

Technologie Komponentowe

Piotr Łukasik

p. 373

email: plukasik@agh.edu.pl / lukasik.pio@gmail.com

www.lukasikpiotr.com

Warunki uzyskania zaliczenia

Ocena końcowa z ćwiczeń posiada następujące składowe:

1. Aktywność
2. Kolokwium

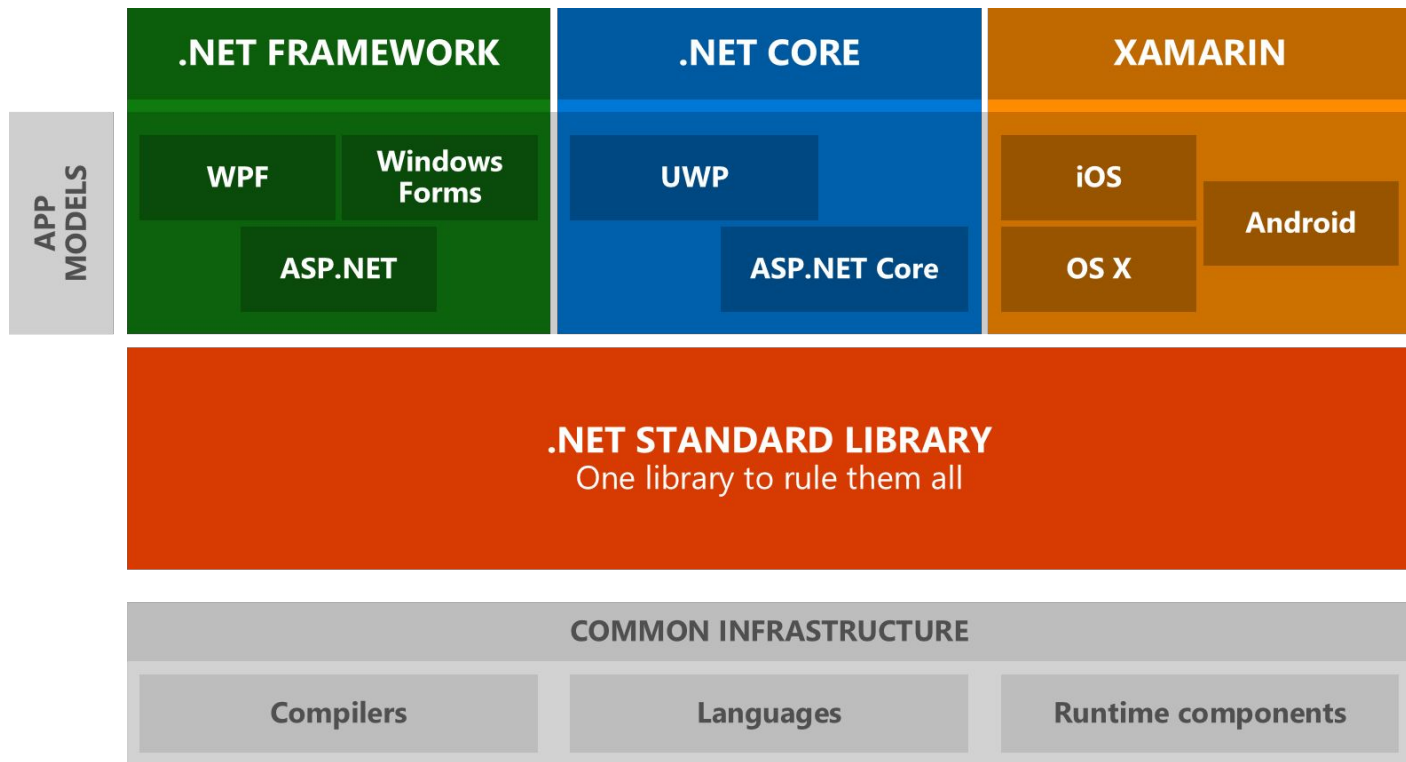
Warunki uzyskania zaliczenia

- W celu uzyskania pozytywnej oceny z ćwiczeń aktywność oraz kolokwium zaliczeniowe muszą być ocenione pozytywnie.
- Kolokwium odbędzie się na ostatnich zajęciach, będą także przewidziane dwa terminy poprawkowe.
- Nieusprawiedliwiona nieobecność na kolokwium zaliczeniowym skutkuje oceną niedostateczną.
- Dodatkowo student ma prawo do maksymalnie jednej nieusprawiedliwionej nieobecności na zajęciach.
- Powyżej 50% nieobecności powoduje nie klasyfikowanie studenta.

Tematyka realizowana na ćwiczeniach

1. C#, .Net Core / .Net Standard
2. WebApi / Swagger
3. Microservices
4. CQRS
5. Propozycje?

.Net what?



Co wybrać?

Use .NET Core for your server application when:

- You have cross-platform needs.
- You are targeting microservices.
- You are using Docker containers.
- You need high-performance and scalable systems.
- You need side-by-side .NET versions per application.

Co wybrać?

Use .NET Framework for your server application when:

- Your app currently uses .NET Framework (recommendation is to extend instead of migrating).
- Your app uses third-party .NET libraries or NuGet packages not available for .NET Core.
- Your app uses .NET technologies that aren't available for .NET Core.
- Your app uses a platform that doesn't support .NET Core.

Przygotowanie środowiska pracy

1. Pobranie .NET Core SDK for Windows:

<https://www.microsoft.com/net/download/windows>

2. Visual Studio Code:

<https://code.visualstudio.com/>

+ rozszerzenie: Omnisharp

3. Git

Przy instalacji proszę zwrócić uwagę na opcje:

a) Use Git from the Windows Command Prompt

b) Use Windows default console window

<https://git-scm.com/download/win/>

Przy próbie instalacji, Windows poprosi Państwa o hasło, proszę wtedy o zgłoszenie się do mnie.

Sprawdzenie instalacji

1. Otwórz: cmd
2. `dotnet --version`
3. Rezultat: 2.1.403
4. `git --version`
5. Rezultat: 2.19.1

Gitlab



1. Utworzenie konta na platformie GitLab:
https://gitlab.com/users/sign_in
2. Username stwórz jako: nick+nrindeksu (np. user250000) WAŻNE!
3. Po zalogowaniu przejdź do: <https://gitlab.com/lukasikpiotr/introduction2018>
4. Po uzyskaniu dostępu kliknij: 
5. Dodaj fork do swojego konta poprzez kliknięcie na swój avatar z username.

Dlaczego fork? Czym różni się fork od branch?

Praca z repozytorium

1. git clone `url_repozytorium`
2. cd `folder_z_repozytorium`
3. git pull (nie jest wymagane, warto jednak sprawdzić)
4. git config user.name "`Ten sam co w gitlab`"
5. git config user.email "`Twoj email`"

VS Code i tworzenie Merge Request

1. Otwórz folder repozytorium w VSCode.
2. Zrób określone zmiany w plikach (polecenia na ćwiczeniach)
3. Wykonaj commit i push do **swojego** repozytorium.
4. Zaloguj się na stronie gitlab.com
5. Wejdź do swojego projektu(Ważne! W zakładce "Projects" widnieją co najmniej dwa projekty, jeden z nich poprzedzony jest nickiem prowadzącego,drugi Twoim, wybierz właściwy)
6. Kliknij w:  Merge Requests 
7. Utwórz merge requesta, gdzie "source branch" to Twój branch, a "target branch" to projekt z którego tworzony był fork.

VS Code

- a) Po sklonowaniu repozytorium, otwórz VS Code.
- b) W VS Code w menu górnym kliknij w File -> Open, otwierając folder z pobranym repozytorium.
- c) Otwórz konsolę: View -> Integrated Terminal.
- d) W celu budowania projektu wpisz (znajdując się w folderze: introduction2018):
`dotnet build`
- e) W celu uruchomienia testów wpisz (znajdując się w folderze: ServiceTest):
`dotnet test`

Zadanie

Opis:

Podane są dwie tablice stringów a1 oraz a2.

Każdy string w tych tablicach tworzony jest przez litery od a do z.

Znajdź:

$\max(\text{abs}(\text{length}(x) - \text{length}(y)))$

gdzie:

- x - string z tablicy a1
- y - string z tablicy a2

Jeżeli a1 lub a2 są pustymi tablicami, zwróć -1.