

جواب تمرین جلسه قبل:

```
private void button1_Click(object sender, EventArgs e)
{
    int x;
    Random r = new Random();
    do{
        x = r.Next(55);
        listBox1.Items.Add(x);
    }while(x!=50);
}
```

روش تعریف کردن آرایه ها :

برای تعریف آرایه دو روش وجود دارد:

روش اول : مقدار دهی عناصر آرایه در زمان تعریف آرایه بدون مشخص کردن طول آرایه:

```
string[] student = { "ahmad", "reza"};
```

روش دوم: تعیین طول دقیق آرایه:

```
string[] student = new string[5];
```

تمرین: برنامه ای بنویسید که در یک جعبه متن از کاربر متنی را دریافت کند. این متن میتواند شامل حروف یا اعداد باشد. (مثلا 3m1ah3di) در برنامه شما یک دکمه وجود داشته باشد که در اثر هر بار فشردن آن، از انتهای متن یک کاراکتر حذف شده و بسته به اینکه نوع آن حرف یا عدد است در یک برچسب جداگانه قرار گیرد. (در انتها انتظار می رود که در یک برچسب Mahdi و در دیگری 313 باشد).

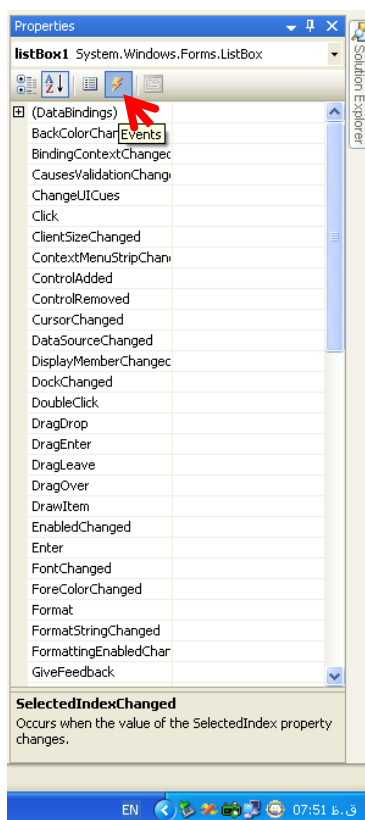
فصل ششم : برنامه های ویندوزی

تعریف رویداد (Event) :

هر رویداد یا حادثه ای که در برنامه اتفاق می افتد ، میتواند منجر به اجرای دستور یا دستوراتی گردد. یعنی میتوان به هر رویداد مجموعه دستوری را منتسب نمود.

در ادامه چند رویداد مهم مطرح شده اند:

نکته : برای مشاهده ی رویدادهای یک ابزار، کافیست پس از انتخاب آن ابزار، در پنل Properties روی علامت زرد رنگ به شکل صاعقه کلیک کنیم.



معرفی چند رویداد مهم:

- ۱- رویداد **click** برای دکمه های فرمان
- ۲- رویداد **MouseEnter** برای دکمه های فرمان : در اثر ورود ماوس به محدوده ی دکمه اتفاق می افتد.
- ۳- رویداد **MouseLeave** برای دکمه های فرمان : در اثر خروج ماوس از محدوده ی دکمه فراخوانی میشود.
- ۴- رویداد **TextChanged** برای جعبه متن: در اثر تغییر متن درون جعبه متن رخ میدهد.
- ۵- رویداد **selectedIndexChanged** برای لیست باکس: در اثر تغییر آیتم انتخاب شده در لیست باکس رخ میدهد.
- ۶- رویداد **Leave** : در اثر ترک کردن تمرکز یا **Focus** کاربر از روی هرکدام از ابزارها رخ میدهد.

مثال : برنامه ای ایجاد کنید که شامل یک دکمه باشد و در اثر وارد شدن ماوس به محدوده ی دکمه و خروج از محدوده ی دکمه، متن دکمه تغییر کند.

```
private void button1_MouseEnter (object sender, EventArgs e)
{
    button1.Text = "شما اینجا هستید";
}
```

```
private void button1_MouseLeave (object sender, EventArgs e)
{
```

```

        button1.Text = "شما رفتید";
    }

```

مثال: تعداد حروف یک متن را به صورت پویا به دست آورید.

```

private void textBox1_TextChanged(object sender, EventArgs e)
{
    string t;
    t = textBox1.Text;
    int x;
    x = t.Length;
    label1.Text = x.ToString();
// label1.Text = textBox1.Text.Length+"کاراکتر";
}

```

مثال فوق را با استفاده از یک متد بازنویسی میکنیم:

```

void shomaresheHarf()
{
    string t;
    t = textBox1.Text;
    int x;
    x = t.Length;
    label1.Text = x + " حرف";
}

```

یک متد دیگر برای شمارش تعداد کلمات متن مینویسیم:

```

void shomaresheWord()
{
    string t;
    t = textBox1.Text;
    string[] str = t.Split(' ');
    int x;
    x = str.Length;
    label1.Text = x + " کلمه";
}

```

حال برای اینکه کاربر بتواند بین شمارش کلمات و شمارش حروف، انتخاب داشته باشد، از دو دکمه ی رادیویی یا Radio Button به نامهای rdbChars و rdbWord به ترتیب برای شمارش حروف و شمارش کلمات استفاده میکنیم.

حال با استفاده از تابعی به نام updateScreen() تشخیص میدهیم که کدامیک از دکمه های رادیویی فعال است و متد مربوطه را فراخوانی میکنیم:

```

void updateScreen()
{
    if (rdbChar.Checked == true)
        shomaresheHarf();
}

```

```

else
    shomaresheWord();
}

```

حال در رویداد تغییر متن مربوط به جعبه متن، کافیسف فقط متد `updateScreen()` را صدا بزنیفم..

```

private void textBox1_TextChanged(object sender, EventArgs e)
{
    updateScreen();
}

```

همچنین برای بهتر کار کردن برنامه در رویداد مربوط به تغییر حالت دکمه های رادیویی مجدداً متد `updateScreen` را صدا میزنیم. بدین منظور روی هر کدام از دکمه های رادیویی دابل کلیک کرده و ..

```

private void rdbChar_CheckedChanged(object sender, EventArgs e)
{
    updateScreen();
}

```

```

private void rdbWord_CheckedChanged(object sender, EventArgs e)
{
    updateScreen();
}

```

کد برنامه به صورت یکجا:

```

private void shomaresheHorooof()
{
    label1.Text = textBox1.Text.Length + " حرف";
}
private void shomaresheKalamat()
{
    string text;
    text = textBox1.Text;
    string[] word = text.Split(' ');
    label1.Text = word.Length + " کلمه";
}
private void update()
{
    if (rdbhorooof.Checked == true)
        shomaresheHorooof();
    else
        shomaresheKalamat();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    update();
}
private void rdbkalamaat_CheckedChanged(object sender, EventArgs e)
{

```

```

        update () ;
    }

    private void rdbhorroof_CheckedChanged(object sender, EventArgs e)
    {
        update () ;
    }

```

مثال: برنامه ای که معدل دو نمره ی یک دانشجو را حساب کند. اگر نمره وارد شده نامعتبر بود، به محض ترک کردن جعبه متن، جعبه ی متن نمره قرمز شده و پیغام مناسبی نمایش داده شود، سپس پس از کلیک کاربر روی پیغام، مجدداً کاربر به جعبه متن مربوط به نمره هدایت شود و جعبه ی متن آماده ی دریافت نمره شود.

بررسی معتبر بودن یک نمره را در رویداد ترک کردن جعبه متن انجام می‌دهیم.. (Leave)

```

private void button1_Click(object sender, EventArgs e)
{
    float x, y, z;
    x = float.Parse(textBox1.Text);
    y = float.Parse(textBox2.Text);
    z = (x + y) / 2;
    MessageBox.Show("معدل = " + z);
}

private void textBox1_Leave(object sender, EventArgs e)
{
    float x;
    x = float.Parse(textBox1.Text);
    if (x < 0 || x > 20)
    {
        textBox1.BackColor = Color.Red;
        MessageBox.Show("نمره وارد شده نامعتبر است");
        textBox1.Text = "";
        textBox1.BackColor = Color.White;
        textBox1.Focus();
    }
}

private void textBox2_Leave(object sender, EventArgs e)
{
    float x;
    x = float.Parse(textBox2.Text);
    if (x < 0 || x > 20)
    {
        textBox2.BackColor = Color.Red;
        MessageBox.Show("نمره وارد شده نامعتبر است");
        textBox2.Text = "";
        textBox2.BackColor = Color.White;
        textBox2.Focus();
    }
}

```

}

تکلیف: متنی را از کاربر دریافت کنید و آن را از آخر به اول دوباره در جعبه متن قرار دهید.

مثال: برنامه‌ای که فضاهای خالی اضافی درون متن را از بین ببرد. می‌باشد.

(ابتدا متن را به کلمات تجزیه کنید، سپس فقط خانه‌هایی از آرایه را که "" نیستند، به هم بچسبانید و در یک متغیر دیگر قرار دهید)

```
private void button1_Click(object sender, EventArgs e)
{
    string s;
    s = textBox1.Text;
    string[] arr = s.Split(' ');
    string jadid="";
    foreach (string a in arr)
    {
        if (a != "")
            jadid += a;
    }
    textBox1.Text = jadid;
}
```

تمرین: نرم افزار هم کد در اندروید را بنویسید. (کاربر شماره تلفن قدیمی خود را وارد کند و شما شماره جدید را به صورت دینامیک به او نمایش دهید).

641528 → 614255 (در اینترنت جستجو شود).

تمرین: همزمان با وارد کردن شماره موبایل توسط کاربر، چهار رقم اول، سه رقم دوم و چهار رقم سوم جدا شوند(یا با space یا -) (کاری شبیه maskedTextBox)

دستور خروج از برنامه: Application.Exit();

دستور بستن فرم جاری: this.Close();

مثال: برنامه ای بنویسید که کاراکتر پنجم از رشته ی موجود در جعبه متن داده شده را از درون آن حذف کرده و در یک messageBox نمایش دهد.

```
string text,harf;
text = textBox1.Text;
harf = text.Substring(4, 1);
MessageBox.Show(harf);
textBox1.Text = text.Substring(0, 4) + text.Substring(5);
```

مثال: برنامه ویرایشگر متن شامل نوار ابزار (tool Strip) و نوار وضعیت (status strip) برای انجام کارهای ساده ای مانند: `about - lowercase -uppercase- blue - red-clear` این ابزارها درون دسته ی `all windows form` می باشند.

بعد از اضافه کردن هر دکمه روی نوار ابزار، میتوان از قسمت `Display Style` نوع نمایش دکمه را انتخاب نمود. همچنین از بخش `text` میتوان متن نمایش داده شده برای هر دکمه را تعیین کرد.

برای کد نویسی کردن هر دکمه، کافیسست روی آن دابل کلیک کرد و کد نویسی را انجام داد:

کد دکمه ی `clear` برای تمیز کردن جعبه متن:

```
textBox1.Text = "";
textBox1.ForeColor = Color.Black;
```

کد دکمه ی `red` برای قرمز کردن رنگ متن:

```
textBox1.ForeColor = Color.Red;
```

کد دکمه ی `blue` برای آبی کردن رنگ متن:

```
textBox1.ForeColor = Color.Blue;
```

کد دکمه ی `UpperCase` برای بزرگ کردن حروف متن :

```
string s;
s = textBox1.Text;
textBox1.Text = s.ToUpper();
```

کد دکمه ی `LowerCase` برای کوچک کردن حروف متن:

```
string s;
s = textBox1.Text;
textBox1.Text = s.ToLower();
```

فراخوانی یک فرم درون یک فرم دیگر (برای بخش `about`)

برای طراحی یک فرم `About`، از مسیر زیر اقدام میکنیم:

در پنجره ی `Solution Explorer` روی نام پروژه راست کلیک کرده و گزینه ی `Add سپس Windows Form` را انتخاب میکنیم.

حال آیتم **About Box** را انتخاب کرده و نام مناسبی برای آن انتخاب نمایید.

برای تغییر محتویات این پنجره، کفیسست روی نام پروژه راست کلیک کرده و گزینه ی **Properties** را انتخاب کنیم. حال در پنجره ی باز شده، روی گزینه ی **Assembly Information** کلیک میکنیم و در پنجره ی باز شده، اطلاعات شرکت خود را ثبت می نماییم.

برای نمایش فرم **About** دستورات زیر را در محل مربوطه قرار میدهم:

```
// تعریف کردن شی a از نوع کلاس AboutBox1
AboutBox1 a = new AboutBox1 ();
// فراخوانی شی ایجاد شده
a.ShowDialog ();
```

نکته: تفاوت دو متد **Show** و **ShowDialog** در این است که اگر از متد **ShowDialog** برای نمایش یک فرم استفاده کنیم، تا زمانیکه فرم جدید را نبسته ایم، اجازه ی دسترسی به فرم قبلی را نداریم ولی اگر از متد **show** استفاده کنیم، همزمان به فرم قدیم و جدید دسترسی خواهیم داشت.

مثال: برنامه ای که شامل بیش از یک فرم باشد و بتوان توسط یکی از فرمها، فرم دیگر را فراخوانی و اجرا نمود.

نکته: برای اضافه کردن یک فرم معمولی، کفیسست از منوی **Project** گزینه ی **Add Windows Form** را کلیک کنیم.

```
private void button1_Click(object sender, EventArgs e)
{
    form11 a = new form11 ();
    a.Show ();
}

private void button2_Click(object sender, EventArgs e)
{
    Form12 b = new Form12 ();
    b.Show ();
}

private void button3_Click(object sender, EventArgs e)
{
    Form13 c = new Form13 ();
    c.Show ();
}
```

نکته: دستور **this.Close()** فقط فرم جاری (فعلی) را می بندد درحالیکه دستور **Application.Exit()** تمام فرمها را میبندد و برنامه را خاتمه میدهد.

نکته: رویداد **FormClosed** مربوط به فرم ها زمانی اجرا میشود که یک فرم بسته میشود.