

# Środowisko programistyczne na przykładzie Visual Studio

Piotr Dobosz

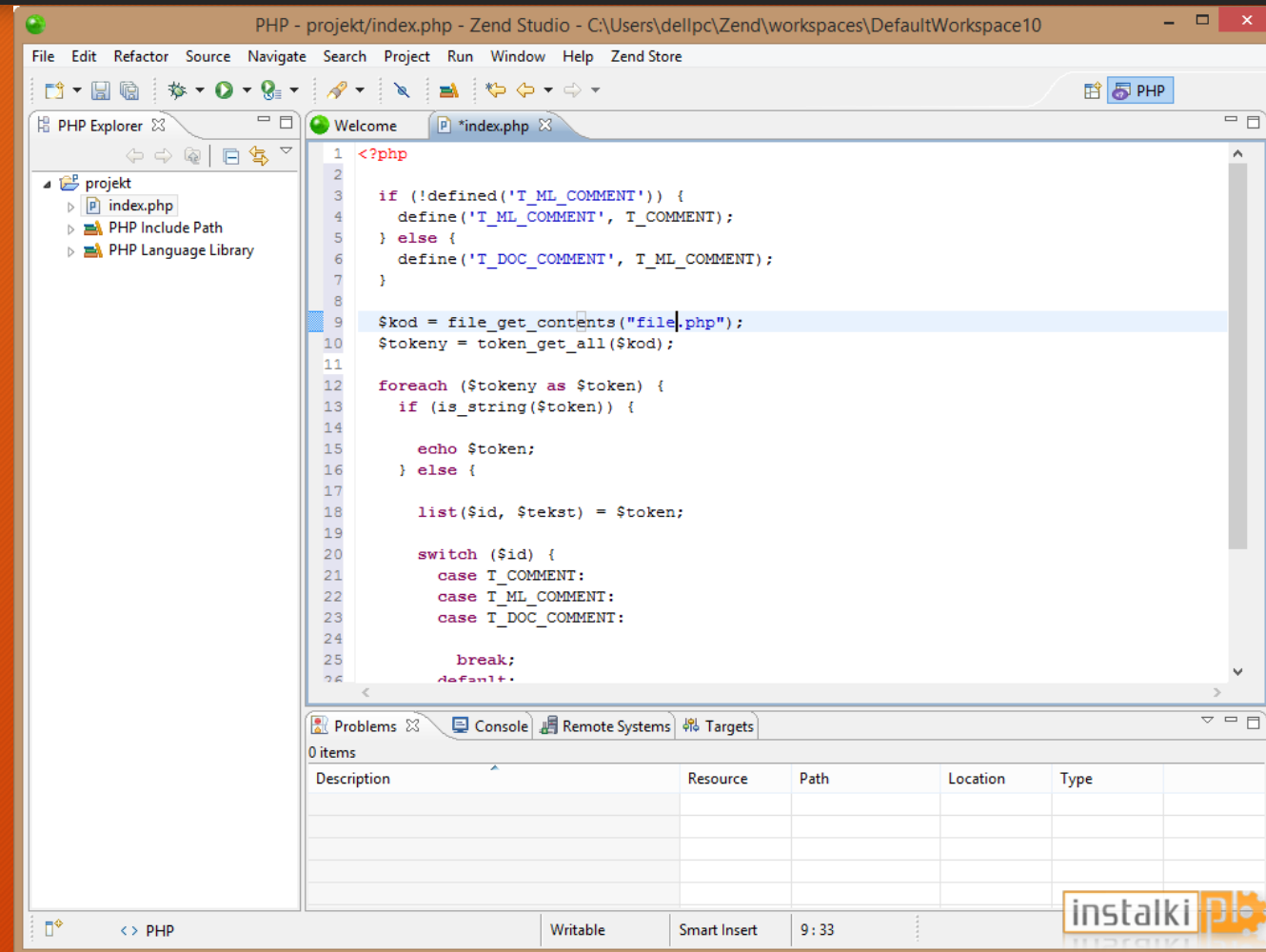
# Środowisko programistyczne

- Program wspomagający tworzenie oprogramowania
- Bazowo "musi" wspierać kompilację/uruchamianie
- Wsparcie dla czytelności kodu
- Elementy ułatwienia pisania kodu

# Zintegrowane środowisko programistyczne (IDE - Integrated Development Environment)

- Program wspierający programowanie
- Posiadający szereg narzędzi...
- ... lub możliwości współpracy z narzędziami minimum jednego języka programowania
- Niemal każde środowisko tego typu posiada tzw. Intelligent code completion (bardziej znane jako IntelliSense)
- Posiadanie integracji z kompilatorem/środowiskiem uruchomieniowym
- Odpluskwiacz (Debugger)

# Zend Studio



# JetBrains

The screenshot shows the IntelliJ IDEA IDE interface. The top toolbar includes the IDEA logo, a search icon, and several utility icons. The breadcrumb navigation shows the path: `intellij-community > platform > core-api > src > com > intellij > lang > folding > LanguageFolding`. The left sidebar displays a project tree for `core-api [intellij.platform.core]`, with the `LanguageFolding` class selected under the `com.intellij.lang.folding` package. The main editor window displays the source code for `LanguageFolding.java`, which includes a package declaration, imports, and several methods. A tooltip is visible over the `builder.buildFoldRegions` call on line 79, showing the return type `FoldingDescriptor[]` and a link to the documentation.

```
49  /**
50   * Only queries base language results if there are no extensions for originally requested
51   */
52   @NotNull
53   @Override
54   public List<FoldingBuilder> allForLanguage(@NotNull Language language) {
55     for (Language l = language; l != null; l = l.getBaseLanguage()) {
56       List<FoldingBuilder> extensions = forKey(l);
57       if (!extensions.isEmpty()) {
58         return extensions;
59       }
60     }
61     return Collections.emptyList();
62   }
63
64   @NotNull
65   public static FoldingDescriptor[] buildFoldingDescriptors(@Nullable FoldingBuilder builder
66     if (!DumbService.isDumbAware(builder) && DumbService.getInstance(root.getProject()).isDumb
67     return FoldingDescriptor.EMPTY;
68   }
69
70   if (builder instanceof FoldingBuilderEx) {
71     return ((FoldingBuilderEx)builder).buildFoldRegions(root, document, quick);
72   }
73   final ASTNode astNode = root.getNode();
74   if (astNode == null || builder == null) {
75     return FoldingDescriptor.EMPTY;
76   }
77
78   return |
79     builder.buildFoldRegions(ASTNode node, Document document) FoldingDescriptor[]
80     FoldingDescriptor.EMPTY (com.intellij.lang... FoldingDescriptor[]
81     Dot, space and some other keys will also close this lookup and be inserted into editor >>
```

# Eclipse

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows a project structure with folders like 'bin', 'docs', and 'src'. Under 'src', there is a 'Container' folder containing 'Exception', 'ApplicationFactory.php', 'ErrorHandlerFactory.php', and 'Zend\Expressive\Container'. The 'WhoopsFactory.php' file is selected.
- Code Editor:** Displays the code for the `__invoke` method in `WhoopsFactory.php`. The code is as follows:

```
45     @param ContainerInterface $container
46     @return Whoops
47
48     /-
49     /-
50     public function __invoke(ContainerInterface $container)
51     {
52         $config = $container->has('config') ? $container->get('config') : [];
53         $config = isset($config['whoops']) ? $config['whoops'] : [];
54
55         $whoops = new Whoops();
56         $whoops->writeToOutput(false);
57         $whoops->allowQuit(false);
58         $whoops->pushHandler($container->get('Zend\Expressive\WhoopsPageHandler'));
59         $this->registerJsonHandler($whoops, $config);
60         $whoops->register();
61     }
62 }
```
- Code Completion Popup:** A popup is visible over the `$whoops->allowQuit(false);` line. It lists several methods from the `Whoops\Run` class. The `allowQuit($exit) : bool - Whoops\Run` method is highlighted. The popup also shows the method signature: `Whoops\Run::allowQuit(bool|int $exit=null) : bool`, a description: "Should Whoops allow Handlers to force the script to quit?", and the parameter: `bool|int $exit`. The return type is `bool`.
- Bottom Bar:** Shows the file type as 'PHP', the editor mode as 'Writable', and the 'Smart Insert' feature. The cursor position is indicated as '59 : 18'.

# Visual Studio

Test Explorer

UserSentiment (18 tests) 1 failed		
UserSentimentAnalysis... (18) 1 sec	53	
UserSentimentAnal... (18) 1 sec	54	
EmojiTests (6) 244 ms	56	
EmojiClothingTest 1 ms	57	
EmojiExtraSpecial... < 1 ms	58	
EmojiFaceSearchTest 77 ms	59	
EmojiHeartsTest 1 ms	60	
SpecificEmojiNullT... 153 ms	61	
TearsOfJoyTest 10 ms	62	
HomeControllerTe... (6) 1 sec	63	
TwitterDataModel... (6) 1 ms	64	

```

Match match = new Regex(emojiToSearch).Match(text);
if (match.Success)
{
    numberFound++;
}

return numberFound;
}

3 references | 1/1 passing | kendrahavens, 35 days ago | 1 author, 1 change
public string TearsOfJoySearch(List<string> tweetsToSearch)
{
    string specificEmoji = tweetsToSearch.Find(text => text.Contains("😭"));
    return specificEmoji;
}

5 references | 4/5 passing | Allison Buchholtz-Au, 29 days ago | 2 authors, 4 changes
public int SpecificEmoji(List<string> emojiToSearch, string text)
{
    int numberFound = 0;
    for (int i = 0; i < emojiToSearch.Count; i++)
    {
        Match match = new Regex(emojiToSearch[i]).Match(text);
        if (match.Success)
        {
            numberFound++;
        }
    }

    return numberFound;
}

```

SpecificEmojiNullTest Copy All

Source: EmojiTests.cs line 101

SpecificEmojiNullTest

Message:

125% No issues found

Team Explorer

Home | VS2019

Azure DevOps

VS2019

Project

Web Portal Task Board

- Changes
- Branches
- Pull Requests
- Sync
- Tags
- Work Items
- Builds
- Settings

Solutions

New... | Open... | Show Folder View

UserSentiment.sln

Solution Explorer Team Explorer

# Visual Studio - wstęp

- Pozwala na pisanie aplikacji:
  - konsolowych
  - sieciowych
  - graficznych
  - mobilnych
  - chmurowych
  - bazodanowych
  - wielu innych, możliwych do napisania

# Visual Studio - warianty

- Społecznościowy (Community) - darmowa; podobna do płatnej wersji, jednak z pewnymi ograniczeniami - zarówno organizacyjnymi w firmie, jak i technologicznymi
- Profesjonalna (Professional) - wersja płatna, zawierająca dodatkową obsługę języków opisowo-bazodanowych oraz zapewnia pełną obsługę programów bazodanowych i chmurowych (Azure)
- Organizacyjna (Enterprise) - główną różnicą od poprzedniej wersji jest skala zastosowania - możliwa jest praca grupowa

# Wersje Visual Studio

Obsługiwane funkcje	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
	<a href="#">Pobierz bezpłatnie</a>	<a href="#">Kup</a>	<a href="#">Kup</a>
⊖ Obsługiwane scenariusze użycia	●●●○	●●●●	●●●●
Indywidualni deweloperzy	●	●	●
Środowiska edukacyjne	●	●	●
Badania akademickie	●	●	●
Udział w projektach typu „open source”	●	●	●
Organizacje niebędące przedsiębiorstwami, <sup>1</sup> do 5 użytkowników	●	●	●
Enterprise		●	●

# Wersje Visual Studio

Obsługiwane funkcje	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
	<a href="#">Pobierz bezpłatnie</a>	<a href="#">Kup</a>	<a href="#">Kup</a>
⊕ Obsługiwane scenariusze użycia	●●●○	●●●●	●●●●
Obsługa platformy deweloperskiej <sup>2</sup>	●●●●	●●●●	●●●●

# Wersje Visual Studio

Obsługiwane funkcje	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
	Pobierz bezpłatnie	Kup	Kup
⊕ Obsługiwane scenariusze użycia	●●●○	●●●●	●●●●
Obsługa platformy deweloperskiej <sup>2</sup>	●●●●	●●●●	●●●●
⊖ Zintegrowane środowisko projektowe	●●●○	●●●○	●●●●
Weryfikacja aktywnych zależności			●
Architektoniczne diagramy warstwowe			●
Weryfikacja architektury			●
Klonowanie kodu			●
CodeLens	●	●	●
Peek Definition	●	●	●
Refaktoryzacja	●	●	●
One-Click Web Deployment	●	●	●
Model Resource Viewer	●	●	●
Wizualizowanie rozwiązań za pomocą wykresów zależności i map kodu	● <sup>3</sup>	● <sup>3</sup>	●
Wielowersyjność kodu	●	●	●

# Wersje Visual Studio

Zaawansowane debugowanie i diagnostyka ⊖	●●○○	●●○○	●●●●
IntelliTrace			●
Integracja mapy kodu z debugerem			●
Analiza zrzutu pamięci na platformie .NET			●
Metryki kodu	●	●	●
Debugowanie grafiki	●	●	●
Statyczna analiza kodu	●	●	●
Centrum wydajności i diagnostyki	●	●	● <sup>4</sup>
Debugger migawki			●
Debugowanie z przechodzeniem w czasie (wersja zapoznawcza)			●

# Wersje Visual Studio

⊖ Narzędzia testowania	●○○○	●○○○	●●●●
Testowanie jednostek na żywo			●
IntelliTest			●
Microsoft Fakes (izolacja testu jednostki)			●
Pokrycie kodu			●
Testowanie jednostek	●	●	●

# Wersje Visual Studio

Programowanie dla wielu platform	●●○○	●●○○	●●●●
Osadzone zespoły			●
Xamarin Inspector			●
Xamarin Profiler			●
Zdalny symulator systemu iOS dla systemu Windows	●	●	●
Udostępnianie kodu między systemami Android i iOS przy użyciu środowiska Xamarin	●	●	●
Natywni projektanci interfejsu użytkownika dla systemów iOS i Android	●	●	●
Xamarin.Forms	●	●	●

# Wersje Visual Studio

⊖ Funkcje i narzędzia współpracy	●●●●	●●●●	●●●●
Tworzenie scenorysów programu PowerPoint	●	●	●
Przegląd kodu	●	●	●
Wstrzymywanie/wznawianie zadań	●	●	●
Team Explorer (obsługa narzędzi deweloperskich innych firm)	●	●	●
Visual Studio Live Share	●	●	●

# Wersje Visual Studio

- Przypisy:
- 1) Przedsiębiorstwa to organizacje z więcej niż 250 komputerami lub rocznym przychodem przekraczającym 1 mln dolarów.
- 2) Aplikacje klasyczne systemu Windows, aplikacje uniwersalne systemu Windows, Internet (ASP.NET), Office 365, aplikacje biznesowe, Azure Stack, tworzenie bibliotek dla wielu platform w języku C++, Python, Node.js, .NET Core, narzędzia platformy Docker
- 3) Może otwierać diagramy wygenerowane w innych wersjach programu Visual Studio w trybie tylko do odczytu.
- 4) Obejmuje funkcję Tier Interaction Profiling.

# Okno Visual Studio

The image shows a screenshot of the Visual Studio IDE with several callout boxes pointing to specific features:

- Create a new project**: Points to the 'Start' button in the top toolbar.
- Run your code**: Points to the 'Start' button in the top toolbar.
- Launch Live Share**: Points to the 'Live Share' icon in the top right corner.
- Send feedback**: Points to the 'Send Feedback' icon in the top right corner.
- Add controls to your UI**: Points to the 'Toolbox' on the left side of the IDE.
- Manage your Azure resources**: Points to the 'Cloud Explorer' on the left side of the IDE.
- Manage files, projects, and solutions**: Points to the 'Solution Explorer' on the right side of the IDE.
- Collaborate on code projects with your team**: Points to the 'Team Explorer - Connect' panel on the right side of the IDE.

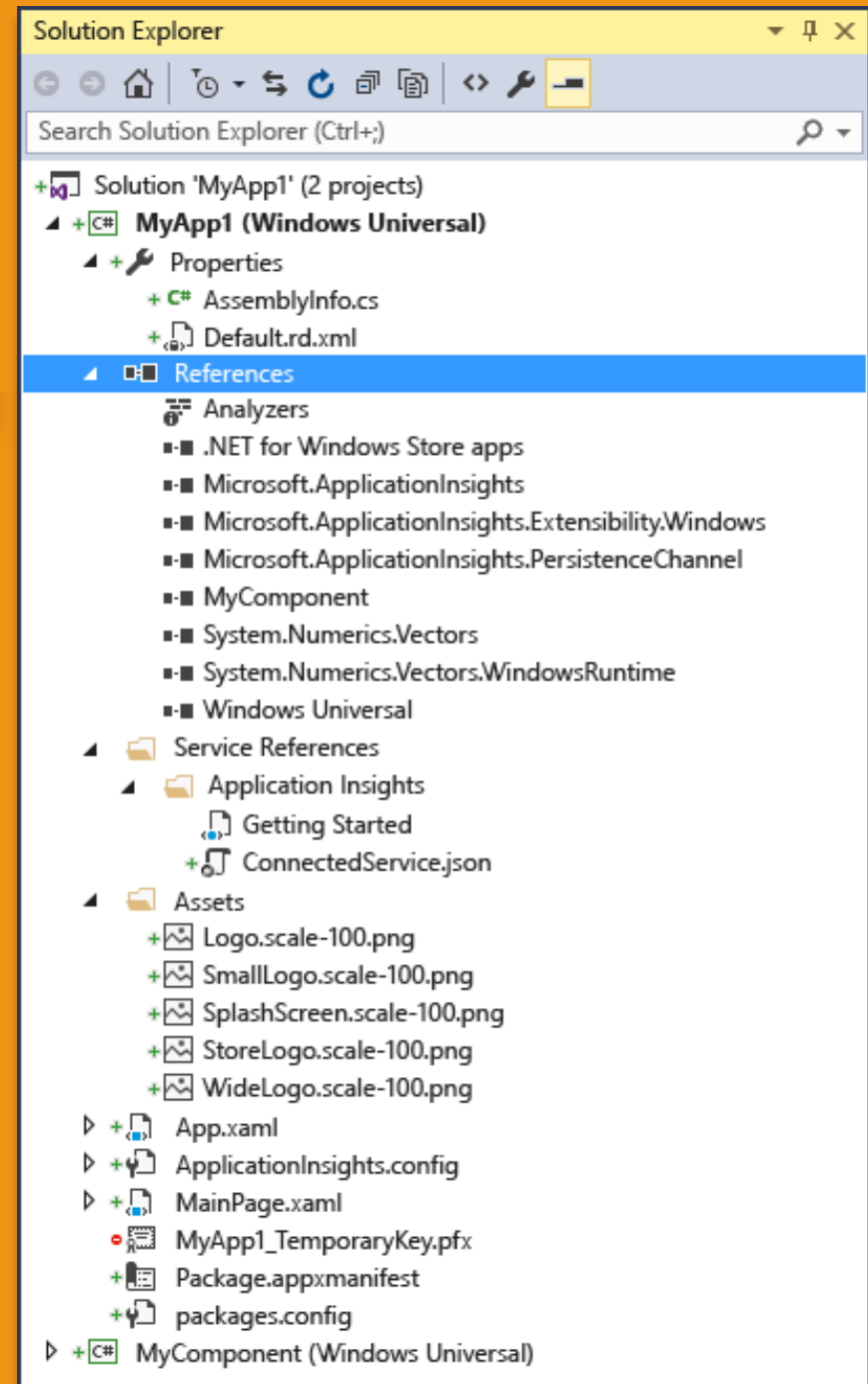
The central editor shows a C# code file named 'Calendar.cs' with the following content:

```
1 using System;
2 using System.Runtime.CompilerServices;
3
4 [assembly: InternalsVisibleTo("QuickTest")]
5 namespace QuickDate
6 {
7     1 reference
8     internal class Calendar
9     {
10        0 references
11        static void Main(string[] args)
12        {
13            DateTime now = GetCurrentDate();
14            Console.WriteLine($"Today's date is {now}");
15            Console.ReadLine();
16        }
17        2 references
18        internal static DateTime GetCurrentDate()
19        {
20            return DateTime.Now.Date;
21        }
22    }
23 }
```

# Okno Visual Studio - opis

- Solution Explorer - pozwala na zarządzanie projektem oraz plikami kodu
- Team Explorer - narzędzie umożliwia pracę grupową/zarządzanie rozwiązaniami pracy grupowej (np. poprzez git)

# Solution Explorer



# Edytor kodu - możliwości

- Syntax Coloring
- Error and Warning Marks
- Brace Matching
- Structure Visualizer
- Line Numbers
- Change Tracking
- Selecting Code and Text
- Zoom
- Virtual Space
- Printing
- Global Undo and Redo

# Zaawansowane możliwości edytora

- Format Document
- Format Selection
- Tabify Selected Lines
- Untabify Selected Lines
- Make Uppercase/Lowercase
- Move selected Lines Up/Down
- Delete Horizontal White Space
- View White Space
- Word Wrap
- Comment/Uncomment Selection
- Increase/Decrease Line Indent
- Select Tag (Content)

# Podkreślanie/szybka akcja

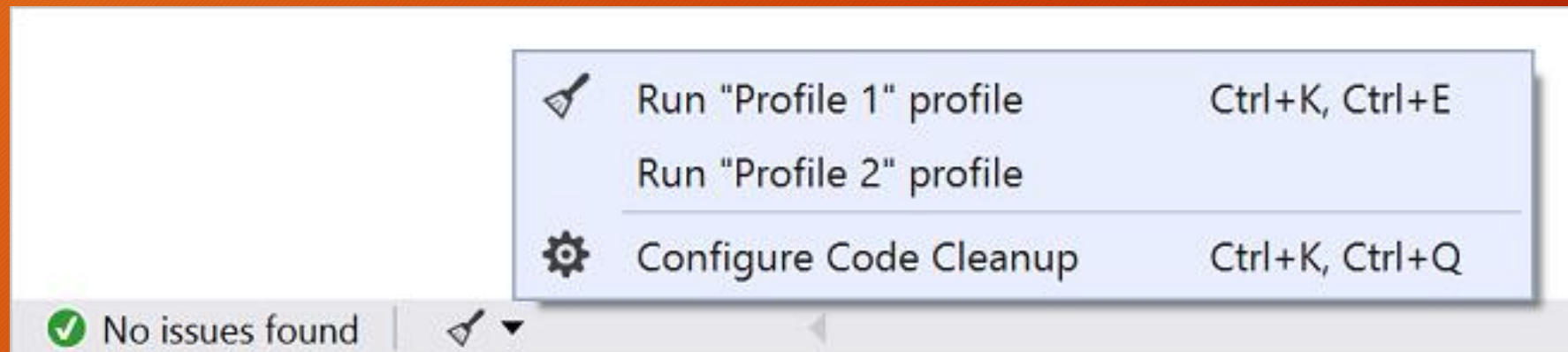
```
public int Calculate() => 3 * Math.Log8
```



'Math' does not contain a definition for 'Log8'

Show potential fixes (Alt+Enter or Ctrl+.)

# Code cleanup



The screenshot displays the Code Cleanup tool interface. At the bottom, a status bar shows a green checkmark icon followed by the text "No issues found". To the right of the status bar is a toolbar with a paintbrush icon and a dropdown arrow. A context menu is open, listing three actions:

- Run "Profile 1" profile (Ctrl+K, Ctrl+E)
- Run "Profile 2" profile
- Configure Code Cleanup (Ctrl+K, Ctrl+Q)

# Refactoring

```
Frame rootFrame = Window.Current.Content as Frame;

// Do not repeat
// just ensure
if (rootFrame
{
    // Create
    rootFrame
    rootFrame.

    if (e.Previous
{
```

	View Designer	Shift+F7
	Quick Actions and Refactorings...	Ctrl+.
	Rename...	Ctrl+R, Ctrl+R
	Organize Usings	
	Insert Snippet...	Ctrl+K, Ctrl+X
	Surround With...	Ctrl+K, Ctrl+S
	Peek Definition	Alt+F12
	Go To Definition	F12

# IntelliSense

```
DateTime now = DateTime.
```

- FromOADate
- IsLeapYear
- MaxValue
- MinValue
- Now**
- Parse
- ParseExact
- ReferenceEquals
- SpecifyKind



```
DateTime DateTime.Now { get; }
```

Gets a `DateTime` object that is set to the current date and time on this computer, expressed as the local time.

# Visual Studio Search

The screenshot shows the Visual Studio interface with a search for 'nuget' in the Help menu. The search results are displayed in a dropdown menu on the right side of the screen. The search term 'nuget' is highlighted in a red box in the search input field.

The search results are categorized as follows:

- Menus (4)**
  - Project → Manage NuGet Packages...
  - Tools → NuGet Package Manager → Package Manager Console
  - Tools → NuGet Package Manager → Manage NuGet Packages for Solution...
  - Tools → NuGet Package Manager → Package Manager Settings
- Options (3)**
  - NuGet Package Manager → General
  - NuGet Package Manager → Package Sources
  - Text Editor → C# → Advanced (Suggest usings for types in NuGet packages,...
- Install (1)**
  - NuGet targets and build tasks (Individual Component)
- NuGet Packages (1)**
  - Search Online for NuGet Packages matching 'nuget'

At the bottom of the search results, there is a note: "Press Ctrl+Q to show 'Menus' results." and a help icon.

```
1 using System;
2
3 namespace ConsoleApp1
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello World!");
10        }
11    }
12 }
13
```

# Live share

- Pozwala na zdalne udostępnianie kodu
- Możliwość współdzielenia się rozwiązaniem projektu poprzez sieć
- Rozwiązania głosowe, terminalowe, sieciowe i inne - wszystko dostępne w czasie rzeczywistym

# Call hierarchy

Call Hierarchy

My Solution

- GetFilePathFromLink(string) (NotInToc.Program)
  - Calls To 'GetFilePathFromLink'
    - IsInToc(System.IO.FileInfo, System.IO.FileInfo) (NotInToc.Program)
      - Calls To 'IsInToc'
        - ListOrphanedTopics(System.Collections.Generic.List<System.IO.FileInfo>, System.Collections.Generic.Dictionary<string, System.Collections.Generic.List<System.IO.FileInfo>>) (NotInToc.Program)
          - Calls To 'ListOrphanedTopics'
          - ListPopularFiles(System.Collections.Generic.List<System.IO.FileInfo>, System.Collections.Generic.Dictionary<string, System.Collections.Generic.List<System.IO.FileInfo>>) (NotInToc.Program)
          - ListOrphanedImages(string, System.Collections.Generic.Dictionary<string, System.Collections.Generic.List<System.IO.FileInfo>>) (NotInToc.Program)

Call Sites	Location ▲
relativePath = GetFilePathFromLink(line);	Program.cs - (299, 40)
relativePath = GetFilePathFromLink(line);	Program.cs - (308, 40)

Package Manager Console Error List Output Test Results Call Hierarchy

# CodeLens

▲ FabrikamFiber.Web.Tests\Controllers\CustomersControllerTest.cs (2) 

🔗 28 : controller.Create(new Customer());

🔗 38 : controller.Create(null);

▲ FabrikamFiber.Web\Helpers\GuardHelper.cs (1)

🔗 16 : controller.Create(new Customer());

[Show on Code Map](#) | [Collapse All](#)

3 references |  0/2 passing | Francis Totten, 3 hou

```
public ActionResult Create(Customer
{
    if (customer == null)
    {
```






GuardHelper.cs (16,18)

🔗 FabrikamFiber.Web.Helpers.GuardHelper.GuardCustomer()

```
14 {
15     CustomersController controller = new CustomersController(null);
16     controller.Create(new Customer());
17
18 }
```

# Go to definitions

```
public double Calculate() => 3 * Math.Log(10);
```


	Quick Actions and Refactorings...	Ctrl+.
	Rename...	F2
	Remove and Sort Usings	Ctrl+R, Ctrl+G
	Peek Definition	Alt+F12
	Go To Definition	F12
	Go To Implementation	Ctrl+F12
	Find All References	Ctrl+K, R
	View Call Hierarchy	Ctrl+K, Ctrl+T

# Peek definition

```
class Program
{
    static void Main(string[] args)
    {
        double number = Calculate();
    }
}

public static double Calculate() => 3 * Math.Log(10);
}
```

Program.cs\* [minimize] [close]

15  
16  public static double Calculate() => 3 \* Math.Log(10);  
17 }  
18 }

# Refaktoryzacja kodu

```
0 references
7 static void Main(string[] args)
8 {
9     Console.WriteLine("\nWhat is your name?");
10    var username = Console.ReadLine();
11
12    Console.WriteLine($"Hello {name}!");
13    Console.WriteLine($"Hello {username}!");
14 }
15 ...
16
```

Rename 'name' to 'username' ▶

Use discard '\_'

Suppress IDE0059 ▶

Preview changes

# Refaktoryzacja kodu

```
12  
13 DateTime now = DateTime.Now;  
14  
15  
16 Console.WriteLine(now.DayOfYear);  
17 Console.WriteLine(now.DayOfYear);  
18 }  
19 }  
20 }  
21
```

Inline temporary variable ▶

...

DateTime now = DateTime.Now;  
int day = now.DayOfYear;  
int day = DateTime.Now.DayOfYear;

...

Preview changes

# Odpluskwanie

```
Program.cs  + X
C# HelloWorld
1  using System;
2
3  namespace HelloWorld
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("\nWhat is your name?");
10             var username = Console.ReadLine();
11             Console.WriteLine($"Hello {username}!");
12
13             int day = DateTime.Now.DayOfYear;
14
15             Console.Write("Day of year: ");
16             Console.WriteLine(day);
17         }
18     }
19 }
```

# Odpluskwianie

```
static void Main(string[] args)
{
    Console.WriteLine("\nWhat is your name?");
    ▶ var username = Console.ReadLine();
    Console.WriteLine(username | 🔍 "Georgette" ⇄ ");
```

# Materiały i źródła

- <https://www.geeksforgeeks.org/introduction-to-visual-studio/>
- <https://visualstudio.microsoft.com/pl/vs/compare/>
- <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- <https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2019>
- <https://docs.microsoft.com/en-us/visualstudio/ide/writing-code-in-the-code-and-text-editor?view=vs-2019>

# Grafiki

- [https://www.jetbrains.com/idea/img/screenshots/idea\\_overview\\_5\\_1@2x.png](https://www.jetbrains.com/idea/img/screenshots/idea_overview_5_1@2x.png)
- [https://www.instalki.pl/components/com\\_djcatalog/images/f8d308885d48f18a944591d34894350czend.png](https://www.instalki.pl/components/com_djcatalog/images/f8d308885d48f18a944591d34894350czend.png)
- <https://www.eclipse.org/pdt/img/shot6-min.png>