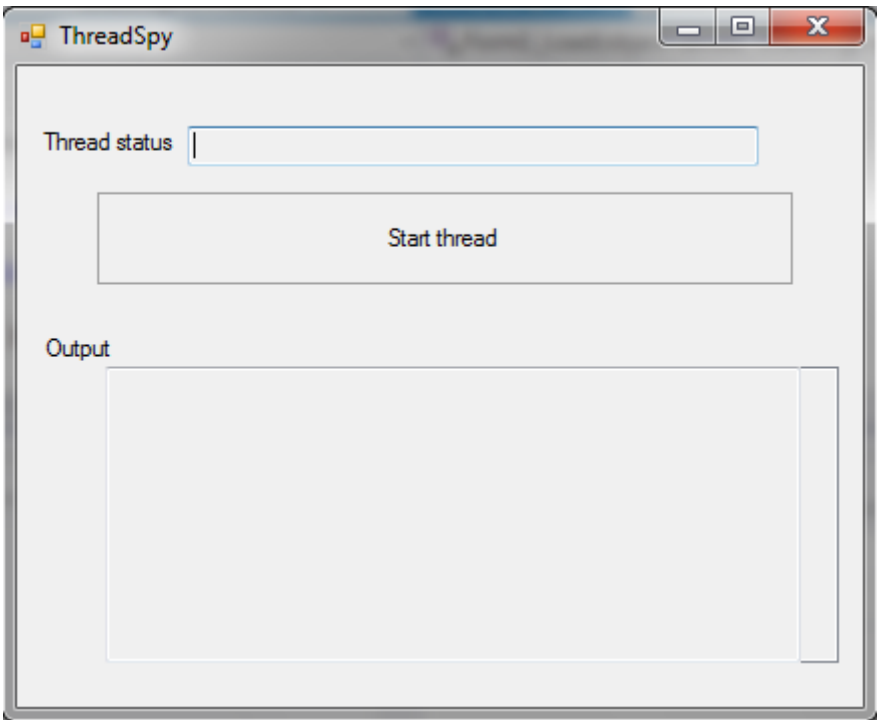


RESOLUTION DU DEVOIR

Dans cette partie, nous ferons des expériences avec plusieurs threads dans un processus.

Fenêtre



Codes des références

Example du method Run1	Example du method Run2
<pre>public sub Run() for I As Integer = 0 to 19 Thread.Sleep(300) TextBoxHelper.AddChar(tb, c) Next End sub</pre>	<pre>public sub Run() for I As Integer = 0 to 19 Thread.Sleep(300) tb.Text += c; wrong, because this would generate an exception Next End sub</pre>

<pre>private sub ButtonStartThread_Click () textBox1.Text = "creation de processus en cours de 300 seconde"; drawingThread.Start ()</pre>	<pre>private sub timer1_Tick() if (drawingThread<> null) this.TextBoxStatus.Text = drawingThread.ThreadState.ToString () end sub</pre>
--	---

end sub	
---------	--

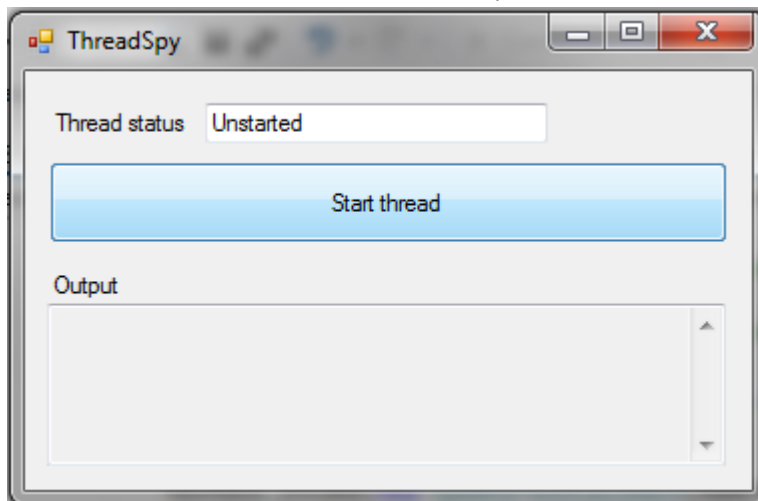
<pre>DrawingRunnable dr = new DrawingRunnable(this.TextBoxOutput, c) ThreadStart ts = new ThreadStart(dr.Run) drawingThread = new Thread(ts)</pre>
--

Question 1. Démarrez l'application.

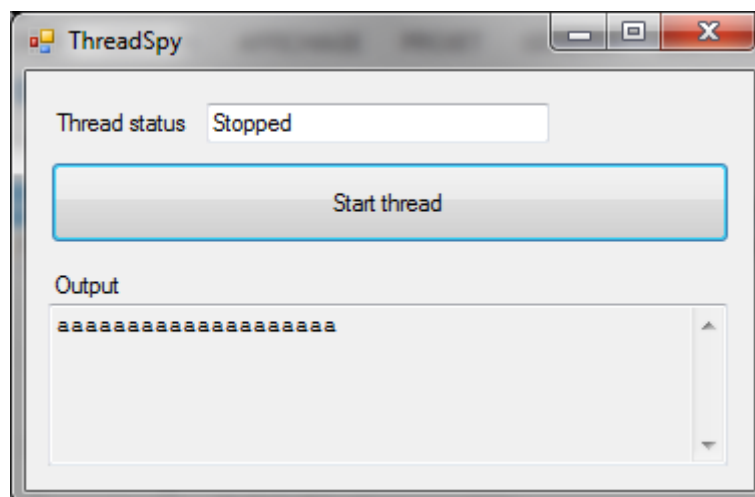
Affichage des 20 caractères au click dans une nouvelle thread

Solution 1.

1. Interface d'un état en cours des processus



2. Interface d'un état de fin de traitement des processus



1. Code source modifié de l'exemple
 - A. Classe DrawingRunnable

```
Imports System.Threading

namespace ThreadSpy

    class DrawingRunnable

        private tb As TextBox
        private a As char
        private b As char
        private c As char

        public sub DrawingRunnable(TextBox kg, char a, char b, char c)

            this.kg = kg
            this.a = a
            this.b = b
            this.c = c
        end sub

        /// <summary>
        /// The method to be executed by the new thread
        /// It will print 20 times the character c into TextBox tb
        /// </summary>

        public sub Run()
            for K As Int = 0 to 1
                Thread.Sleep(300)
                TextBoxHelper.AddChar(tb, c, c, c)
            Next
            for K As Int = 0 to 2
                Thread.Sleep(300)
                TextBoxHelper.AddChar(tb, b, b, b)
            Next
            for K As Int = 0 to 3
                Thread.Sleep(300)
                TextBoxHelper.AddChar(tb, a, a, a)
            Next
        End sub
    End sub
End class
```

B. Avec la class TextBoxHelper

```
Imports System.Threading

Imports System.Windows.Forms;

namespace ThreadSpy
    class TextBoxHelper
        static private TextBox textbox
        public delegate sub UpdateTextCallback(char a, char b, char c)

        //

        /// <summary>
        /// This method will add the char c into the textbox tb
        /// </summary>
        /// <param name="tb"></param> The TextBox where the char will be added
        /// <param name="c"></param> The char to add
    End class
```

```

static public sub AddChar(TextBox tb, char a, char b, char c)
    textbox = tb
    textbox.Invoke(new UpdateTextCallback(AddCharSave), a,b,c)

end sub
static private sub AddCharSave(char a, char b, char c)

    textbox.Text += b

end sub
end sub
end class
C. Code du method modifier

```

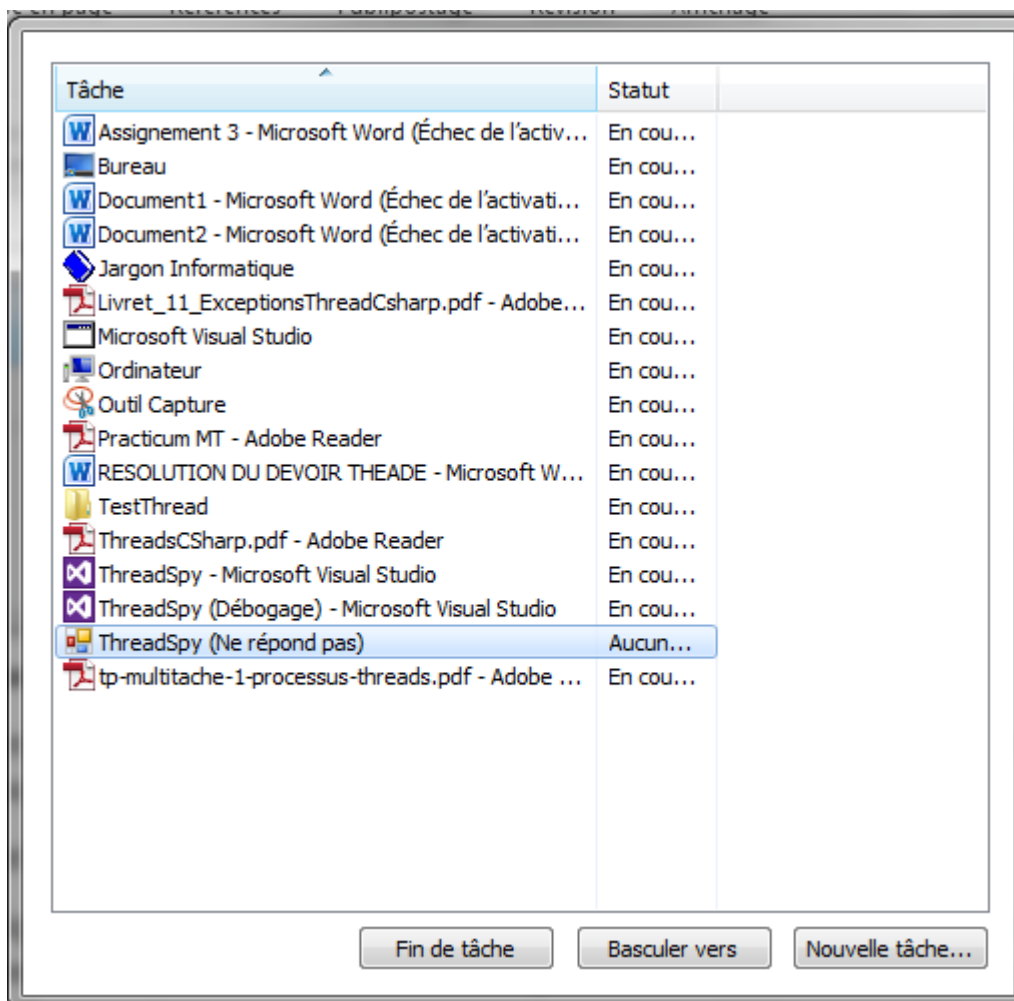
```

public sub Run()
    for K As Int = 0 to 3
        Thread.Sleep(300)
        TextBoxHelper.AddChar(tb, c)
    next

```

Question 2. Utiliser le Gestionnaire des tâches pour afficher le nombre de threads du noyau dans cette application.

Solution



Question 3. Le thread AC # ne peut pas être démarré deux fois.

Pour vérifier cela, fermez l'application et redémarrez-la.

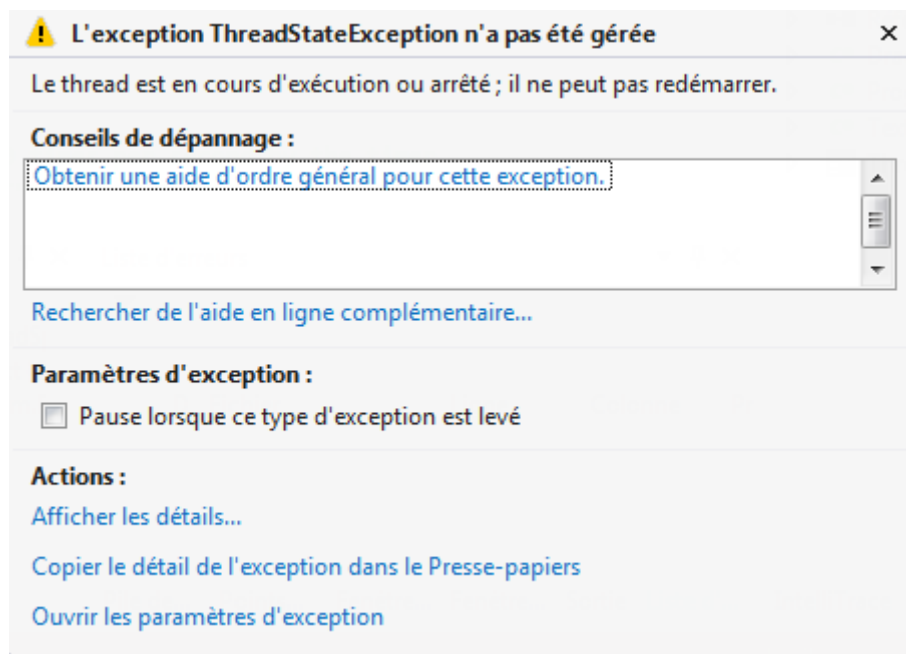
Cliquez sur le bouton "Démarrer le fil" et recommencez pendant que le fil est encore fonctionnement. Quel type d'exception est levé?

Fermez l'application et redémarrez-la.

Cliquez sur le bouton «Démarrer le fil», et recommencez lorsque le fil est "Arrêté". Quel type d'exception est levé?

L'application ne s'arrête pas correctement.

Changez le nouveau fil de façon à ce qu'il se ferme immédiatement et correctement lorsque le l'application est fermée.



Question 4. Changez le programme pour qu'un nouveau thread démarre à chaque fois lorsque vous cliquez sur le bouton «Démarrer le fil».

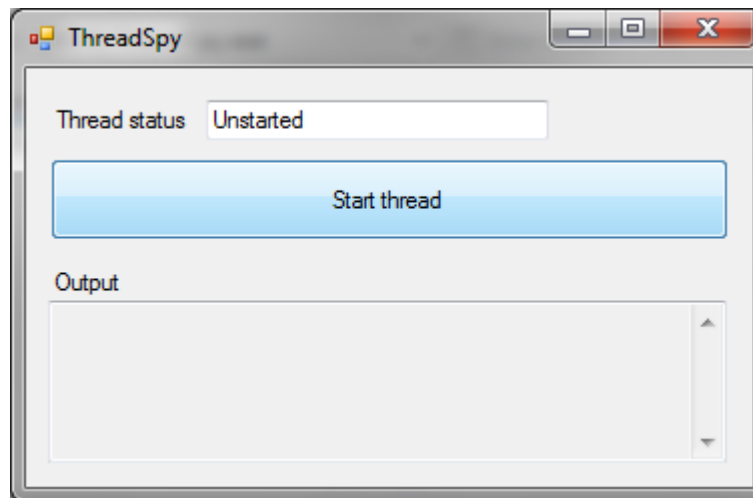
Pour voir la différence entre les fils, laissez chaque fil en imprimer un autre caractère dans la zone de texte: le premier fil doit imprimer 20 a, le deuxième fil

20 b, le troisième fil 20 c, et ainsi de suite.

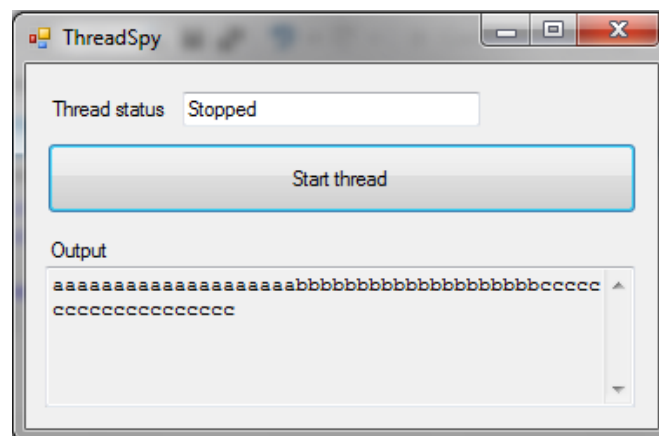
Bien entendu, le "Statut du thread" ne peut être affiché que pour un seul thread. Faites-le tel que toujours le statut du dernier thread créé est affiché. Vérifiez que vous pouvez cliquer maintenant sur " Start thread " plusieurs fois sans obtenir une exception.

Solution 4.

1. Interface d'un état en cours des processus



2. Interface d'un état de fin de traitement des processus



2. Code source modifié de l'exemple

```
public sub Run()  
    for K As Int = 0 to 19  
        Thread.Sleep(300)  
        TextBoxHelper.AddChar(tb, c,c,c)  
    Next  
    for K As Int = 0 to 19  
        Thread.Sleep(300)  
        TextBoxHelper.AddChar(tb, b,b,b)  
    Next  
    for K As Int = 0 to 19  
        Thread.Sleep(300)  
        TextBoxHelper.AddChar(tb, a,a,a)  
    Next  
End sub
```

Question 5. Changez le programme de sorte que chaque nouveau fil en imprime un de plus caractère que le fil précédent.

- Le premier fil imprime 1 'a',
- le deuxième fil imprime 2 'b's,
- le troisième fil imprime 3 'c's,
- et ainsi de suite.

Pour ce faire, ajoutez un int `NrOfChars` privé à la classe

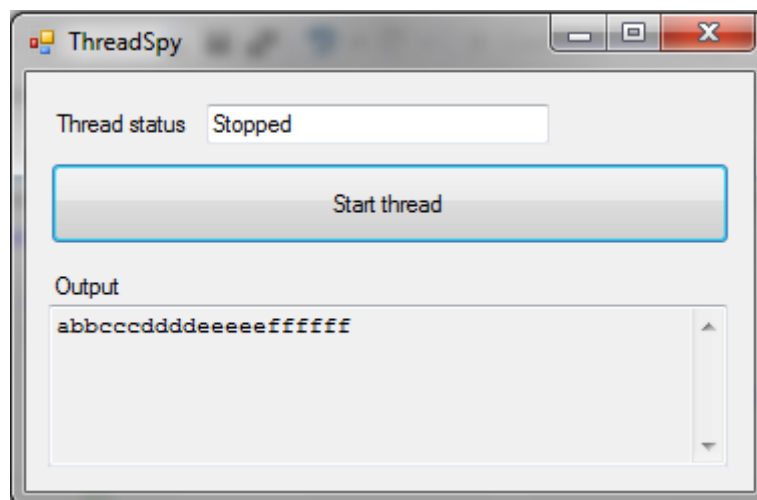
`ThreadSpyForm` et assurez-vous qu'il augmente à chaque fois qu'un nouveau le fil est créé. Testez votre application pour vous assurer qu'elle s'exécute toujours et se termine correctement.

Remarque: essayez d'appuyer plusieurs fois rapidement sur le bouton `` Démarrer le fil ", puis vous devriez toujours voir 1 'a', 2 'b's, 3 ' c's, etc. Ces caractères doivent être mélangés, mais le numéro de chaque caractère doit être comme ça.

Par exemple: `abcbccddeedeffffef`.

Solution.

Interface d'un état de fin des processus



Code source modifié de l'exemple

A. Classe `TextBoxHelper`

```
Imports System.Threading
namespace ThreadSpy

    class TextBoxHelper

        static private TextBox textbox;
        public delegate sub UpdateTextCallback(char a, char b, char c,
        char d, char e, char f)
```

```

    static public sub AddChar(TextBox kg, char a, char b, char c, char d,
char e, char f)
        textbox = kg;
        textbox.Invoke(new UpdateTextCallback(AddCharSave), a, b,
c, d, e, f)

        static private sub AddCharSave(char a, char b, char c, char d,
char e, char f)
        {
            textbox.Text += b
        }
    End sub

```

B. Classe DrawingRunnable

```

Imports System.Threading
namespace ThreadSpy
    class DrawingRunnable
        private TextBox kg
        private a As char
        private b As char
        private c As char
        private d As char
        private e As char
        private f As char

        public sub DrawingRunnable(TextBox kg, char a, char b, char c,
char d, char e, char f)
            this.tb = kg
            this.a = a
            this.b = b
            this.c = c
            this.d = d
            this.e = e
            this.f = f
        end sub

        /// <summary>
        /// The method to be executed by the new thread
        /// It will print 20 times the character c into TextBox kg
        /// </summary>
        public sub Run()

            for K As Int = 0 to 1
                Thread.Sleep(300)
                TextBoxHelper.AddChar(kg, f,f,f,f,f,f)
            Next
            for K As Int = 0 to 2
                Thread.Sleep(300)
                TextBoxHelper.AddChar(kg,e,e,e,e,e,e)
            Next
            for K As Int = 0 to 3
                Thread.Sleep(300)
                TextBoxHelper.AddChar(kg,d,d,d,d,d,d)
            Next
            for K As Int = 0 to 4
                Thread.Sleep(300)
                TextBoxHelper.AddChar(kg, c,c,c,c,c,c)
            Next
            for K As Int = 0 to 5
                Thread.Sleep(300)

```



```

        TextBoxHelper.AddChar(kg, b,b,b,b,b,b)
    Next
for K As Int = 0 to 6
    Thread.Sleep(300)
    TextBoxHelper.AddChar(kg, a, a, a, a, a, a);
Next
End sub

```

C. Méthode Run

```

public sub Run()
    for K As Int = 0 to 1
        Thread.Sleep(300)
        TextBoxHelper.AddChar(kg, c,c,c)

        Next
    for K As Int = 0 to 2
        Thread.Sleep(300)
        TextBoxHelper.AddChar(kg, b,b,b)
    Next
    for K As Int = 0 to 3
        Thread.Sleep(300)
        TextBoxHelper.AddChar(kg, a, a, a)
    Next
End sub

```