**Virtual methods - StringsProcessor classes**

The aim of this exercise will be to make this code work:

A computer screen shot of a program code

Description automatically generated

Currently, the **StringsProcessor**, **StringsTrimmingProcessor**, and **StringsUppercaseProcessor**classes are not implemented.

Let's understand what happens in the **ProcessAll**method. It takes a List of strings as a parameter. Inside, it has a collection of StringsProcessor objects. Those objects expose a Process method, which also takes a List of strings and returns the same type.

The **StringsTrimmingProcessor**class has a method that takes a collection of strings and, as a result, returns an identical collection, but with each word trimmed by half. So, for example, for the following input:

**"bobcat", "wolverine", "grizzly"**

It shall return:

**"bob", "wolv", gri"**

To cut a string in half, you can use the [Substring](https://learn.microsoft.com/en-us/dotnet/api/system.string.substring?view=net-8.0)method.

The **StringsUppercaseProcessor**class has a method that takes a collection of strings and, as a result, returns an identical collection, but with each word made uppercase. So, for example, for the following input:

**"bobcat", "wolverine", "grizzly"**

It shall return:

**"BOBCAT", "WOLVERINE", "GRIZZLY"**

Because both those transformations will be applied to the collection of strings in the **ProcessAll**method, it should return the collection like the input collection, but with each word both trimmed and made uppercase.

For example, for the following input:

**"bobcat", "wolverine", "grizzly"**

The result of the ProcessAll method should be:

**"BOB", "WOLV", "GRI"**

Your job is to create the implementations of the following classes:

* **StringsProcessor (base)**
* **StringsUppercaseProcessor (derived)**
* **StringsTrimmingProcessor (derived)**

Try to avoid code duplication as much as possible.