



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



wsei-csharp201 / [cs-lab02-Implementacja-IEquatable-IComparable-IComparer](#) Public

[Watch](#) 1

[Fork](#) 18

[Star](#) 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

[main](#) 1 branch 0 tags

[Go to file](#)

[Add file](#)

[Code](#)

[kmlenda](#) Update README.md

eca3685 on 18 Oct 2020 7 commits

docs	update links	17 months ago
.gitignore	Initial commit	17 months ago
README.md	Update README.md	17 months ago

README.md

## Implementacja interfejsów `IEquatable`, `IComparable`, `IComparer`

- Autor: Krzysztof Molenda
- Wersja: 2019-10-03

Celem ćwiczenia jest nabycie umiejętności w zakresie tworzenia "dobrze uformowanych typów" (*well formed types*) i implementowania podstawowych interfejsów umożliwiających porównywanie obiektów i stosowanie bibliotecznych algorytmów (sortowanie, wyszukiwanie).

Po wykonaniu ćwiczenia powinieneś:

- umieć implementować równość obiektów (w sensie "taki sam") - implementacja `IEquatable<T>`, przesłonięcie `Equals`, `GetHashCode`,
- umieć określać naturalny porządek dla projektowanego typu - implementacja `IComparable` i wykorzystać go w procesie sortowania i wyszukiwania,
- umieć określać porządek w zbiorze elementów danego typu poprzez wykorzystanie interfejsu `IComparer` oraz delegata `Comparison` i wykorzystać go w procesie sortowania i wyszukiwania,
- umieć zaprogramować prostą metodę generyczną (na przykładzie metody sortującej),
- umieć wykorzystać w praktyce implementację algorytmu wyszukiwania binarnego (metoda `BinarySearch` w klasie `Array` oraz `List<T>`),
- poznać środowisko programowania VisualCode w aspekcie programowania w C#.

Ćwiczenie wykonaj wykorzystując lekkie środowisko VS Code z dodatkiem C# ( [Get started with C# and Visual Studio Code](#) ). Należy zainstalować dodatek: [C# for Visual Studio Code \(powered by OmniSharp\) by Microsoft](#). Dodatkowo, możesz zainstalować dodatki:

- [vscode-solution-explorer](#) by Fernando Escobar
- [.NET Core Test Explorer](#)

Wszystkie polecenia tworzenia czy uruchamiania możesz jednak wydawać z poziomu terminala. Więcej informacji: <https://docs.microsoft.com/en-US/dotnet/core/tools/> (polecenia: `dotnet new sln`, `dotnet new console`, `dotnet new classlib`, `dotnet new mstest`, `dotnet new xunit`, `dotnet add ...`, `dotnet run`, `dotnet test`, ...).

Opisy innych rozszerzeń VS Code przydatnych dla programisty .Net znajdziesz w [Top 5 VS Code Extensions for .Net Core](#). Więcej informacji o pracy z C# z wykorzystaniem VS Code znajdziesz w dokumentacji Microsoft: [VS Code. Working with C#](#)

Obowiązkowa literatura: Michaelis, Rozdział 10. Well formed types

Wykonaj kolejno podane kroki.

### About

Ćwiczenie w zakresie implementowania tożsamości instancji (`IEquatable`) oraz porównywania (`IComparable`, `IComparer`, `Comparison`)

[Readme](#)

0 stars

1 watching

18 forks

### Releases

No releases published

### Packages

No packages published



© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)