

# APBD - Ćwiczenia 1

Wykładowca: [pgago@pja.edu.pl](mailto:pgago@pja.edu.pl)

## Zadanie 1

W trakcie pierwszego zadania należy przygotować nowe konto w serwisie GitHub

**`https://github.com/`**. Będziecie wykorzystywać niniejsze narzędzie do końca semestru.

- Należy utworzyć nowe konto wykorzystując do tego mail szkolny. Nazwa konta powinna składać się z numeru indeksu, a w opisie konta należy umieścić imię, nazwisko i numer grupy.
- Dołączenie się do Github classroom (z pomocą linku przekazanego przez prowadzącego ćwiczenia)

1) Wybieramy z listy swój identyfikator

---

Join the classroom:

**APBD\_Zaoczne**

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

Can't find your name? [Skip to the next step →](#)

Identifiers	
lacrit	>

1) Następnie klikamy „Accept the assignment”

---

Your account is linked to lacrit on the roster. If this is wrong, please reach out to your instructor. ×

APBD\_Zaoczne

## Accept the assignment — Cwiczenia\_1


Once you accept this assignment, you will be granted access to the `cwiczenia_1-lacrit12` repository in the [PJATK-Database-department](#) organization on GitHub.

---

Accept this assignment

2) Po kilku minutach wygeneruje nam się repozytorium


---




## You're ready to go!

You accepted the assignment, **Cwiczenia\_1**.

Your assignment repository has been created:

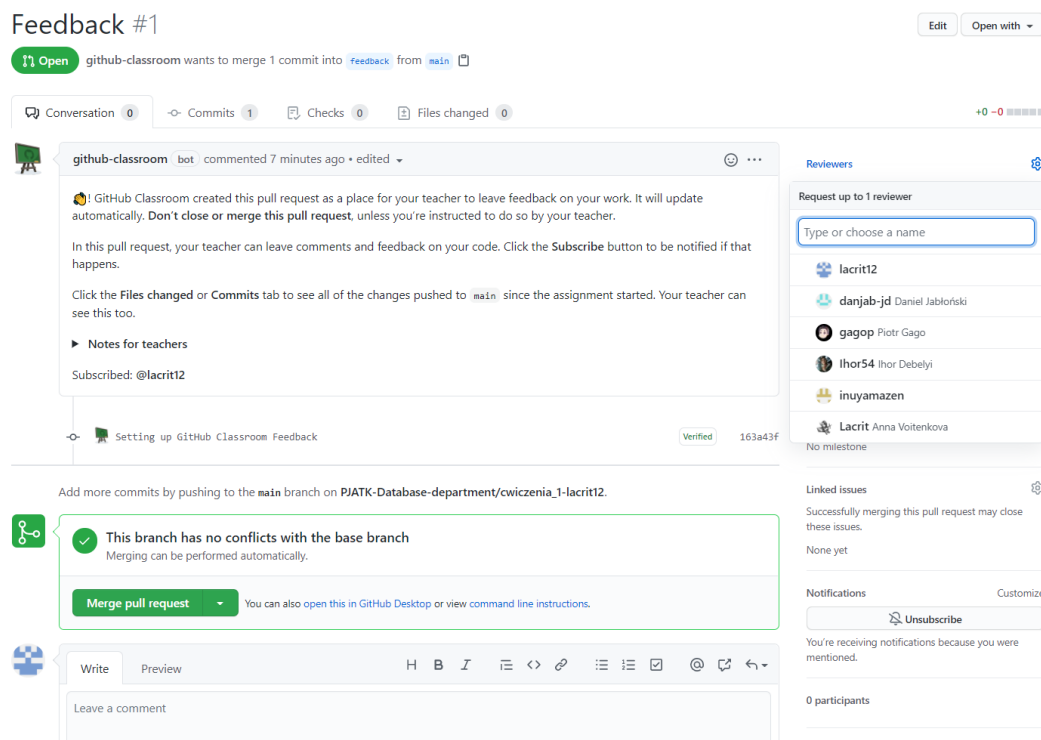
 [https://github.com/PJATK-Database-department/cwiczenia\\_1-lacrit12](https://github.com/PJATK-Database-department/cwiczenia_1-lacrit12)

We've configured the repository associated with this assignment ([update](#)).

 Your assignment is due by **Mar 13, 2021, 23:00 CEST**

## Praca na utworzonym repozytorium

Wygenerowane repozytorium posiada dwie gałęzie. Gałąź „feedback” utworzona w celu otrzymywania odpowiedzi od prowadzącego. Automatycznie został utworzony „pull request” z gałęzi „main” (wcześniej „master”) na „feedback”.



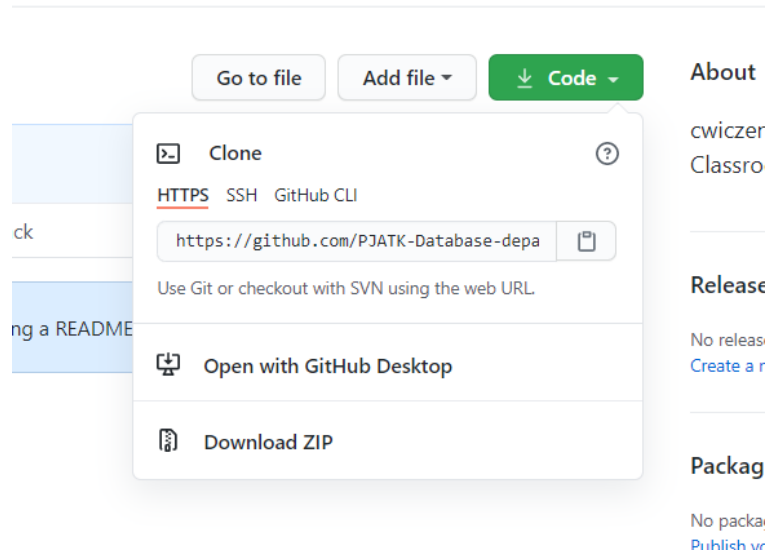
Oznacza to, że każdy zpushowany commit na tą gałąź będzie widoczny wraz ze wszystkimi zmodyfikowanymi plikami.

Po prawej stronie można zaznaczyć prowadzącego, a na dole dodać komentarz (również komentarze można dodawać do konkretnych linii w kodzie w danym pliku).

Żeby zacząć prace nad dzisiejszym projektem należy stworzyć sobie lokalną kopię repozytorium za pomocą komendy

```
git clone [adres repozytorium]
```

Adres repozytorium zobaczymy po kliknięciu w przycisk „Code”:



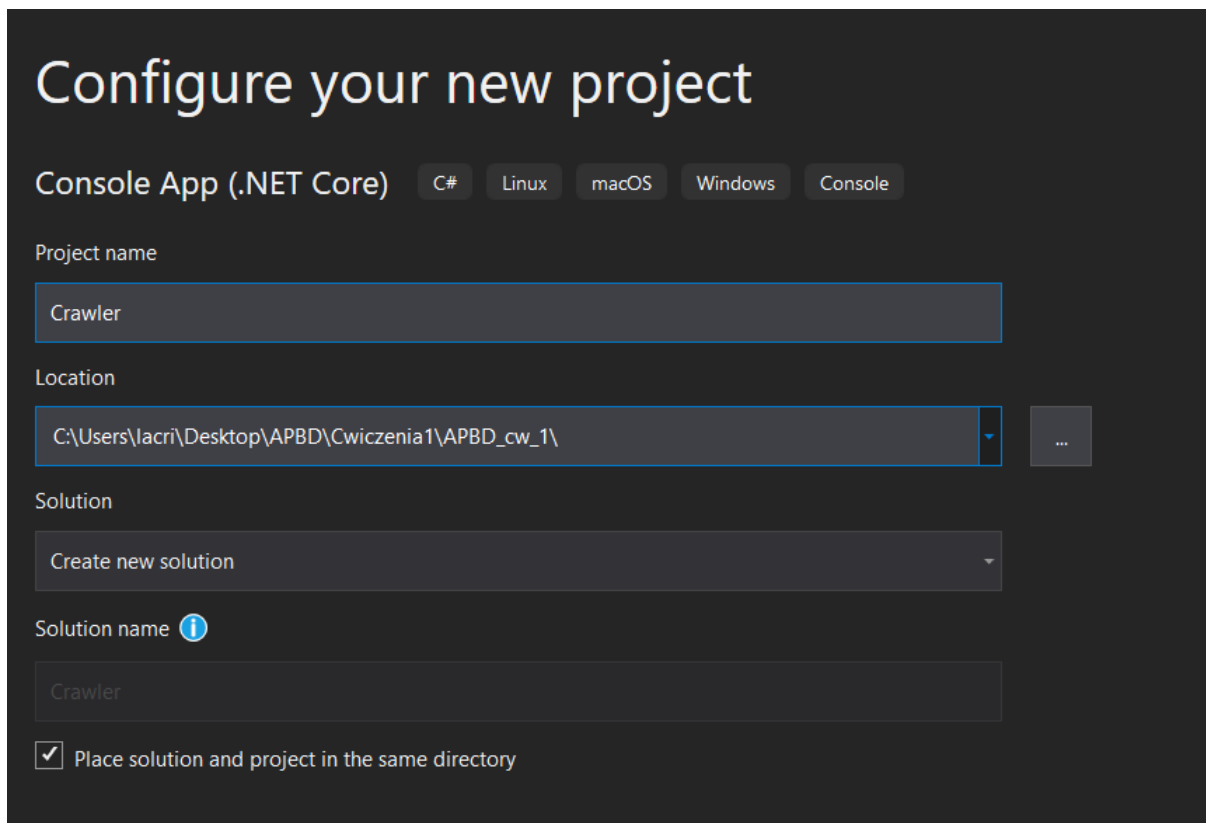
Oczywiście, żeby ta komenda zadziałała musimy zainstalować git (<https://git-scm.com/downloads>)

## Zadanie 2

Niniejsze ćwiczenie będziemy realizować za pomocą Microsoft Visual Studio 2019. Link do pobrania znajdziecie tutaj (<https://visualstudio.microsoft.com/>)

Wszystkie rzuty ekranów są tylko przykładem, jeśli ktoś woli wykonywać ćwiczenia używając innego narzędzia (np. Rider), jest to możliwe, natomiast w tym przypadku musicie znaleźć odpowiednie instrukcje (np. tworzenie nowego projektu itd.).

- Należy stworzyć nowy projekt aplikacji konsolowej wykorzystując platformę .NET Core 5.0. Jako lokalizację projektu wybieracie katalog ze sklonowanym repozytorium



Configure your new project

Console App (.NET Core) C# Linux macOS Windows Console

Project name

Crawler

Location

C:\Users\Iacri\Desktop\APBD\Cwiczenia1\APBD\_cw\_1\

Solution

Create new solution

Solution name ⓘ

Crawler

☒ Place solution and project in the same directory

- Po stworzeniu projektu warto go uruchomić i sprawdzić czy widzicie w konsoli tekst "Hello World!".
- Ten projekt wrzucamy do repo (repozytorium) za pomocą konsoli na gałąź „main”:

```
git status
git add .
git status
git commit -m "Pierwszy commit"
git push
```

```

lacri@DESKTOP-LRGN20F MINGW64 ~/Desktop/APBD/Cwiczenia1/cwiczenia_1-lacrit12 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .vs/
        Crawler/

nothing added to commit but untracked files present (use "git add" to track)

lacri@DESKTOP-LRGN20F MINGW64 ~/Desktop/APBD/Cwiczenia1/cwiczenia_1-lacrit12 (main)
$ git add .

lacri@DESKTOP-LRGN20F MINGW64 ~/Desktop/APBD/Cwiczenia1/cwiczenia_1-lacrit12 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .vs/VSSWorkspaceState.json
        new file:   .vs/cwiczenia_1-lacrit12/v16/.suo
        new file:   .vs/slnx.sqlite
        new file:   Crawler/.vs/Crawler/v16/.suo
        new file:   Crawler/Crawler.csproj
        new file:   Crawler/Crawler.sln
        new file:   Crawler/Program.cs
        new file:   Crawler/obj/Crawler.csproj.nuget.dgspec.json
        new file:   Crawler/obj/Crawler.csproj.nuget.g.props
        new file:   Crawler/obj/Crawler.csproj.nuget.g.targets
        new file:   Crawler/obj/Debug/netcoreapp3.1/.NETCoreApp,Version=v3.1.AssemblyAttributes.cs
        new file:   Crawler/obj/Debug/netcoreapp3.1/Crawler.AssemblyInfo.cs
        new file:   Crawler/obj/Debug/netcoreapp3.1/Crawler.AssemblyInfoInputs.cache
        new file:   Crawler/obj/Debug/netcoreapp3.1/Crawler.assets.cache
        new file:   Crawler/obj/Debug/netcoreapp3.1/Crawler.csprojAssemblyReference.cache
        new file:   Crawler/obj/project.assets.json
        new file:   Crawler/obj/project.nuget.cache

lacri@DESKTOP-LRGN20F MINGW64 ~/Desktop/APBD/Cwiczenia1/cwiczenia_1-lacrit12 (main)
$ git commit -m "Pierwszy commit"
[main 0dad39b] Pierwszy commit
17 files changed, 252 insertions(+)
create mode 100644 .vs/VSSWorkspaceState.json
create mode 100644 .vs/cwiczenia_1-lacrit12/v16/.suo
create mode 100644 .vs/slnx.sqlite
create mode 100644 Crawler/.vs/Crawler/v16/.suo
create mode 100644 Crawler/Crawler.csproj
create mode 100644 Crawler/Crawler.sln
create mode 100644 Crawler/Program.cs
create mode 100644 Crawler/obj/Crawler.csproj.nuget.dgspec.json
create mode 100644 Crawler/obj/Crawler.csproj.nuget.g.props
create mode 100644 Crawler/obj/Crawler.csproj.nuget.g.targets
create mode 100644 Crawler/obj/Debug/netcoreapp3.1/.NETCoreApp,Version=v3.1.AssemblyAttributes.cs
create mode 100644 Crawler/obj/Debug/netcoreapp3.1/Crawler.AssemblyInfo.cs
create mode 100644 Crawler/obj/Debug/netcoreapp3.1/Crawler.AssemblyInfoInputs.cache
create mode 100644 Crawler/obj/Debug/netcoreapp3.1/Crawler.assets.cache
create mode 100644 Crawler/obj/Debug/netcoreapp3.1/Crawler.csprojAssemblyReference.cache
create mode 100644 Crawler/obj/project.assets.json
create mode 100644 Crawler/obj/project.nuget.cache

lacri@DESKTOP-LRGN20F MINGW64 ~/Desktop/APBD/Cwiczenia1/cwiczenia_1-lacrit12 (main)
$ git push
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (29/29), 25.22 KiB | 1.94 MiB/s, done.
Total 29 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.

```

- Sprawdzamy czy zmiany są widoczne w repo (i pull requestcie)

main 2 branches 0 tags Go to file Add file Code

Lacrit Pierwszy commit 0dad39b 22 seconds ago 3 commits

.github	GitHub Classroom Feedback	1 hour ago
.vs	Pierwszy commit	22 seconds ago
Crawler	Pierwszy commit	22 seconds ago

Help people interested in this repository understand your project by adding a README. Add a README

## Feedback #1

Open github-classroom wants to merge 2 commits into feedback from main

Conversation 0 Commits 2 Checks 0 Files changed 17

github-classroom bot commented 1 hour ago • edited

👋 GitHub Classroom created this pull request as a place for your teacher to leave feedback on your work. It will update automatically. **Don't close or merge this pull request**, unless you're instructed to do so by your teacher.

In this pull request, your teacher can leave comments and feedback on your code. Click the **Subscribe** button to be notified if that happens.

Click the **Files changed** or **Commits** tab to see all of the changes pushed to `main` since the assignment started. Your teacher can see this too.

► Notes for teachers

Subscribed: @lacrit12

Setting up GitHub Classroom Feedback Verified 163a43f

+ New changes since you last viewed View changes

Pierwszy commit 0dad39b

Add more commits by pushing to the `main` branch on `PJATK-Database-department/cwiczenia_1-lacrit12`.

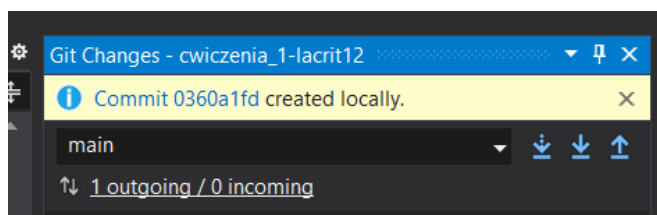
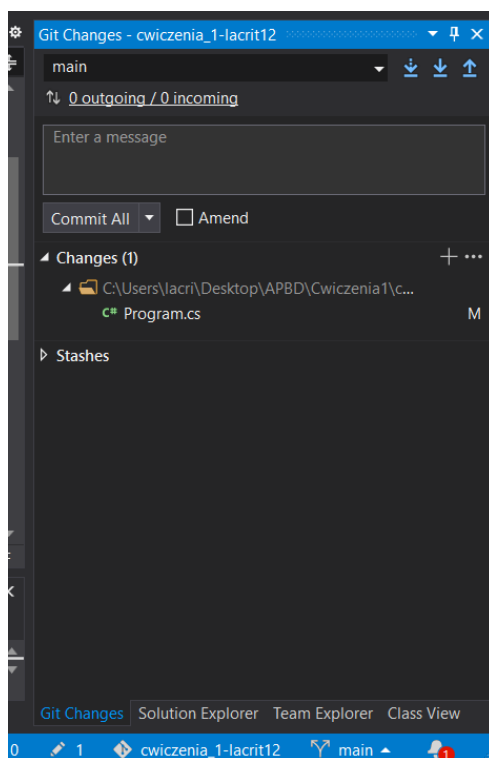
🔗 This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

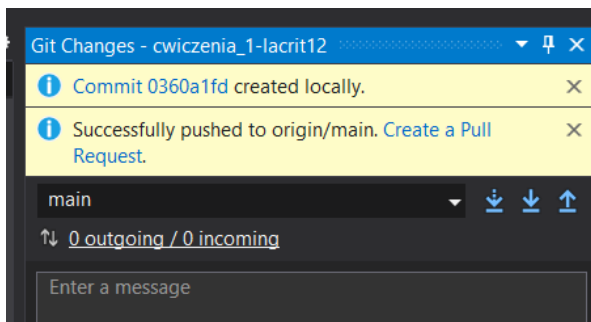
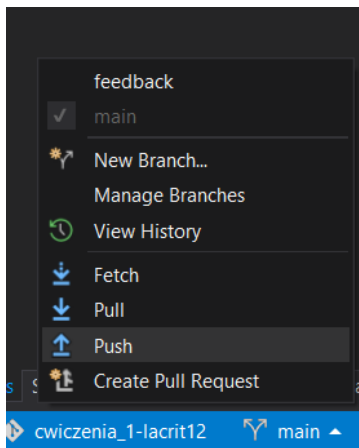
### Zadanie 3

Waszym zadaniem jest przygotowanie prostego "crawlera", który będzie przeszukiwać wybraną stronę WWW i odnajdywać na niej wszystkie adresy email.

- Program powinien przyjmować pojedynczy parametr, który przechowuje adres URL przeszukiwanej strony.
- Następnie za pomocą klasy HttpClient wykonujemy żądanie HTTP GET i pobieramy zawartość kodu źródłowego strony.
- W ostatnim kroku przeszukujemy zawartość strony i wypisujemy na konsoli wszystkie adresy email, które znaleźliśmy.
- Proszę obejrzeć zawartość projektu w oknie Solution Explorer. Ponadto proszę uruchomić aplikację w trybie debugowania i przestudiować działanie punktów przerwania (tzw. breakpoint'ów), które są niezwykle przydatne w procesie debugowania.
- Po zakończeniu należy wykonać za pomocą IDE:





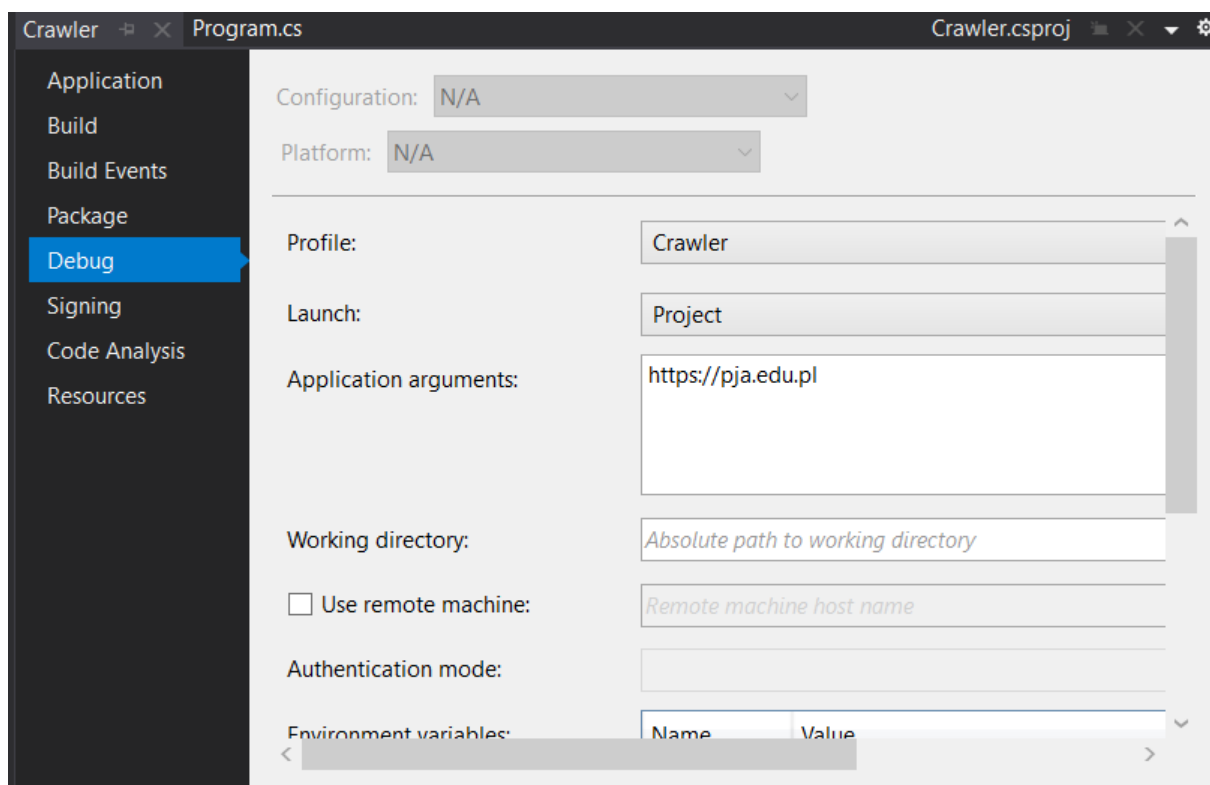


- Proszę się upewnić na stronie GitHub, że Wasze commit'y są widoczne.

W trakcie ćwiczeń proszę zawsze pamiętać o "wypychaniu" (komenda push) swoich commit'ów na serwer. Wasze prace będą sprawdzane wyłącznie poprzez Wasze repozytorium i nie są widoczne dopóki nie zrobicie push'a.

```
namespace Crawler
{
    0 references
    public class Program
    {
        0 references
        public static async Task Main(string[] args)
        {
            string websiteUrl = args[0];
            HttpClient httpClient = new HttpClient();
            HttpResponseMessage response = await httpClient.GetAsync(websiteUrl);
            // ...
        }
    }
}
```

Proszę sprawdzić, jak wygląda zawartość skompilowanej aplikacji na dysku. Proszę przejrzeć zawartość folderu bin aplikacji wykonanej w zadaniu 3. Następnie z pomocą konsoli proszę uruchomić aplikację i przekazać parametr w postaci adresu URL (albo umieścić parametr w ustawieniach projektu Crawler.sln -> Properties -> Debug)



#### Zadanie 4 (Praca domowa)

Dostaliśmy dodatkowe wytyczne związane z aplikacją napisaną w zadaniu 2. W tym celu musimy zmodyfikować kod z zadania 3.

- W sytuacji, kiedy parametr 1 nie został przekazany, powinniśmy zgłosić błąd `ArgumentNullException`
- W sytuacji kiedy przekazany parametr nie jest poprawnym adresem URL, powinniśmy zgłosić `ArgumentException`
- Powinniśmy w poprawny sposób zwalniać zasoby (wykorzystanie metody `Dispose()`) związane z wykorzystaniem klasy `HttpClient`
- W sytuacji, kiedy podczas pobierania wystąpił błąd - wyświetlamy informację "Błąd w czasie pobierania strony".
- W sytuacji, kiedy nie znaleziono żadnych adresów email - wyświetlamy informację "Nie znaleziono adresów email".
- W sytuacji, kiedy znaleźliśmy adresy - wyświetlamy je na konsoli. Chcielibyśmy wyświetlić wyłącznie unikalne adresy email.

## Dodatkowe materiały

Git:

<https://www.youtube.com/watch?v=D6EI7EbEN4Q>

C#

<https://docs.microsoft.com/pl-pl/dotnet/csharp/index?view=aspnetcore-2.1>

[tutorialspoint.com/csharp/index.htm](http://tutorialspoint.com/csharp/index.htm)