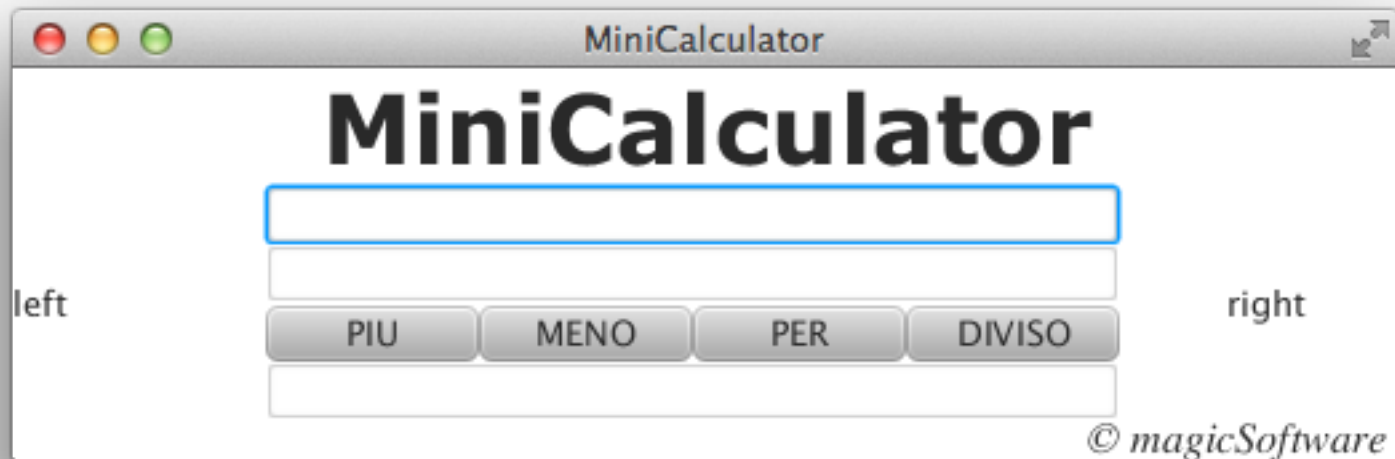


Esercizio



Soluzione - 1

```
public class MiniCalculator extends Application {  
    final TextField input1 = new TextField("");  
    final TextField input2 = new TextField("");  
    final TextField output = new TextField("");  
  
    public void start(Stage primaryStage) {  
        primaryStage.setTitle("MiniCalculator");  
        ...  
    }  
  
    public static void main(String[] args) {  
        Application.launch(args);  
    }  
  
    class MyListener implements EventHandler {  
        public void handle(Event t) {  
            ...  
        }  
    }  
}
```



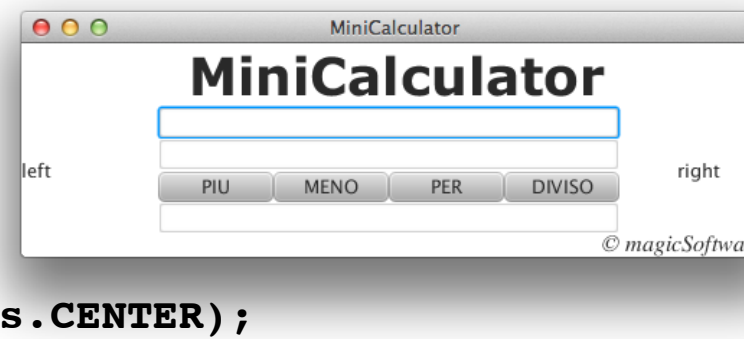
Soluzione - 2

```
public void start(Stage primaryStage) {
    primaryStage.setTitle("MiniCalculator");
    BorderPane borderP = new BorderPane();

    // ===== Top
    Label lt = new Label("MiniCalculator");
    lt.setFont(Font.font("Verdana", FontWeight.BOLD, 36));
    borderP.setTop(lt);
    BorderPane.setAlignment(lt, Pos.CENTER);

    // ===== Right
    Label labright = new Label("right");
    labright.setMinWidth(100.0);
    borderP.setRight(labright);
    BorderPane.setAlignment(labright, Pos.CENTER);

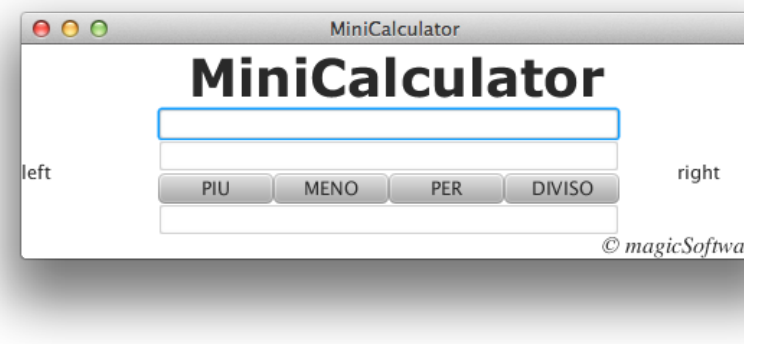
    // ===== Left
    Label lableft = new Label("left");
    lableft.setMinWidth(100.0);
    borderP.setLeft(lableft);
    BorderPane.setAlignment(lableft, Pos.CENTER_LEFT);
}
```



Soluzione - 3

```
// ===== Bottom
Label lb = new Label("© magicSoftware ");
lb.setFont(Font.font("Times", FontPosture.ITALIC, 16));
borderP.setBottom(lb);
BorderPane.setAlignment(lb, Pos.BOTTOM_RIGHT);

// ===== Center
// --- BOX
TilePane box = new TilePane();
box.setPrefColumns(1);
box.setMinWidth(320.0);
box.setMaxWidth(320.0);
// --- Campi
```



Soluzione - 4

```
// --- Campi
```

```
// --- Bottoni
```

```
TilePane hb = new TilePane();
```

```
hb.setPrefWidth(320.0);
```

```
hb.setAlignment(Pos.CENTER);
```

```
Button sum = new Button("PIU");
```

```
sum.setId("+");
```

```
sum.setMaxWidth(Double.MAX_VALUE);
```

```
sum.setPrefWidth(80);
```

```
Button divide = new Button("DIVISO");
```

```
divide.setPrefWidth(80);
```

```
divide.setId("/");
```

```
Button multiply = new Button("PER");
```

```
multiply.setPrefWidth(80);
```

```
multiply.setId("*");
```

```
Button subtract = new Button("MENO");
```

```
subtract.setPrefWidth(80);
```

```
subtract.setId("-");
```

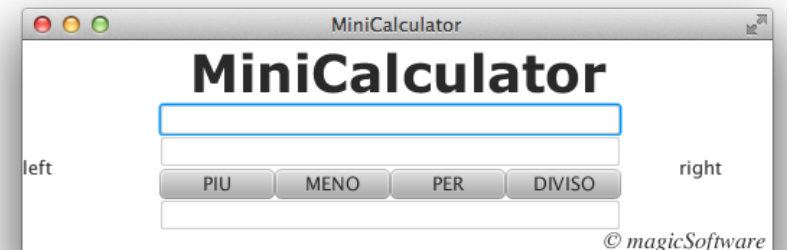


Soluzione - 5

```
hb.getChildren().addAll(sum, subtract, multiply, divide);
box.getChildren().addAll(input1, input2, hb, output);

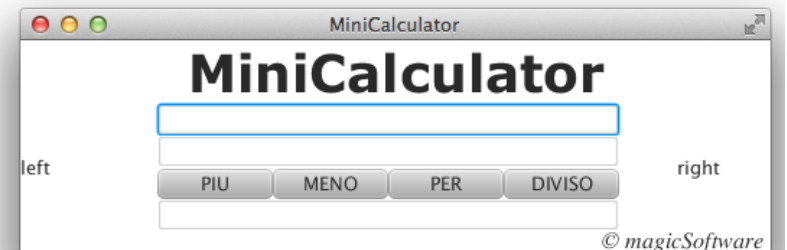
// ===== Behaviour
MyListener list = new MyListener();
multiply.addEventFilter(ActionEvent.ACTION, list);
sum.addEventFilter(ActionEvent.ACTION, list);
divide.addEventFilter(ActionEvent.ACTION, list);
subtract.addEventFilter(ActionEvent.ACTION, list);

borderP.setCenter(box);
Scene scene = new Scene(borderP);
primaryStage.setScene(scene);
primaryStage.sizeToScene();
primaryStage.show();
}
```



Soluzione - 7

```
private class MyListener implements EventHandler {  
    public void handle(Event t) {  
        String i1 = input1.getText();  
        String i2 = input2.getText();  
        double o1 = 0;  
        double o2 = 0;  
        try {  
            o1 = Double.parseDouble(i1);  
            o2 = Double.parseDouble(i2);  
        } catch (NumberFormatException e) {  
            Label msg = new Label("Errore - Not A Number!");  
            StackPane g = new StackPane();  
            g.getChildren().add(msg);  
            Scene stageScene = new Scene(g, 200, 200);  
            Stage errorStage = new Stage();  
            errorStage.setScene(stageScene);  
            errorStage.show();  
            return;  
        }  
    }  
}
```



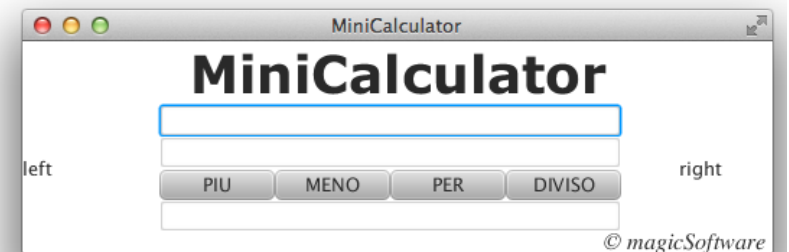
Soluzione - 8

```
String s = ((Button) t.getTarget()).getId();
switch (s) {
    case "+":
        output.setText("" + (o1 + o2));
        break;
    case "*":
        output.setText("" + (o1 * o2));
        break;
    case "-":
        output.setText("" + (o1 - o2));
        break;
    case "/":
        output.setText("" + (o1 / o2));
        break;
}
```

```
}
```

```
}
```

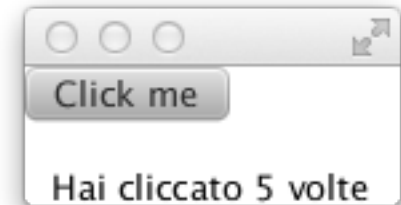
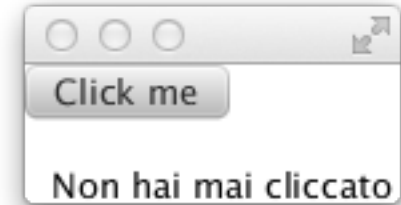
```
}
```



Come organizzare gli
ascoltatori/osservatori

Listener Esterno

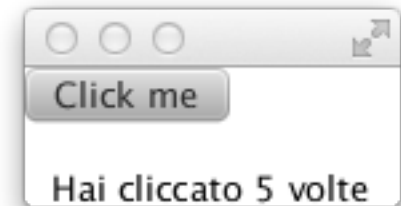
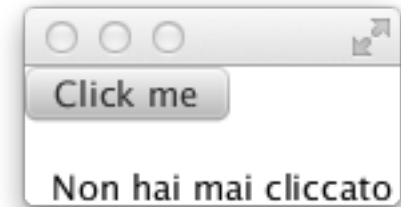
```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
        text=new Text(10,50,"Non hai mai cliccato ");
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener(this);
        btn.addEventHandler(ActionEvent.ACTION, a);
        Group root = new Group(btn);
        root.getChildren().add(text);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }
    public void updateText(int n){
        text.setText("Hai cliccato "+n
            +" volte");
    }
    public static void main(String[] args) {
        Application.launch(args);
    }
}
```



```
class Listener
    implements EventHandler{
    AppWithEvents1 awe=null;
    int counter=0;
    Listener1(AppWithEvents1 a){
        awe=a;
    }
    public void handle(Event t) {
        awe.updateText(++counter);
    }
}
```

Listener Interno

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
        text=new Text(10,50,"Non hai mai cliccato ");
        Button btn = new Button();
        btn.setText("Click me");
        Listener a=new Listener(this);
        btn.addEventHandler(ActionEvent.ACTION, a);
        Group root = new Group(btn);
        root.getChildren().add(text);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }
    public void updateText(int n){
        text.setText("Hai cliccato "+n
            +" volte");
    }
    public static void main(String[] args) {
        Application.launch(args);
    }
}
```



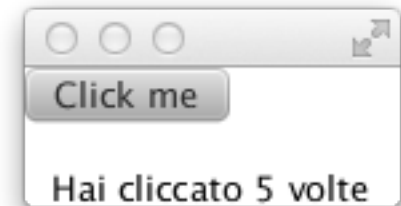
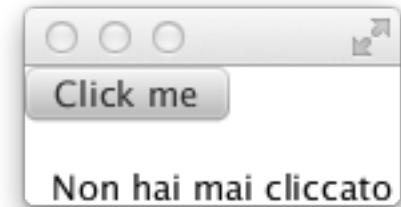
```
class Listener
    implements EventHandler{
AppWithEvents1 awe=null;
    int counter=0;
Listener1(AppWithEvents1 a){
    awe=a;
}
    public void handle(Event t) {
        awe.updateText(++counter);
    }
}
```

Listener Interno

```
public class AppWithEvents1 extends Application {  
    Text text=null;  
    public void start(Stage stage) {  
        text=new Text(10,50,"Non hai mai cliccato ");  
        Button btn = new Button();  
        btn.setText("Click me");  
        Listener1 a=new Listener();  
        btn.addEventHandler(ActionEvent.ACTION, a);  
        Group root = new Group(btn);  
        root.getChildren().add(text);  
        Scene scene = new Scene(root);  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

```
class Listener  
    implements EventHandler{  
        int counter=0;  
        public void handle(Event t) {  
            updateText(++counter);  
        }  
    }
```

```
public void updateText(int n){  
    text.setText("Hai cliccato"  
        +n+" volte");  
}  
public static void main(  
    String[] args) {  
    Application.launch(args);  
}}
```

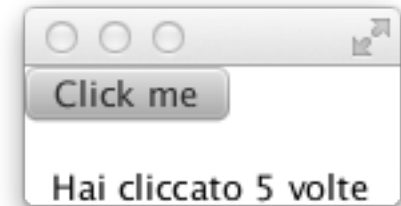
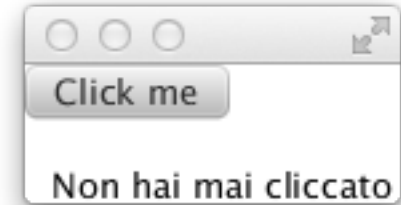


Listener Interno Anonimo

```
public class AppWithEvents1 extends Application {
    Text text=null;
    public void start(Stage stage) {
        text=new Text(10,50,"Non hai mai cliccato ");
        Button btn = new Button();
        btn.setText("Click me");
        Listener1 a=new EventHandler(){
            int counter=0;
            public void handle(Event t) {
                updateText(++counter);
            };
        };
        btn.addEventHandler(ActionEvent.ACTION, a);
        Group root = new Group(btn);
        root.getChildren().add(text);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }
}

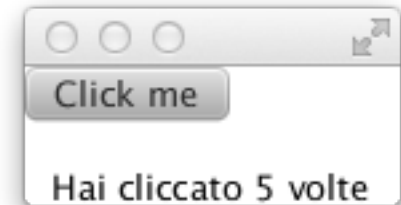
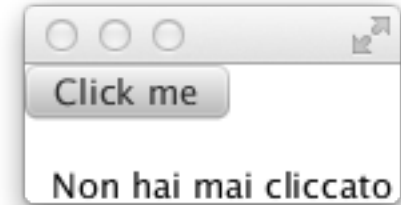
public void updateText(int n){
    text.setText("Hai cliccato"
        +n+" volte");
}

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



Self Listener

```
public class AppWithEvents
    extends Application implements EventHandler {
    Text text=null;
    int counter=0;
    public void start(Stage stage) {
        text=new Text(10,50,"Non hai mai cliccato ");
        Button btn = new Button();
        btn.setText("Click me");
        btn.addEventHandler(ActionEvent.ACTION, this);
        Group root = new Group(btn);
        root.getChildren().add(text);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }
    public void handle(Event t) {
        updateText(++counter);
    }
```

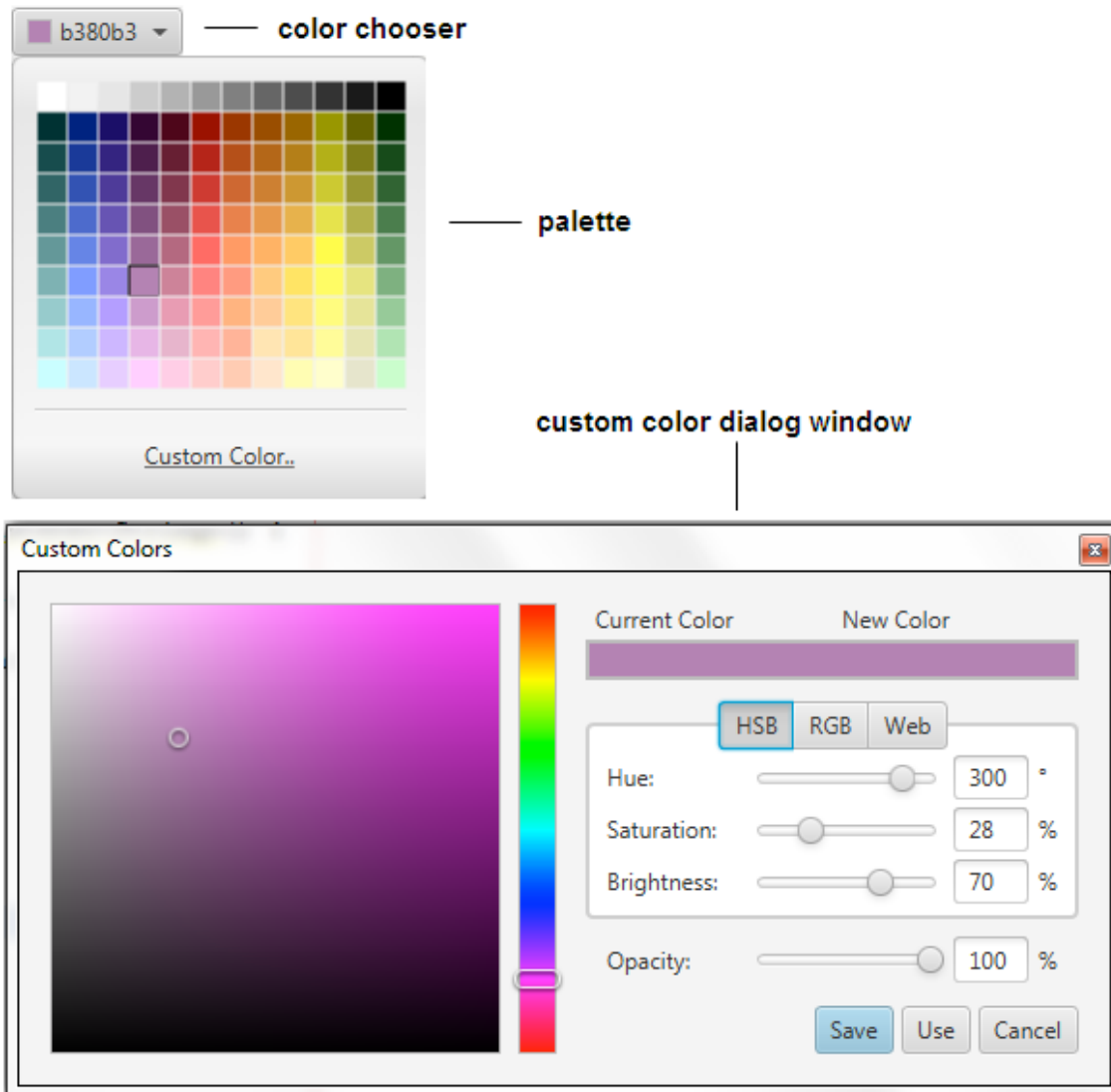


```
public void updateText(int n){
    text.setText("Hai cliccato"
        +n+" volte");
}
public static void main(
    String[] args) {
    Application.launch(args);
}}
```

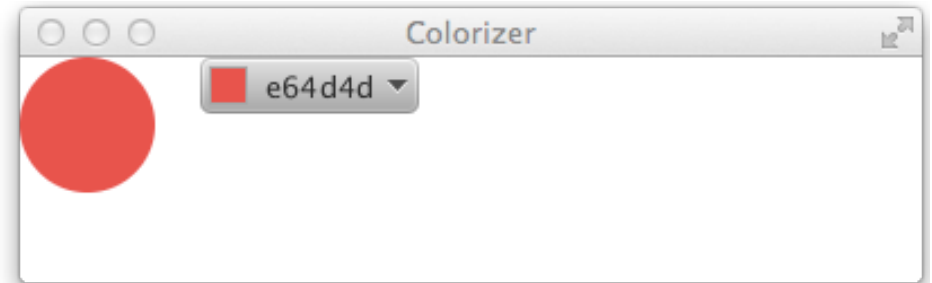
Due questioni:

- ColorPicker
- Convenience Methods

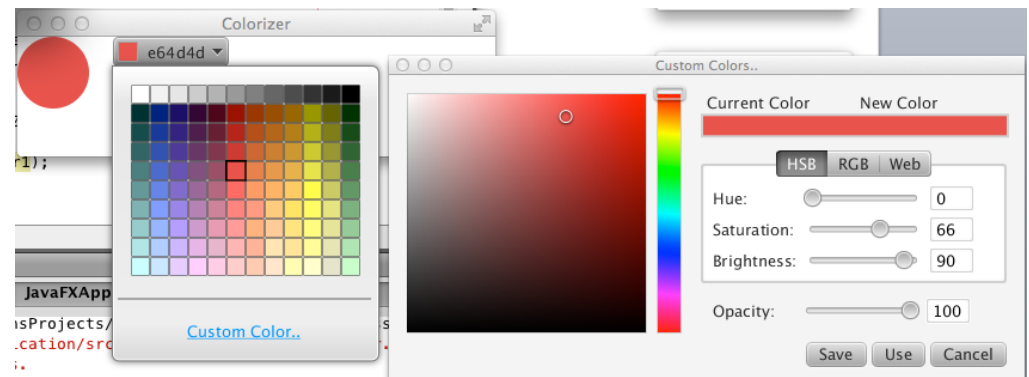
ColorPicker



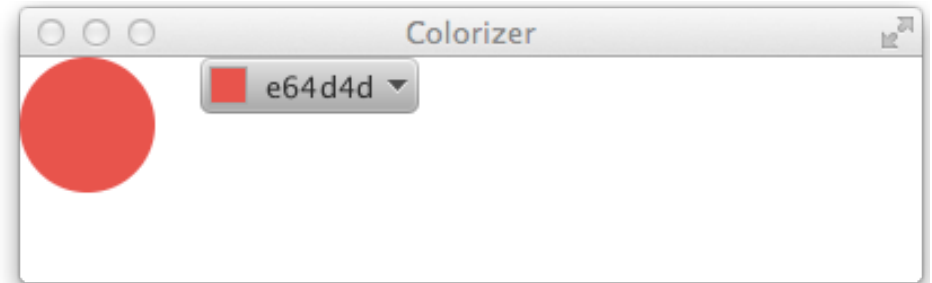
ColorPicker



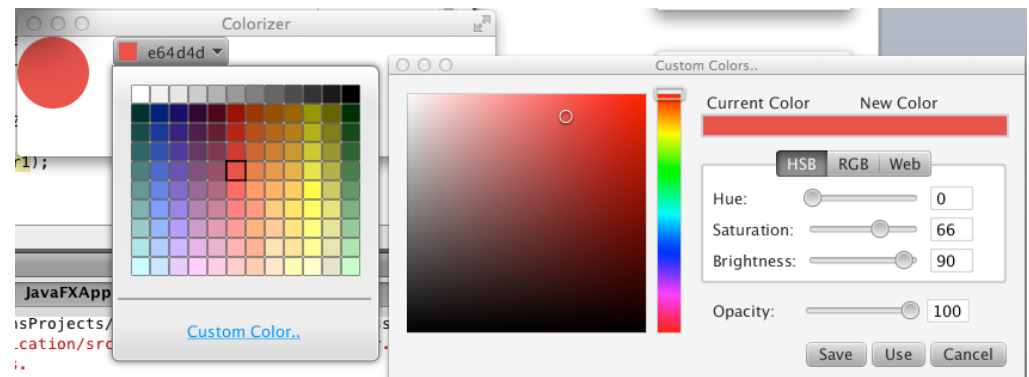
```
public class Colorizer extends Application {  
    public void start(final Stage stage) {  
        final Circle circ = new Circle(40, 40, 30);  
        final ColorPicker colorPicker1 = new ColorPicker(Color.BLACK);  
  
        colorPicker1.addEventHandler(ActionEvent.ACTION, new EventHandler() {  
            @Override  
            public void handle(Event t) {  
                System.out.println(t.getEventType());  
                circ.setFill(colorPicker1.getValue());  
            }  
        });  
        Scene scene = new Scene(new HBox(20), 400, 100);  
        HBox box = (HBox) scene.getRoot();  
        box.getChildren().addAll(circ, colorPicker1);  
        stage.setScene(scene);  
        stage.show();  
    }  
    ...  
}
```



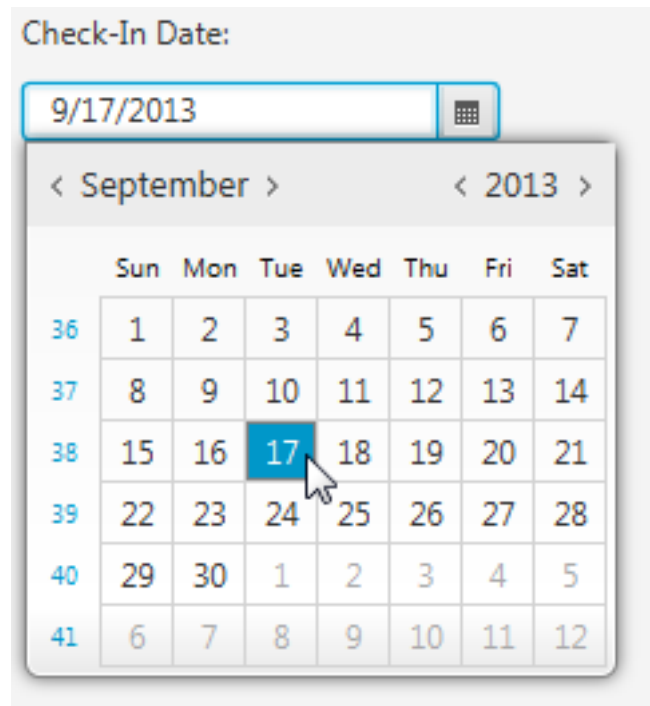
Convenience Methods



```
public class Colorizer extends Application {  
    public void start(final Stage stage) {  
        final Circle circ = new Circle(40, 40, 30);  
        final ColorPicker colorPicker1 = new ColorPicker(Color.BLACK);  
        colorPicker1.setOnAction(new EventHandler() {  
            // colorPicker1.addEventHandler(ActionEvent.ACTION, new EventHandler() {  
            @Override  
            public void handle(Event t) {  
                System.out.println(t.getEventType());  
                circ.setFill(colorPicker1.getValue());  
            }  
        });  
        Scene scene = new Scene(new HBox(20), 400, 100);  
        HBox box = (HBox) scene.getRoot();  
        box.getChildren().addAll(circ, colorPicker1);  
        stage.setScene(scene);  
        stage.show();  
    }  
    ...  
}
```



DatePicker



<http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/date-picker.htm#CCHHJBEA>

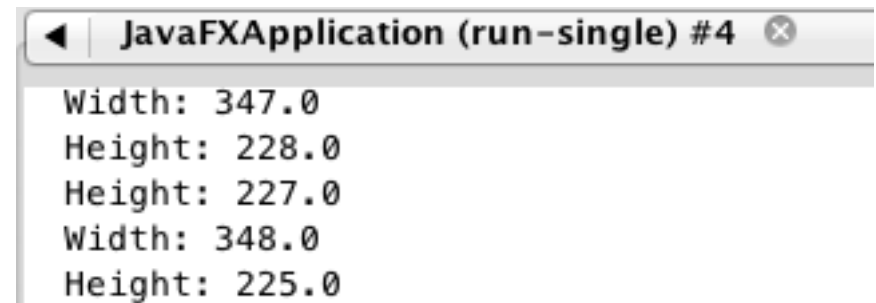
Ridimensionamento della finestra

Come si impara a programmare?

Leggendo (tanto) codice

Cattura gli eventi di resize della scena

```
public class WindowResize extends Application {  
    public void start(Stage stage) {  
        Node circ = new Circle(40, 40, 30);  
        Parent root = new Group(circ);  
        Scene scene = new Scene(root, 400, 300);  
        scene.widthProperty().addListener(new ChangeListener<Number>() {  
            public void changed(ObservableValue<? extends Number>  
                observableVal, Number oldWidth, Number newWidth) {  
                System.out.println("Width: " + newWidth);  
            }  
        });  
        scene.heightProperty().addListener(new ChangeListener<Number>() {  
            public void changed(ObservableValue<? extends Number>  
                observableVal, Number oldHeight, Number newHeight) {  
                System.out.println("Height: " + newHeight);  
            }  
        });  
        stage.setScene(scene);  
        stage.show();  
    }  
    ...  
}
```



Scacchiera bidimensionale

```
public void start(Stage stage) {
    int w = 400; int h = 699;
    int size = 8;
    GridPane board = new GridPane();
    Scene scene = new Scene(board, w, h);
    NumberBinding rectsAreaSize =
        Bindings.min(board.heightProperty(), board.widthProperty());
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            Rectangle rect;
            if ((i+j) % 2 == 1) {
                rect = new Rectangle(15.0, 15.0, Color.WHITE);
            } else {
                rect = new Rectangle(15.0, 15.0, Color.BLACK);
            }
            rect.widthProperty().bind(rectsAreaSize.divide(size));
            rect.heightProperty().bind(rectsAreaSize.divide(size));
            board.add(rect, i, j);
            GridPane.setHalignment(rect, HPos.CENTER);
        }
    }
    stage.setScene(scene); stage.show();
}
```

