

YAPILAN İŞ : Buton oluşturarak programın ismi ve arka plan renginin değiştirilmesi ve EXIT butonu ile programın kapatılması.

```
4 ApplicationWindow {
5     id: mainWindow
6     width: 640
7     height: 480
8     visible: true
9     title: qsTr("Hello World")
10
11     Row{
12         id: buttonRow
13         spacing: 5
14         Button{
15             id: button1
16             width: 80
17             height: 80
18             text: 'buton1'
19             onClicked: {
20                 mainWindow.title = "programtitle1"
21                 mainWindow.color = 'black'
22             }
23         }
24         Button{
25             id: button2
26             width: 80
27             height: 80
28             text: 'buton2'
29             onClicked: {
30                 mainWindow.title = "programtitle2";
31                 mainWindow.color = 'purple'
32             }
33         }
34     }
35     Button{
36         id: button3
37         width: 80
38         height: 80
39         text: 'exitButton'
40         onClicked: {
41             Qt.quit();
42         }
43     }
44 }
```

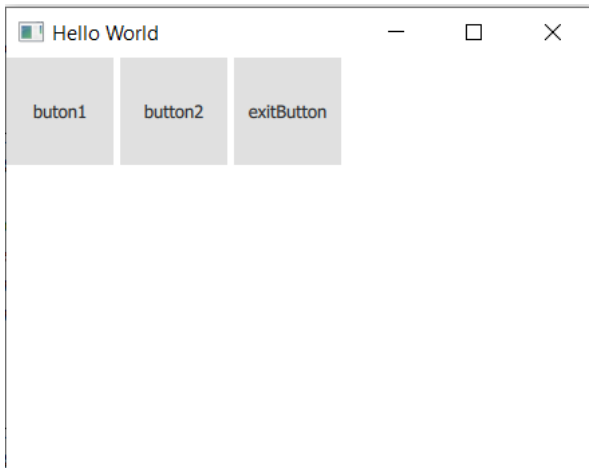
NOT 1 : ROW fonksiyonu oluşturulan öğelerin yan yana oluşturulmasını sağlar.

NOT 2 : SPACING komutu oluşturulan butonların arasındaki mesafe ayarlanmıştır.

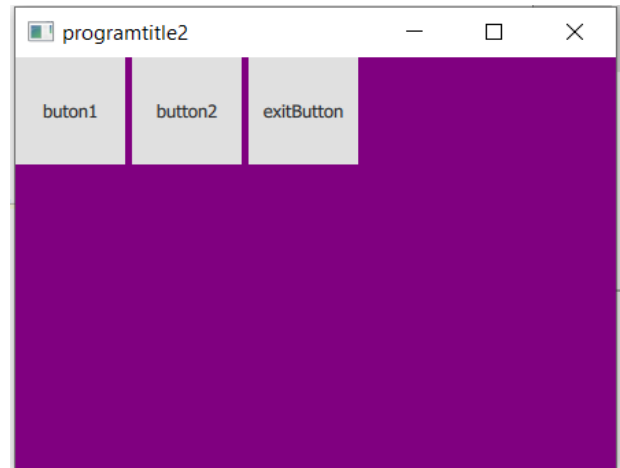
NOT 3 : onClicked komutu ile butona basıldığında ne yapmak isteniyorsa o işlevler girilir. Burada Buton 1 ve Buton 2 de arka plan rengi ve program ismi değiştirilirken Buton 3 ile programdan çıkış uygulanmıştır.

NOT 4 : Burada arka plan rengi değiştirilmek istediğinde ekranın id 'si kullanılarak değiştirilmiştir. Belirli bir alanın rengi değiştirilmek istenirse ilk olarak o alan anchors komutu ile belirlenir. Ardından renk değiştirilmesi yapılabilir.

GÖRSEL 1 -> Program



GÖRSEL 2-> Program açıldığındaki hali

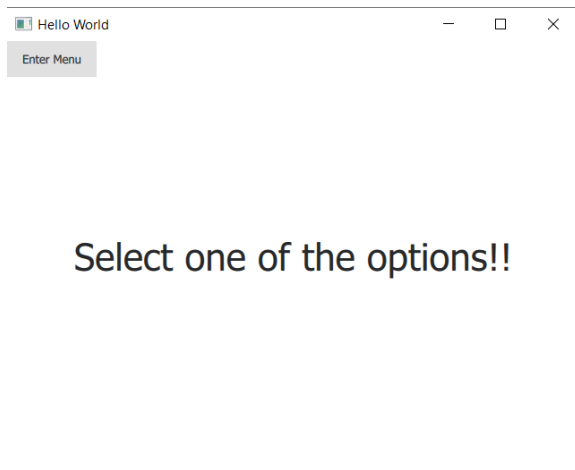


GÖRSEL 3-> Buton 2 basıldığında

YAPILAN İŞ : Menu Oluşturma

```
5 Window {
6   width: 640
7   height: 480
8   visible: true
9   title: qstr("Hello World")
10  Button {
11    id : openmenu
12    text : 'Enter Menu'
13    onClicked: {
14      menu.open();
15    }
16  }
17  Label {
18    id : label
19    anchors.centerIn: parent
20    text: 'Select one of the options!!'
21    font.pointSize: 25
22  }
23  Menu {
24    id: menu
25    y : openmenu.height
26    MenuItem{
27      text: 'Option1'
28      onClicked: {
29        label.text = 'Selected Option 1'
30      }
31    }
32    MenuItem{
33      text: 'Option2'
34      onClicked: {
35        label.text= 'Selected Option 2'
36      }
37    }
38    MenuItem{
39      text: 'Option3'
40      onClicked: {
41        label.text= 'Selected Option 3'
42      }
43    }
44  }
45 }
```

GÖRSEL 4 -> PROGRAM



GÖRSEL 5-> Program Açıldığında

NOT 1 : İlk olarak bir buton oluşturularak bu butona basıldığında menü açılması komutu verilmiştir.

NOT 2 : LABEL fonksiyonu ile ekranda label oluşturulmuştur. 19. Satırda yer alan anchors.centerIn:parent komutu ile oluşturulan label tam merkezde oluşturulmuştur.

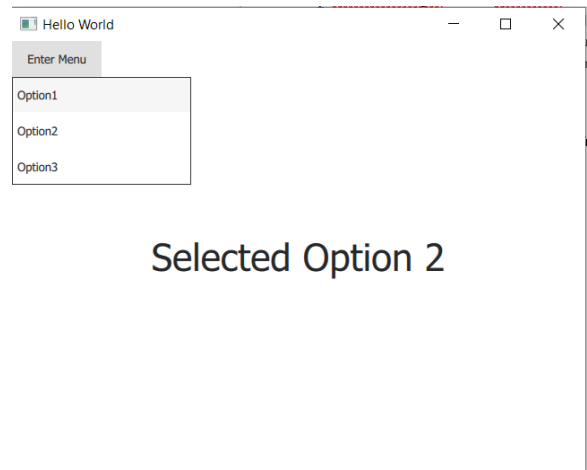
Anchors komutu kullanılacak herhangi bir ögenin bölgesini belirlemek için kullanılır.

Parent komutu ise Item çeşididir. Görsel öğeyle ilgili özellikleri tutar. Burada Label fonksiyonunda nesne belirsiz olduğu için ve daha sonra birden fazla fonksiyon içinde kullanılacağı için parent kullanılmıştır. Kullanılacağı zaman oluşturulan fonksiyonun ID 'si yardımı ile kullanılabilir.

NOT 3 : Menünün oluşturulması klasik olarak tasarlanmıştır. Buradaki önemli nokta 25. Satırdaki

y: openmenu.height komutu openmenu butonu basıldığında bu butonun kaybolmasını engeller. Menu içindeki seçenekler openmenu butonunun altında yer alır. Eğer bu satır olmazsa Enter menü butonuna basıldığında bu buton kaybolur ve onun yerine seçenekler alt alta belirlenir.

NOT 4: Buton yerleri belirlenmediği için sol üstte yer alır.



GÖRSEL 6 -> Option 2 seçilip tekrar Enter Menu butonuna basıldığında

Yapılan İş : Bir Repeater kullanılarak alt satırlarda ilk satırlardaki buton dizilimini oluşturmak.

```
1 import QtQuick 2.9
2 import QtQuick.Controls 2.5
3 import QtQuick.Layouts 1.3
4
5 ▼ ApplicationWindow {
6     width: 640
7     height: 480
8     visible: true
9     title: qsTr("Hello World")
10
11
12 ▼     header :Rectangle{
13         width: parent.width
14         height: 50
15         color: 'blue'
16 ▼         Label {
17             id : mytitle
18             anchors.centerIn: parent
19             color : 'white'
20             font.pointSize: 20
21         }
22     }
23 ▼     ColumnLayout {
24 ▼         anchors {
25             left: parent.left
26             right: parent.right
27             bottom: parent.bottom
28             top: header.bottom
29         }
30
31         spacing: 5
32 ▼         Repeater {
33             model :5
34 ▼             RowLayout{
35                 Layout.fillHeight:true
36                 width: parent.width
37                 height: 10
38 ▼                 Repeater {
39                     model : ListModel{
40 ▼                         ListElement{
41                             mytext:'Button 1'
42                         }
43 ▼                         ListElement{
44                             mytext:'Button 2'
45                         }
46 ▼                         ListElement{
47                             mytext:'Button 3'
48                         }
49 ▼                         ListElement{
50                             mytext:'Button 4'
51                         }
52 ▼                         ListElement{
53                             mytext:'Button 5'
54                         }
55                     }
56 ▼                 Button{
57                     Layout.fillWidth: true
58                     text: mytext
59 ▼                     onClicked: {
60                         mytitle.text = text;
61                     }
62                 }
63             }
64         }
65     }
66 }
```

NOT 1 : 12. Satırda header komutu oluşturulan dikdörtgenin en üstte olmasını sağlar.

NOT 2 : Bu headerı oluşturmak için diğer örneklerdeki import QtQuick.Window silinmelidir ve Window, ApplicationWindow olarak değiştirilmelidir. Son olarak da satır 2 deki QtQuick.Control import edilmelidir.

NOT 3 : Satır 13 'teki width : parent.width satırı ekranın tüm genişliğini kullanır.

NOT 4 : Satır 18 de bulunan anchors.centerIn : parent satırı oluşturulan Label'ın ,Rectangle içinde tam merkezde oluşturulmasını sağlar.

NOT 5 : ColumnLayout oluşturulacak sütunlar için kullanılır.

NOT 6: Satır 28'deki oluşturulacak satırların olduğu bölgenin en üst sınırını en üstte header ile oluşturulan bölgenin alt sınırı bahsedilen bölgenin üst sınırı olması için yazılmıştır.

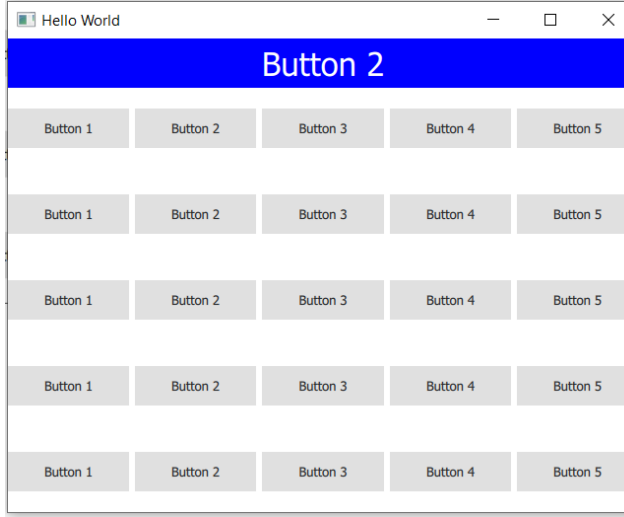
NOT 7 :Satır 33 'deki model ile kaç satır oluşturulacağı belirtilir.

NOT 8 : Satır 35 ve Satır 57 ile oluşturulacak satır ve sütunların bölgenin genişliğine göre tam sığacak şekilde ayarlar.

GÖRSEL 6 -> Repeater Programı



GÖRSEL 7 -> Program Çalıştırıldığında



GÖRSEL 8 -> Herhangi bir Buton 2 'ye basıldığında



GÖRSEL 9 -> Herhangi bir Buton 3 'e basıldığında

YAPILAN İŞ : Programda Slider Bar ve Progress Bar oluşturulmuştur. Bu Slider Bar'daki değer aynı zamanda Text olarak yazdırılmıştır. Bir diğer özellik Slider Bar'daki değer Progress Bar olarak tanımlanmıştır. Bir buton konulmuştur. Bu butonun özelliği o butona basıldığında SliderBar ve ProgressBar 'da 30 değerini göstermiştir. (30 örnek olarak konulmuştur.)

```
1 import QtQuick 2.15
2 import QtQuick.Controls 2.5
3 import QtQuick.Layouts 1.3
4 ApplicationWindow {
5     width: 640
6     height: 480
7     visible: true
8     title: qsTr("Hello World")
9
10    ColumnLayout{
11        anchors.fill: parent
12        ProgressBar {
13            id:myprogressbar
14            Layout.fillWidth: true
15            from:myslider.from
16            to:myslider.to
17            value:myslider.value
18        }
19        Slider{
20            Layout.fillWidth: true
21            id:myslider
22            from:0
23            to:100
24            stepSize: 1
25            value:50
26        }
27        Text{
28            Layout.fillWidth: true
29            text: myslider.value
30            wrapMode: Text.WrapAtWordBoundaryOrAnywhere
31            font.pointSize: 20
32            horizontalAlignment: Text.AlignHCenter
33            verticalAlignment: Text.AlignVCenter
34        }
35        Button{
36            id:sliderbutton
37            text: '30'
38            onClicked: {
39                myslider.value = 30;
40            }
41        }
42    }
43 }
```

NOT 1 : ProgressBar'daki değerler Slider'ın id' si kullanılarak alınır.

NOT 2 : Satır 20 'deki Layout.fillWidth : true satırı tüm genişliği kullanılmasını sağlar.Aynı zamanda ortalamaya da kullanılır.

NOT 3 : Satır 32 ve Satır 33 'deki Allignment komutları yatay ve dikey hizalama için kullanılır.

NOT 4 : wrapMode metin sığmadı zaman ekranın genişlemesi için kullanılır.Ancak burada denenmek için kullanılmıştır. Programda herhangi bir etkisi yoktur.

NOT 5 : Başlangıç değeri öylesine belirlenmiştir. Programdan değiştirilebilir.

NOT 6 : Bu programda asıl amaç Slider Bar ve ProgressBar oluşturmak olduğu için butonla ilgili Ayarlamalar yapılmamıştır.

GÖRSEL 10-> Slider Bar & Progress Bar programı

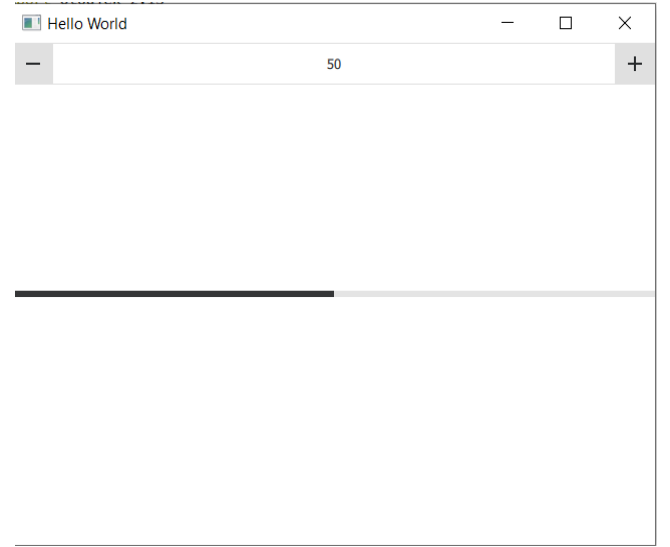


GÖRSEL 11 -> Program Açıldığında

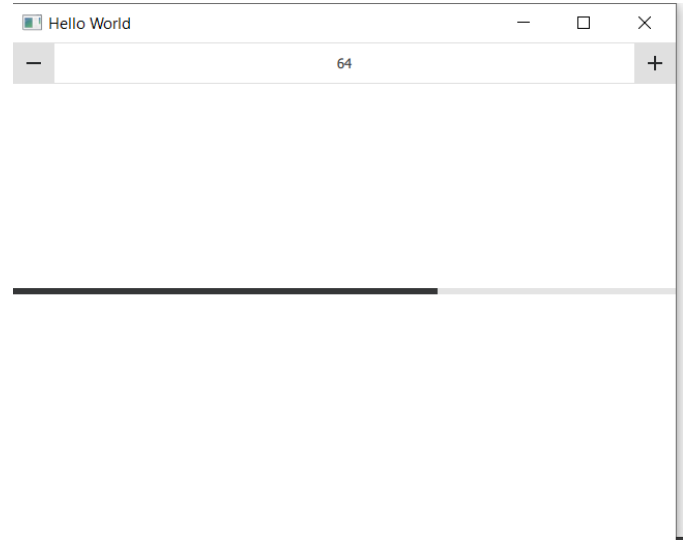
YAPILAN İŞ : SpinBox oluşturulmuştur. Bu SpinBox ile ProgressBar entegre edilmiştir.

```
1  import QtQuick 2.15
2  import QtQuick.Controls 2.5
3  import QtQuick.Layouts 1.3
4
5  ▼ ApplicationWindow {
6      width: 640
7      height: 480
8      visible: true
9      title: qsTr("Hello World")
10     ▼ ColumnLayout{
11         anchors.fill: parent
12         ▼ ProgressBar {
13             id:myprogress
14             Layout.fillWidth: true
15             from:myspinbox.from
16             to:myspinbox.to
17             value:myspinbox.value
18         }
19     }
20     ▼ SpinBox{
21         id:myspinbox
22         from:0
23         to:100
24         stepSize: 1
25         value:50
26         ▼ anchors{
27             left: parent.left
28             right: parent.right
29         }
30     }
31 }
```

GÖRSEL -> 12 SpinBox Programı



GÖRSEL -> 13 Program Açıldığında



GÖRSEL ->14 Değer Değiştirildiğinde

YAPILAN İŞ: Swipe View Oluşturulmuştur

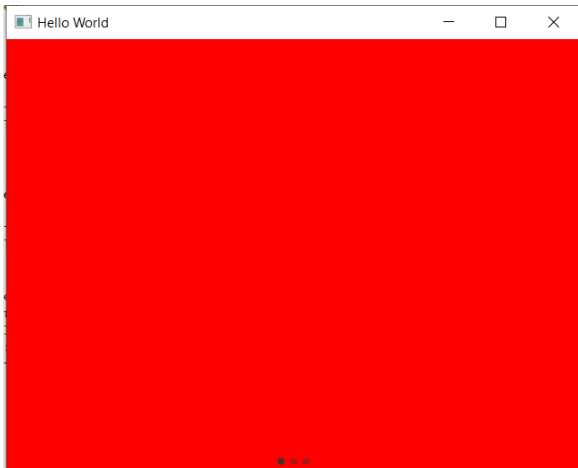
```
5  Window {
6      width: 640
7      height: 480
8      visible: true
9      title: qstr("Hello World")
10     SwipeView {
11         id: view
12         anchors.fill: parent
13         currentIndex: 0
14         Item {
15             id: primerElement
16             Rectangle {
17                 anchors.fill: parent
18                 color: 'red'
19             }
20         }
21         Item {
22             id: seconElement
23             Rectangle {
24                 anchors.fill: parent
25                 color: 'blue'
26             }
27         }
28         Item {
29             id: thirdElement
30             Rectangle {
31                 anchors.fill: parent
32                 color: 'black'
33             }
34             Button {
35                 anchors.centerIn: parent
36                 id: exitButton
37                 text: 'EXIT!!'
38                 onClicked: {
39                     Qt.quit();
40                 }
41             }
42         }
43     }
44     PageIndicator {
45         id: sayac
46         count: view.count
47         currentIndex: view.currentIndex
48         anchors.bottom: view.bottom
49         anchors.horizontalCenter: parent.horizontalCenter
50     }
51 }
```

NOT 1 : Satır 13 'deki ve Satır 47 'deki currentIndex komutları programın alt kısmındaki sayacın ilerlemesini sağlar.

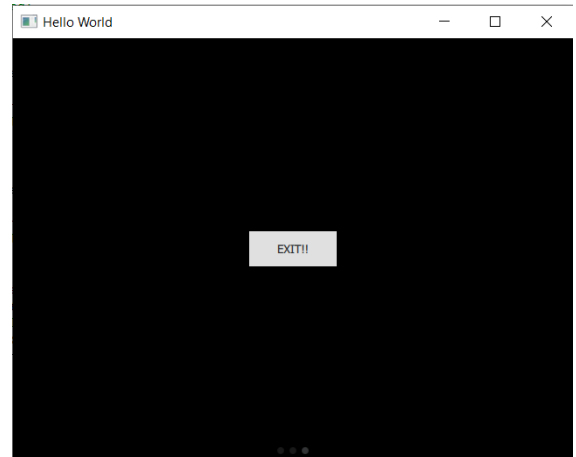
NOT 2 : PageIndicator fonksiyonu sayfanın alt kısmındaki sayaç için oluşturulmuştur.

NOT 3 : SwipeView 'da yazılan Itemlerde tekrar tekrar Rectangle yerine dışarda bir Rectangle oluşturulup Itemlerin içinde o fonksiyon kullanılabilir.

GÖRSEL -> 15 SwipeView Programı



GÖRSEL->16 Program Çalıştırıldığında



GÖRSEL -> 17 3.Swipe ekranı

YAPILAN İŞ : TAB BAR oluşturma

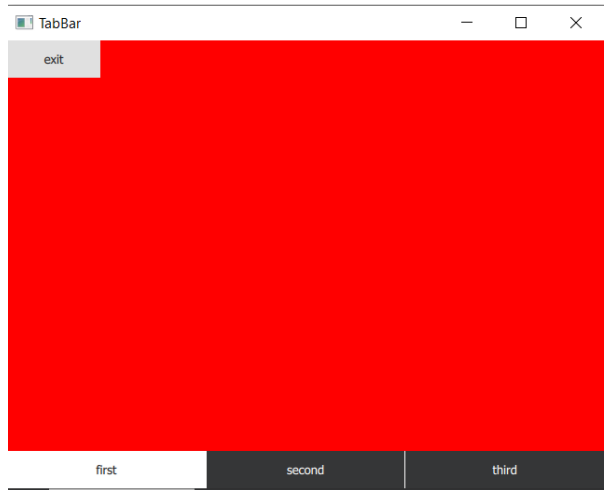
```
5 ApplicationWindow {
6     width: 640
7     height: 480
8     visible: true
9     title: qsTr("TabBar")
10    footer: TabBar{
11        id: bar
12        TabButton{
13            text: 'first'
14        }
15        TabButton{
16            text: 'second'
17        }
18        TabButton{
19            text: 'third'
20        }
21    }
22    StackLayout {
23        anchors.fill: parent
24        currentIndex: bar.currentIndex
25        Item {
26            id: firstTab
27            Rectangle {
28                color: 'red'
29                anchors.fill: parent
30            }
31            Button{
32                id: exitButton
33                text: 'exit'
34                onClicked: {
35                    Qt.quit();
36                }
37            }
38        }
39        Item {
40            id: secondTab
41            Rectangle{
42                color: 'blue'
43                anchors.fill:parent
44            }
45        }
46        Item {
47            id: thirdTab
48            Rectangle{
49                color: 'purple'
50                anchors.fill: parent
51            }
52        }
53    }
54 }
55 }
```

GÖRSEL 18->TabBar programı

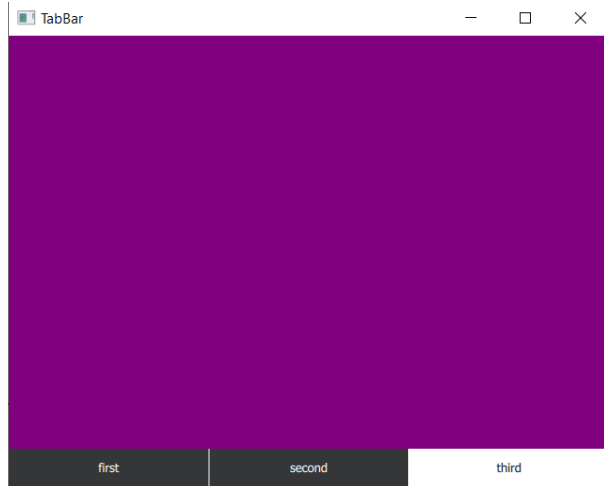
NOT 1 : footer komutu TabBar'ın en alt kısmında oluşmasını sağlar.

NOT 2: Satır 24 'deki currentIndex komutu TabBar da başka bir sekme basıldığında sayfanın değişmesini sağlar. Eğer silinirse Tabbar'da basılsa bile renk değişmez.

NOT 3: StackLayout ie aynı anda 1 ögenin görülebileceği bir Stack oluşturulur.



GÖRSEL 19-> Program Çalıştırıldığında



GÖRSEL 20-> 3. Tab basıldığında