Ce Este JavaFX ?

JavaFx este o platforma software folosita pentru a crea aplicatii multi-platforma. Aceasta a fost creata pentru a permite crearea si rularea unor aplicatii web bogate in continut media imersiv, platforma JavaFx asigurand astfel ca aceste aplicatii(RIA-rich internet applications- aplicatii web ce au unele dintre caracteristicile unor aplicatii desktop) arata si ruleaza in acelasi mod pe diferite dispozitive de diferite marimi.

Sa incepem sa cunoastem JavaFx

Aceasta sectiune ne invata cum sa cream prima aplicatie JavaFx folosind NetBeans IDE 6.5 .

Astfel, vom crea o sfera cu text care isi schimba culoarea, forma si marimea intr-o anumita perioada de timp. De asemenea sfera va putea fi miscata cu mouse-ul.

Figura 1: Sfera in JavaFX

1.Asigurati-va ca aveti NetBeans IDE 6.5 instalat.

2. Porniti aplicatia NetBeans

- pentru Windows selectati : Start > All Programs > NetBeans > NetBeans IDE 6.5

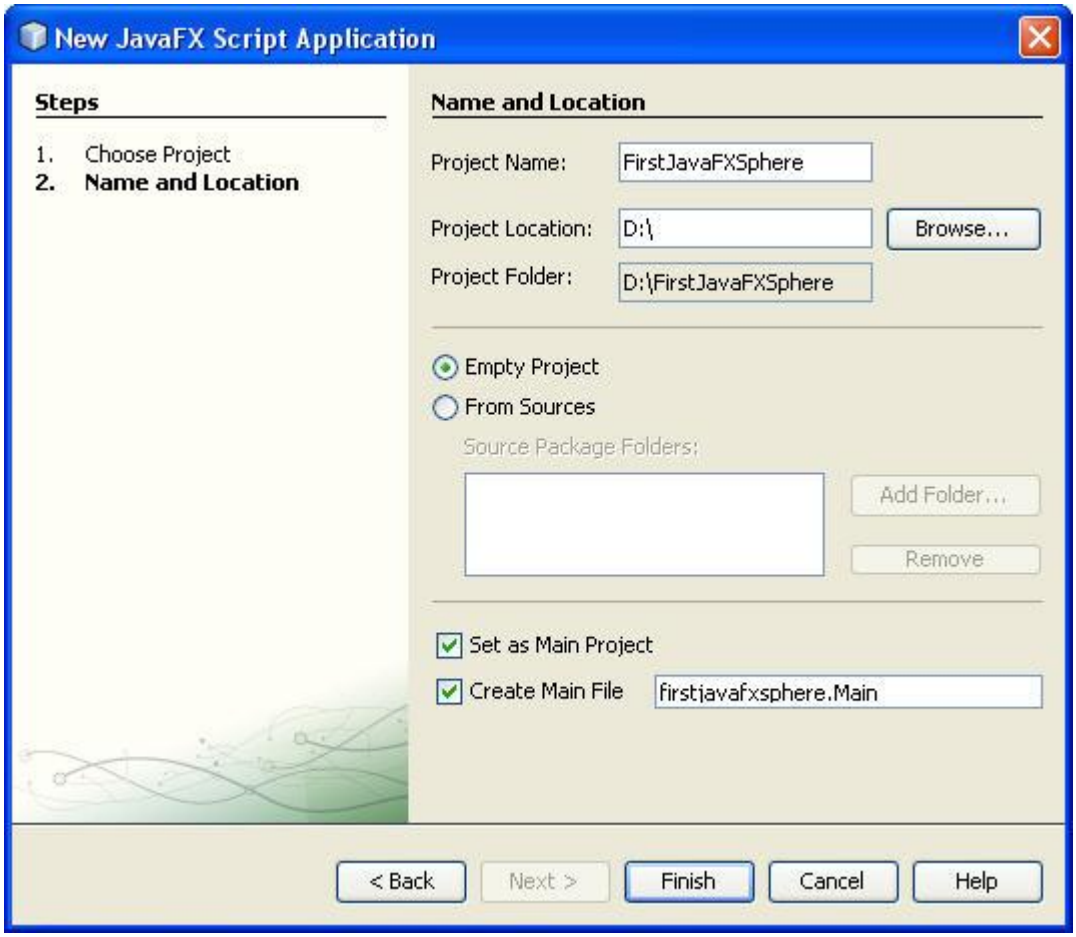
- pentru Mac OS X selectati: Applications > NetBeans > NetBeans 6.5

3. Alegeti : Create a JavaFX Script Application project;

i) File>New Project (Ctrl-Shift-N)

ii) In fereastra **New Project** , selectati **JavaFX** in panoul "Categorii" si **JavaFX Script Application** in panoul "**Proiecte**". Click Next.

iii) In pagina **Name and Location**, pentru numele proiectului (Project name) tastati **FirstJavaFXSphere**, selectati locatia proiectului dumneavoastra din campul **Project Location** si lasati celelalte valori neschimbate dupa cum va este aratat in figura urmatoare.



iv) dati click pe finish.

Proiectul FirstJavaFXSphere se va deschide atat in ferestra **Proiecte** (Projects) cat si in fereastra **Fisiere** (Files), iar fisierul **Main.fx** va deschide editorul de cod, dupa cum arata figura 3.

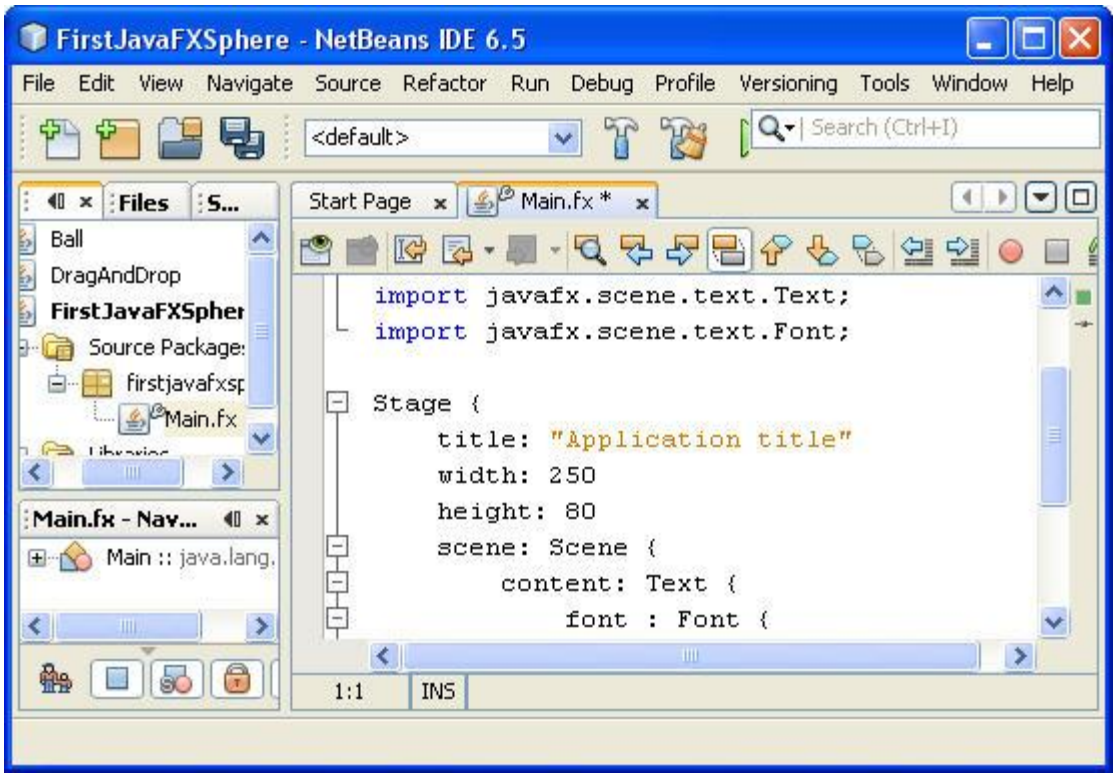


Figura 3.

Se poate observa ca, codul sursa JavaFX este inclus in fisierul **Main.fx** in mod automat. Acest cod include cateva declaratii de import si cateva valori fixe ale codului sursa-un exemplu aproximativ sunt clasele/entitatile - (cum ar fi **Stage**). Aceste “clase”/entitati sunt concepte cheie in aplicatia JavaFX si sunt descrise in urmatorul tabel.

Table 1: Object Literals Created by Default

Object Literal	Description
`Stage`	Cea mai importanta fereastra de continut folosita pentru a afisa orice obiect JavaFX. Variabilele de baza title,width si height definesc textul ce va aparea pe marginea superioara a ferestrei precum si inaltimea si grosimea acestuia. Variabila **scene** defineste o instanta a “clasei” **scene** care defineste suprafata in care se poate plasa obiectul din JavaFX.
`Scene`	Este similar cu a desena suprafata pentru continutul graphic al aplicatiei dumneavoastra. Variabila de instanta **scene** are un continut variabil care este folosit pentru a stoca elementele grafice JavaFX si a defini continutul grafic al aplicatiei. Variabilele width si height definesc latimea si inaltimea

Table 1: Object Literals Created by Default

Object Literal	Description
	suprafetei in care se lucreaza. Pentru mai multe informatii despre clasa scene cititi Presenting UI Objects in a Graphical Scene, a lesson in Building GUI Applications With JavaFX

De retinut: Pentru mai multe informatii cititi: Using Objects, a lesson in Learning the JavaFX Script Programming Language.

---

4. Modificati entitatea **Stage** astfel in cat fereastra sa contina titlul de mai jos.Modificati entitatea **Scene** pentru a preciza atat marimea suprafetei de continut cat si textul ce va fi continut de aplicatie. De asemenea trebuie sa declarati continutul pentru clasa **TextAlignment**.

Puteti copia in programul dumneavoastra liniile ingrosate de mai jos :

Source Code

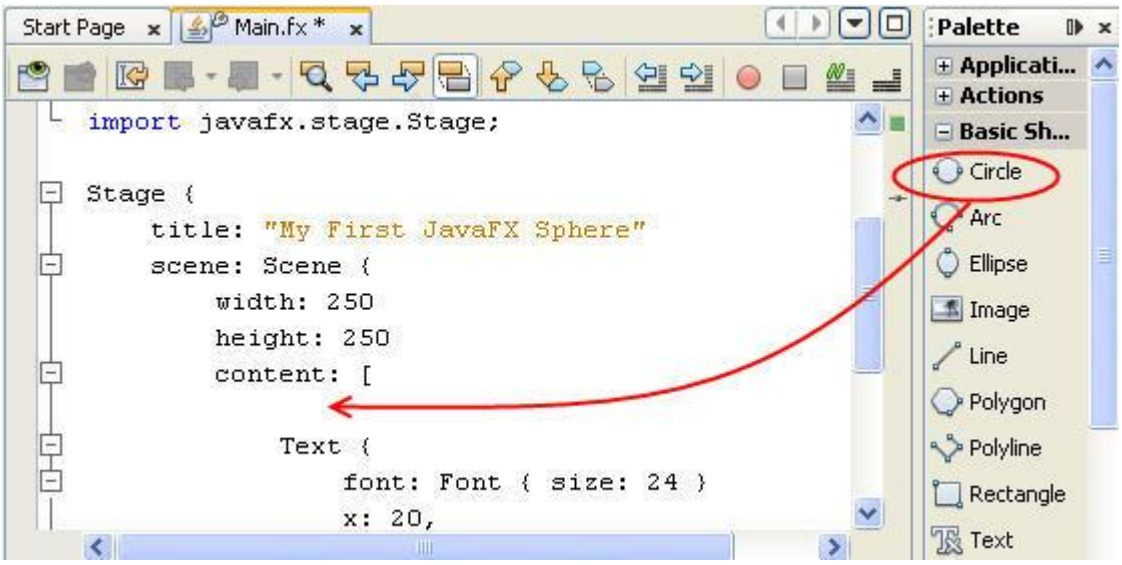
```
Stage {
     title: "My First JavaFX Sphere"
     scene: Scene {
         width: 250
         height: 250
         content: [
             Text {
                font: Font { size: 24 }
                x: 20, y: 90
               textAlignment: TextAlignment.CENTER
               content:"Welcome to \nJavaFX  World"
             }  //Text
        ] // content
     } // Scene

} // Stage
```

Pentru a repara eroarea raportata in caz ca nu ati declarat clasa TextAlignment, dati click dreapta in orice spatiu liber din editor si selectati **Fix Imports** (Ctrl-Shift-I)

Urmatoarea formulare trebuie adaugata in la inceputul fisierului **Main.fx** :

```
import javafx.scene.text.TextAlignment;
```

5) Definiti cercul din care veti crea sfera expanzand sectiunea **Basic Shapes** din fereastra **Palette** si tragand entitatea **Circle** in linia de cod de deasupra codului Text; dupa cum va este aratat in figura:



6) Modificati variabilele de instanta ale cercului astfel incat sa aiba marimea potrivita pentru a contine textul. De asemenea adaugati o noua setare **RadialGradient** care ii da cercului adancimea si il face sa arate ca o sfera.

i) Modificati codul sursa folosind urmatorul cod. Puteti copia liniile ingrosate incluzant parantezele drepte si acoladele pentru variabila **fill**.

Source Code

```
Stage {
 title: "My First JavaFX Sphere"
 width: 250
 height: 250

 scene: Scene {
     content: [
         Circle {
             centerX: 100,
             centerY: 100
             radius: 90

             fill: RadialGradient {
                     centerX: 75
                     centerY: 75
                     radius: 90
                     proportional: false
```

```
                        stops: [
                            Stop {
                                offset: 0.0
                                color: Color.RED},
                            Stop {
                                offset: 1.0
                                color: Color.DARKRED}
                        ] // stops
                    } // RadialGradient
                } // Circle
                Text {
                  font: Font {
                          size: 24
                  }
                  x: 20, y: 90
                  textAlignment: TextAlignment.CENTER
                  content:"Welcome to \nJavaFX  World"
                }  //Text
            ] // content
        } // Scene
} // Stage
```

ii) Pentru a repara erorile evidentiate din cauza lipsei declaratiilor de import pentru cateva clase introduse in pasul anterior,dati click dreapta intr-un loc liber din editorul de cod si selectati **Fix Imports** (Ctrl-Shift-I).

Astfel urmatoarele declaratii vor fi adaugate la inceputul fisierului Main.fx

Source Code

```
import javafx.scene.paint.RadialGradient;
import javafx.scene.paint.Stop;
```

7) Porniti fereastra preview dand click pe butonul Enable Preview din toolbar-ul editor

O noua fereastra Design Preview va afisa cercul pe l-ati modificat pentru a arata ca o sfera. Particularitatea **feature** va permite sa vizualizati progresul dumneavoastra in crearea Interfetei Grafice (GUI). Atata timp cat codul sursa modificat nu contine erori, aceasta fereastra va afisa progresul dumneavoastra.

Sfera in fereastra Design Preview.

8) Urmati viitorii pasi pentru a schimba culoare textului in galben:

i) Adaugati variabilele **color** si **scale** cu valorile lor initiale. Aceste variabile vor fi folosite in animatia pe care o veti crea.

Source Code

```
var scale = 1.0;
var color = Color.YELLOW;


Stage {
   title: "My First JavaFX Sphere"
   scene: Scene {
       width: 250
       height: 250
```

ii) Modificati entitatea **Text** astfel incat liniile ce definesc variabilele de instanta fill si transforms sunt adaugate ca in exemplu. Puteti copia liniile de cod ingrosate.

Source Code

```
  Text {
     font: Font {
         size: 24
     }

     textAlignment: TextAlignment.CENTER
     content:"Welcome to \nJavaFX  World"
     fill: bind color

     transforms: [
         Translate{
             x: bind 20
             y: bind 90
```

```
        }//Translate
        Scale{
            x: 1
            y: bind scale
            pivotX: 0
            pivotY: 5
        } // Scale
    ]

} // Text
```

Variabila **fill** “uneste” culoare textului cu variabila color. Entitatea **Translate** reprezinta o translatie definita de variabilele x si y. Acestea definesc o mutare atat pe axa x cat si pe axa y. Entitatea **Scale** reprezinta o transformare scalara definita de variabilele x,y,pivotX si pivotY. Variabilele x si y definesc factorul cu care aceste coordonate sunt scalate pe axa x respectiv y. Variabilele pivotX si pivotY definesc coordonatele X si Y unde are loc scalarea.

iii) Pentru a corecta erorile datorate lipsei unor declaratii de date pentru cele doua clase introduse, dati click dreapta intr-un loc liber si selectati **Fix Imports** (Ctrl-Shift-I)

Selectati javafx.scene.transform.Scale din prima fereastra, ca in figura 5, si apasati enter. In fereastra nou aparuta selectati javafx.scene.transform.Translate si apasati enter.

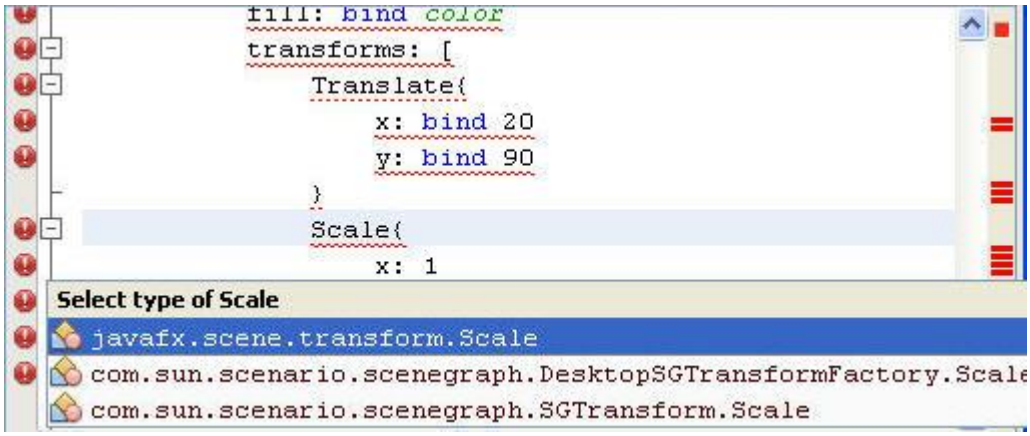


Figura 6.

Urmatoarele declaratii sunt adaugate la inceputul fisierului **Main.fx** :

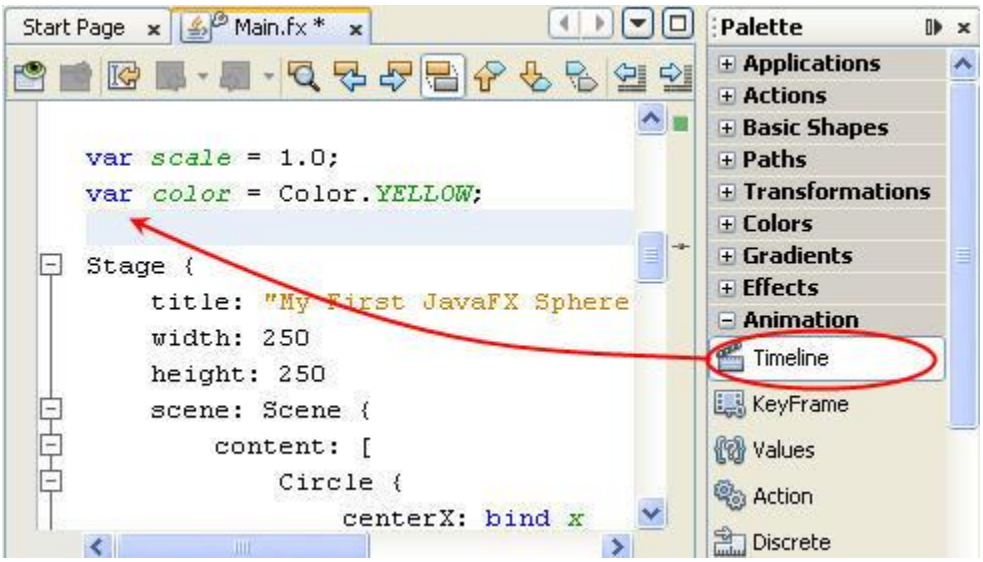
Source Code

```
import javafx.scene.transform.Scale;
import javafx.scene.transform.Translate;
```

Puteti observa in fereastra Design Preview ca textul si-a schimbat culoarea in galben, exact ca in figura urmatoare.

9)Adaugati animatii astfel incat textul sa-si schimbe culaore in verde sis a se invarta.

iv) Mariti/extindeti sectiunea **Animation** din fereastra **Palette**, selectati **Timeline** si trageti entitatea cu codul **Timeline** in linia de deasupra sectiunii de cod **Stage**, dupa cum vedeti in figura urmatoare:



Animatia are loc de-a lungul unei perioade de timp reprezentata de entitatea **javafx.animation.Timeline**. Fiecare perioada de timp contine una sau mai multe imagini (frame) reprezentate de **javafx.animation.KeyFrame**. Pentru mai multe informatii vizitati
Creating Animated Objects, a lesson in Building GUI Applications With JavaFX .

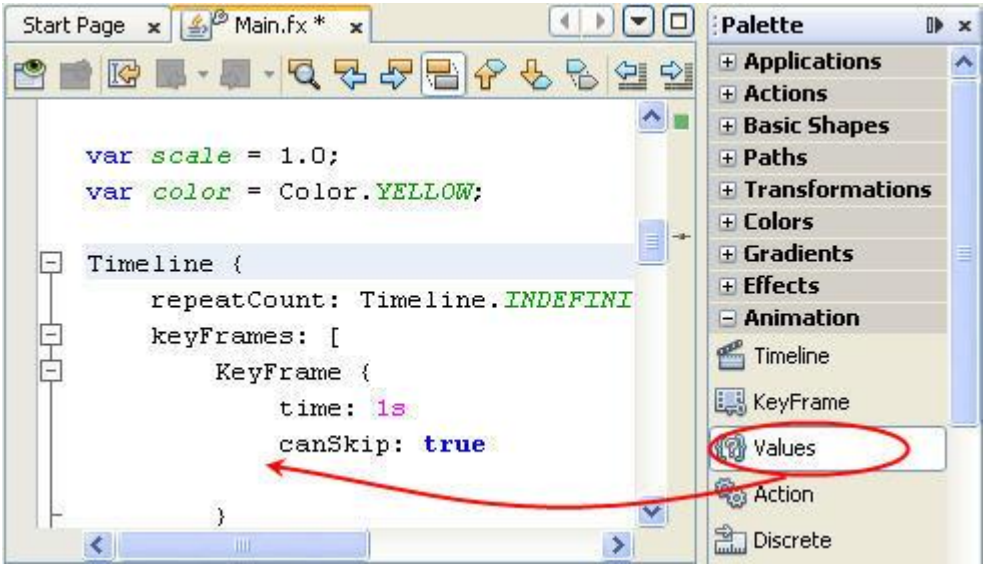
v) Schimbati valoare variabile time de la 1s la 5s astfel incat noul Timeline sa apara astfel :

Source Code

```
Timeline {
    repeatCount: Timeline.INDEFINITE
    keyFrames : [
        KeyFrame {
            time : 5s
            canSkip: true

        } // KeyFrame
    ] // keyFrames
} // Timeline
```

Aceasta perioada de timp se repeat la nesfarsit la un interval de 5 secunde. Aceasta valoare defineste timpul trecut dupa care imaginile se vor schimba. Urmatorii pasi din tutorial vor defini acele valori care trebuie introduse.

vi) Trageti entitatea **Values** din fereastra **Animation** exact deasupra variabilei time, exact ca in imagine :



Values defineste lista variabilelor tinta si valorile dorite care ar trebui interpolate la momentul specific din KeyFrame.

Copiati liniile de cod ingrosate in editorul codului sursa pentru a modifica instata values. Acest cod schimba culoarea textului din galben in verde.

Source Code

```
 values : [
      scale => -1.0 tween Interpolator.EASEBOTH
      color => Color.GREEN

 ] // values
```

Valorile dorite ale variabilelor scale si color sunt definite in intervalul de 5 secunde din keyframe. Operatorul => produce o notatie speciala care face mai usoara evidentierea valorile interpolate sau a proprietatilor obiectelor.

In acest caz folosim operatorul **tween** pentru a construi valorile interpolate pentru o scala de la 1.0 la -1.0 pentru a crea iluzia unei rotiri. Interpolator.EASEBOTH este interpolatorul inclus. De asemenea in perioada de 5 secunde culoarea se va schimba din galben in verde.

Click dreapta in orice spatiu liber si selectati **Format** (Alt-Shift-F) pentru a alinia corect noile linii de cod adaugate.

Adaugati **.play()** ; la sfarsitul entitatii Timeline dupa cum va este aratat mai jos. Aceasta metoda ruleaza perioada de timp ca un intreg cum este descrisa.

Source Code

```
 Timeline {
         repeatCount: Timeline.INDEFINITE
         keyFrames: [
             KeyFrame {
                 time: 5s
                 canSkip: true
                 values : [
                     scale => -1.0 tween Interpolator.EASEBOTH

                     color => Color.GREEN
                 ] // values
             } // KeyFrame
       ] // keyFrames
}.play();
```

Puteti observa in fereastra Design Preview ca textul isi schimba culoare in mod continuu de la galben la verde in timp ce se roteste. Daca nu puteti vedea aceste modificari apasati butonul Reset Preview.

10) Adaugati un comportament de “tarare” sferei impreuna cu textul.

6. Adaugati variabilele x si y si setati valorile implicite la 100.

Aceste variabile vor fi folosite pentru a uni pozitia sferei cu pozitia mouse-ului.

Source Code

```
var x = 100;
var y = 100;


Timeline {
    repeatCount: Timeline.INDEFINITE
    keyFrames: [
```

2. Uniti pozitia entitatii Circle cu variabilele x si y dupa cum va este arata mai jos :

Source Code

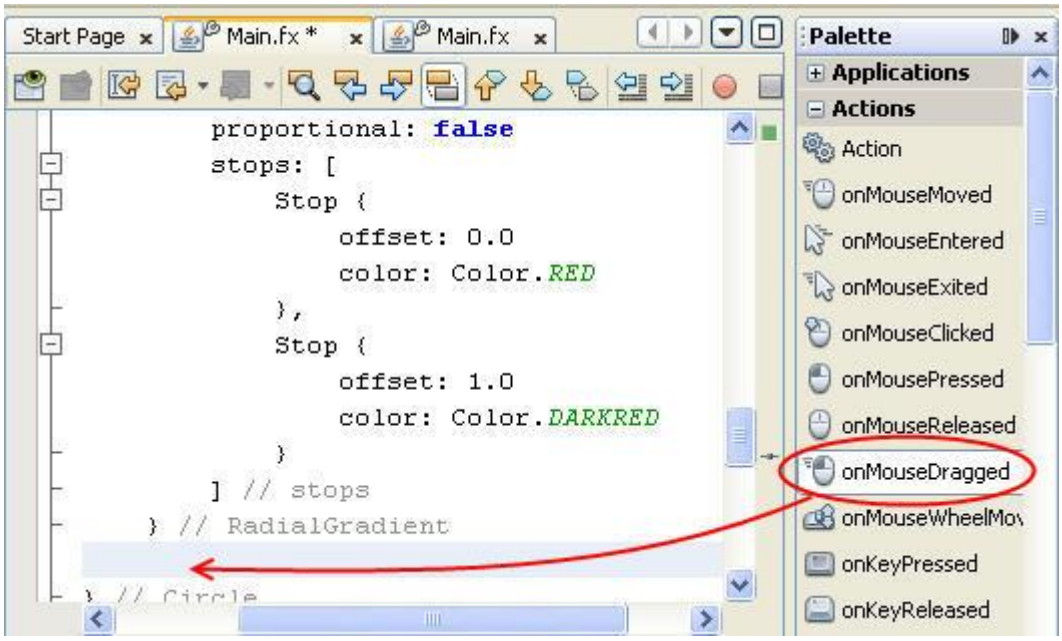
```
scene: Scene {
    content: [
       Circle {
           centerX: bind x
           centerY: bind y

           radius: 90
```

3. uniti variabilele x si y ale entitatii Translate cu variabilele globale x si y dupa cum va este aratat mai jos:

Source Code

```
transforms: [
    Translate{
        x: bind x - 80
        y: bind y - 10

    }
```

4. Trageti entitatea onMouseDragged din Action > onMouseDragged exact inainte de acolada pentru entitatea Circle



5) Copiati liniile de cod ingrosate de mai jos in editorul de cod pentru a defini comanda executata in cazul comenzii Mouse Dragged.

Source Code

```
onMouseDragged: function( e: MouseEvent ):Void {
x = e.x;

y = e.y;
}
```

11. Rulati programul.

0. Inchideti fereastra Design Preview.

In fereastra Projects, dati click dreapta pe FirstJAvaFXSphere si selectati Run Project.

IDE va compila proiectul si va pregati fisierele necesare rularii aplicatiei folosind modelul de executie Desktop. Dupa ce programul este compilat cu succes, o sfera animata, similara cu cea din figura 1 va fi afisata. Culoarea Textului at trebui sa se schimbe de la galben la verde in timp ce se roteste.

Felicitari, tocmai ati creat prima aplicatie in JavaFX !

Rularea aplicatiei JavaFX pe un emulator mobil (Mobile Emulator)

Aceasta sectiune va arata programatorilor cum sa ruleze o aplicatie JavaFX pe un emulator mobil(Mobile Emulator) folosind NetBeans IDE 6.5 pentru JavaFX 1.1.

In sectiunea anterioara, ati construit o aplicatie JavaFX numita FirsJavaFXSphere si ati rulat-o ca un program folosind modelul de executie standard (Standard Execution model).

In aceasta sectiune, veti castiga experienta ruland acelasi cod. Pentru a face acest lucru, folositi aplicatia FirstJavaFXSphere si rulati codul pe emulatorul mobil (Mobile Emulator).

Interfata de programare JavaFX contine profile desktop si comune (desktop and common). Profilul desktop contine clase specifice desktop ce nu pot fi rulate in aplicatii mobile. Cand folositi profilul comun (common) pentru a crea aplicatia JavaFX, practic va asigurati ca aplicatiile pot fi rulate atat pe computere cat si pe dispozitive mobile.

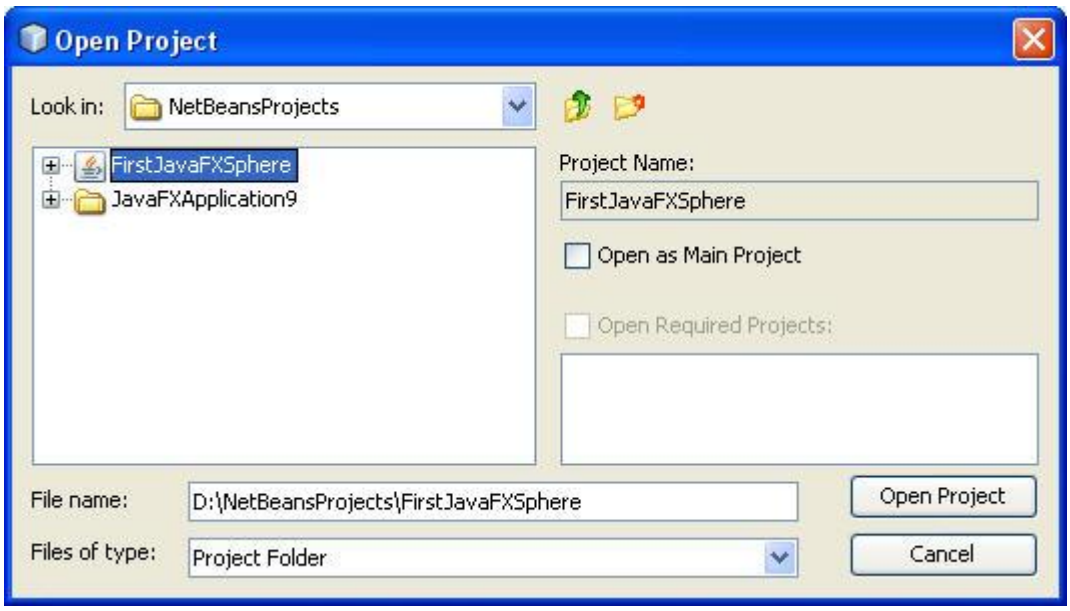
1. Porniti programul NetBeans IDE.
    a. - pentru Windows selectati : Start > All Programs > NetBeans > NetBeans IDE 6.5

    b. – pentru Mac OS X selectati : Applications > NetBeans > NetBeans 6.5

2. Deschideti primul proiect FirstJavaFXSphere pe care l-ati creat mai devreme :

    1. alegeti : File > Open Project (Ctrl-Shift-O).

    2. In fereastra Open Project selectati proiectul FirstJavaFXSphere.

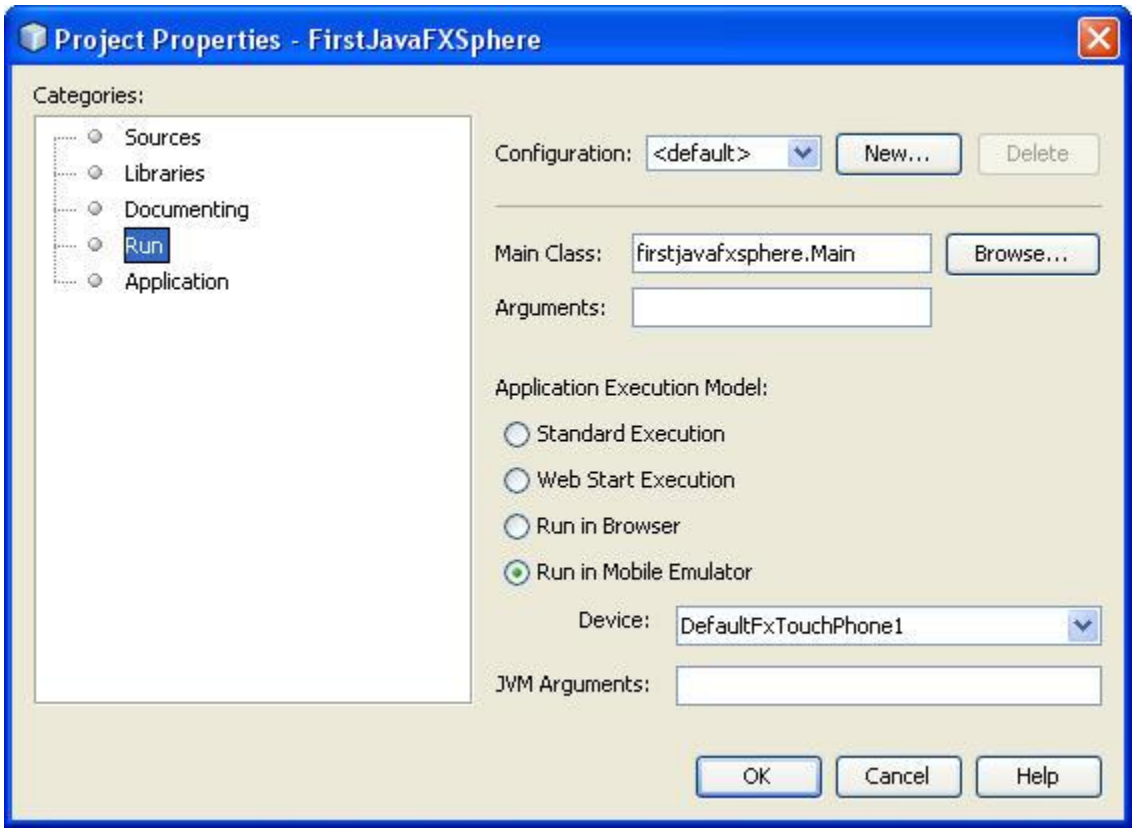


3.Selectati Open Project.

3. Selectati Run on Mobile Emulator

0. In fereastra Projects , dati click dreapta pe FirstJavaFXSphere si selectati Properties.

1. In fereastra de dialog Project Properties selectati Run.



2.Selectati Run in Mobile Emulator.

3. Selectati un dispozitiv din lista de dispositive.
4. click ok.

4. Rulati programul
   In fereastra Projects , click dreapta pe proiectul FirstJavaFXSphere si selectati Run Project.

   IDE va compila programul si va pregati fisierele necesare rularii acestuia folosind modelul de executie Run in Mobile Emulator. Dupa ce proiectul va fi compilat cu succes, o sfera animata va fi afisata pe ecranul unui dispozitiv mobil in emulatorul mobil (Mobile Emulator).


DefaultFxTouchPhone1 (ID: 1)
Application
View
Help
Phone number: 123456789
JavaFx
Welcome to
JavaFX World
Vibration is off