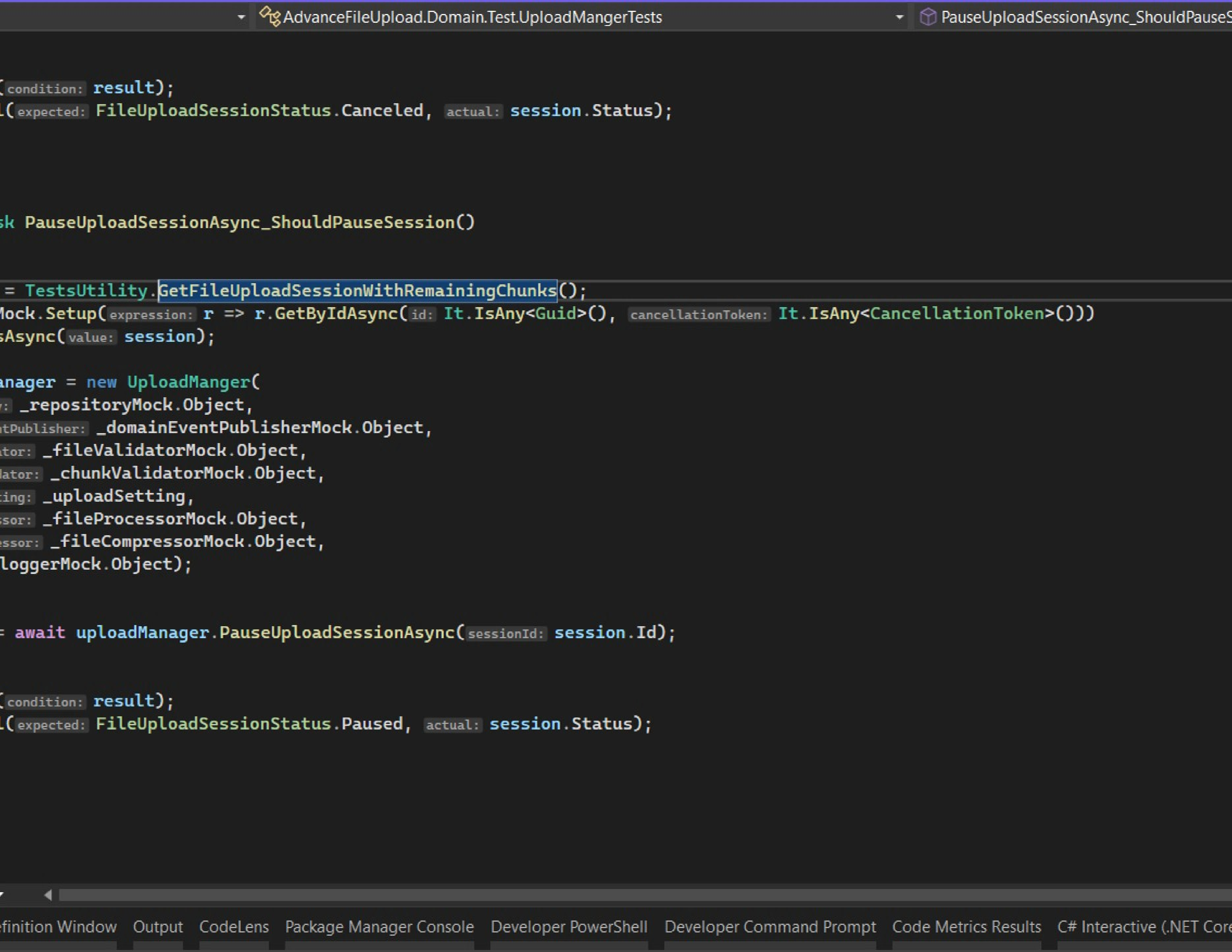
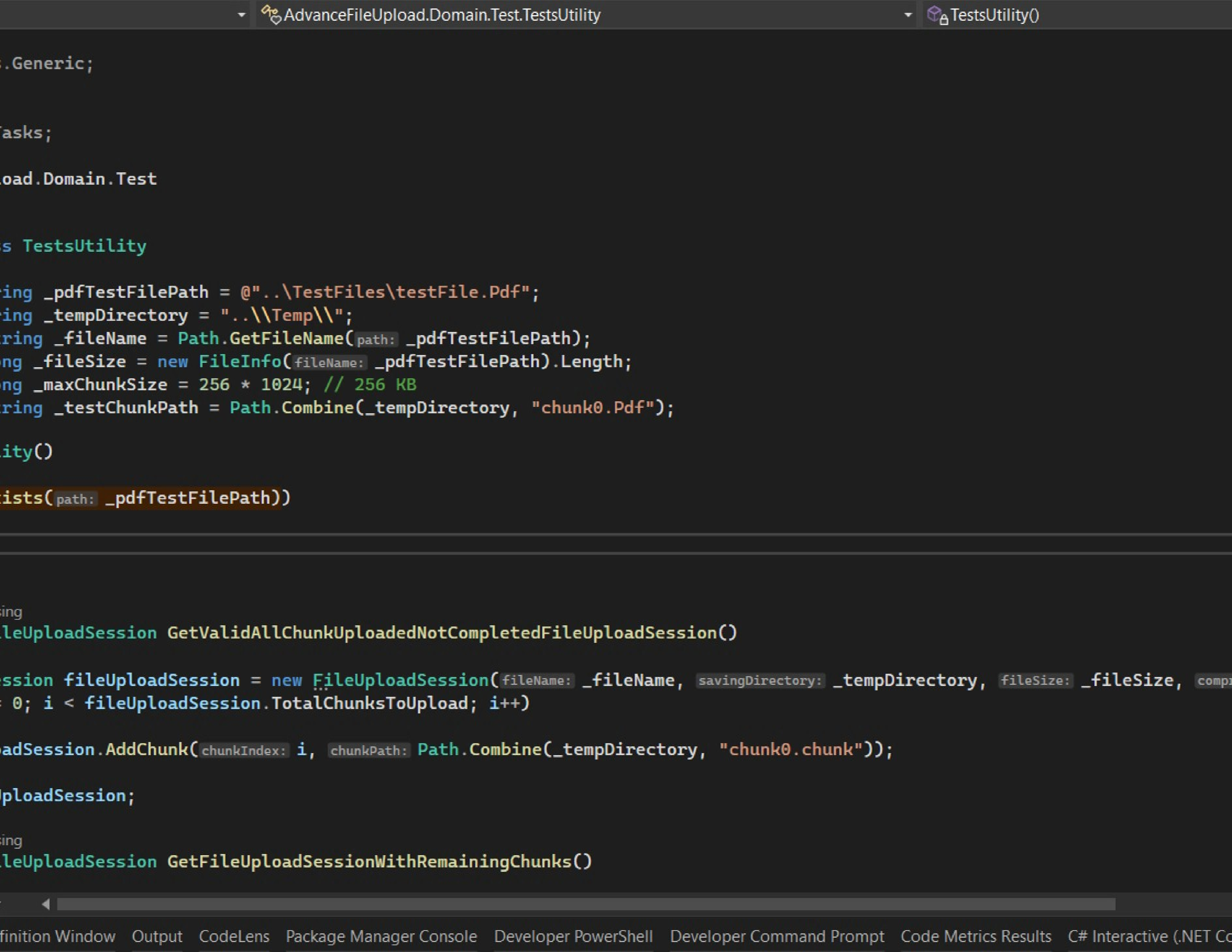


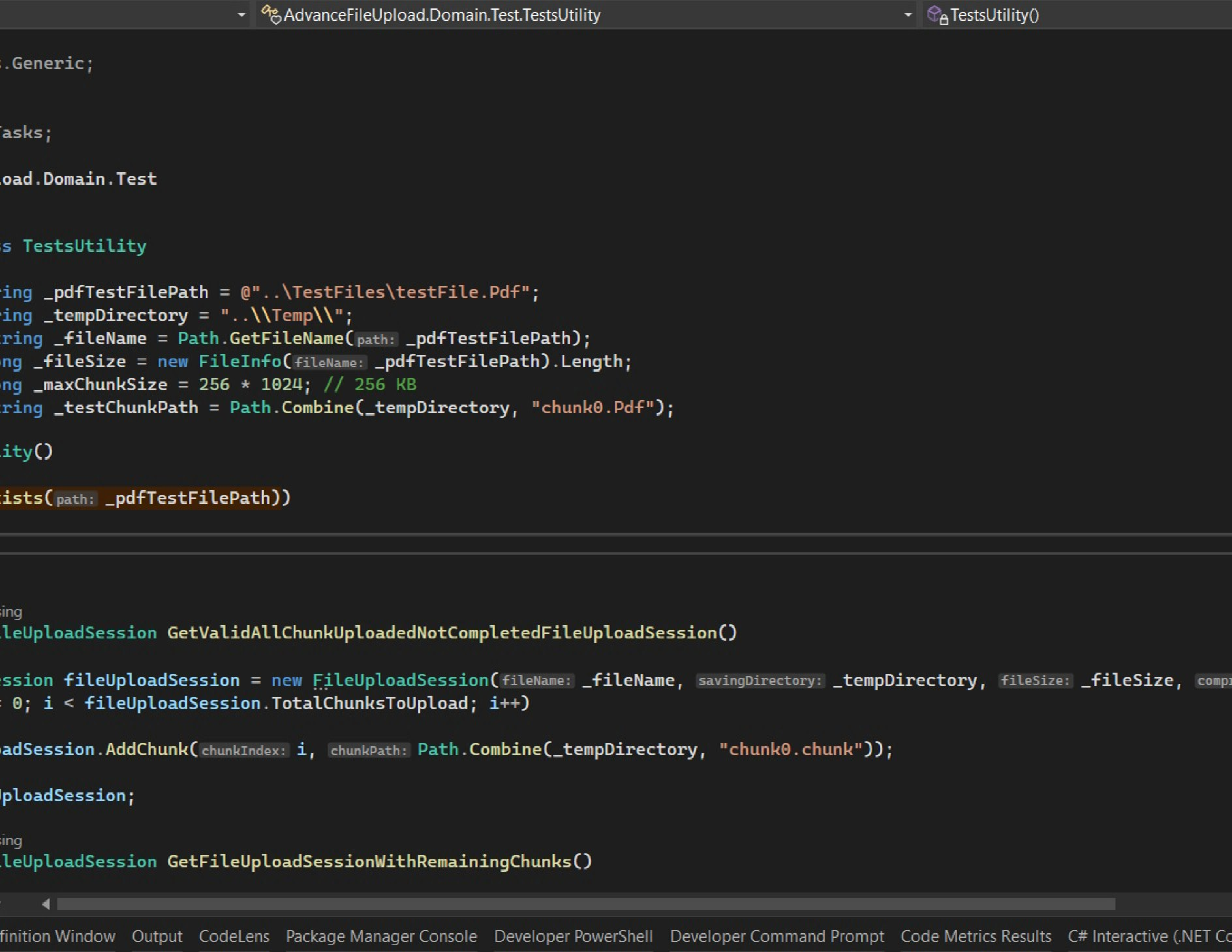
```
(condition: result);
l(expected: FileUploadSessionStatus.Canceled, actual: session.Status);




sk PauseUploadSessionAsync_ShouldPauseSession()


 = TestsUtility.GetFileUploadSessionWithRemainingChunks();
Mock.Setup(expression: r => r.GetByIdAsync(id: It.IsAny<Guid>(), cancellationToken: It.IsAny<CancellationToken>()))
sAsync(value: session);

anager = new UploadManger(
y: _repositoryMock.Object,
tPublisher: _domainEventPublisherMock.Object,
ator: _fileValidatorMock.Object,
dator: _chunkValidatorMock.Object,
ting: _uploadSetting,
ssor: _fileProcessorMock.Object,
essor: _fileCompressorMock.Object,
loggerMock.Object);


= await uploadManager.PauseUploadSessionAsync(sessionId: session.Id);


(condition: result);
l(expected: FileUploadSessionStatus.Paused, actual: session.Status);
```


AdvanceFileUpload.Domain.Test.TestsUtility   TestsUtility()

```
.Generic;


asks;

oad.Domain.Test


s TestsUtility

ing _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ing _tempDirectory = "..\\Temp\\";
ring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ng _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ng _maxChunkSize = 256 * 1024; // 256 KB
ring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

ity()

ists(path: _pdfTestFilePath))
```

```
ing
leUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, comp
 0; i < fileUploadSession.TotalChunksToUpload; i++)

adSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

ploadSession;

ing
leUploadSession GetFileUploadSessionWithRemainingChunks()
```

finition Window   Output   CodeLens   Package Manager Console   Developer PowerShell   Developer Command Prompt   Code Metrics Results   C# Interactive (.NET Co

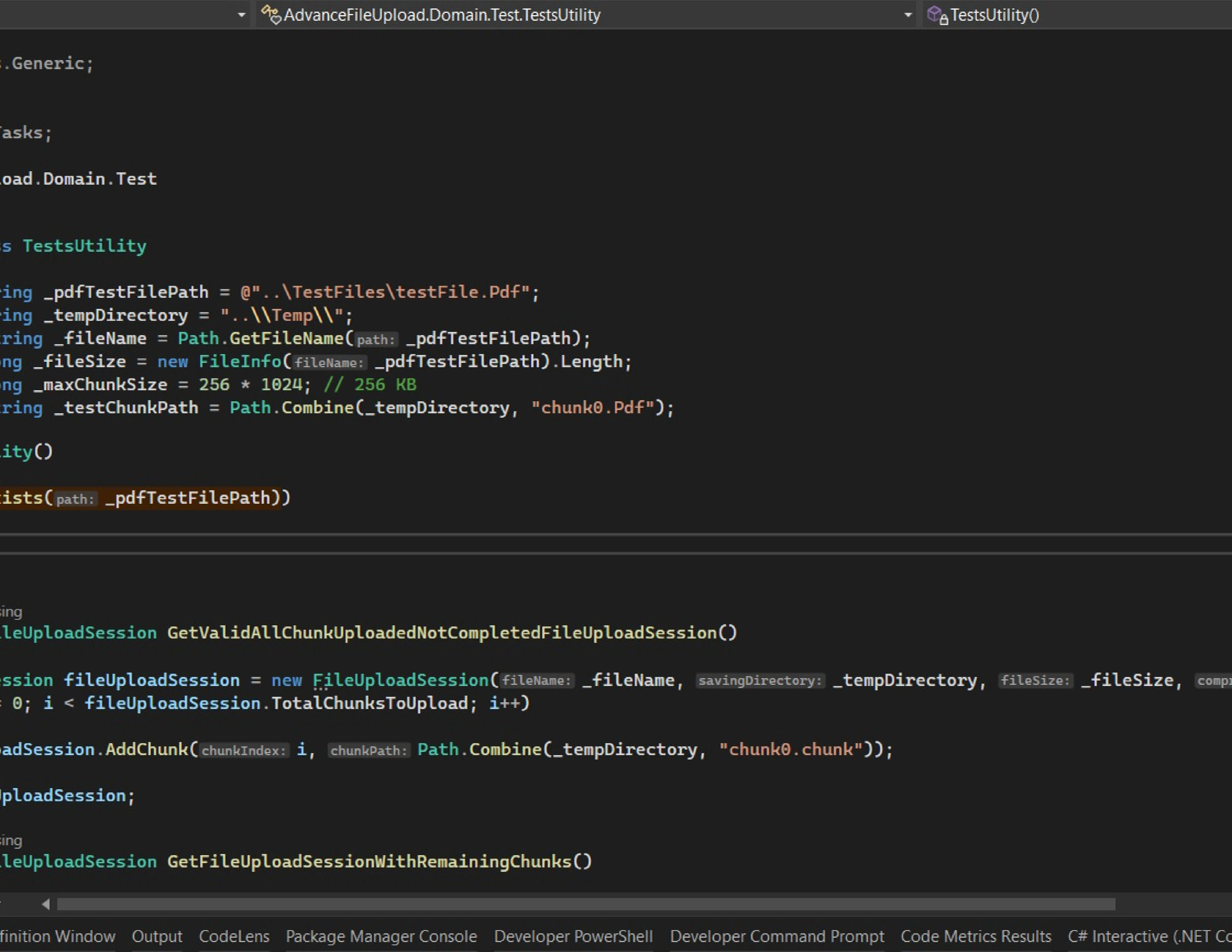


AdvanceFileUpload.Domain.Test.TestsUtility TestsUtility()

```
.Generic;

Tasks;

oad.Domain.Test

s TestsUtility

ing _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ing _tempDirectory = "..\\Temp\\";
ring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ng _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ng _maxChunkSize = 256 * 1024; // 256 KB
ring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

ity()

ists(path: _pdfTestFilePath))
```

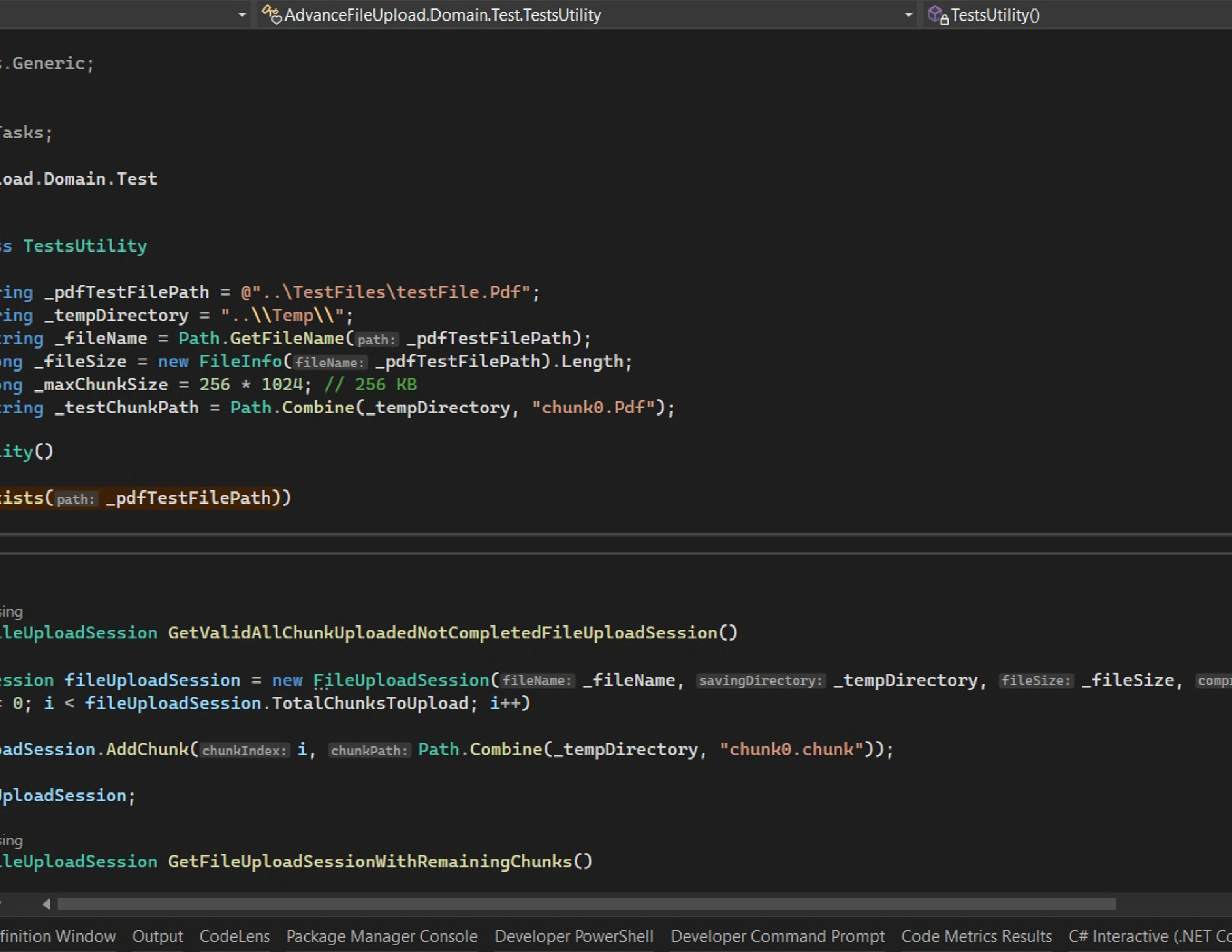
```
ing
leUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, compr
 0; i < fileUploadSession.TotalChunksToUpload; i++)

adSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

ploadSession;

ing
leUploadSession GetFileUploadSessionWithRemainingChunks()
```

finition Window Output CodeLens Package Manager Console Developer PowerShell Developer Command Prompt Code Metrics Results C# Interactive (.NET Co



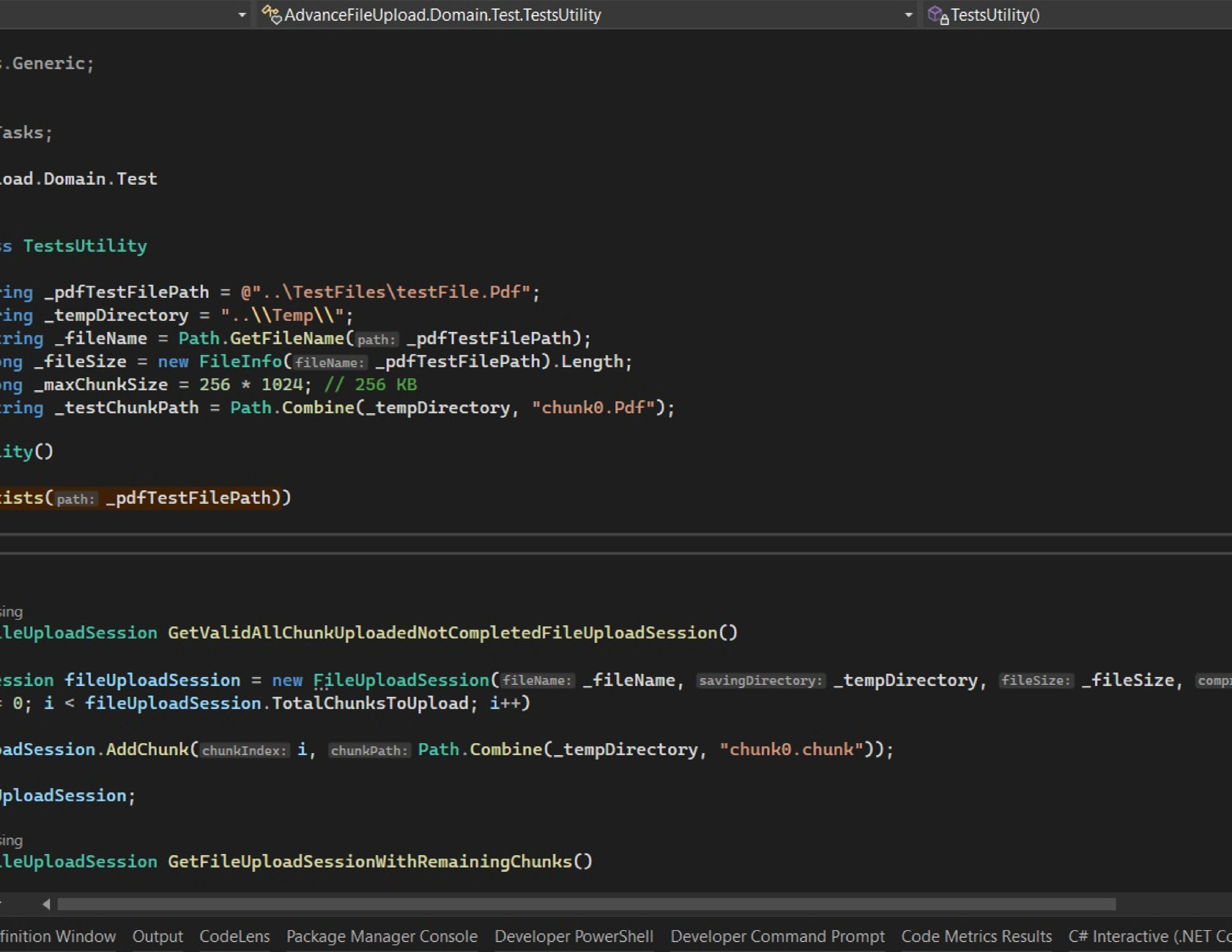
```
s.Generic;

Tasks;

oad.Domain.Test

s TestsUtility

ing _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ing _tempDirectory = "..\\Temp\\";
ring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ng _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ng _maxChunkSize = 256 * 1024; // 256 KB
ring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

ity()

ists(path: _pdfTestFilePath))
```

```
sing
leUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, compr
 0; i < fileUploadSession.TotalChunksToUpload; i++)

adSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

ploadSession;

sing
leUploadSession GetFileUploadSessionWithRemainingChunks()
```



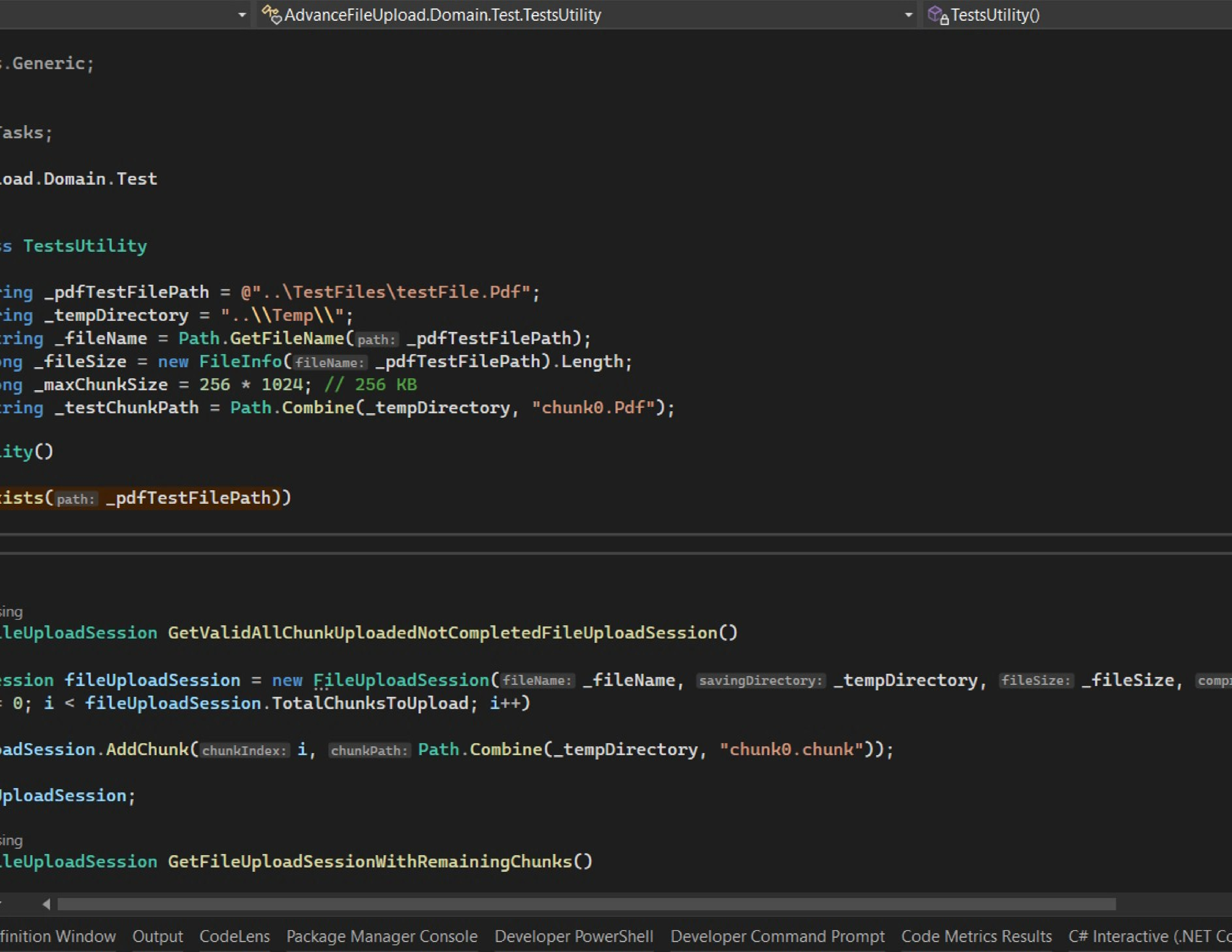
```
s.Generic;


Tasks;

Load.Domain.Test


ss TestsUtility

ring _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ring _tempDirectory = "..\\Temp\\";
tring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ong _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ong _maxChunkSize = 256 * 1024; // 256 KB
tring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

lity()

xists(path: _pdfTestFilePath))
```

```
sing
ileUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, compr
= 0; i < fileUploadSession.TotalChunksToUpload; i++)

oadSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

UploadSession;

sing
ileUploadSession GetFileUploadSessionWithRemainingChunks()
```



```
s.Generic;

Tasks;

Load.Domain.Test

ss TestsUtility

ring _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ring _tempDirectory = "..\\Temp\\";
tring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ong _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ong _maxChunkSize = 256 * 1024; // 256 KB
tring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

lity()

xists(path: _pdfTestFilePath))
```

```
sing
ileUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, comp
= 0; i < fileUploadSession.TotalChunksToUpload; i++)

oadSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

UploadSession;

sing
ileUploadSession GetFileUploadSessionWithRemainingChunks()
```

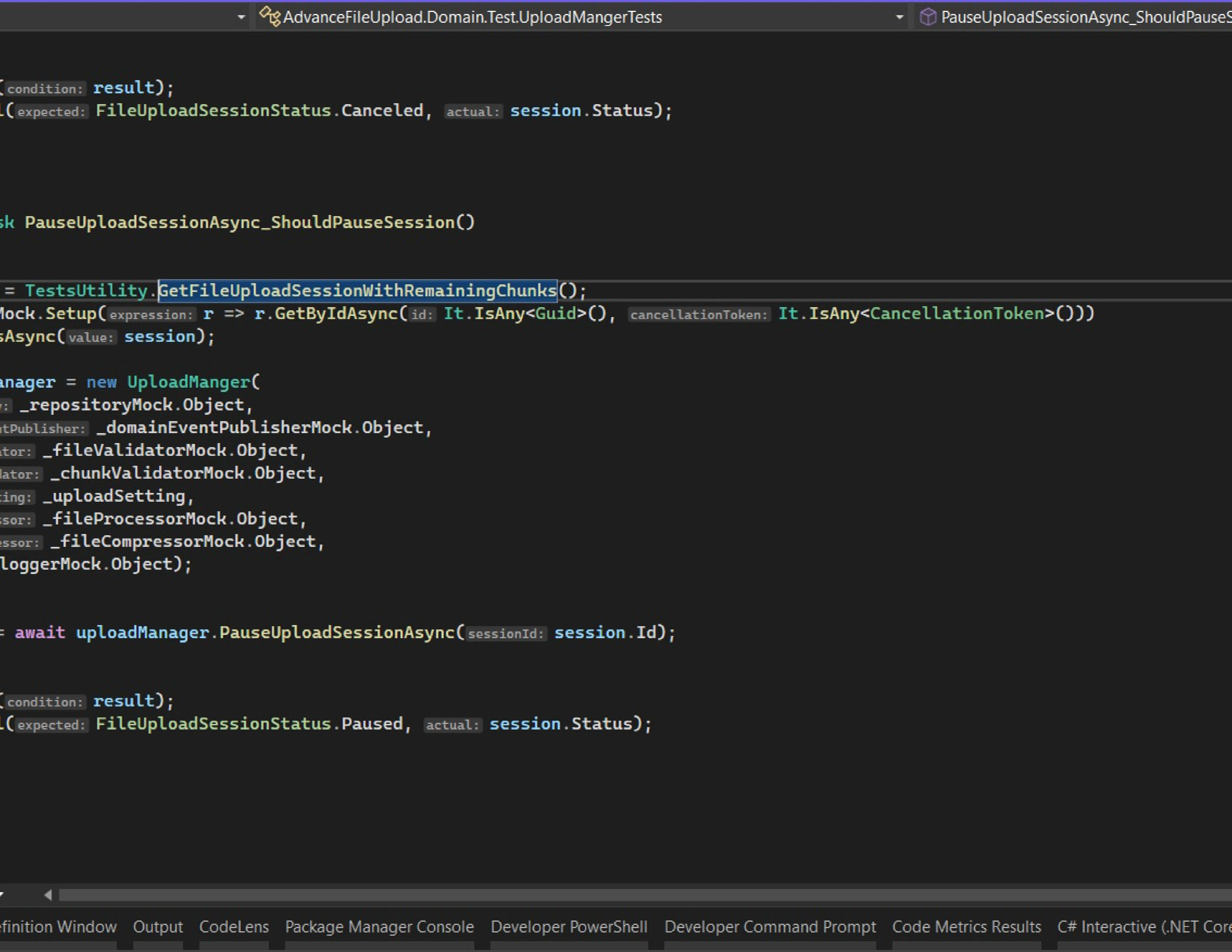


AdvanceFileUpload.Domain.Test.TestsUtility   TestsUtility()

```
s.Generic;


Tasks;

Load.Domain.Test


ss TestsUtility

ring _pdfTestFilePath = @"..\TestFiles\testFile.Pdf";
ring _tempDirectory = "..\\Temp\\";
tring _fileName = Path.GetFileName(path: _pdfTestFilePath);
ong _fileSize = new FileInfo(fileName: _pdfTestFilePath).Length;
ong _maxChunkSize = 256 * 1024; // 256 KB
tring _testChunkPath = Path.Combine(_tempDirectory, "chunk0.Pdf");

lity()

xists(path: _pdfTestFilePath))
```

```
sing
leUploadSession GetValidAllChunkUploadedNotCompletedFileUploadSession()

ession fileUploadSession = new FileUploadSession(fileName: _fileName, savingDirectory: _tempDirectory, fileSize: _fileSize, comp
= 0; i < fileUploadSession.TotalChunksToUpload; i++)

oadSession.AddChunk(chunkIndex: i, chunkPath: Path.Combine(_tempDirectory, "chunk0.chunk"));

UploadSession;

sing
leUploadSession GetFileUploadSessionWithRemainingChunks()
```

finition Window   Output   CodeLens   Package Manager Console   Developer PowerShell   Developer Command Prompt   Code Metrics Results   C# Interactive (.NET Co

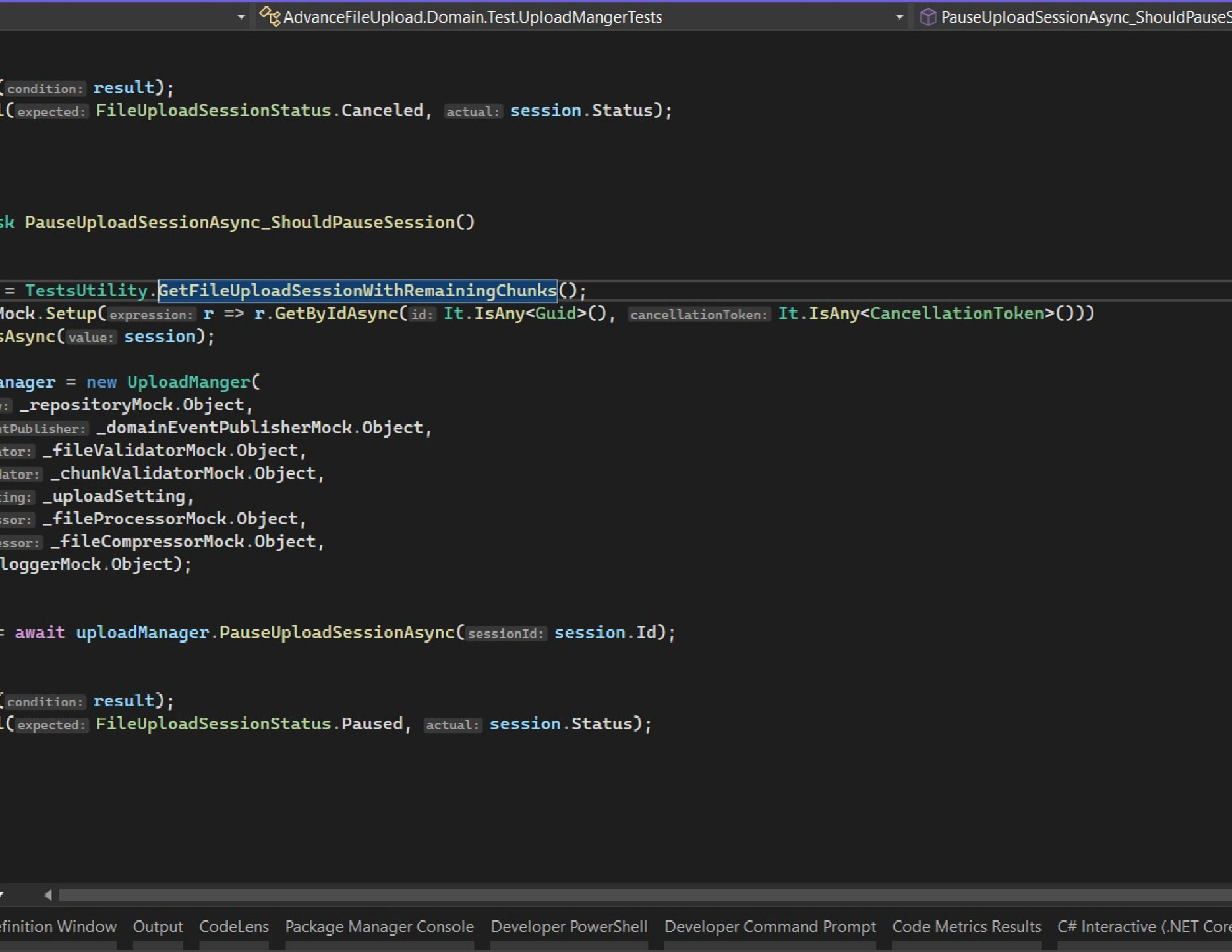


AdvanceFileUpload.Domain.Test.UploadMangerTests

PauseUploadSessionAsync_ShouldPauseS

```
(condition: result);
l(expected: FileUploadSessionStatus.Canceled, actual: session.Status);




sk PauseUploadSessionAsync_ShouldPauseSession()


 = TestsUtility.GetFileUploadSessionWithRemainingChunks();
Mock.Setup(expression: r => r.GetByIdAsync(id: It.IsAny<Guid>(), cancellationToken: It.IsAny<CancellationToken>()))
sAsync(value: session);

anager = new UploadManger(
y: _repositoryMock.Object,
tPublisher: _domainEventPublisherMock.Object,
ator: _fileValidatorMock.Object,
dator: _chunkValidatorMock.Object,
ting: _uploadSetting,
sor: _fileProcessorMock.Object,
essor: _fileCompressorMock.Object,
loggerMock.Object);


= await uploadManager.PauseUploadSessionAsync(sessionId: session.Id);


(condition: result);
l(expected: FileUploadSessionStatus.Paused, actual: session.Status);
```

efinition Window   Output   CodeLens   Package Manager Console   Developer PowerShell   Developer Command Prompt   Code Metrics Results   C# Interactive (.NET Core



AdvanceFileUpload.Domain.Test.UploadMangerTests
PauseUploadSessionAsync_ShouldPauseS

```
(condition: result);
l(expected: FileUploadSessionStatus.Canceled, actual: session.Status);




sk PauseUploadSessionAsync_ShouldPauseSession()


 = TestsUtility.GetFileUploadSessionWithRemainingChunks();
Mock.Setup(expression: r => r.GetByIdAsync(id: It.IsAny<Guid>(), cancellationToken: It.IsAny<CancellationToken>()))
sAsync(value: session);

anager = new UploadManger(
y: _repositoryMock.Object,
tPublisher: _domainEventPublisherMock.Object,
ator: _fileValidatorMock.Object,
dator: _chunkValidatorMock.Object,
ting: _uploadSetting,
sor: _fileProcessorMock.Object,
essor: _fileCompressorMock.Object,
loggerMock.Object);


= await uploadManager.PauseUploadSessionAsync(sessionId: session.Id);


(condition: result);
l(expected: FileUploadSessionStatus.Paused, actual: session.Status);
```

efinition Window Output CodeLens Package Manager Console Developer PowerShell Developer Command Prompt Code Metrics Results C# Interactive (.NET Core