Windows form

Bài 15

backgroundWorker - progressBar

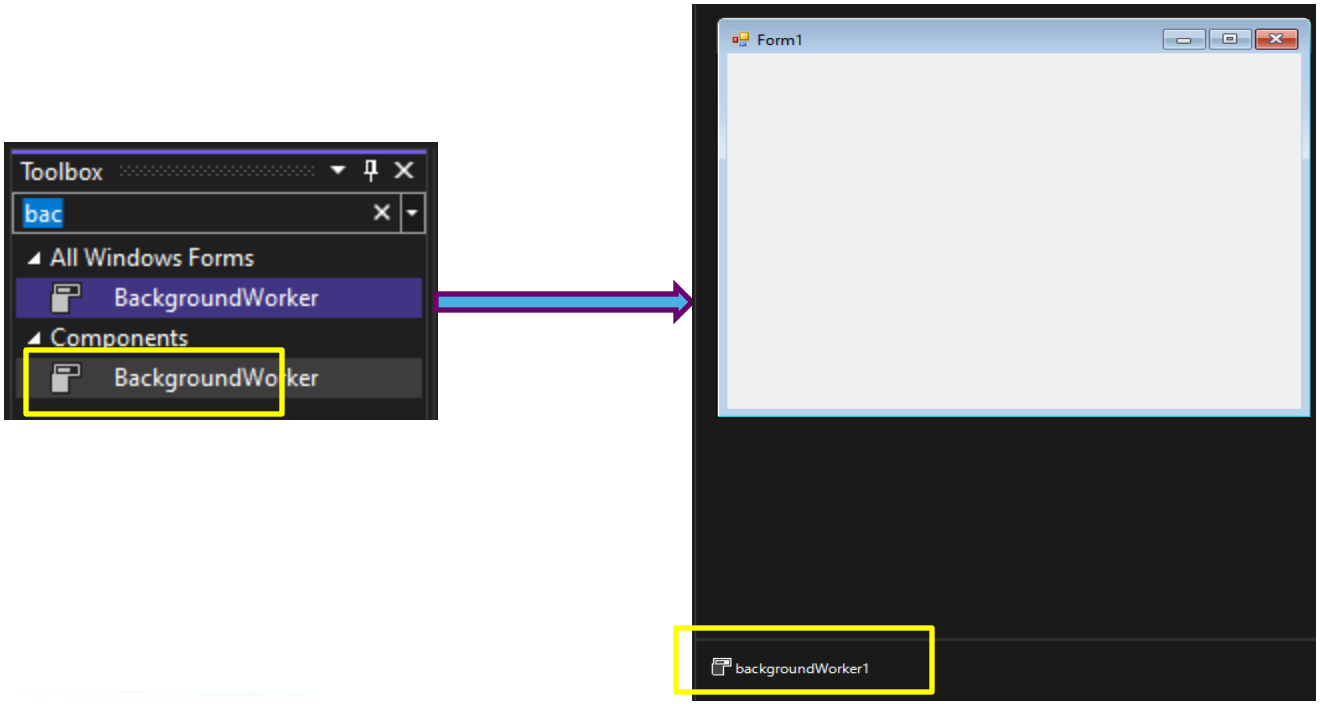
backgroundWorker

❑ backgroundWorker :

✓ *Khi nào cần dùng ? Có những công việc xử lý mất thời gian, trong thời gian phần mềm đang xử lý mà người dùng lại hoạt động các tác vụ khác thì có thể sinh ra lỗi, hoặc treo phần mềm.*

✓ *Dùng tương tác và hiển thị tiến trình xử lý 1 công việc nào đó*

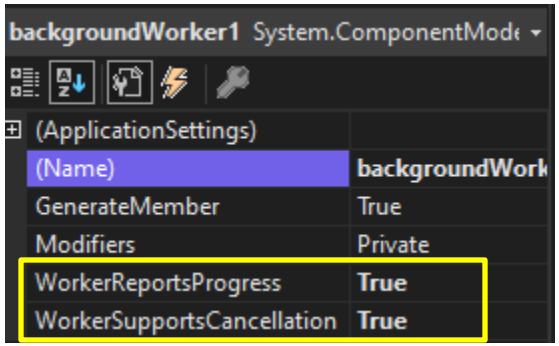
✓ *Ngoài ra ta có thể can thiệp dừng tiến trình khi đang chạy dở*



backgroundWorker

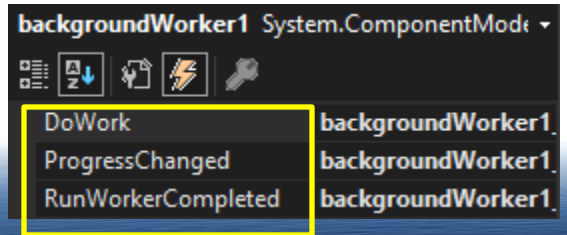
❑ Properties :

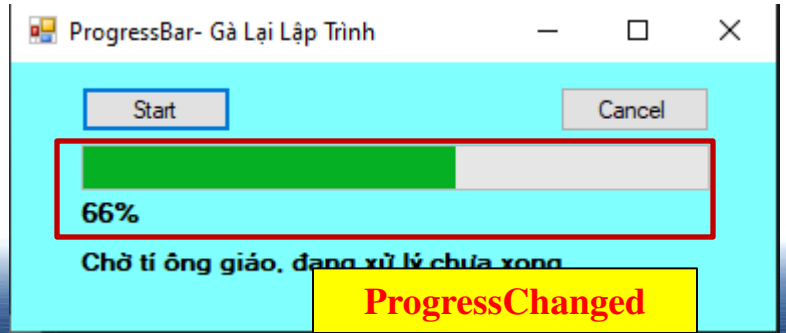
- ✓ *WorkerReportsProgress = true* => *Bật report tiến trình hoạt động nền*
- ✓ *WorkerSupportsCancellation = true* =>*Bật hỗ trợ hủy tiến trình chạy nền*



❑ Event :

- ✓ *DoWork : Event chứa code xử lý công việc chính (tốn thời gian chạy)*
- ✓ *ProgressChanged : Nơi để viết code cập nhật tiến trình xử lý*
- ✓ *RunWorkerCompleted : event này sẽ tự động gọi khi*
  - ✓ *Người dùng bấm cancel tiến trình*
  - ✓ *Quá trình chạy bị lỗi*
  - ✓ *Quá trình thực sự hoàn thành*





backgroundWorker

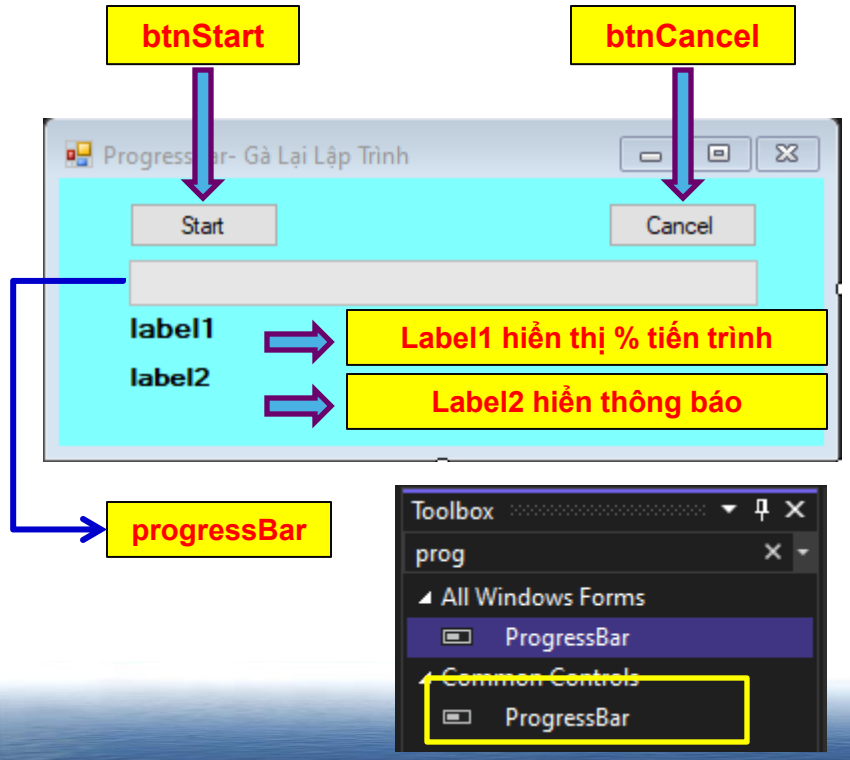
❑ Bài tập vận dụng :

✓ *Tính tổng từ 0 đến 100 và update tiến trình hiển thị quá trình xử lý qua progressBar*

✓ *Start để bắt đầu chạy*

✓ *Cancel để hủy tiến trình*

✓ *Nếu đang xử lý mà bấm lại start thì báo là ứng dụng đang chạy*



backgroundWorker

❑ Bài tập vận dụng :

```
1 reference
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    // công việc cần làm
    int sum = 0;
    for (int i = 0; i <= 100; i++)
    {
        Thread.Sleep(100);
        sum += i;
        //gọi sự kiện ProgressChanged
        backgroundWorker1.ReportProgress(i);

        if(backgroundWorker1.CancellationPending)
        {
            e.Cancel = true;
            backgroundWorker1.ReportProgress(0);
            return;
        }
    }
    e.Result = sum;
}
```

Event DoWork

```
private void backgroundWorker1_RunWorkerCompleted(object sender,
    RunWorkerCompletedEventArgs e)
{
    if(e.Cancelled)
    {
        label2.Text = "Tiến trình bị hủy ông giáo ạ";
    }
    else if (e.Error !=null)
    {
        label1.Text=e.Error.Message;
    }
    else
    {
        label1.Text ="Kết quả của ông giáo là: " +e.Result.ToString();
    }
}
```

Event RunWorkerCompleted

```
1 reference
private void backgroundWorker1_ProgressChanged(object sender,
    ProgressChangedEventArgs e)
{
    progressBar1.Value = e.ProgressPercentage;
    label1.Text=e.ProgressPercentage.ToString()+"%";
}
```

Event ProgressChanged

backgroundWorker

❑ Bài tập vận dụng :

```
private void btnStart_Click(object sender, EventArgs e)
{
    label2.Text = "";
    // check backworker có đang busy hay không ?
    if (!backgroundWorker1.IsBusy) // nếu worker o bận
    {
        backgroundWorker1.RunWorkerAsync();
    }
    else
    {
        label2.Text = "Chờ tí ông giáo, đang xử lý chưa xong";
    }
}
```

Code btnStart

```
1 reference
private void btnCancel_Click(object sender, EventArgs e)
{
    //Nếu worker đang chạy thì CancelAsync để dừng
    if (backgroundWorker1.IsBusy)
        backgroundWorker1.CancelAsync();
    else
    {
        label2.Text = "Em đang không chạy nên không cần hủy ông giáo ạ";
    }

}
```

Code btnCancel