

Programlama Dilleri 2

Ders Notları



Burak Kuyumaz

Object Oriented Programming (OOP)

Bir sınıfın iki tane önemli özelliği vardır.

Özellikler → Değişken,
Davranışlar → Metotlar

Bir class nasıl yazılır

Bir isim verilir, önce özellikler tanımlanır sonra davranışlar tanımlanır.
bu şekilde yeni bir veri tipi tanımlanmış olur.

class isimleri büyük harfle başlamalıdır birden fazla kelimeden oluşuyorsa her kelimenin ilk harfi büyük harfle başlayacak şekilde yazılmalıdır.

```
isim degiskenIsmi = new isim();
```

→ bu şekilde bir class dan yeni bir nesne oluşturulmuş olur.

new operatörü kullanılıyorsa bu bir class dır..

ÖRNEK:

```
class Nokta{
    int x;
    int y;
    // nokta veri tipine ait iki ten özellik var.
    public void degerGoster(){
        System.out.println("x="+x+"y"+y);
    }
}
```

bir nesnenin içerisindeki bir metot veya özelliğe ulaşmak istenirse,

NesneIsmi.ulaşılacakIstenenDeğerVeyaÖzellik

```
package javaapplication1;
public class JavaApplication1 {
    public static void main(String[] args) {
        Nokta n=new Nokta();
        n.x=10;
        n.y=20;
        n.degerGoster();
    }
}
```

Nokta n=new Nokta(); burada parantez açıp kapayarak bir metot çağırılır..

Constructor(kurucu metot) : görevi class içerisinde tanımlanan değişkenlere ilk değer atamaktır.

sahip olması gereken özellikler:

- 1-kurucu metot ismi sınıf ismi ile aynı olmalıdır.(en önemli kural)
- 2-kurucu metot void anahtar sözcüğü veya geri dönüş değer tipi içeremez.
- 3- kesinlikle public olmalıdır.
- 4-overload edilebilir. (aynı isimde birden fazla değişken tanımlanabilir.)

değer atanmazsa 'Default Constructor' devreye girer..

```
class Nokta{
int x;
int y;
    public Nokta(){
        x=-1;
        y=-1;
        public Nokta(int a, int b){
            x=a; y=b;
        }
    }
    public void degerGoster(){
        System.out.println("x="+x+"y"+y);
    }
}
```

```
package javaapplication1;
public class JavaApplication1 {
    public static void main(String[] args) {
        Nokta n=new Nokta();
        n.degerGoster();
        Nokta n1=new Nokta();
        n1.degerGoster(2,3);
    }
}
```

Aynı isimli iki değişken varsa global olana ulaşmak için this operatörü kullanılır.

```
class Nokta{
int x;
int y;
    public Nokta(){
        x=-1;
        y=-1;
    }
    public Nokta(int x, int y){
        this.x=x; this.y=y;
    }
    public void degerGoster(){
        system.out.println("x="+x+"y"+y);
    }
}
```

Erişim belirleyiciler (Access modifiers)

public	:	herkes erişebilir demek
private	:	sınıfa özel demek
protected	:	kalıtımda kullanılır kalıtım yoksa private olarak kullanılır.
(boş)	:	bu özelliklere aynı paket içerisinde erişilebilir demektir.

bir metot veya bir özelliğin yanında bulunabilirler

```
class Nokta{
int x;
int y;p
    public Nokta(){
        public x=-1;
        public y=-1;
        public Nokta(int x, int y){
            this.x=x; this.y=y;
        }
    }
    public void degerGoster(){
        system.out.println("x="+x+"y"+y);
    }
}
```

```
package javaapplication1;
public class JavaApplication1 {
    public static void main(String[] args) {
        ogrenci ogr = new ogrenci();
        ogr.adSoyad = "ali al";
        ogr.yas=-10;
        ogr.bilgiGoster();
    }
}

public class ogrenci {
    public String adSoyad;
    public int yas;
    public void bilgiGoster(){
        System.out.println("Adı Soyadı = " + adSoyad
        +", "+"Yas"+yas);
    }
}
```

Kullanıcının class içerisindeki verilere istediği gibi erişmesinin engellenmesi gerekir. Bu yüzden her değişken public olarak tanımlanmaz.

```
public class ogrenci {
    private String adSoyad;
    private int yas;
    public void bilgiGoster(){
        System.out.println("Adı Soyadı = " + adSoyad
        +", "+"Yas"+yas);
    }
}
```

main class dan yas ve adSoyad fonksiyonlarını değiştirmek istersek erişemediğini söyler...

getter → private bir değişkenin değerini öğrenmeye yarar..

setter → private bir değişkenin değerini değiştirmeye yarar.

```
public class ogrenci {
    private String adSoyad;
    private int yas;

    public void setAdSoyad(){
        this.adSoyad = adSoyad;
    }
    public void setAdSoyad(String adSoyad){
        this.adSoyad=adSoyad;
    }
}
```

```
    }  
    public void setYas(){  
        this.yas = yas;  
    }  
    public void setYas(int yas){  
        if (yas<0){  
            this.yas =0;  
        }  
        else {  
            this.yas=yas;  
        }  
    }  
  
    public void bilgiGoster(){  
        System.out.println("Adı Soyadı = " + adSoyad  
+", "+"Yas"+yas);  
    }  
}
```

```
package javaapplication1;  
public class JavaApplication1 {  
    public static void main(String[] args) {  
        ogrenci ogr = new ogrenci();  
        ogr.setAdSoyad("Merhaba");  
        ogr.setYas(20);  
        ogr.bilgiGoster();  
    }  
}
```

Getter ve setter için kısa yol (Netbeans)

sağ tık Refactor → Encapsulate. getter ve setter seçilebilir ..

Kalıtım (Inheritance)

Bir class ı başka bir class dan türetme işlemine kalıtım denir.

```
class isim extends temel(super)_class_ismi{  
    ...  
}
```

bilinmesi gereken temel şey temel class dan türemiş class a sadece public ve protected özellik ve metotlar aktarılır. private olanları erişilemez.

ÖRNEK:

```
public class Nokta {  
    public int x;  
    public int y;  
}
```

```
public class Dikdortgen extends Nokta{  
    public int x1;  
    public int x2;  
    int en; int boy;  
    public void alanHesapla(){  
        boy = x1 -x;  
        en  = x2 -y;  
        System.out.println("");  
    }  
    public void cevreHesapla(){  
        boy = x1-x;  
        en =x2-y;  
        System.out.println("Çevre = "+(boy + en)*2);  
    }  
}
```

JAVA FX

JavaFx neden çıktı?

Eskiden desktop için swing kullanılırdı fakat pek iyi değildi. Javafx bu konuda css kullanmaya da izin verdiği için daha iyi bulundu.

xml veriyi biçimlendirmek için kullanılan bir işaretleme dilidir. Html gibidir fakat html den farkı kendi taglarını yazabilmeye olanak sağlar.

Çalışma Mantığı

Fxml dosyası Fxml document dosyasındaki kodlara göre çalışır. Yani .fxml dosyasına çift tıklayınca 'fxscene Builder' açılır...

Wrap → text yazı sığmadığı zaman aşağı atmaya sağlar..

her bir nesneyi diğer nesnelerden farklı yapan id sidir.. aynı container üzerindeki nesnelerin id leri birbirinden farklı olmak zorundadır..

Button da on action eventi, buton üzerine tıklanınca çalışan olaydır yani ne yapılması istendiğidir.. butona tıklanınca birşey yapılması isteniyorsa on action doldurulmalıdır.

handleButtonAction default olarak program tarafından yazılan bir fonksiyondur eğer kullanılmıyorsa silinmesi sıkıntı yaratmaz.

```
JOptionPane.showMessageDialog(null, "Merhaba dünya");
```

null değeri pencerenin ekranın tam ortasında görünmesini sağlar. Eğer null yerine scene ismini verirsen scene i ortalara.

```
private void clickMe(ActionEvent event) {  
    Random rnd=new Random();  
    int x=rnd.nextInt(571);  
    int y=rnd.nextInt(354);  
    btn_tikla.setLayoutX(x);  
    btn_tikla.setLayoutY(y);  
}
```

butonun yerini random olarak değiştirir.

Label

içerisindeki veriyi değiştirmek için setText, öğrenmek için getText kullanılır..

FXML Controller dosyası//

```
package ilkjavafxml;
```

```
import java.net.URL;
import java.util.Random;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javax.swing.JOptionPane;
```

```
public class FXMLDocumentController implements Initializable {
    int i;
    private Label label;
    @FXML
    private Button btn_tikla;
    @FXML
    private Label lbl_text;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void clickMe(ActionEvent event) {
        lbl_text.setText(Integer.toString(i));
        i++;
    }
}
```

→ Butona her basılışta lbl_text id sine sahip Label içerisindeki sayıyı bir artırır.

TextField

TextField içine veri girişi yaptırır. Eğer matematiksel işlem yapılacaksa önce int'e çevirmek gerekir..

NOT :

id'si olmayan elemanlar 'Make Controller' yapınca fxml controller dosyasına aktarılmaz..

```
package ilkjavafxml;

import java.net.URL;
import java.util.Random;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javax.swing.JOptionPane;

public class FXMLDocumentController implements Initializable {
    int i=1;
    private Label label;
    @FXML
    private Button btn_tikla;
    private Label lbl_text;
    @FXML
    private TextField islem1;
    @FXML
    private TextField islem2;
    @FXML
    private TextField islem3;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void clickMe(ActionEvent event) {
        int sayi1 = Integer.parseInt(islem1.getText());
        int sayi2 = Integer.parseInt(islem2.getText());
        if (i==1){
            islem3.setText(Integer.toString(sayi1 + sayi2));
            i=0;
        }
        else{
            islem3.setText(Integer.toString(sayi1 - sayi2));
        }
    }
}
```

```
        i=1;  
    }  
  
}  
  
}
```

→ ilk tıklamada toplamını ikinci tıklamada farkını veren program..

UYGULAMA :

ComboBox ları fxml içerisinde tanımladığımız zaman `private ComboBox<?>` şeklinde aktarılır. Buradaki '?' yerine ComboBox ın içine gelecek verinin tipi girilir. Örnek : `private ComboBox<String>`.

initialize FXML Controller dosyasında ilk derlenen kısımdır.

Program: Bir bilet satın alma programı yapın...

```
package biletal;

import java.net.URL;
import java.util.Observable;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
public class FXMLDocumentController implements Initializable {
    private Label adLabel;
    @FXML
    private TextField adText;
    private Label tarihLabel;
    @FXML
    private CheckBox koltuk1;
    @FXML
    private CheckBox koltuk2;
    @FXML
    private CheckBox koltuk4;
    @FXML
    private CheckBox koltuk3;
    @FXML
    private Button btn1;
    @FXML
    private Button btn_clear;
    @FXML
    private Label biletLabel;
    @FXML
    private TextField tarihText;
    @FXML
    private ComboBox<String> neredenCombo;
    @FXML
    private ComboBox<String> nereyeCombo;

    ObservableList<String> nereden
=FXCollections.observableArrayList("Antalya","İzmir","Ankara","İst
anbul");// <> içerisinde hangi tür veriler gireceğini girmen
gerekir.
    ObservableList<String> nereye
=FXCollections.observableArrayList("Paris","Londra","Sochi","Beşik
```

```
taş");

@Override
public void initialize(URL url, ResourceBundle rb) {
    btn_clear.setVisible(false); // ilk olarak temizleme
    butonu görünmez yap
    neredenCombo.setItems(nereden); // üstteki listeyi combo-
    box a gömdük
    nereyeCombo.setItems(nereye);
}

@FXML
private void click_btn1(ActionEvent event) {
    String isim, tarih, nereden2, nereye2, koltukNo = null;
    isim = adText.getText();
    tarih = tarihText.getText();
    nereden2 = neredenCombo.getValue();
    nereye2 = nereyeCombo.getValue();

    if (koltuk1.isSelected() == true){koltukNo =
koltuk1.getText();koltuk1.setDisable(true);}
    if (koltuk2.isSelected() == true){koltukNo =
koltuk2.getText();koltuk2.setDisable(true);}
    if (koltuk3.isSelected() == true){koltukNo =
koltuk3.getText();koltuk3.setDisable(true);}
    if (koltuk4.isSelected() == true){koltukNo =
koltuk4.getText();koltuk4.setDisable(true);}

    biletLabel.setText("Bilet Bilgileri:\nAd Soyad:
\t"+isim+"\nTarih:\t\t"+tarih+"\nNereden:\t\t"+nereden2+"\nNereye:
\t\t"+nereye2+"\nKoltuk No:\t"+koltukNo);
    btn1.setVisible(false);
    btn_clear.setVisible(true);
}

@FXML
private void click_btn_clear(ActionEvent event) {
    adText.clear();
    tarihText.clear();
    nereyeCombo.setValue("");
    neredenCombo.setValue("");
    biletLabel.setText("");
    btn1.setVisible(true);
    btn_clear.setVisible(false);
}
}
```

JavaFX - Devamı

```
@FXML
private void basildi(ActionEvent event) {
    Button b = (Button)event.getSource();
    lbl_txt.setText(b.getText());
}
```

event parametresi ile bu fonksiyonun hangi nesne tarafından kullanıldığı anlaşılabilir.

FXMLDocumentController

```
@FXML
private void basildi(ActionEvent event) {
    Button b = (Button)event.getSource();
    lbl_txt.setText(b.getText());
}

}
```

FXMLDocument

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

```
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
```

```
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="237.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="ders.FXMLDocumentController">
    <children>
        <Button layoutX="72.0" layoutY="110.0" mnemonicParsing="false" text="Buton 1"
onAction="#basildi" />
        <Button layoutX="211.0" layoutY="110.0" mnemonicParsing="false" text="Buton 2"
onAction="#basildi" />
        <Button layoutX="331.0" layoutY="110.0" mnemonicParsing="false" text="Buton 3 "
onAction="#basildi" />
        <Button layoutX="462.0" layoutY="110.0" mnemonicParsing="false" text="Buton 4"
onAction="#basildi" />
        <Label fx:id="lbl_txt" layoutX="282.0" layoutY="55.0" text="Label" />
    </children>
</AnchorPane>
```

Label in Arkaplanını Değiştirme

```
private void basildi(ActionEvent event) {
    Button b = (Button)event.getSource();
    if (b.getText().equals("Buton 1"))
        lbl_txt.setStyle("-fx-background-color:blue");
    else if (b.getText().equals("Buton 2"))
        lbl_txt.setStyle("-fx-background-color:yellow");
    else if (b.getText().equals("Buton 3"))
        lbl_txt.setStyle("-fx-background-color:green");
    else
        lbl_txt.setStyle("-fx-background-color:black");
}

}

// Bu fonksiyon çalıştığında ilk olarak event parametresi
// sayesinde hangi buton tarafından kullanıldığını bulur. Ardından bu
// butonlara göre if kalıbıyla farklı farklı renk verilebilir.

// lbl_txt.setStyle("-fx-background-color:green");
// Label in arkaplanını yeşil yapar.
```

Liste Kullanımı

```
public void initialize(URL url, ResourceBundle rb) {

    lst_list.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);
    ObservableList<String> liste = FXCollections.observableArrayList();
    for(int i=0;i<10;i++)
        liste.add(Integer.toString(i));
    lst_list.setItems(liste);
}

/*lst_list.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);
*
*bu kod satırını ListView aracı içerisindeki nesnelerde çoklu seçim
*yapılabilir.
*/
```

```
/*ObservableList<String> liste = FXCollections.observableArrayList();
*
*kod satırı içinde String barındıran liste isminde bir liste oluşturur.
* bu listenin özelliği boyutunun içinde barındırdığı nesneye göre
* değişmesi
*/

/*liste.add(Integer.toString(i));
*
* listeye eleman atamaya yarar.
*/
```

Listede seçilenleri Label a atama

```
@FXML
private void secildi(MouseEvent event) {

lbl.setText(lst_list.getSelectionModel().getSelectedItems());
}
```

ListView da seçili olan sayıların toplamının veren program:

```
public class FXMLDocumentController implements Initializable {

    private Label lbl_txt;
    @FXML
    private Label lbl;
    @FXML
    private ComboBox<String> cmb_liste;
    @FXML
    private ListView<String> lst_list;

    @Override
    public void initialize(URL url, ResourceBundle rb) {

lst_list.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);

        ObservableList<String> liste =
FXCollections.observableArrayList();
        for(int i=0;i<10;i++)
            liste.add(Integer.toString(i));
        cmb_liste.setItems(liste);
        lst_list.setItems(liste);
    }

    private void basildi(ActionEvent event) {
```



```
}

@FXML
private void degisti(ActionEvent event) {
    lbl.setText(cmb_liste.getValue());
}

@FXML
private void secildi(MouseEvent event) {
    lbl.setText(lst_list.getSelectionModel().getSelectedItem());
}

@FXML
private void toplama(ActionEvent event) {
    ObservableList<String> secilen =
        lst_list.getSelectionModel().getSelectedItem();
    int toplam=0;
    for (String e:secilen)
        toplam+=Integer.parseInt(e);
    lbl.setText(String.valueOf(toplam));
}

@FXML
private void secildi(MouseEvent event) {
}

}
```

Uygulama

Radio Button Ayarları

```
ToggleGroup grubum = new ToggleGroup();  
rb1.setToggleGroup(grubum);  
rb2.setToggleGroup(grubum);
```

ToggleGroup grubum = new ToggleGroup(); ile ToggleGroup sınıfından grubum isminde Toggle grup oluşturulur ve radio buttonlar .setToggleGroup(grubum); fonksiyonu ile birbirine bağlanır. Bu sayede bir radio button seçili iken aynı anda başkası seçilemez.

NOT:

Uygulamada yapılan programı ekte bulabilirsiniz.

JavaFX

check Box:

```
public class FXMLDocumentController implements Initializable {

    private Label label;
    @FXML
    private CheckBox chkuyumak;
    @FXML
    private CheckBox chkgezmek;
    @FXML
    private CheckBox chkkitap;
    @FXML
    private CheckBox chkmuzik;
    @FXML
    private Button btnsecilen;

    private void handleButtonAction(ActionEvent event) {
        System.out.println("You clicked me!");
        label.setText("Hello World!");
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void secilen(ActionEvent event) {
        String secili = "";
        if (chkuyumak.isSelected()){
            secili+="Uyumayı ";
        }
        if (chkgezmek.isSelected()){
            secili+="Gezmeyi ";
        }
        if (chkkitap.isSelected()){
            secili+="Kitap Okumayı ";
        }
        if (chkmuzik.isSelected()){
            secili+="Müzik Dinlemeyi ";
        }

        secili += " severim";
        JOptionPane.showMessageDialog(null, secili);
    }
}
```

```
}  
}
```

// burada 4 tane checkBox gelecek ve hangisini seçersek ona göre başka bir pencerede yazı çıkacak.

checkboxboxlara da onAction atanabilir. İşaretlendiğinde onAction'a yazılan fonksiyon çalışır:

```
@FXML  
private void secildi(ActionEvent event) {  
    if (chkmuzik.isSelected())  
        chkmuzik.setText("Secili");  
    else  
        chkmuzik.setText("Secili Değil");  
}
```

Radio Button:

kullanıcıya yine birçok seçenek sunulur fakat farkı sadece bir tanesini seçmeye izin verilir. ToggleGroup yapısı ile kullanılır.

```
@FXML  
private Button btnsecilen;  
@FXML  
private ToggleGroup sevemeyenler;  
@FXML  
private RadioButton rbt;  
@FXML  
private RadioButton rbn;  
@FXML  
private RadioButton rbs;  
@FXML  
private RadioButton rbc;  
@FXML  
private Label lbltext;  
  
@Override  
public void initialize(URL url, ResourceBundle rb) {  
    // TODO  
}  
  
@FXML  
private void secilen(ActionEvent event) {
```

```
        if (rbt.isSelected())
            System.out.println("Tabiki");
        else if (rbc.isSelected())
            System.out.println("Cekimser");
        else if (rbn.isSelected())
            System.out.println("nefret");
        else if (rbs.isSelected())
            System.out.println("sabah");
    }
}
```

Ödev: checkbox ları radio button gibi kullanmaya yarayan programı yazın..

Scrollbar

Min : alabileceği minimum değer

Max: alabileceği maximum değer

Unit Increment : Aldığı değerın kaçar kaçar arttığı veya azaldığı

Value : değer

onValueChange : değiştiğinde yapılacak işlere görev atanır. (~.fxml dosyasından yazabiliriz. Scene Builder da yok.)

```
public class FXMLDocumentController implements Initializable {
```

```
    @FXML
    private Label lbltext;
    @FXML
    private ScrollBar scrr;
    @FXML
    private ScrollBar scrg;
    @FXML
    private ScrollBar scrb;

    @Override
    public void initialize(URL url, ResourceBundle rb) {

    }

    @FXML
    private void degisti(Event event) {
        Color c = Color.rgb((int)scrr.getValue(),
(int)scrg.getValue(),(int)scrb.getValue());
        lbltext.setStyle("-fx-background-
color:"+c.toString().replace("0x","#"));
    }

}
```

Animation

Oracle ın sitesinde animasyonla ilgili detaylı notlar var. Google a JavaFX Animation yazarsanız detaylı bilgi alabilirsiniz.

Birden fazla animasyon oluşturulabilir ve bu animasyonlar paralel veya sıralı çalıştırılabilir.

Transitions

Fade Transition

Görünürlük ayarı yapmaya yarar.

setArcHeight / setArcWeight : köşeleri yuvarlar

```
@FXML
private void baslat(ActionEvent event) {
    FadeTransition ft = new
    FadeTransition(Duration.millis(1000),rect);
    ft.setFromValue(1.0);// hangi value dan (1.0 en görünür)
    ft.setToValue(0.1);// hangi value ya (0.0 saydam)
    ft.setCycleCount(6);// sonsuza kadar devam eder (5) olursa
    2.5 kere yapar.
    ft.setAutoReverse(true);// otomatik tekrar
    ft.play();// oynat
}
```

.fxml Dosyası ---

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

<?import javafx.scene.shape.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

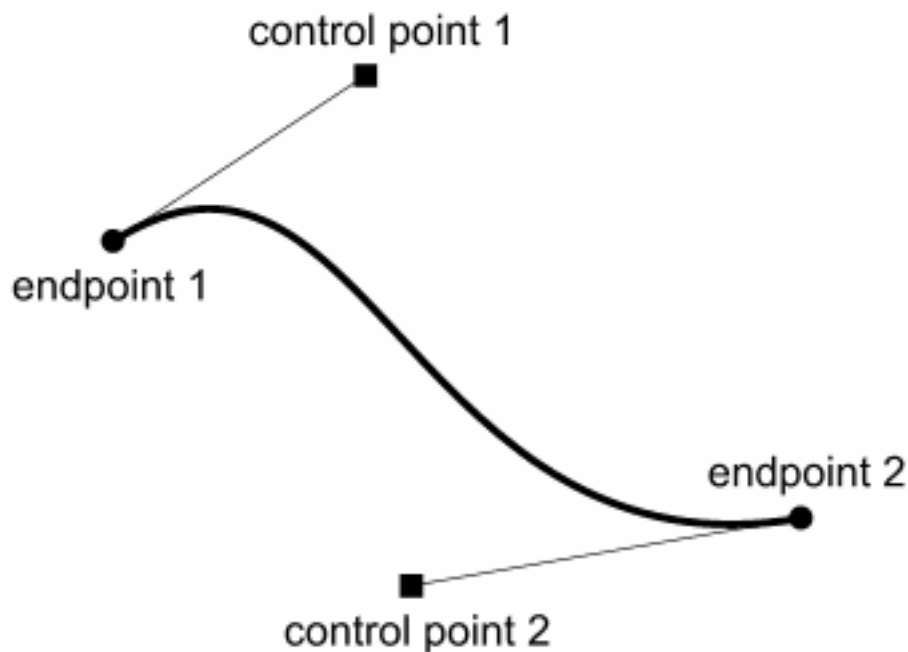
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0"
prefWidth="600.0" xmlns="http://javafx.com/javafx/8"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="animasyon.FXMLDocumentController">
    <children>
        <Rectangle fx:id="rect" arcHeight="20.0" arcWidth="20.0"
```

```
fill="DODGERBLUE" height="90.0" layoutX="78.0" layoutY="27.0"
stroke="BLACK" strokeType="INSIDE" width="90.0" />
    <Button fx:id="basla" layoutX="269.0" layoutY="275.0"
mnemonicParsing="false" onAction="#baslat" text="Başla" />
</children>
</AnchorPane>
```

Path transitions

bir nesneyi bir yerden başka bir yere taşımaya yarar.

cubikcurve Bézier eğrisine göre çalışır..



```
public class FXMLDocumentController implements Initializable {

    @FXML
    private Rectangle rect;
    @FXML
    private Button basla;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }
}
```

```
}

@FXML
private void baslat(ActionEvent event) {
    Path path = new Path(); //yeni bir yol oluşturuldu.
    path.getElements().add(new MoveTo(20,20));
    path.getElements().add(new
CubicCurveTo(380,0,980,120,200,120));
    path.getElements().add(new
CubicCurveTo(0,120,0,240,380,240));
    PathTransition pathTransition = new PathTransition();
    pathTransition.setDuration(Duration.millis(4000)); // süre 4 sn
    pathTransition.setPath(path);
    pathTransition.setNode(rect);

    pathTransition.setOrientation(PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT); // gidilen eğri boyunca taşınan nesneyi yola paralel götürmeyi sağlar. NONE olursa şekil düz bir şekilde yolda ilerler.
    pathTransition.setCycleCount(Timeline.INDEFINITE);
    pathTransition.setAutoReverse(true);
    pathTransition.play();

}

}
```

Uygulama Dosyasını .zip dosyası içerisinde bulabilirsiniz.

Java FX

Scale Transition

```
private void baslat(ActionEvent event) {
    ScaleTransition sc = new ScaleTransition(Duration.millis(2000),
rect);
    sc.setToX(2f);
    sc.setToY(2f);
    sc.setCycleCount(4);
    sc.setAutoReverse(true);
    sc.play();
}
```

// şekli büyütüp küçülten program

RotateTransition

```
private void baslat(ActionEvent event) {
    RotateTransition rc = new RotateTransition(Duration.millis(2000),
rect);
    rc.setByAngle(180f);
    rc.setCycleCount(4);
    rc.setAutoReverse(true);
    rc.play();
}
```

// şekli döndüren program

Parallel Transition

```
FadeTransition ft = new FadeTransition(SEC_3);
ft.setFromValue(1.0f);
ft.setToValue(0.3f);
ft.setCycleCount(2f);
ft.setAutoReverse(true);

TranslateTransition tt = new TranslateTransition(SEC_2);
tt.setFromX(-100f);
tt.setToX(100f);
tt.setCycleCount(2f);
tt.setAutoReverse(true);

RotateTransition rt = new RotateTransition(SEC_3);
rt.setByAngle(180f);
rt.setCycleCount(4f);
rt.setAutoReverse(true);

ScaleTransition st = new ScaleTransition(SEC_2);
st.setByX(1.5f);
st.setByY(1.5f);
st.setCycleCount(2f);
st.setAutoReverse(true);

ParallelTransition pt = new ParallelTransition(rect, ft, tt, rt, st);
```

```
pt.play();
```

Sequential Transition

```
FadeTransition ft = new FadeTransition(SEC_3);  
ft.setFromValue(1.0f);  
ft.setToValue(0.3f);  
ft.setCycleCount(2f);  
ft.setAutoReverse(true);
```

```
TranslateTransition tt = new TranslateTransition(SEC_2);  
tt.setFromX(-100f);  
tt.setToX(100f);  
tt.setCycleCount(2f);  
tt.setAutoReverse(true);
```

```
RotateTransition rt = new RotateTransition(SEC_3);  
rt.setByAngle(180f);  
rt.setCycleCount(4f);  
rt.setAutoReverse(true);
```

```
ScaleTransition st = new ScaleTransition(SEC_2);  
st.setByX(1.5f);  
st.setByY(1.5f);  
st.setCycleCount(2f);  
st.setAutoReverse(true);
```

```
SequentialTransition seqT = new SequentialTransition (rect, pt, ft, tt, rt, st);  
seqT.play();
```

GridPane'e Eleman Atama:

```
Button[] btns = new Button[49];
@Override
public void initialize(URL url, ResourceBundle rb) {

    int row =0, col = 0;
    for (int i=0;i<49;i++){
        btns[i]=new Button(Integer.toString(i+1));
        sahne.add(btns[i], row, col);
        col++;
        if(col==7){
            col =0;
            row++;
        }

    }

}

// 1 den 49 a kadar üzerinde sayı yazan buton oluşturan program
```

ÖDEV: Adam Asmaca oyunu yapılacak. Fakat darağacının her parçası animasyonlu bir şekilde ekrana gelecek.

İpucu:

Elimizde bir dizi olacak. Bu diziden seçilen kelimenin harf sayısı kadar label yazdırılacak.

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package uckasim;

import java.net.URL;
import java.util.Random;
import java.util.ResourceBundle;
import javafx.animation.FadeTransition;
import javafx.animation.ParallelTransition;
import javafx.animation.RotateTransition;
import javafx.animation.ScaleTransition;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
```

```
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.GridPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.util.Duration;

/**
 *
 * @author burak
 */
public class FXMLDocumentController implements Initializable {
    @FXML
    private GridPane sahne;

    Button[] btns = new Button[49];
    @Override
    public void initialize(URL url, ResourceBundle rb) {

        int row =0, col = 0;
        for (int i=0;i<49;i++){
            btns[i]=new Button(Integer.toString(i+1));
            btns[i].setDisable(true);
            sahne.add(btns[i], row, col);
            row++;
            if(row==7){
                row =0;
                col++;
            }

        }

    }

    public boolean varMi(int[] gelen,int sayi){
        int i;
        for (i=0;i<gelen.length;i++){
            if (gelen[i]== sayi)
                break;
        }
        if (i==gelen.length)
            return false;
        else
            return true;
    }

    public int[] sayiUret(){
        int[] sayilar={-1,-1,-1,-1,-1,-1};
        int sayi;
        Random rnd= new Random();
```

```
        for(int i=0;i<6;i++){
            while(true){
                sayi = rnd.nextInt(49)+1;
                if (!varMi(sayilar, sayi)){
                    sayilar[i] = sayi;
                    break;
                }
            }
        }
        return sayilar;
    }

    @FXML
    private void oyna(ActionEvent event) {
        Random rnd = new Random();
        int[] sonuc = sayiUret();
        Color renk = Color.rgb(rnd.nextInt(256), rnd.nextInt(256),
rnd.nextInt(256));
        for (int i=0;i<49;i++){
            btns[i].setDisable(true);
            btns[i].setStyle("-fx-background-color: none");
        }

        for (int i=0;i<sonuc.length;i++){
            btns[sonuc[i]].setDisable(false);
            btns[sonuc[i]].setStyle("-fx-background-color:" +
renk.toString().replace("0x", "#"));
        }
    }

    }

    // Sayısa Loto uygulaması:
    // 1 den 49 a kadar butonlar var bu butonlar arasından rastgele seçilen 6
    buton aktif olur ve //arkaplan renkleri rasgele seçilir.
```

Uygulama Dersinde Yapılan Örnek .zip Dosyası İçerisindedir...

Java FX

Bir Butona derleme anında butona event(onAction) ataması yapmak istersek;

```
btn.setOnAction(this::onClick);
```

fonksiyonunu kullanmamız gerekir. Buradaki `this::onClick` ifadesi, bulunduğu fonksiyon dışındaki `OnClick` isimli fonksiyonu işaret etmektedir.

@FXML

```
private AnchorPane cerceve;
```

@FXML

```
public void onClick(ActionEvent e){  
    Button b = (Button)e.getSource();  
    JOptionPane.showMessageDialog(null,b.getText());  
}
```

@Override

```
public void initialize(URL url, ResourceBundle rb) {  
    String harfler = "ABCÇDEFGĞHIİJKLMNOÖPRSŞTUÜVYZ";  
    int x=50;  
    for (int i=0;i<29;i++){  
        Button btn = new Button(harfler.substring(i,i+1));  
        btn.setOnAction(this::onClick);  
        btn.setLayoutX(x);  
        btn.setLayoutY(50);  
        cerceve.getChildren().add(btn);  
        x+=30;  
    }  
}
```

ImageView

Butona basınca ImageView ögesine tanımlanan resmi gösteren program:

@FXML

```
private AnchorPane cerceve;  
@FXML  
private Button btn_yukle;  
@FXML  
private ImageView img_resim;
```

@Override

```
public void initialize(URL url, ResourceBundle rb) {  
  
}
```

@FXML

```
private void yukle(ActionEvent event) {
    File f = new File("src/img/araba1.png");
    Image araba = new Image(f.toURI().toString()); // dosya yolunu
programın anlayacağı hale getirir.
    img_resim.setImage(araba);
}
```

// paket içine "img" isimli bir klasör oluşturulup resimler bu klasörün içerisine atıldı.

Eğer biz /img klasörü içerisindeki tüm resimleri sırayla görüntülemek istersek ImageView ögesini aşağıdaki gibi kullanabiliriz.

```
File[] dosyalar;
ArrayList<Image> imgList = new ArrayList();
int index = 0;

@Override
public void initialize(URL url, ResourceBundle rb) {
    File dizin = new File("src/img");
    dosyalar = dizin.listFiles();

}

@FXML
private void yukle(ActionEvent event) {
    Image resim = new Image(dosyalar[index].toURI().toString());
    img_resim.setImage(resim);
    index++;
    index = index % dosyalar.length;
}
}
```

// bir dizindeki dosyaların tümünü çağıran ve butona basınca sırayla gösteren program

ColorPicker

```
public class FXMLDocumentController implements Initializable {

    @FXML
    private AnchorPane cerceve;
    @FXML
    private Label lbl_color;
}
```

```
@FXML
private ColorPicker clr_picker;
@FXML
private ColorPicker clr2;

@Override
public void initialize(URL url, ResourceBundle rb) {

}

private void yukle(ActionEvent event) {

}

@FXML
private void renkDegis(ActionEvent event) {
    lbl_color.setStyle("-fx-background-
color:"+clr_picker.getValue().toString().replace("0x", "#")); // bu kısım
arkaplanı değiştiren kısım

    lbl_color.setTextFill(Color.web(clr2.getValue().toString().replace("0x",
"#"))); //bu kısım ise yazının rengini değiştiren kısım
}
}

// 2 tane colorpicker atandı clr2 den yazının rengi, clr_picker dan
arkaplan rengi seçilir.
```

Hyperlink

hyperLink elemanı iki şekilde kullanılabilir.

1 → throws IOException yaparak

2 → try-catch kullanarak

```
@FXML
private AnchorPane cerceve;
@FXML
private Hyperlink hyp;

@Override
```



```
public void initialize(URL url, ResourceBundle rb) {  
  
}
```

```
//--> 1
```

```
@FXML  
private void linkAc(ActionEvent event){  
    Runtime rt =Runtime.getRuntime();  
    try{  
        rt.exec("xdg-open"+hyp.getText());// bağlantıyı varsayılan  
tarayıcıdan açan linux komutu (!!! WINDOWS DA ÇALIŞMAZ !!!)  
    }  
    catch(Exception e){  
        System.out.println(e.getMessage());  
    }  
}  
}
```

```
//-->2
```

```
@FXML  
private void linkAc(ActionEvent event) throws IOException{  
    Runtime rt =Runtime.getRuntime();  
}  
}
```

google chrome da açmak için

```
rt.exec("google-chrome "+hyp.getText());
```

komutu verilmeli..

mouse un gittiği yere butonun gitmesi için anchorpane in mouseMoved kısmına fonksiyon tanımlanıp mouse un x ve y kordinatları alınmalı..

@FXML

```
private void gezdir(MouseEvent event) {  
    btn_gezgin.setLayoutX(event.getX());  
    btn_gezgin.setLayoutY(event.getY());  
}
```

koduyla ayarlama yapılabilir.

Aynısının klavye sağ sol tuşlarıyla yapılmış şekli:

Buradaki fark anchorpane in mouseMoved kısmını değil OnKeyPressed kısmını doldurmamız gerekir.

@FXML

```
private void gezdir(KeyEvent event) {  
    if (event.getCode() == KeyCode.UP)  
        btn_gezgin.setLayoutY(btn_gezgin.getLayoutY() - 5);  
    else if(event.getCode() == KeyCode.DOWN)  
        btn_gezgin.setLayoutY(btn_gezgin.getLayoutY() + 5);  
    else if(event.getCode() == KeyCode.LEFT)  
        btn_gezgin.setLayoutX(btn_gezgin.getLayoutX() - 5);  
    else if(event.getCode() == KeyCode.RIGHT)  
        btn_gezgin.setLayoutX(btn_gezgin.getLayoutX() + 5);  
}
```

Uygulama Dersinde Yapılan Örnek .zip Dosyasını İçerisindedir..

Java FX

ID vermeden bir elemana ulaşma

.FXML dosyasından id vermeden belirli bir elemana ulaşmak için aşağıdaki kod kullanılabilir. Çalışma mantığı container elemanının içerisindeki elemanlar kontrol edilir ve istenilen bir eleman türünün özelliği değiştirilebilir.

```
@FXML
private void bul(ActionEvent event) {
    for (int i=0;i<container.getChildren().size();i++){
        if (container.getChildren().get(i) instanceof CheckBox) //
sıradaki eleman bir CheckBox örneği ise

        ((CheckBox)container.getChildren().get(i)).setDisable(true);
    }
}
```

burada container id'sine sahip anchorPane elemanının içerisindeki CheckBoxlara disable özelliği eklenmiş.

Grafik çizimleri

Pie Chart (Pasta Grafiği)

```
@FXML
private void çiz(ActionEvent event) {
    ObservableList<PieChart.Data> data =
FXCollections.observableArrayList();
    data.add(new PieChart.Data("Portakal", 35));
    data.add(new PieChart.Data("Elma", 13));
    data.add(new PieChart.Data("Kiraz", 32));
    data.add(new PieChart.Data("Çilek", 20));

    piechart.setData(data);
    //piechart.setLegendSide(Side.LEFT); // bölümlerin renklerinin ne
olduklarını belirten iconların yerini belirler.
    //piechart.setLabelLineLength(50); // grafik ismini belirten
çizgilerin boylarını ayarlar
    piechart.setClockwise(false); // tekrar tıklandığında dönüş
yönünü belirtir.. saatYönü/Tersi
// aşağıdaki kısım ise üzerine tıklanan bölümün yüzde değerini verir.
    for (PieChart.Data piedata : piechart.getData()){
        piedata.getNode().addEventHandler(MouseEvent.MOUSE_CLICKED,
            new EventHandler<MouseEvent>(){
                @Override
                public void handle (MouseEvent event){
                    JOptionPane.showMessageDialog(null,
String.valueOf(piedata.getPieValue()));
                }
            });
    }
}
```

Line Chart(Çizgi Grafiği)

.fxml dosyası içerisinde x axis bilgilerini değiştirmemiz gerekmekte

```
<LineChart fx:id="bk" layoutX="28.0" layoutY="172.0" prefHeight="155.0"
prefWidth="194.0">
    <xAxis>
        <CategoryAxis side="BOTTOM" />
    </xAxis>
    <yAxis>
        <NumberAxis side="LEFT" />
    </yAxis>
</LineChart>
```

Burada CategoryAxis Yazı şeklinde bir değeri desteklerken NumberAxis ise sayı cinsinden bir değeri destekler.

@FXML

```
private LineChart<Number,Number> linechart;
```

@FXML

```
private void ciz(ActionEvent event) {
    linechart.getXAxis().setLabel("Aylar");
    linechart.getYAxis().setLabel("Borsa Endeksi");
    linechart.setTitle("2010 Borsa Verileri");

    XYChart.Series veriler = new XYChart.Series();
    veriler.setName("Borsa");
    veriler.getData().add(new XYChart.Data(1,23));
    veriler.getData().add(new XYChart.Data(2,12));
    veriler.getData().add(new XYChart.Data(3,45));
    veriler.getData().add(new XYChart.Data(4,78));
    veriler.getData().add(new XYChart.Data(5,25));
    veriler.getData().add(new XYChart.Data(6,10));
    veriler.getData().add(new XYChart.Data(7,25));
    veriler.getData().add(new XYChart.Data(8,32));
    veriler.getData().add(new XYChart.Data(9,79));
    veriler.getData().add(new XYChart.Data(10,58));
    veriler.getData().add(new XYChart.Data(11,32));
    veriler.getData().add(new XYChart.Data(12,21));
    linechart.getData().add(veriler);
}
```

Uygulama

Pencereye görsellik tanımlamak için bir tane .css dosyası içerisinde arkaplan tanımlanır ve bu arkaplan main.java dosyası içerisine import edilir.

```
@Override
public void start(Stage stage) throws Exception {
    Parent root =
FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

    Scene scene = new Scene(root);

scene.getStylesheets().addAll(this.getClass().getResource("arkaplan.css")
.toExternalForm());

    stage.setScene(scene);
    stage.show();
}

// Mavi kısım sonradan eklenmiştir. Diğer kısımlar proje oluşturulduğu
anda otomatik oluşturulur.
```

Derste yapılan Uygulama Dosyası .zip Dosyası içerisindedir.

Java FX - MySQL

UNIQUE KEY olarak işaretlenen elemandan sadece bir defa yazılabilir. Fakat Primary Key değildir.

```
mysql < employees.sql -u root -p //
```

*** employees.sql dosyası zip dosyası içerisindedir.

bu ifade elimizde hazır bulunan bir sql dosyasını bizim mysql veritabanımıza eklememizi sağlar.

bir veya birden fazla tablodan belirli şartları görmek istersek SELECT kullanılır.

SELECT alan veya alan isimleri **FROM** hangi table/table isimleri
// en temel mantık budur gerisi tercihe bağlıdır.

WHERE şart veya şartlar **ORDER BY** sıralanacak alan veya alanlar.

```
SELECT * FROM departments;
```

```
SELECT dept_no FROM departments;
```

* → tüm alanlar demek..

dept_no → sadece dept_no bilgilerini getir.

```
SELECT * FROM departments WHERE dept_no = 'd001';
```

dept_no = d001 olan kaydın tüm bilgilerini getir.

```
SELECT *FROM employees WHERE emp_no >=10001 and emp_no<=10025;
```

emp_no 1 ile 25 arasındaki elemanların tüm bilgilerini getir.

```
SELECT *FROM employees WHERE emp_no between 10001 and 10025;
```

üsttekinin aynısı sadece farklı yazılımı.

LIKE komutu

sadece metinler üzerinde çalışır.

```
field_name like 'character sequence%' → ile başlayan
```

```
'%character sequence' → ile biten
```

```
'%character sequence%' → içerisinde geçen
```

```
SELECT *FROM employees WHERE first_name like 'Pa%'; → Pa ile başlayanları listele.
```

```
SELECT *FROM employees WHERE first_name like '_a_____'; → 5 harfli ikinci harfi a olanları getir.
```

```
SELECT * FROM employees WHERE first_name in ('Paris','Padma'); ismi paris ve padma olanlar getir..
```

not in yazarsak olmayanları getir demek.

LIMIT Komutu

SELECT * FROM employees LIMIT 0,100;
ilk 100 kaydı getirir. Index mantığıyla çalışır.

ilk 100 kayıt için ;

SELECT * FROM employees LIMIT 100;

komutu da kullanılabilir.

SELECT * FROM employees WHERE emp_no BETWEEN 10001 AND 10100 ORDER BY first_name;

ilk 100 ün ismini harf sırasına koyar.

DESC koyarsak sona ters sıralar.

AGG FUNCTIONS

COUNT : sorgudan etkilenen kayıt sayısı
SUM(alan) : o alandaki toplamı verir.
AVG(alan) : o alandaki aritmetik ortalamayı verir.
MIN(alan) : o alandaki en küçüğü verir
MAX(alan) : o alandaki en büyüğü verir.
STDDEV(alan): o alandaki elemanların standart sapmasını verir.

SELECT COUNT(*) FROM employees;

kaç tane eleman var onu gösterir.

(*) komutu Primary key i gösterir. Yoksa direk alanı yazmamız gerekir.

SELECT SUM(salary) FROM salaries; şirketin verdiği toplam maaşı gösterir.

AVG yerine

sum(salary)/count(*) as ortalama from salaries
yazarsak da aynısını yapar.

as komutu yeni oluşturulacak tablonun ismini belirtir.

Uygulama

```
mysql> create table kitap(
  -> kitapNo int not null auto_increment Primary Key,
  -> kitapAdi varchar(30),
  -> yazari varchar(15),
  -> yayinEvi varchar(15),
  -> fiyat int);

mysql> CREATE TABLE odunc(
  -> oduncNo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> uyeNo INT,
  -> kitapNo INT,
  -> CONSTRAINT fk_uyeNo FOREIGN KEY (uyeNo) REFERENCES uye(uyeNo),
  -> CONSTRAINT fk_kitapNo FOREIGN KEY (kitapNo) REFERENCES kitap(kitapNo)
  -> );

mysql> create table kitap(
  -> kitapNo int not null auto_increment Primary Key,
  -> kitapAdi varchar(30),
  -> yazari varchar(15),
  -> yayinEvi varchar(15),
  -> fiyat int);

mysql> INSERT INTO kitap VALUES('1', 'Kim Korkar Javadan', 'Ahmet', 'Papatya', '30');
Query OK, 1 row affected (0,06 sec)

mysql> INSERT INTO kitap(kitapAdi,yazari,yayinEvi,fiyat) VALUES('C
Programlama','Mehmet','Kodlab','20');
Query OK, 1 row affected (0,04 sec)

mysql> INSERT INTO kitap(kitapAdi,yazari,yayinEvi,fiyat) VALUES('Python Programlamaya
Giris','Veli','Abakus','20');
Query OK, 1 row affected (0,03 sec)

mysql> INSERT INTO kitap(kitapAdi,yazari,yayinEvi,fiyat) VALUES('Hacking
Interface','Hamza','Kodlab','25');
Query OK, 1 row affected (0,04 sec)

mysql> INSERT INTO kitap(kitapAdi,yazari,yayinEvi,fiyat) VALUES('Sosyal
Muhendislik','Kevin','Papatya','45');
Query OK, 1 row affected (0,03 sec)

mysql> select * from kitap;
+-----+-----+-----+-----+-----+
| kitapNo | kitapAdi | yazari | yayinEvi | fiyat |
+-----+-----+-----+-----+-----+
| 1 | Kim Korkar Javadan | Ahmet | Papatya | 30 |
| 2 | C Programlama | Mehmet | Kodlab | 20 |
| 3 | Python Programlamaya Giris | Veli | Abakus | 20 |
| 4 | Hacking Interface | Hamza | Kodlab | 25 |
| 5 | Sosyal Muhendislik | Kevin | Papatya | 45 |
+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql> INSERT INTO uye(uyeNo,adi,soyadi,yasi) VALUES('1','Burak','Kiymaz','21');
Query OK, 1 row affected, 1 warning (0,04 sec)
```



```
mysql> INSERT INTO uye(adi,soyadi,yasi) VALUES('Burak','Aydin','23');
Query OK, 1 row affected (0,04 sec)
```

```
mysql> INSERT INTO uye(adi,soyadi,yasi) VALUES('Nurdan','Aydin','55');
Query OK, 1 row affected (0,04 sec)
```

```
mysql> INSERT INTO uye(adi,soyadi,yasi) VALUES('Burcu','Altinok','20');
Query OK, 1 row affected (0,03 sec)
```

```
mysql> select * from uye;
```

uyeNo	adi	soyadi	yasi
1	Burak	Kiyamaz	21
2	Burak	Aydin	23
3	Nurdan	Aydin	55
4	Burcu	Altinok	20

```
4 rows in set (0,00 sec)
```



JAVA FX**Tek MySQL komutu ile diğer tablolardan veri çekmek**

```
mysql> SELECT e.first_name, e.last_name, d.dept_name, de.from_date
FROM departments d, employees e, dept_emp de WHERE
d.dept_no=de.dept_no AND de.emp_no=e.emp_no LIMIT 30;
```

first_name	last_name	dept_name	from_date
Mary	Sluis	Customer Service	1990-01-22
Huan	Lortz	Customer Service	1989-09-20
Basil	Tramer	Customer Service	1992-05-04
Breanna	Billingsley	Customer Service	1992-11-11
Jungsoo	Syrzycki	Customer Service	1992-03-21
Sreekrishna	Servieres	Customer Service	1989-06-29
Yuichiro	Swick	Customer Service	1998-05-01
Chikara	Rissland	Customer Service	1988-03-03
Kayoko	Valtorta	Customer Service	1985-09-08
Babette	Lamba	Customer Service	1988-06-06
Maren	Hutton	Customer Service	1985-02-18
Abdulah	Thibadeau	Customer Service	1996-10-23
Jagoda	Braunmuhl	Customer Service	2000-05-31
Brendon	Lenart	Customer Service	1994-12-22
Mechthild	Bonifati	Customer Service	1996-08-11
Mihalis	Lowrie	Customer Service	1997-12-26
Yuping	Alpin	Customer Service	1996-02-24
Jackson	Kakkad	Customer Service	1994-10-06
Genta	Kolvik	Customer Service	1993-03-31
Kellie	Chinen	Customer Service	1995-12-14
Shaowen	Desikan	Customer Service	1998-09-09
Susanna	Vesel	Customer Service	1994-08-15
Barton	Jumpertz	Customer Service	1999-07-11
Saddek	Gopalakrishnan	Customer Service	1994-12-24
Falguni	Erie	Customer Service	1998-02-15
Toshimori	Bahi	Customer Service	1999-11-07
Heon	Ranai	Customer Service	1999-12-29
Oksana	Brodie	Customer Service	1991-09-28
Willard	Rosin	Customer Service	1995-04-25
Shirish	Dredge	Customer Service	1999-05-07

30 rows in set (0,09 sec)

“FROM departments d, employees e” komutunda ‘departments d’ bölümü bundan sonra ‘departments’ yerine ‘d’ kullanılacağını belirtir. ‘employees’ için de aynı olay geçerlidir.

20000 nolu işçi şirkette hangi unvanlarla (title) çalışmış?

```
mysql> SELECT e.emp_no, concat(e.first_name," ",e.last_name) as
adsoy, t.title FROM employees e, titles t WHERE e.emp_no = t.emp_no
AND e.emp_no=20000;
```

```
+-----+-----+-----+
| emp_no | adsoy          | title          |
+-----+-----+-----+
| 20000  | Jenwei Matzke  | Senior Engineer |
+-----+-----+-----+
1 row in set (0,08 sec)
```

"d002" numaralı departmana müdürlük yapan kişilerin isim ve soyisimleri nedir?

```
mysql> SELECT e.emp_no, concat(e.first_name," ",e.last_name) as
adsoy FROM employees e, dept_manager dm WHERE dm.emp_no = e.emp_no
AND dm.dept_no='d002';
```

```
+-----+-----+
| emp_no | adsoy          |
+-----+-----+
| 110085 | Ebru Alpin     |
| 110114 | Isamu Legleitner |
+-----+-----+
2 rows in set (0,03 sec)
```

Nested Query(iç içe Sorgu)**Şirketteki çalışanlardan kaç tanesi şirkette hiç müdürlük yapmamış?**

```
SELECT (sonuç yazılacak) COUNT(*) FROM employees WHERE emp_no NOT IN
(SELECT DISTINCT emp_no FROM dept_manager)
```

önce müdürlük yapanların "employees no"ları getiriyor ve aynı olanlar çıkarıyor. Yani müdürlük yapanları getirmiş oluyor.

Sonra tüm elemanlardan müdürlük yapanların sayısını çıkarıyor(not in ile).

Fabrika tarihinde en yüksek maaşı alan kişiyi bulan sorgu?

```
SELECT first_name, last_name FROM employees WHERE emp_no IN (SELECT
emp_no FROM salaries WHERE salary=(SELECT MAX(salary) FROM
salaries));
```

```
+-----+-----+
| first_name | last_name |
+-----+-----+
| Tokuyasu   | Pesch     |
+-----+-----+
```

Ve bu kişinin maaşı?

```
SELECT first_name, last_name, salary FROM employees e, salary s
WHERE e.emp_no=s.emp_no AND salary=(SELECT MAX(salary) FROM
salaries);
```

Bu şirketteki bay bayan sayısını bulan sorgu?

```
select gender, count(*) from employees group by gender;
```

```
+-----+-----+
| gender | count(*) |
+-----+-----+
| M      | 179973   |
| F      | 120051   |
+-----+-----+
```

Ders içerisinde yapılan uygulama

FXMLDocumentController.java Dosyası:

```
public class FXMLDocumentController implements Initializable {

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void sorgula(ActionEvent event) {
        mysqlConnection vt= new mysqlConnection();
        if (!vt.Connect()){
            JOptionPane.showMessageDialog(null, "Bağlatı Yok");
            return; // hata olursa bunun altındaki satırları çalıştırmaz...
        }
        try{
            Statement st = (Statement)vt.getCon().createStatement();
            ResultSet rs = st.executeQuery("Select count(*) from "+
tf_gir.getText());
            rs.next();
            lbl_goster.setText(String.valueOf(rs.getInt(1)));
        }
        catch(SQLException ex){
            System.out.println("Hata " + ex.getMessage());
        }
        finally{
            vt.disconnect();
        }
    }

}
```

mysqlConnection.java Dosyası

```
public class mysqlConnection {
    private Connection con;
    private String user= "root";
    private String pass= "bnmysql";
    private String database_url = "jdbc:mysql://localhost:3306/employees";
    public boolean Connect(){
        try{
            Class.forName("com.mysql.jdbc.Connection");
            con = (Connection)DriverManager.getConnection(database_url,
user, pass);
            return true;
        }
        catch(Exception e){
            return false;
        }
    }
    public boolean disconnect(){
        try{
            getCon().close();
            return true;
        }
        catch(Exception e){
            return false;
        }
    }

    /**
     * @return the con
     */
    public Connection getCon() {
        return con;
    }
}
```



Önemli

Eğer mysql veri tabanını kullanmak istiyorsak "Add Library" kısmından "MySQL JDBC Driver" in import edilmesi gerekmektedir...

JAVA – MySQL CONNECTION

Kurucu Metodun Kuralları:

- Public olacak
- Hiçbir geri dönüş içermeyecek ve
- Parametre alacak ve
- Class ismiyle aynı isimde olacak

DML (Data Manipulation Language)

INSERT, UPDATE, DELETE, SELECT, MERGE, CALL, EXPLAIN PLAN, LOCK TABLE

DDL (Data Definition Language)

CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME

STATEMENT sorgu için kullanıyorsak sorguya parametre gönderilmez. Parametrik çalışacaksa;
Örneğin:

```
INSERT INTO employees(emp_no, first_name, last_name, birth_date, hire_date) VALUES (?, ?, ?, ?, ?);
```

? dışarıdan girilecek bilgiyi gösterir. preparedStatement kullanılır.

Eğer auto-increment özelliği verilmemiş bir özelliği (örnek emp_no) artırmak istersek max emp_no komutu ile en büyük emp_no alınır ve ekleyeceğimiz değere bunun bir fazlasını veririz.

Veri Tabanına Kişi Ekleme

Eleman Ekle

Doğum Tarihi	<input type="text"/>
Adı	<input type="text"/>
Soyadı	<input type="text"/>
Cinsiyet	<input type="radio"/> M <input type="radio"/> F
İşe Başlama Tarihi	<input type="text"/>

Kaydet

DatabaseConnection.java dosyası...

```

public class DatabaseConnection {
    private Connection con;
    private String user= "root";
    private String pass= "bnmsql";
    private String database_url = "jdbc:mysql://localhost:3306/employees";
    public boolean Connect(){
        try{
            Class.forName("com.mysql.jdbc.Connection");
            con = (Connection)DriverManager.getConnection(database_url, user, pass);
            System.out.println("Database Connected...");
            return true;
        }
        catch(Exception e){
            System.out.println("Can not connect database...");
            return false;
        }
    }

    public boolean disConnect(){
        try{
            getCon().close();
            System.out.println("Database Disconneted...");
            return true;
        }
        catch(Exception e){
            System.out.println("Can not disconnect database...");
            return false;
        }
    }

    /**
     * @return the con
     */
    public Connection getCon() {
        return con;
    }
}

```

EmployeeController.java dosyası...

```

public class EmployeeController {

    public int getMaxEmpNo(){
        int no;

        DatabaseConnection dbCon = new DatabaseConnection();
        dbCon.Connect();
        try{
            Statement st = (Statement)dbCon.getCon().createStatement();
            ResultSet rs = st.executeQuery("Select max(emp_no) from employees");
            rs.next();
            return rs.getInt(1) +1;
        }
        catch(Exception e){
            System.out.println(e.getMessage());
            return -1;
        }
        finally{
            dbCon.disConnect();
        }
    }

    public boolean insertData(employee employee){
        DatabaseConnection dbCon = new DatabaseConnection();
        if(!dbCon.Connect())
            return false;
        else{
            try{

```

```

        PreparedStatement ps = (PreparedStatement)dbCon.getCon().prepareStatement("INSERT
INTO employees VALUES (?, ?, ?, ?, ?, ?)");
        int empno = getMaxEmpNo();
        if (empno == -1){
            return false;
        }
        ps.setInt(1, empno);
        ps.setDate(2, employee.getBirthdate());
        ps.setString(3, employee.getFirst_name());
        ps.setString(4, employee.getLast_name());
        ps.setString(5, employee.getGender());
        ps.setDate(6, employee.getHire_date());
        ps.executeUpdate();
        return true;
    }
    catch(Exception e){
        System.out.println(e.getMessage());
        return false;
    }
    finally{
        dbCon.disconnect();
    }
}
}
}

```

FXMLDocumentController.java dosyası...

```

@Override
public void initialize(URL url, ResourceBundle rb) {
    // TODO
}

@FXML
private void kaydet(ActionEvent event){
    String cinsiyet=null;

    cinsiyet = gender.getSelectedToggle().toString();
    if (rb_f.isSelected()){
        cinsiyet = "F";
    }
    else if(rb_m.isSelected()){
        cinsiyet = "M";
    }
    employee emp = new employee(0, java.sql.Date.valueOf(txt_dt.getText()),
txt_adi.getText(), txt_sadi.getText(), cinsiyet, java.sql.Date.valueOf(txt_hire.getText()));
    EmployeeController emc = new EmployeeController();
    emc.insertData(emp);
}

```

Employee.java dosyası...

```

public class employee {
    private int emp_no;
    private Date birthdate;
    private String first_name;
    private String last_name;
    private String gender;
    private Date hire_date;

    public employee(int emp_no, Date birth_date, String first_name, String last_name, String
gender, Date hire_date){// kurucu metot
        this.emp_no = emp_no;
        this.birthdate = birth_date;
        this.first_name = first_name;
    }
}

```



```

        this.last_name = last_name;
        this.gender = gender;
        this.hire_date = hire_date;
    }

    /**
     * @return the emp_no
     */
    public int getEmp_no() {
        return emp_no;
    }

    /**
     * @param emp_no the emp_no to set
     */
    public void setEmp_no(int emp_no) {
        this.emp_no = emp_no;
    }

    /**
     * @return the birthdate
     */
    public Date getBirthdate() {
        return birthdate;
    }

    /**
     * @param birthdate the birthdate to set
     */
    public void setBirthdate(Date birthdate) {
        this.birthdate = birthdate;
    }

    /**
     * @return the first_name
     */
    public String getFirst_name() {
        return first_name;
    }

    /**
     * @param first_name the first_name to set
     */
    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }

    /**
     * @return the last_name
     */
    public String getLast_name() {
        return last_name;
    }

    /**
     * @param last_name the last_name to set
     */
    public void setLast_name(String last_name) {
        this.last_name = last_name;
    }

    /**
     * @return the gender
     */
    public String getGender() {
        return gender;
    }
}

```

```
/**
 * @param gender the gender to set
 */
public void setGender(String gender) {
    this.gender = gender;
}

/**
 * @return the hire_date
 */
public Date getHire_date() {
    return hire_date;
}

/**
 * @param hire_date the hire_date to set
 */
public void setHire_date(Date hire_date) {
    this.hire_date = hire_date;
}
}
```

