Практикум. Сжатие и декомпрессия существующего файла

Упражнение 1. Сжатие существующего файла

В этом упражнении вы сожмете существующий файл, получив новый, сжатый.

1. Создайте новое консольное приложение с именем CompressionDemo.
2. Добавьте директиву using чтобы импортировать в новый проект про­странства имен System.IO и System.IO.Compression.
3. Создайте статический метод с именем CompressFile, прини­мающий в качестве параметров две строки: inFilename и outFilename. Сигнатура мето­да должна выглядеть так:

// C#

static void CompressFile(string inFilename,

string outFilename)

{ }

1. В теле этого метода откройте объект FileStream (с именем sourceFile), открывающий файл, заданный параметром inFilename.
2. Создайте новый объект FileStream (с именем destFile), создающий файл, заданный параметром outFilename.
3. Создайте новый объект GZipStream (с именем compStream), задав destFile в качестве потока для записи сжатых данных. Также укажите, что это поток для сжатия. Полу­ченный код выглядит примерно так:

GZipStream compStream =

new GZipStream(destFile, CompressionMode.Compress);

7. Байт за байтом перенесите данные из исходного файла в поток сжатия:  
// C#

int theByte - sourceFile.ReadByte();

while (theByte != -1)

{

compStream.WriteByte((byte)theByte); theByte = sourceFile. ReadByte();

}

1. Перед выходом из метода закройте все потоки.
2. Из метода Main нового консольного проекта вызовите метод CompressFile, передав ему в качестве параметров имена существующего и нового файла (имя нового файла обычно получают прибавлением .gz к имени исходного). Получится примерно такой код:

CompressFile(@"c:[\boot](file:///boot).ini", @"с:[\boot](file:///boot).ini.gz");

10. Соберите проект и исправьте ошибки. Проверьте, появился ли сжатый файл. В тек-  
стовом редакторе, таком как Блокнот, содержимое файла должно выглядеть бессмыс-  
ленным набором символов.

Упражнение 2. Декомпрессия файла

Сейчас вы откроете файл, созданный в упражнении 1, разуплотните его содержимое и запишете его в новый файл.

1. Откройте проект CompressionDemo, созданный в упражнении 1.
2. Создайте новый статический метод с именем UncompressFile, принимающий в качестве параметров две строки: inFileName и outFileName. Сигнату­ра метода должна выглядеть так:

// С#

static void UncompressFile(string inFilename,

string outFilename)

{ }

1. В теле этого метода откройте объект FileStream (с именем sourceFile), открывающий файл, заданный параметром inFilename, — это должен быть сжатый файл, записан­ный в упражнении 1.
2. Создайте новый объект FileStream (с именем destFile), создающий файл, заданный параметром outFilename.
3. Создайте объект GZipStream (с именем compStream), задав sourceFile в качестве пото­ка, из которого будут считываться сжатые данные. Также укажите, что это поток для декомпрессии. Полученный код выглядит примерно так:

GZipStream compStream =

new GZipStream(sourceFile, CompressionMode.Decompress);

6. Байт за байтом перенесите данные из сжатого файла в результирующий файл. Полу-  
ченный код выглядит примерно так:

// C#

int theByte = compStream.ReadByteO;

while (theByte != -1)

{

destFile.WriteByte((byte)theByte); theByte = compStream. ReadByte();

}

1. Перед выходом из метода закройте все потоки.
2. Из метода Main нового консольного проекта вызовите метод UncompressFile, передав ему в качестве параметров имя сжатого файла, созданного в упражнении, и имя фай­ла для записи несжатых данных. Получится примерно такой код:

// С#

DecompressFile(@"c:[\boot](file:///boot).ini.gz", @"c:[\boot](file:///boot).ini.test");

9. Соберите проект и исправьте ошибки. Убедитесь, что консольное приложение пра­вильно создало несжатый файл, открыв его в Блокноте. Сравните содержимое этого файла с исходным — файлы должны быть идентичными.