Create a console application that downloads images from webpages. For us, let’s assume webpages look like this:

<html>

<body>

<img src="pic1.bmp" alt="First pic" />

<img src="pic2.bmp" alt="Second pic" />

<img src="pic3.bmp" alt="Third pic" />

</body>

</html>

The „alt” attribute must be ignored. An example XML can be found here: <http://users.nik.uni-obuda.hu/cseri/zh2_gyakorlo/simplepage.html>

Create the following classes:

* ImageData: stores data: string URL, string FileName. FileName is the value of the img attribute (e.g.: „pic1.bmp”), the URL should be created from the HTML address (everything until the last slash character) and the filename (e.g.: „http://users.nik.uni-obuda.hu/cseri/zh2\_gyakorlo/pic1.bmp”)
* WebpageProvider: Should have one method: XDocument Download(string url) that will download the XML file.
* ImageCollector: should have a method *public IList<ImageData> GetImages(string url)* that will return the ImageData instances read from the specified URL. The download of the XML file should be done by a WebpageProvider that is specified in the constructor.

Operation

* When starting, the ImageCollector should be used to collect the image datas from the example address.
* For downloading an image, launch a new Task (multiple images should be downloaded at the same time). Use WebClient.DownloadFile() to download and save the image files.
* Every task should have its own line in the console, with its own specified colour (even and odd lines should have different colours). When starting the download, then “[FILENAME] … Downloading” should be displayed, and when the download is finished, then “[FILENAME] … Success”.



* When all tasks are completed, then open all downloaded images using Process instances.

Create a test project that tests the operation of the ImageCollector using a single test with valid test input. During the test, the ImageCollector should NOT use the real WebPageProvider, use a mock object instead.