
Inleiding tot JavaFX

Stage

Als je een applicatie ontwikkelt, dan zal die meestal uit meerdere schermen bestaan. Zo'n scherm noemen we in JavaFX een stage (type Stage).

Het hoofdscherm waarin je terechtkomt als je de applicatie opstart, noemen we de primary stage (parameternaam primaryStage).

Het hoofdscherm (de primary stage) programmeren we in de klasse HelloApplication, voor alle andere schermen (stages) maken we afzonderlijke klassen.

Laten we ons hoofdscherm (primary stage) maken.

```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage primaryStage) throws IOException {  
        primaryStage.setTitle("Hallo!");  
        primaryStage.show();  
    }  
  
    public static void main(String[] args) {  
        launch();  
    }  
}
```

Scene

Een scherm (stage) bevat een venster (in JavaFX een scene genoemd, met een in te stellen breedte en hoogte).

Lay-outcontainer

Aan zo'n venster (scene) dienen we een lay-outcontainer te koppelen.

De lay-outcontainer is de ruimte (de container) waarop we visuele componenten (in JavaFX controls genoemd) zullen kunnen plaatsen. De lay-outcontainer regelt de plaatsing van de controls.

Er zijn verschillende types van lay-outcontainers, o.a.

- VBox,

- HBox,
- GridPane.

Op een VBox (vertical box) worden de controls automatisch onder elkaar geplaatst.

Op een HBox (horizontal box) worden de controls automatisch naast elkaar geplaatst.

Een GridPane werkt als een raster, waarbij voor elke visuele component (control) de plaats (verticaal en horizontaal) dient opgegeven te worden.

Laten we een venster met VBox maken.

```
public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)

        root = new VBox();
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
        primaryStage.setTitle("Hallo!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

We kunnen voor het venster (de scene) ook de kleur instellen, door gebruik te maken van de methode setFill.

Voorbeeld:

```
public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)

        root = new VBox();
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
        scene.setFill(Color.PURPLE);
        primaryStage.setTitle("Hallo!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

Instellen van de VBox

De gewenste spatiëring (ruimte tussen de controls) kan ingesteld worden bij instantiëring.

Voorbeeld: `root = new VBox(5);`

Deze spatiëring kan steeds gewijzigd worden met de methode `setSpacing`.

Voorbeeld: `root.setSpacing(25);`

De padding (de ruimte tussen de rand en de controls) kan ingesteld worden met de methode `setPadding`.

Voorbeeld: `root.setPadding(new Insets(30));`

Om de achtergrond van de VBox in te stellen, gebruik je `setBackground`.

Voorbeeld:

```
root.setBackground(new Background(new BackgroundFill  
    (Color.GREY, new CornerRadii(50), new Insets(20))));
```

Om de uitlijning (alignering) van de visuele componenten (controls genoemd in javafx) op de VBox in te stellen, gebruik je `setAlignment`.

Voorbeeld: `root.setAlignment(Pos.CENTER);`

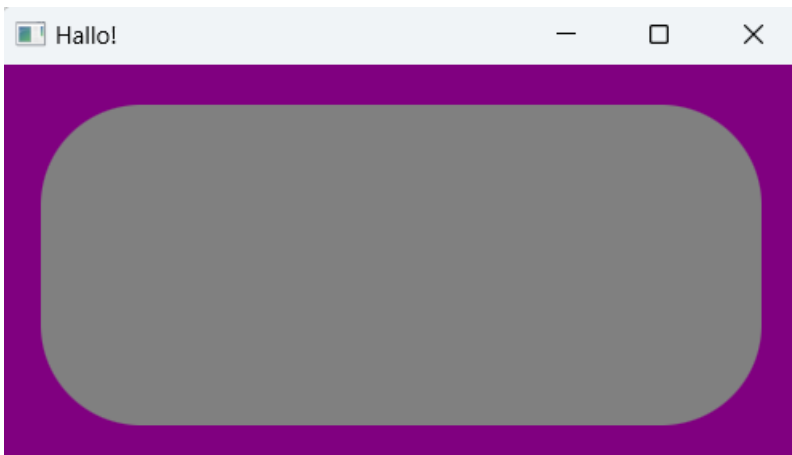
Ons programma tot nu toe ziet er dan als volgt uit:

```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage primaryStage) throws IOException {  
        VBox root; //lay-outcontainer  
        Scene scene; //Venster (scene)  
  
        root = new VBox(5);  
        root.setBackground(new Background(new BackgroundFill  
            (Color.GREY, new CornerRadii(50), new Insets(20))));  
        root.setPadding(new Insets(20));  
        root.setAlignment(Pos.CENTER);  
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200  
        scene.setFill(Color.PURPLE);  
        primaryStage.setTitle("Hallo!");  
    }  
}
```

```
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch();
}
```

Het resultaat van onze code:



Controls: children van de VBox

We zullen in wat volgt heel wat controls (visuele componenten) op de VBox plaatsen. Het is daarbij belangrijk om alle controls op te nemen in de lijst van children van de VBox.

Dat doen we als volgt:

```
root.getChildren().addAll(lijst van controls);
```

Controls: Label

In het onderstaande voorbeeld plaatsen we een label (type Label) op de VBox, en stellen we hiervoor de fontgrootte en de tekstkleur in.

```
public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)
        Label label;

        //*****Lay-outcontainer (VBox)*****
        root = new VBox(5);
        root.setBackground(new Background(new BackgroundFill
            (Color.GREY, new CornerRadii(50), new Insets(20))));
        root.setPadding(new Insets(20));
        root.setAlignment(Pos.CENTER);
        //*****Controls (de visuele elementen)*****
        label = new Label("Dit is een tekst.");
        label.setFont(new Font(30));
        label.setTextFill(Color.GREEN);
        //*****De controls op de VBox plaatsen*****
        root.getChildren().addAll(label);
        //*****Scene*****
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
        scene.setFill(Color.PURPLE);
        primaryStage.setTitle("Hallo!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

Controls: Button (knop)

In het onderstaande voorbeeld wordt een knop (type Button) op de VBox geplaatst, en wordt hiervoor het volgende ingesteld:

- de grootte,
- de achtergrond,
- de tekst.

```
public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)
        Label label;
        Button knop;

        //*****Lay-outcontainer (VBox)*****
        root = new VBox(5);
        root.setBackground(new Background(new BackgroundFill
            (Color.GREY, new CornerRadii(50), new Insets(20))));
        root.setPadding(new Insets(20));
        root.setAlignment(Pos.CENTER);
        //*****Controls (de visuele elementen)*****
        label = new Label("Dit is een tekst.");
        label.setFont(new Font(30));
        label.setTextFill(Color.GREEN);
        knop = new Button("OK");
        knop.setMinWidth(50);
        knop.setMinHeight(30);
        knop.setBackground(new Background(new BackgroundFill
            (Color.RED, new CornerRadii(5), new Insets(5))));
        knop.setTextFill(Color.WHITE);
        //*****De controls op de VBox plaatsen*****
        root.getChildren().addAll(label, knop);
        //*****Scene*****
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
        scene.setFill(Color.PURPLE);
        primaryStage.setTitle("Hallo!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

Event-handling (het afhandelen van een gebeurtenis)

Als op de knop geklikt wordt, willen we dat er code wordt uitgevoerd. Het feit dat op de knop geklikt wordt, is een gebeurtenis (in de ICT-wereld een event genoemd). We kunnen nu instellen wat er moet gebeuren als dit event (een muisklik) zich voordoet. Dit noemen we event-handling.

Voorbeeld:

```
public void start(Stage primaryStage) throws IOException {
    VBox root; //lay-outcontainer
    Scene scene; //Venster (scene)
    Label label;
    Button knop;

    //*****Lay-outcontainer (VBox)*****
    root = new VBox(5);
    root.setBackground(new Background(new BackgroundFill
        (Color.GREY, new CornerRadii(50), new Insets(20))));
    root.setPadding(new Insets(20));
    root.setAlignment(Pos.CENTER);
    //*****Controls (de visuele elementen)*****
    label = new Label("Dit is een tekst.");
    label.setFont(new Font(30));
    label.setTextFill(Color.GREEN);
    knop = new Button("OK");
    knop.setMinWidth(50);
    knop.setMinHeight(30);
    knop.setBackground(new Background(new BackgroundFill
        (Color.RED, new CornerRadii(5), new Insets(5))));
    knop.setTextFill(Color.WHITE);
    //*****Event-handling*****
    knop.setOnMouseClicked(new EventHandler<MouseEvent>() {
        @Override
        public void handle(MouseEvent mouseEvent) {
            label.setText("Er werd op de knop geklikt.");
            label.setTextFill(Color.PAPAYAWHIP);
        }
    });
    //*****De controls op de VBox plaatsen*****
    root.getChildren().addAll(label, knop);
    //*****Scene*****
    scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
    scene.setFill(Color.PURPLE);
    primaryStage.setTitle("Hallo!");
    primaryStage.setScene(scene);
    primaryStage.show();
}

public static void main(String[] args) {
    launch();
}
```

Event-handling bij een hyperlink

In het onderstaande voorbeeld plaatsen we een hyperlink op de VBox, en voorzien we via event-handling dat bij een klik op de hyperlink de betrokken webpagina in de webbrowser geopend wordt.

```
public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)
        Label label;
        Button knop;
        Hyperlink hyperlink;

        //*****Lay-outcontainer (VBox)*****
        root = new VBox(5);
        root.setBackground(new Background(new BackgroundFill
            (Color.GREY, new CornerRadii(50), new Insets(20))));
        root.setPadding(new Insets(20));
        root.setAlignment(Pos.CENTER);
        //*****Controls (de visuele elementen)*****
        label = new Label("Dit is een tekst.");
        label.setFont(new Font(30));
        label.setTextFill(Color.GREEN);
        knop = new Button("OK");
        knop.setMinWidth(50);
        knop.setMinHeight(30);
        knop.setBackground(new Background(new BackgroundFill
            (Color.RED, new CornerRadii(5), new Insets(5))));
        knop.setTextFill(Color.WHITE);
        hyperlink = new Hyperlink("www.bazandpoort.be");
        hyperlink.setTextFill(Color.YELLOW);
        //*****Event-handling*****
        knop.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent mouseEvent) {
                label.setText("Er werd op de knop geklikt.");
                label.setTextFill(Color.PAPAYAWHIP);
            }
        });
        hyperlink.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent mouseEvent) {
                getHostServices().showDocument("http://www.bazandpoort.be");
            }
        });
        //*****De controls op de VBox plaatsen*****
        root.getChildren().addAll(label, knop, hyperlink);
        //*****Scene*****
        scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
        scene.setFill(Color.PURPLE);
        primaryStage.setTitle("Hallo!");
        primaryStage.setScene(scene);
    }
}
```



```

    primaryStage.show();
}

public static void main(String[] args) {
    launch();
}
}

```

Binding

Binding houdt in dat we eigenschappen van verschillende visuele componenten (controls) aan elkaar kunnen linken. In JavaFX is deze binding mogelijk gemaakt via methodes met *Property* in de naam.

Zo kunnen we bijvoorbeeld de spatiëring (spacing) van de VBox (de ruimte tussen de controls) linken aan de hoogte:

```
root.spacingProperty().bind(root.heightProperty().divide(10));
```

Cirkel met binding

Nog een voorbeeld van binding: een cirkel waarvan de straal gelinkt is aan de breedte van de VBox.

```

public class HelloApplication extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        VBox root; //lay-outcontainer
        Scene scene; //Venster (scene)
        Label label;
        Button knop;
        Hyperlink hyperlink;
        Circle cirkel;

        //*****Lay-outcontainer (VBox)*****
        root = new VBox(5);
        root.setBackground(new Background(new BackgroundFill
            (Color.GREY, new CornerRadii(50), new Insets(20))));
        root.setPadding(new Insets(20));
        root.setAlignment(Pos.CENTER);
        //*****Controls (de visuele elementen)*****
        label = new Label("Dit is een tekst.");
        label.setFont(new Font(30));
        label.setTextFill(Color.GREEN);
        knop = new Button("OK");
        knop.setMinWidth(50);
        knop.setMinHeight(30);
        knop.setBackground(new Background(new BackgroundFill
            (Color.RED, new CornerRadii(5), new Insets(5))));
        knop.setTextFill(Color.WHITE);
        hyperlink = new Hyperlink("www.bazandpoort.be");
        hyperlink.setTextFill(Color.YELLOW);
    }
}

```

```

cirkel = new Circle();
cirkel.radiusProperty().bind(root.widthProperty().divide(15));
//*****Event-handling*****
knop.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        label.setText("Er werd op de knop geklikt.");
        label.setTextFill(Color.PAPAYAWHIP);
    }
});
hyperlink.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        getHostServices().showDocument("http://www.bazandpoort.be");
    }
});
//*****De controls op de VBox plaatsen*****
root.getChildren().addAll(label, knop, hyperlink, cirkel);
//*****Scene*****
scene = new Scene(root, 400, 200); // venster met breedte 400 en hoogte 200
scene.setFill(Color.PURPLE);
primaryStage.setTitle("Hallo!");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch();
}
}

```