

Дәріс 6. Ағын жұмысын тоқтату. Ағынның күйі. Негізгі ағынмен жұмыс.

Ағындарды тоқтату

Мысал 1. Ағынды тоқтату үшін *Abort()* әдісін қолдану.

```
using System;
using System.Threading;

class MyThread
{
    public Thread Thrd;

    public MyThread(string name)
    {
        Thrd = new Thread(this.Run);
        Thrd.Name = name;
        Thrd.Start();
    }
    // Ағынға кіру нүктесі.
    void Run()
    {
        Console.WriteLine(Thrd.Name + " басталды.");
        for (int i = 1; i <= 1000; i++)
        {
            Console.Write(i + " ");
            if ((i % 10) == 0)
            {
                Console.WriteLine();
                Thread.Sleep(250);
            }
        }
        Console.WriteLine(Thrd.Name + " аяқталды.");
    }
}

class StopDemo
{
    static void Main()
    {
        MyThread mt1 = new MyThread("Ағын 1");
        Thread.Sleep(1000); // туынды ағынның орындалуына рұқсат беру
        Console.WriteLine("Ағын жұмысын тоқтату.");
        mt1.Thrd.Abort();
        mt1.Thrd.Join(); // ағын жұмысының тоқтатылуын күту
        Console.WriteLine("Негізгі ағын жұмысы тоқтатылды.");
    }
}
```

Мысал 2. *Abort()* әдісінің баламалы формасы

```
using System;
using System.Threading;

class MyThread
{
    public Thread Thrd;

    public MyThread(string name)
    {
        Thrd = new Thread(this.Run);
        Thrd.Name = name;
        Thrd.Start();
    }
}
```

```

    }
    // Ағынға кіру нүктесі.
    void Run()
    {
        try
        {
            Console.WriteLine(Thrd.Name + " басталды.");
            for (int i = 1; i <= 1000; i++)
            {
                Console.Write(i + " ");
                if ((i % 10) == 0)
                {
                    Console.WriteLine();
                    Thread.Sleep(250);
                }
            }
            Console.WriteLine(Thrd.Name + " калыпты аяқталды.");
        }
        catch (ThreadAbortException exc)
        {
            Console.WriteLine("Ағын тоқтатылды, аяқтау коды " + exc.ExceptionState);
        }
    }
}

class UseAltAbort
{
    static void Main()
    {
        MyThread mt1 = new MyThread("Ағын 1");
        Thread.Sleep(1000); // туынды ағынның орындалуына рұқсат беру
        Console.WriteLine("Ағын жұмысын тоқтату.");
        mt1.Thrd.Abort(100);
        mt1.Thrd.Join(); // ағын жұмысының тоқтатылуын күту
        Console.WriteLine("Негізгі ағын жұмысы тоқтатылды.");
    }
}

```

Мысал 3. *ResetAbort()* әдісін пайдалану

```

using System;
using System.Threading;

class MyThread
{
    public Thread Thrd;
    public MyThread(string name)
    {
        Thrd = new Thread(this.Run);
        Thrd.Name = name;
        Thrd.Start();
    }

    void Run()
    {
        Console.WriteLine(Thrd.Name + " басталды");
        for (int i = 1; i <= 1000; i++)
        {
            try
            {
                Console.Write(i + " ");
                if ((i % 10) == 0)
                {

```

```

        Console.WriteLine();
        Thread.Sleep(250);
    }
}
catch (ThreadAbortException exc)
{
    if ((int)exc.ExceptionState == 0)
    {
        Console.WriteLine("Ағын токтатылуынын алдын алу! " + "Аяқталу
коды " + exc.ExceptionState);
        Thread.ResetAbort();
    }
    else
        Console.WriteLine("Ағын токтатылды, аяқтау коды " +
exc.ExceptionState);
    }
}
Console.WriteLine(Thrd.Name + " калыпты аяқталды.");
}
}

class ResetAbort
{
    static void Main()
    {
        MyThread mt1 = new MyThread("Туынды Ағын");
        Thread.Sleep(1000);
        Console.WriteLine("Ағынды токтату.");
        mt1.Thrd.Abort(0); // ағын тоқтатылмайды
        Thread.Sleep(1000); // туынды ағынның ұзағырақ орындалуына мүмкіндік беру
        Console.WriteLine("Ағынды токтату.");
        mt1.Thrd.Abort(100); // ағын тоқтатылады
        mt1.Thrd.Join(); // ағынның тоқтатылуын күту
        Console.WriteLine("Негізгі ағын токтатылды.");
    }
}

```

Ағын күйі

Ағын күйі Thread сыныбында қол жетімді ThreadState сипатынан алынуы мүмкін. Төменде осы сипаттың жалпы пішіні келтірілген.

```
public ThreadState ThreadState{ get; }
```

Ағын күйі ThreadState тізімінде анықталған мән түрінде қайтарылады.

Негізгі ағын

Мысал 4. Негізгі ағынды басқару.

```

using System;
using System.Threading;

class UseMain
{
    static void Main()
    {
        Thread Thrd;
        // Негізгі ағынды алу.
        Thrd = Thread.CurrentThread;
    }
}

```

```

// Негізгі ағынның атын көрсету.
if (Thrd.Name == null)
    Console.WriteLine("Негізгі ағынның атауы жок.");
else
    Console.WriteLine("Негізгі ағынның атауы: " + Thrd.Name);
// Негізгі ағынның басымдығын көрсету.
Console.WriteLine("Басымдығы: " + Thrd.Priority);
Console.WriteLine();
// Атау мен басымдықты орнату.
Console.WriteLine("Атау мен басымдықты орнату.\n");
Thrd.Name = "Негізгі Ағын";
Thrd.Priority = ThreadPriority.AboveNormal;
Console.WriteLine("Енді негізгі ағын атауы: " + Thrd.Name);
Console.WriteLine("Енді басымдық: " + Thrd.Priority);
}
}

```

Мысал 5. Жаңа процесті іске қосу

```

using System;
using System.Diagnostics;

class StartProcess
{
    static void Main()
    {
        Process newProc = Process.Start("wordpad.exe");
        Console.WriteLine("Жаңа процесс іске қосылды.");
        newProc.WaitForExit();
        newProc.Close(); // бөлінген ресурстарды босату
        Console.WriteLine("Жаңа процесс аяқталды.");
    }
}

```