStage.setScene(scene);
17         primaryStage.show();
18         primaryStage.setResizable(false);
19     }
20
21     public static void main(String[] args) {
22         launch(args);
23     }
24 }

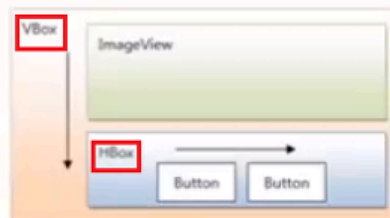
```

Colored by Color Scripter CS

17.4.2 HBox와 VBox 컨테이너

- 수평과 수직으로 컨트롤을 배치하는 컨테이너
- HBox는 컨트롤의 높이 확장하고, 컨트롤의 폭은 유지
- VBox는 컨트롤의 폭 확장하고 컨트롤의 높이는 유지

HBox와 VBox에서 사용할 수 있는 주요 설정은 다음과 같다.



태그 및 속성	설명	적용
prefWidth	폭을 설정	HBox, VBox
prefHeight	높이를 설정	HBox, VBox
alignment	컨트롤의 정렬을 설정	HBox, VBox
spacing	컨트롤의 간격을 설정	HBox, VBox
fillWidth	컨트롤의 폭 확장 여부 설정	VBox
fillHeight	컨트롤의 높이 확장 여부 설정	HBox
<children>	컨트롤을 포함	HBox, VBox
<HBox.hgrow> <Priority fxconstant="ALWAYS"/> </HBox.hgrow>	HBox의 남은 폭을 채움	컨트롤
<VBox.vgrow> <Priority fxconstant="ALWAYS"/> </VBox.vgrow>	VBox의 남은 높이를 채움	컨트롤



```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.*?>
4 <?import javafx.scene.image.*?>
5 <?import javafx.scene.layout.*?>
6 <?import javafx.scene.control.*?>
7 <?import java.lang.*?>
8

```

CS

```

9 <VBox xmlns:fx="http://javafx.com/fxml">
10 <padding>
11 <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
12 </padding>
13
14 <children>
15 <ImageView fitWidth="200" preserveRatio="true"> // 그림의 비율에 맞게 높이를 설정
16 <image>
17 <Image url="@images/javafx.jpg" /> // 현재 경로 기준으로 상대경로로 파일 지정
18 </image>
19 </ImageView>
20
21 <HBox alignment="CENTER" spacing="20.0">
22 <children>
23 <Button text="이전" />
24 <Button text="다음"> // 오른쪽 남은 공간을 버튼이 모두 채우도록 설정
25 <HBox.hgrow><Priority fx:constant="ALWAYS"/></HBox.hgrow>
26 <maxWidth><Double fx:constant="MAX_VALUE"/></maxWidth>
27 </Button> // 버튼의 폭을 자동으로 확장하기 위해 설정
28 </children>
29 <VBox.margin>
30 <Insets top="10.0" />
31 </VBox.margin>
32 </HBox>
33 </children>
34 </VBox>

```

Colored by Color Scripter

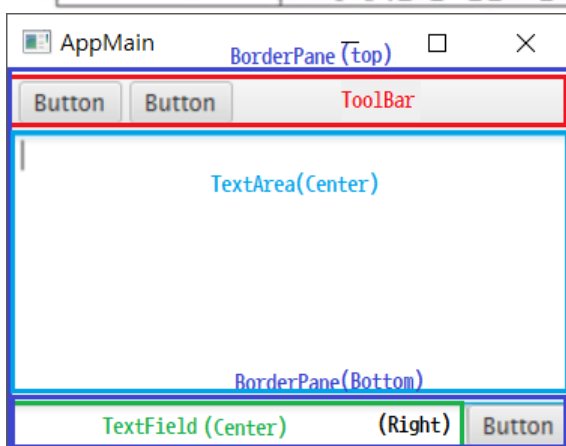
17.4.3 BorderPane 컨테이너

- top, bottom, left, right, center 셀에 컨트롤을 배치하는 컨테이너
- 각 셀에는 하나의 컨트롤 또는 컨테이너만 배치
- top, bottom, left, right에 배치하지 않으면 center에 배치된 컨트롤이 확장



다음은 BorderPane 에서 사용할 수 있는 태그 및 속성들이다.

태그 및 속성	설명	적용
prefWidth	폭을 설정	BorderPane
prefHeight	높이를 설정	BorderPane
<top>	top 에 배치될 컨트롤을 포함	BorderPane
<bottom>	bottom 에 배치될 컨트롤을 포함	BorderPane
<right>	right 에 배치될 컨트롤을 포함	BorderPane
<left>	left 에 배치될 컨트롤을 포함	BorderPane



```

1 <?xml version="1.0" encoding="UTF-8"?>
2

```

```

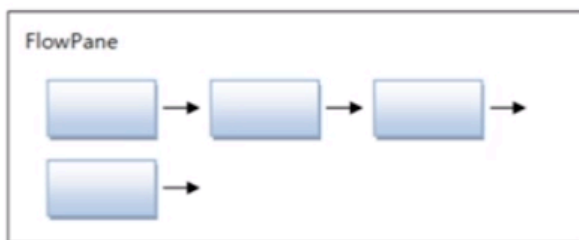
3 <?import javafx.scene.layout.*?>
4 <?import javafx.scene.control.*?>
5
6 <BorderPane xmlns:fx="http://javafx.com/fxml" prefHeight="200.0" prefWidth="300.0" >
7   <top>
8     <ToolBar>
9       <items>
10        <Button text="Button" />
11        <Button text="Button" />
12      </items>
13    </ToolBar>
14  </top>
15
16  <center>
17    <TextArea/>
18  </center>
19
20  <bottom>
21    <BorderPane>
22      <center>
23        <TextField/>
24      </center>
25      <right>
26        <Button text="Button"/>
27      </right>
28    </BorderPane>
29  </bottom>
30 </BorderPane>

```

Colored by Color Scripter

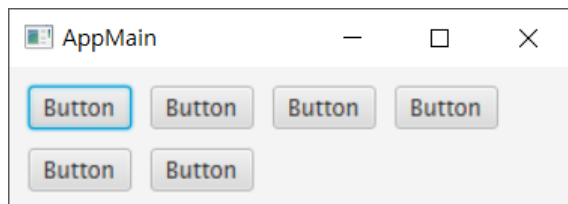
17.4.4 FlowPane 컨테이너

➤ 행으로 컨트롤을 배치하되 공간이 부족하면 새로운 행에 배치하는 컨테이너



다음은 FlowPane 에서 사용할 수 있는 태그와 속성들이다.

태그 및 속성	설명	적용
prefWidth	폭을 설정	FlowPane
prefHeight	높이를 설정	FlowPane
hgap	컨트롤의 수평 간격을 설정	FlowPane
vgap	컨트롤의 수직 간격을 설정	FlowPane
<children>	컨트롤을 포함	FlowPane



```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.layout.*?>
4 <?import javafx.geometry.*?>
5 <?import javafx.scene.control.*?>
6
7 <FlowPane xmlns:fx="http://javafx.com/fxml"
8   prefWidth="300.0" prefHeight="70.0" hgap="10.0" vgap="10.0" >
9   <padding>                                //수평 간격 //수직간격
10    <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
11  </padding>

```

CS

```

12
13 <children>
14   <Button text="Button" />
15   <Button text="Button" />
16   <Button text="Button" />
17   <Button text="Button" />
18   <Button text="Button" />
19   <Button text="Button" />
20 </children>
21 </FlowPane>

```

Colored by Color Scriptor

17.4.5 TilePane 컨테이너

- 그리드로 컨트롤을 배치하되 고정된 셀(타일) 크기를 갖는 컨테이너
- 오른쪽에 컨트롤을 배치할 공간이 부족하면 새로운 행에 컨트롤을 배치



다음은 TilePane 에서 사용할 수 있는 태그와 속성들이다.

태그 및 속성	설명	적용
prefWidth	폭을 설정	TilePane
prefHeight	높이를 설정	TilePane
prefTileWidth	타일의 폭을 설정	TilePane
prefTileHeight	타일의 높이를 설정	TilePane
<children>	컨트롤을 포함	TilePane



```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.layout.*?>
4 <?import javafx.scene.image.*?>
5
6 <TilePane xmlns:fx="http://javafx.com/fxml" prefTileHeight="100" prefTileWidth="100" >
7   <children>
8     <ImageView>
9       <image><Image url="@images/fruit1.jpg" /></image>
10    </ImageView>
11    <ImageView>
12      <image><Image url="@images/fruit2.jpg" /></image>
13    </ImageView>
14    <ImageView>
15      <image><Image url="@images/fruit3.jpg" /></image>
16    </ImageView>
17    <ImageView>
18      <image><Image url="@images/fruit4.jpg" /></image>
19    </ImageView>
20    <ImageView>
21      <image><Image url="@images/fruit5.jpg" /></image>
22    </ImageView>
23  </children>
24 </TilePane>

```

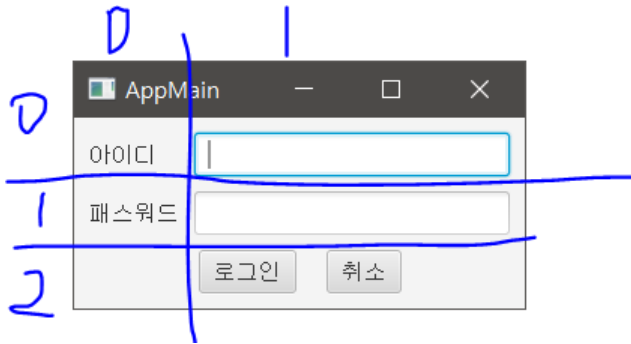
Colored by Color Scriptor CS

17.4.6. GridPane 컨테이너

- 그리드로 컨트롤을 배치하되 셀의 크기가 고정적이지 않고 유동적인 컨테이너
- 셀 병합이 가능하기 때문에 다양한 입력폼 화면을 만들 때 매우 유용



태그 및 속성	설명	적용
prefWidth	폭을 설정	GridPane
prefHeight	높이를 설정	GridPane
hgap	수평 컨트롤 간격을 설정	GridPane
vgap	수직 컨트롤 간격을 설정	GridPane
<children>	컨트롤을 포함	GridPane
GridPane.rowIndex	컨트롤이 위치하는 행 인덱스를 설정	컨트롤
GridPane.columnIndex	컨트롤이 위치하는 컬럼 인덱스를 설정	컨트롤
GridPane.rowSpan	행 병합 수를 설정	컨트롤
GridPane.columnSpan	컬럼 병합 수를 설정	컨트롤
GridPane.hgrow	수평 빈 공간 채우기를 설정	컨트롤
GridPane.vgrow	수직 빈 공간 채우기를 설정	컨트롤
GridPane.halignment	컨트롤의 수평 정렬을 설정	컨트롤
GridPane.valignment	컨트롤의 수직 정렬을 설정	컨트롤



```

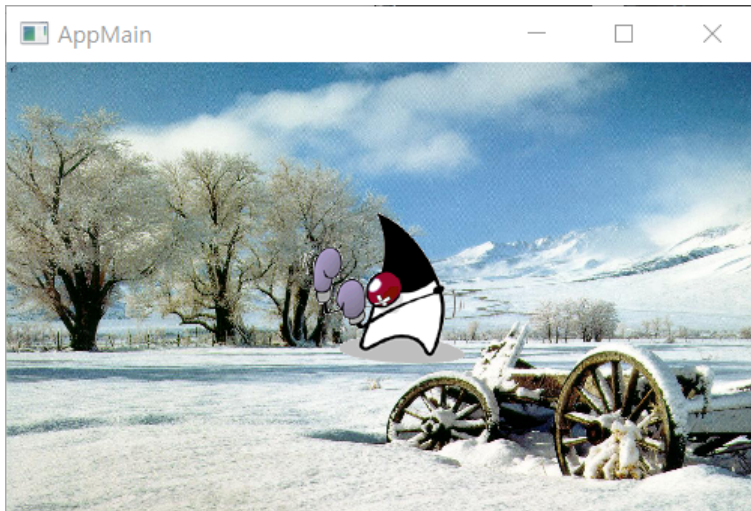
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.layout.*?>
4  <?import javafx.geometry.*?>
5  <?import javafx.scene.control.*?>
6
7  <GridPane xmlns:fx="http://javafx.com/fxml"
8      prefWidth="300.0" hgap="10.0" vgap="10.0" >
9      <padding>
10         <Insets topRightBottomLeft="10.0"/>
11     </padding>
12     <children>
13         <Label text="아이디" GridPane.rowIndex="0" GridPane.columnIndex="0"/>
14         <TextField GridPane.rowIndex="0" GridPane.columnIndex="1"
15             GridPane.hgrow="ALWAYS" /> // 오른쪽 빈공간까지 확장
16
17         <Label text="패스워드" GridPane.rowIndex="1" GridPane.columnIndex="0"/>
18         <TextField GridPane.columnIndex="1" GridPane.rowIndex="1"
19             GridPane.hgrow="ALWAYS" />
20
21         <HBox GridPane.rowIndex="2" GridPane.columnIndex="0"
22             GridPane.columnSpan="2" GridPane.hgrow="ALWAYS" // GridPane.columnSpan은 컬럼 2개 병합
23             alignment="CENTER" spacing="20.0" >
24             <children>
25                 <Button text="로그인" />
26                 <Button text="취소" />
27             </children>
28         </HBox>
29     </children>
30 </GridPane>
31

```

Colored by Color Scripter cs

❖ StackPane 컨테이너

- 컨트롤을 겹쳐 배치하는 컨테이너
- 위에 있는 컨트롤이 투명이라면 밑에 있는 컨트롤이 겹쳐 보임



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.layout.*?>
4 <?import javafx.scene.image.*?>
5
6 <StackPane xmlns:fx="http://javafx.com/fxml">
7   <children>
8     <ImageView fitWidth="500" fitHeight="300" > // 가로비와 세로비 상관없이 고정 길이로 설정, 먼저 배치된것이 밑에
9       <image>
10        <Image url="@images/snow.jpg" />
11      </image>
12    </ImageView>
13    <ImageView preserveRatio="true"> // 가로비와 세로비를 유지
14      <image>
15        <Image url="@images/duke.gif" />
16      </image>
17    </ImageView>
18  </children>
19 </StackPane>
```

Colored by Color Scriptor CS