APBD - Ćwiczenia 1

pgago@pja.edu.pl

29 lutego 2020

1 Zadanie 1

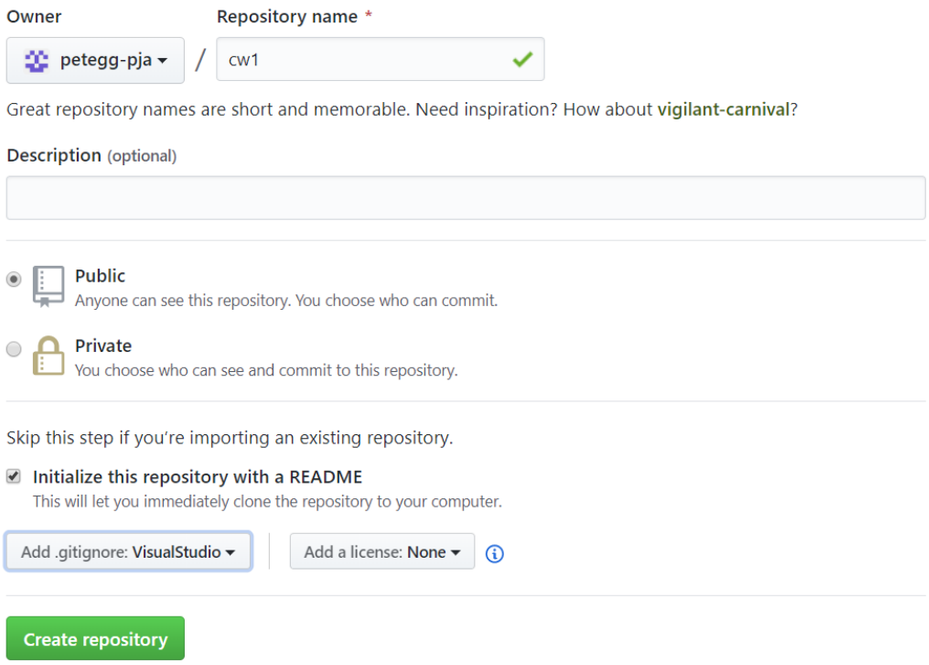
W trakcie pierwszego zadania należy przygotować nowe konto w serwisie GitHub (`https://github.com/`). Będziecie wykorzystywać niniejsze narzędzie do końca semestru.

- Należy utworzyć nowe konto wykorzystując do tego mail szkolny. Nazwa konta powinna składać się z numeru indeksu.
- Następnie należy stworzyć repozytorium o nazwie 'cw1' dla kodu C/. W repozytorium należy umieścić odpowiedni plik **.gitignore** wraz z plikiem '**README.MD**'.
- W pliku **readme** należy umieścić imię, nazwisko, numer grupy studenckiej i numer indeksu.
- Następnie za pomocą poniższej komendy należy sklonować repozytorium na pulpit. W tym celu należy uruchomić konsolę PowerShell. Komendę należy uruchomić przechodząc na pulpit ekranu.

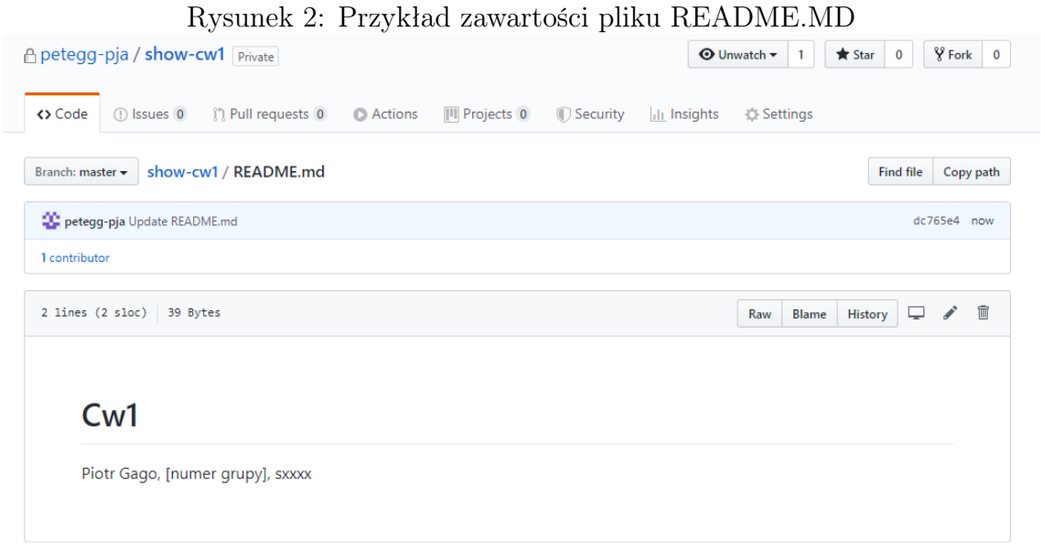
```
git clone [adres repozytorium]
```

- Po ukończeniu ćwiczenia na zakładce Ćommits"powinniście widzieć wykonane ćommity".

Rysunek 1: Ekran tworzenia repozytorium



Rysunek 2: Przykład zawartości pliku README.MD

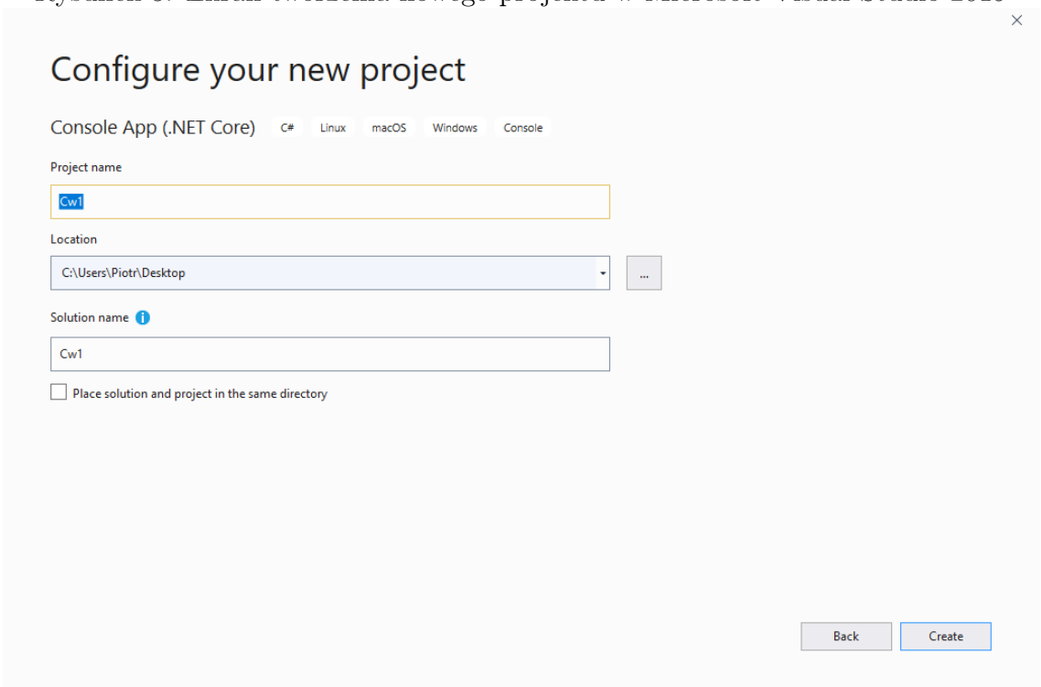


2 Zadanie 2

Niniejsze ćwiczenie będziemy realizować za pomocą Microsoft Visual Studio 2019.

- Należy stworzyć nowy projekt aplikacji konsolowej wykorzystując platformę .NET Core 3.1 (ewentualnie 2.2).

Rysunek 3: Ekran tworzenia nowego projektu w Microsoft Visual Studio 2019



- Nowy projekt należy umieścić w folderze, który jest pod nadzorem systemu kontroli wersji. Jest to folder, który sklonowaliście w poprzednim zadaniu.
- Po stworzeniu projektu warto go uruchomić i sprawdzić czy widzicie w konsoli tekst "Hello World!".
- Waszym zadaniem jest przygotowaniem prostego ćrawlera", 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 konsoli Powershell ćommit". W tym celu należy otworzyć konsolę Powershell i przejść do folderu stanowiącego Wasze repozytorium. Następnie należy skorzystać w odpowiedni sposób z poniższych komend.

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

- 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.

Rysunek 4: Przykładowy fragment kodu

```
namespace Cw1
{
    0 references
    public class Program
    {
        0 references
        public static async Task Main(string[] args)
        {
            var httpClient = new HttpClient();
            var response = await httpClient.GetAsync(args[0]);
            // ...
        }
    }
}
```

3 Zadanie 3

W tym zadaniu proszę sprawdzić jak wygląda zawartość skompilowanej aplikacji na dysku. Proszę przejrzeć zawartość folderu bin aplikacji wykonanej w zadaniu 2. Następnie z pomocą konsoli Powershell proszę uruchomić aplikację i przekazać parametr w postaci adresu URL.

4 Zadanie 4

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

- 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 Ńie znaleziono adresów email".
- W sytuacji kiedy znaleźliśmy adresy - wyświetlamy je na konsoli. Chcielibyśmy wyświetlić wyłącznie unikalne adresy email.

5 Zadanie 5

Na końcu kod z zadania 4 umieszczamy na osobnej gałęzi w naszym repozytorium. W tym momencie nie będziemy już korzystać z konsoli w celu pracy z narzędziem Git. Skorzystamy z okna "Team Explorer". Git jest narzędziem konsolowym, jednak niemal każde IDE posiada wbudowany, graficzny klient Git. Warto jednak zawsze pamiętać, że narzędzia graficzne pozwalają zazwyczaj wykorzystać jedynie niewielki procent wszystkich komend, które oferuje Git.

- Tworzymy nową gałąź o nazwie "dodatkowe-wymagania".
- Umieszczamy nasz kod jako ćommitńa gałęzi "dodatkowe-wymagania".
- wykonujemy komendę "push"i umieszczamy gałąź na serwerze.
- Po wykonaniu zadania proszę sprawdzić na stronie Github.com, czy w zakładce "Branches"widzicie nowo dodaną gałąź.