



Assembly.

Assembly to skompilowana i spakowana reprezentacja kodu zawartego w danym projekcie stworzona w .net.

W procesie kompilacji możemy wytwarzać pliki exe lub dll w zależności od typu projektu jaki wybierzemy. Wszystkie projekty typu library będą podczas procesu kompilacji tworzyć pliki DLL natomiast projekty uruchamialne np. Console application wytworzą plik exe.



trzeba pamiętać że do tworzenia assembly w C# nie konieczne jest VisualStudio. Poniżej przedstawiona została procedura stworzenia prostego programu poza VS.

W notatniku tworzymy plik keto.cs jak poniżej

```
keto.cs - Notepad
File Edit Format View Help
public class Program
{
    static void Main()
    {
        System.Console.WriteLine("I Like Keto");
    }
}
```

Znajdujemy lokalizację w której zainstalowany jest .net framework a następnie testujemy czy mamy csc.exe. po wpisaniu csc powinna pojawić się następująca wiadomość.

```
Command Prompt
C:\Windows\Microsoft.NET\Framework>v4.0.30319\csc
Microsoft (R) Visual C# Compiler version 4.6.1586.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
supports language versions up to C# 5, which is no longer the latest version. Fo
r compilers that support newer versions of the C# programming language, see http
://go.microsoft.com/fwlink/?LinkID=533240

warning CS2008: No source files specified
error CS1562: Outputs without source must have the /out option specified

C:\Windows\Microsoft.NET\Framework>
```

Następnie komendą `csc ścieżka\ket.cs` uruchamiamy kompilację pliku keto

```
Administrator: Command Prompt
C:\>c:\Windows\Microsoft.NET\Framework\v4.0.30319\csc keto.cs
Microsoft (R) Visual C# Compiler version 4.6.1586.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only s
upports language versions up to C# 5, which is no longer the latest version. For
compilers that support newer versions of the C# programming language, see http://
go.microsoft.com/fwlink/?LinkID=533240

C:\>dir keto*
Volume in drive C is OSDisk
Volume Serial Number is F232-B880

Directory of C:\

02/08/2019  02:24 PM                114 keto.cs
02/08/2019  02:39 PM            3,584 keto.exe
               2 File(s)            3,698 bytes
               0 Dir(s) 83,943,116,800 bytes free

C:\>
```

Efektem będzie powstaniem pliku exe. Proszę sprawdzić co by się stało gdyby nasz keto nie zawierał metody **main**. Wszystkie pliki dll nie są uruchamiane ale mogą być dodawane do projektu jako referencja bądź wczytywane z kodu a następnie używane tak jakby były fragmentem danego projektu. Standardowe assembly .net są częścią (GAC) Global

Assembly Cache dll zlokalizowane są:

```
Administrator: Command Prompt
C:\Windows\assembly\GAC>dir
Volume in drive C is OSDisk
Volume Serial Number is F232-B880

Directory of C:\Windows\assembly\GAC

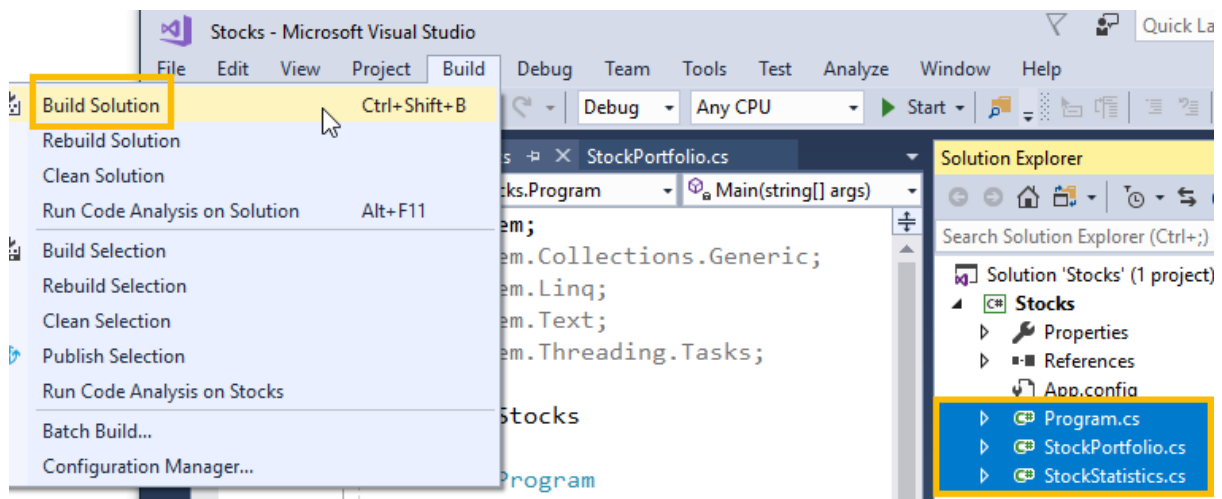
05/23/2018  08:05 AM    <DIR>          .
05/23/2018  08:05 AM    <DIR>          ..
05/23/2018  08:04 AM    <DIR>          ADODB
05/23/2018  08:04 AM    <DIR>          Extensibility
05/23/2018  08:04 AM    <DIR>          Microsoft.Ink
05/23/2018  08:05 AM    <DIR>          Microsoft.mshtml
05/23/2018  08:05 AM    <DIR>          Microsoft.StdFormat
05/23/2018  08:05 AM    <DIR>          mscomctl
05/23/2018  08:05 AM    <DIR>          MSDATASRC
05/23/2018  08:05 AM    <DIR>          stdole
               0 File(s)                0 bytes
              10 Dir(s) 83,922,280,448 bytes free

C:\Windows\assembly\GAC>
```

Wszystkie assembly z GAC mogą być normalnie dodawane poprzez referencje do programów pisanych .net

Assembly z poziomu VS.

Po stworzeniu projektu stock typu console application i skompilowaniu go uzyskamy następujące pliki.



```
C:\Users\users\source\repos\Stocks\Stocks\bin\Debug>dir
Volume in drive C is OSDisk
Volume Serial Number is F232-B880
```

```
Directory of C:\Users\users\source\repos\Stocks\Stocks\bin\Debug
```

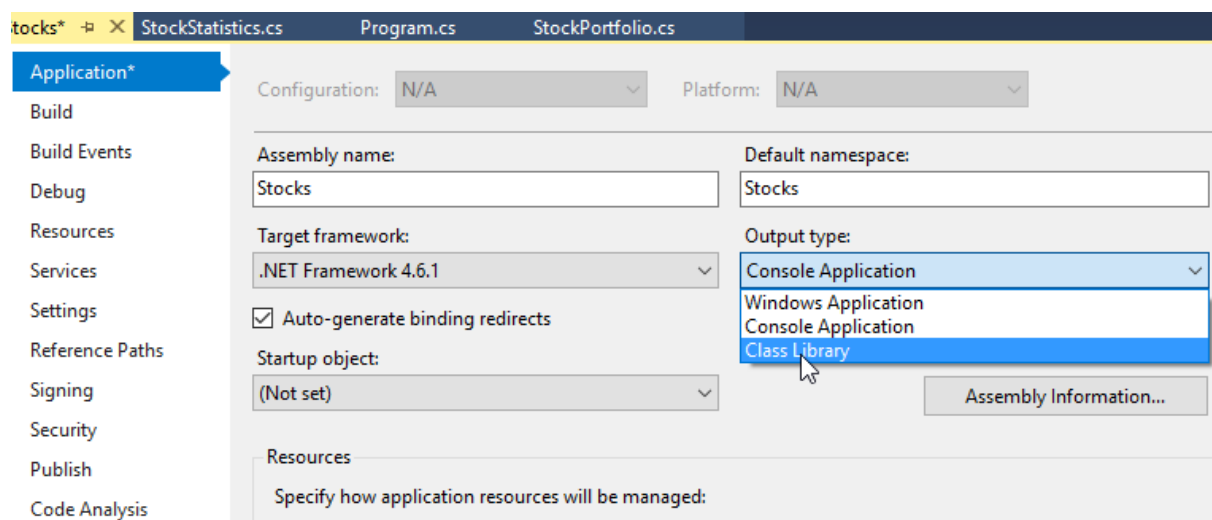
02/05/2019 01:43 PM

02/05/2019 01:43 PM

```
02/06/2019 05:29 PM          5,632 Stocks.exe
02/04/2019 04:46 PM          189 Stocks.exe.config
02/06/2019 05:29 PM        22,016 Stocks.pdb
          3 File(s)          27,837 bytes
```

2 Dir(s) 83,911,729,152 bytes free

jeżeli chcemy stworzyć plik dll to możemy to zrobić w sposób dwojaki. Po pierwsze możemy zmienić typ projektu w jego właściwościach jak poniżej.



albo stworzyć nowy projekt typu .netFramework library. Zmieniając typ projektu stocks.exe dostaniemy następujący efekt.

```
C:\Users\user\sourcerepos\Stocks\Stocks\bin\Debug>dir
```

```
Volume in drive C is OSDisk
```

```
Volume Serial Number is F232-B880
```

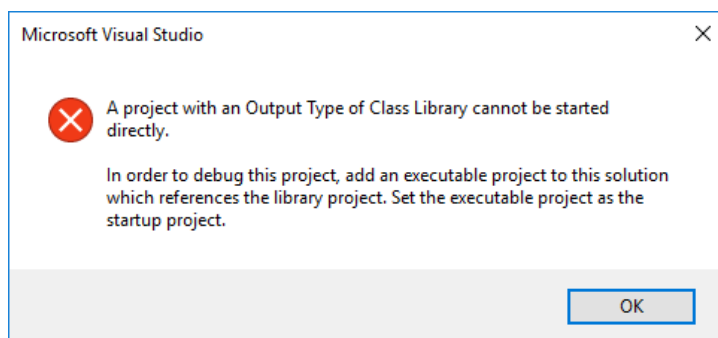
```
Directory of C:\Users\user\sourcerepos\Stocks\Stocks\bin\Debug
```

02/08/2019 04:53 PM

.02/08/2019 04:53 PM

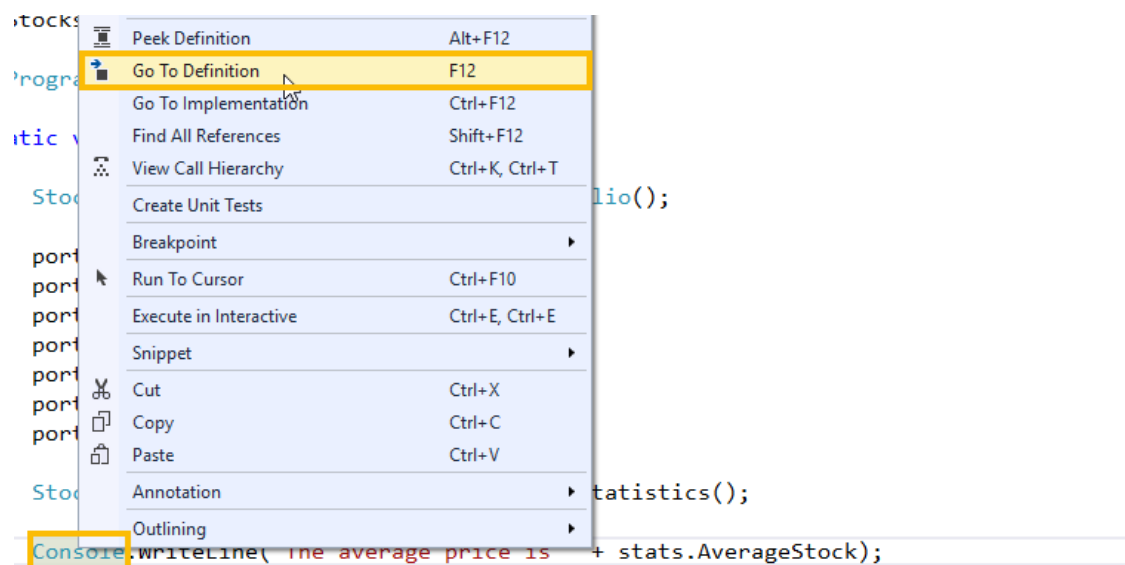
```
02/08/2019 04:53 PM          5,120 Stocks.dll
02/04/2019 04:46 PM          189 Stocks.dll.config
02/08/2019 04:53 PM        19,968 Stocks.pdb
          3 File(s)          25,277 bytes
          2 Dir(s) 83,904,868,352 bytes free
```

Uruchomienie takiego programu jest niemożliwe. ponieważ nie jest on już plikiem wykonywalnym.

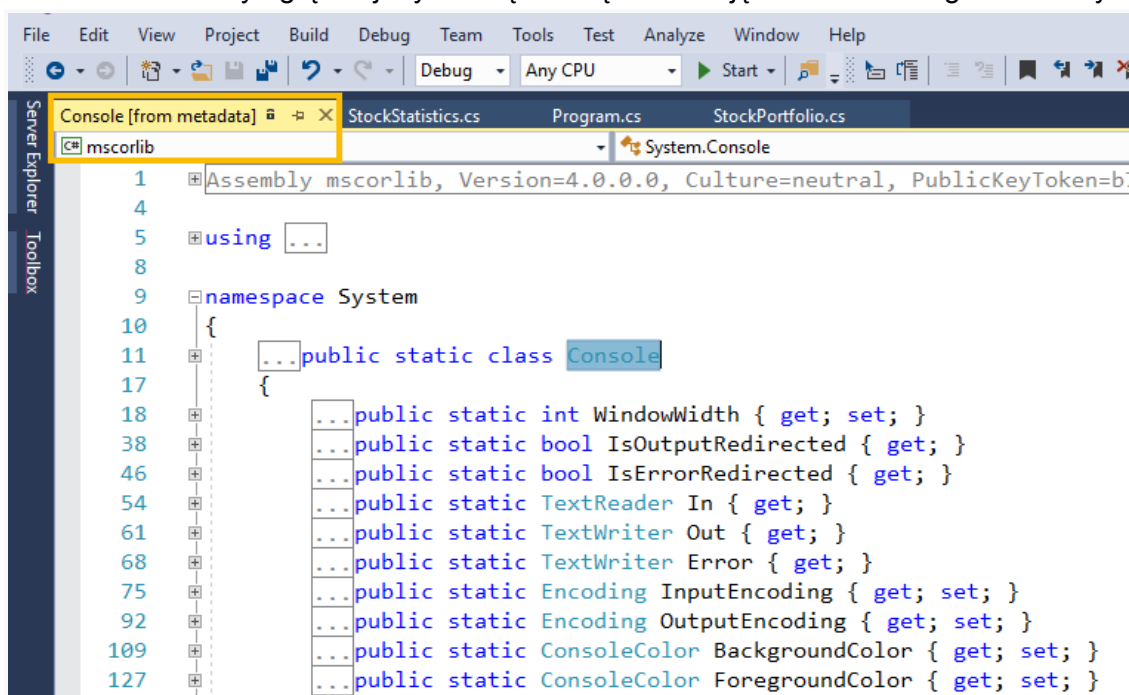


Ciekawostka.

w .net możemy podglądać praktycznie każde assembly. wchodząc do definicji.

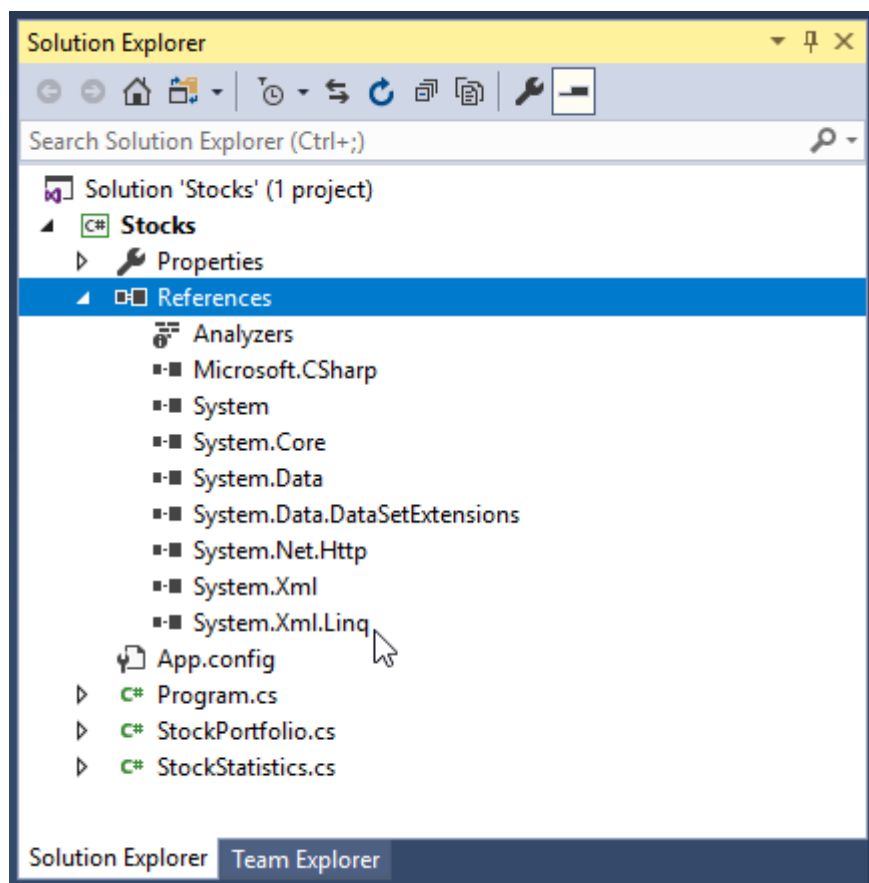


W środku możemy oglądać jedynie częściową informację o kodzie danego assembly



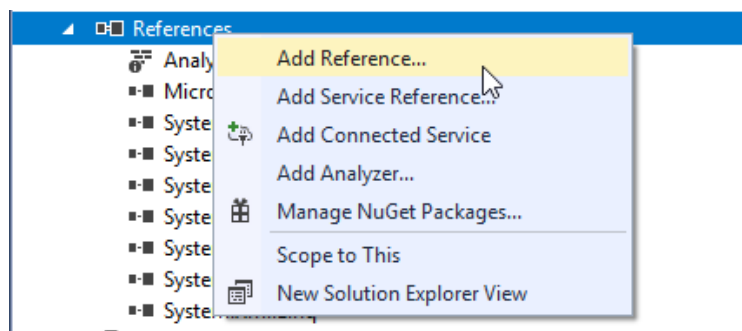
Assembly/referencje

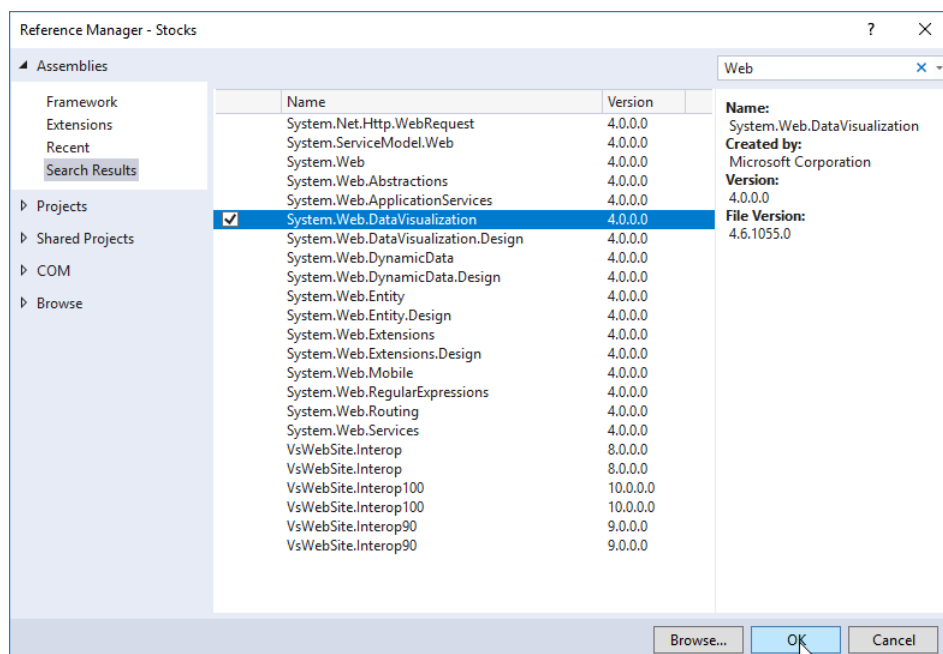
Dodawanie referencji do projektu może się odbywać poprzez bezpośrednią edycję pliku cs.proj lub dodawanie assembly do referencji w projekcie.



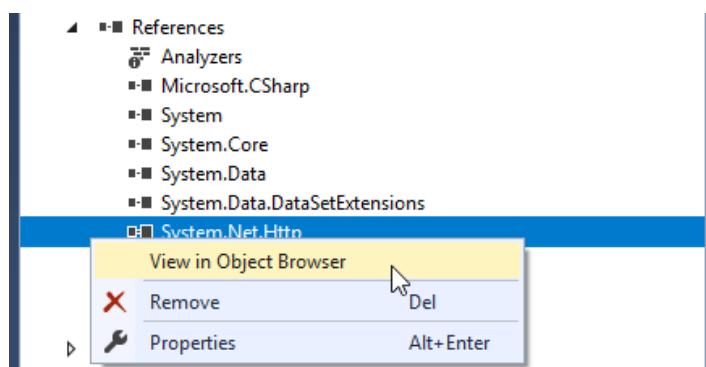
Można dodawać usuwać a także podglądać lokalizację pliku.

Dodawanie.

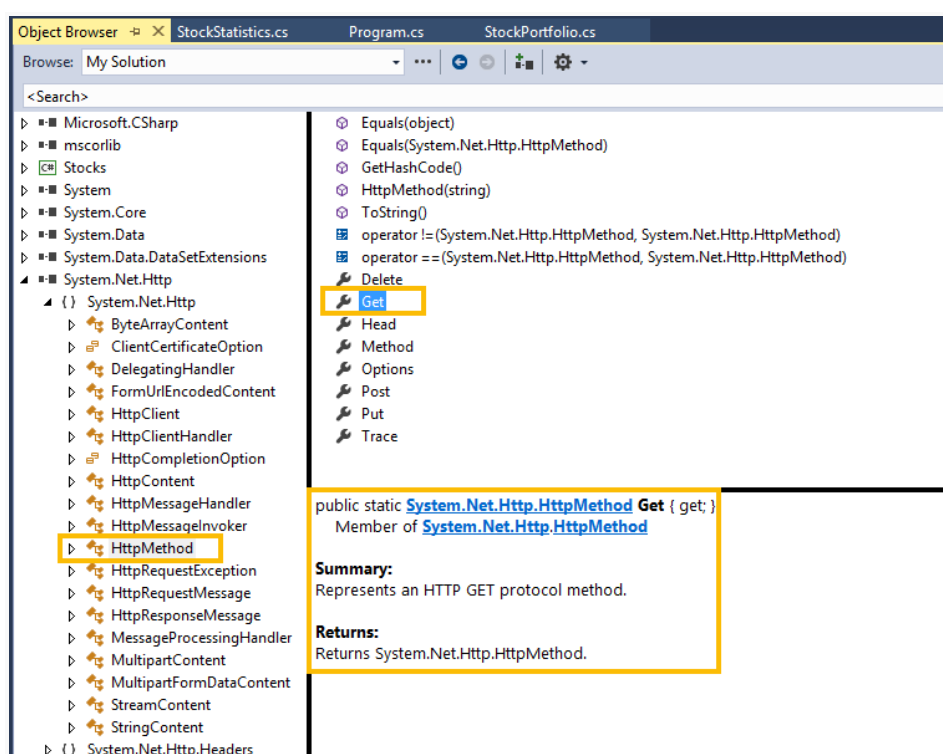




Lokalizacja na dysku.



Podglądanie metod.



Na tym etapie jako przykład do zadania domowego przesyłam państwu projekt.

StudenstPLaygorund. Jako zadanie domowe proszę:

Zadanie

Proszę stworzyć 4 projekty typu .net class Library

1. **Common** gdzie będzie znajdował się interfejs **Izlecieenie**
2. **BeerProcessor** gdzie będzie znajdowała się klasa **zlecieeniePiwo**, która ma implementować interfejs **Izlecieenie**
3. **SandwichProcessor** gdzie będzie znajdowała się klasa **zlecieenieKanapka**, która ma implementować interfejs **Izlecieenie**.
4. Oraz jako ostatni projekt Fabryka, który ma być projektem wykonywalnym, gdzie za pomocą refleksji będą państwo uruchamiać z assembly **BeerProcessor** oraz **SandwichProcessor** metodę **process** związaną ze zleceniem.

Czyli W obu projektach **BeerProcessor** oraz **SandwichProcessor** należy stworzyć klasę reprezentującą zlecenie. Zależne od swojej roli to będzie albo **zlecieenieKanapka/zlecieeniePiwo**. Obie klasy powinny zawierać także właściwość typu string o nazwie **Tytuł**. Interfejs niech zawiera metodę **Process**. która ma być implementowana w klasach **zlecieenieKanapka/zlecieeniePiwo**.

W przypadku **SandwichProcessor** metoda **process** ma wyświetlić tytuł zlecenia oraz w odstępach co 1 sekundę etapy wytwarzania kanapki z salami. (proszę wymyślić kilka np. krojenie smarowanie w moim wypadku wyrzucanie do kosza ab o mary ze mnie kucharz).

W przypadku **BeerProcessor** metoda **process** ma wyświetlić tytuł zlecenia oraz w odstępach co 2 sekundy etapy wytwarzania West coast IPA. Przepis przykładowy:
<https://wkpd.waw.pl/receptury-west-coast-india-pale-ale/>

Następnie mechanizmami refleksji oraz uruchamiania assembly proszę stworzyć program który na podstawie podanej na wejściu ścieżki do assembly **SandwichProcessor** lub **BeerProcessor** stworzy zlecenie wytworzenia Produktu. Jako parametr wejściowy do programu niech też będzie przekazywana nazwa zlecenia. Program po wczytaniu danych ma utworzyć X obiektów implementujących interfejs **Izlecieenie** dla każdego z nich ustawić tytuł a następnie uruchomić metodę **Process**.

Przykład:

Fabryka.exe "c:\\BeerProcessor.dll;Piwo dla Franka" "c:\\SandwichProcessor .dll;Kanapka dla Franka" "c:\\SandwichProcessor .dll;Kanapka Janiny"

Program wraz z przykładowym plikiem bat. zawierającym prezentację działania należy dostarczyć w formie skompresowanej zip.

Zabrania się dodawania w projektach referencji innych niż do projektu **Common**. z wyjątkiem systemowy lub tych znajdujących się w standardowej bibliotece