The method has the async modifier, the compiler generates IL including the state machine structure.

```
[CompilerGenerated]
private sealed class <Main>d__0 : IAsyncStateMachine
{
    public int <>1__state;

    public AsyncTaskMethodBuilder <>t__builder;

    public string[] args;

    private Downloader <downloader>5__1;

    private string <url>5__2;

    private string <temp>5__3;

    private string <folderName>5__4;

    private IOException <ioex>5__5;

    private TaskAwaiter <>u__1;
```

The first variable <>1_state stores the number of the reached await statement. As long as no await is encountered, the value of this variable is -1.

```
awaiter = <downloader>5__1.DownloadImage(<url>5__2, string.Concat("c:\\", <temp>5__3)).GetAwaiter();
if (!awaiter.IsCompleted)
{
    num = (<>1__state = 0);
    <>u__1 = awaiter;
    <Main>d__0 stateMachine = this;
    <>t__builder.AwaitUnsafeOnCompleted(ref awaiter, ref stateMachine);
    return;
}
```

Once it reached the first await, the temporary object is waiting for the asynchronous task to complete. If tasks are not completed then state equals = 0;

```
        catch (Exception exception)
        {
            <>1__state = -2;
            <downloader>5__1 = null;
            <url>5__2 = null;
            <temp>5__3 = null;
            <>t__builder.SetException(exception);
            return;
        }
        <>1__state = -2;
        <downloader>5__1 = null;
        <url>5__2 = null;
        <temp>5__3 = null;
        <>t__builder.SetResult();
    }
```

In the case of Exception state is equals -2;

As the class implements the IAsyncStateMachine interface, it will have two methods:

1.MoveNext

```
void IAsyncStateMachine.MoveNext()
{
    //ILSpy generated this explicit interface implementation from .override directive in MoveNext
    this.MoveNext();
}
```

The MoveNext method moves the state machine to its next state.

2.SetStateMachine

```
void IAsyncStateMachine.SetStateMachine(IAsyncStateMachine stateMachine)
{
    //ILSpy generated this explicit interface implementation from .override directive in SetStateMachine
    this.SetStateMachine(stateMachine);
}
```

Those methods are used by the compiler.

After completion state is equaled to -2;

```
    <>1__state = -2;
    <>t__builder.SetResult();
```