

> AWT(Abstract Window Toolkit)

- 운영 체제가 제공하는 네이티브 UI 컴포넌트를 이용
- 운영 체제에 따라 UI의 모양이 서로 달랐고, 종류도 제한적

> Swing

- 모든 운영체제상에서 동일한 UI를 갖도록 자신만의 UI(look and feel)을 가짐
- 사용자는 애니메이션이 추가된 더 이쁘진 운영 체제의 네이티브 UI를 더 선호
 - ✓ 네이티브 UI로 보여지도록 자신의 UI를 다시 페이팅
 - ✓ 실행 성능이 느려지고, 메모리를 더 많이 사용

> JavaFX

- 데스크탑 뿐만 아니라 임베디드까지 적용할 수 있도록 가볍고 풍부한 UI 제공
- 레이아웃, 스타일, 애플리케이션 로직을 분리 개발
- 자바7 업데이트6버전부터 JavaFX2.2를 JDK와 JRE에 포함



17.2 JavaFX 애플리케이션 개발 시작

// JavaFX 애플리케이션을 개발하려면 제일 먼저 메인 클래스를 작성해야 한다.
// 메인 클래스는 JavaFX 애플리케이션을 시작하는 관문이다

17.2.1 메인 클래스

// JavaFX 애플리케이션을 시작시키는 메인 클래스는 추상 클래스인 Application을 상속받고
// Application에 있는 start() 메소드를 재정의해야 한다
// start() 메소드는 UI(User Interface : 사람과 사물간의 상호작용)를 생성하고 윈도우를 생성하는 메소드이다

// start() 메소드는 코드에서 직접 호출할 수는 없고, 메인클래스에서 Application이 가지고 있는 메소드인 launch를 호출함으로써, 내부적으로 start() 메소드를 호출하게 해야 한다

// JavaFX는 window를 Stage라는 용어로 사용한다

// 매개변수 값으로 제공되는 primaryStage를 show 함으로써 윈도우를 보여준다

[AppMain.java] 메인 클래스

```
1 public class AppMain extends Application {  
2     @Override  
3     public void start(Stage primaryStage) throws Exception {  
4         primaryStage.show(); //윈도우 보여주기  
5     }  
6  
7     public static void main(String[] args) {  
8         launch(args); //AppMain 객체 생성 및 메인 윈도우 생성  
9     }  
10 }
```



```
1 package sec02.exam01_application_start;  
2  
3 import javafx.application.Application;  
4 import javafx.stage.Stage;  
5  
6 public class AppMain extends Application { // Application 클래스는 추상 클래스이므로  
7     @Override  
8     public void start(Stage primaryStage) throws Exception {  
9         primaryStage.show(); //윈도우 보여주기  
10    }  
11  
12    public static void main(String[] args) {
```

CS

```

13 | launch(args); //AppMain 객체 생성 및 메인 윈도우 생성
14 | }
15 | }

```

Colored by Color Scripter

17.2.2 JavaFX 라이프사이클

// launch() 메소드가 호출이되면 내부적으로 두 개의 쓰레드가 호출이되는데
 // JavaFX Application Thread와 JavaFX Launcher이다
 // JavaFX Application Thread는 UI를 생성하고 변경을 하는 쓰레드고,
 // JavaFX Launcher는 init()만 실행해 주는 쓰레드이다



```

1 | package sec02.exam02_application_lifecycle;
2 |
3 | import javafx.application.Application;
4 | import javafx.stage.Stage;
5 |
6 | public class AppMain extends Application {
7 |     public AppMain() { // 기본 생성자를 이용해서 객체 생성함
8 |         System.out.println(Thread.currentThread().getName()+" : AppMain() 호출");
9 |     }
10 |
11 |     @Override
12 |     public void init() throws Exception {
13 |         System.out.println(Thread.currentThread().getName()+" : init() 호출");
14 |     } // init()(JavaFX Launcher)안에는 UI생성 코드를 넣으면 안된다,
15 |         // JavaFX Application Thread에만 넣는다
16 |     @Override
17 |     public void start(Stage primaryStage) throws Exception {
18 |         System.out.println(Thread.currentThread().getName()+" : start() 호출");
19 |         primaryStage.show();
20 |     }
21 |
22 |     @Override
23 |     public void stop() throws Exception {
24 |         System.out.println(Thread.currentThread().getName()+" : stop() 호출");
25 |     }
26 |
27 |     public static void main(String[] args) {
28 |         System.out.println(Thread.currentThread().getName()+" : main() 호출");
29 |         launch(args);
30 |     }
31 | }

```

Colored by Color Scripter CS

17.2.3 메인 클래스 실행 매개값 얻기

```
java AppMain --ip=192.168.0.5 --port=50001
```

➤ init() 메소드에서 다음 두가지 방법으로 매개값을 얻을 수 있다.

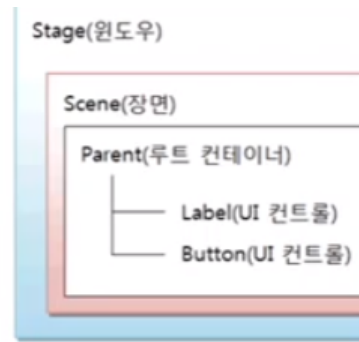
```

Parameters params = getParameters();
List<String> list = params.getRaw();           //①
Map<String, String> map = params.getNamed();   //②

```

- 무대(Stage)는 하나의 장면을 가질 수 있다.
- 장면은 `javafx.scene.Scene` 으로 표현한다.

```
Scene scene = new Scene(Parent root);
primaryStage.setScene(scene);
primaryStage.show();
```



17.2.4 무대(Stage)와 장면(Scene)

```

1 package sec02.exam03_stage_scene;
2
3 import javafx.application.Application;
4 import javafx.application.Platform;
5 import javafx.geometry.Pos;
6 import javafx.scene.Scene;
7 import javafx.scene.control.Button;
8 import javafx.scene.control.Label;
9 import javafx.scene.layout.VBox;
10 import javafx.scene.text.Font;
11 import javafx.stage.Stage;
12
13 public class AppMain extends Application {
14     @Override
15     public void start(Stage primaryStage) throws Exception {
16         VBox root = new VBox(); //Parent 하위 객체인 VBox 생성
17         root.setPrefWidth(350); //VBox의 폭 설정
18         root.setPrefHeight(150); //VBox의 높이 설정
19         root.setAlignment(Pos.CENTER); //컨트롤을 중앙으로 정렬
20         root.setSpacing(20); //컨트롤의 수직 간격
21
22         Label label = new Label(); //Label 컨트롤 생성
23         label.setText("Hello, JavaFX"); //text 속성 설정
24         label.setFont(new Font(50)); //font 속성 설정
25
26         Button button = new Button(); //Button 컨트롤 생성
27         button.setText("확인"); //text 속성 설정
28         button.setOnAction(event -> Platform.exit()); //클릭 이벤트 처리
29
30         root.getChildren().add(label); //VBox에 Label 컨트롤 추가
31         root.getChildren().add(button); //VBox에 Button 컨트롤 추가
32
33         Scene scene = new Scene(root); //VBox를 루트 컨테이너로 해서 Scene 생성
34
35         primaryStage.setTitle("AppMain"); //윈도우의 제목 설정
36         primaryStage.setScene(scene); //윈도우에 장면 설정
37         primaryStage.show(); //윈도우 보여주기
38     }
39
40     public static void main(String[] args) {
41         launch(args);
42     }
43 }

```

Colored by Color Scripter CS

