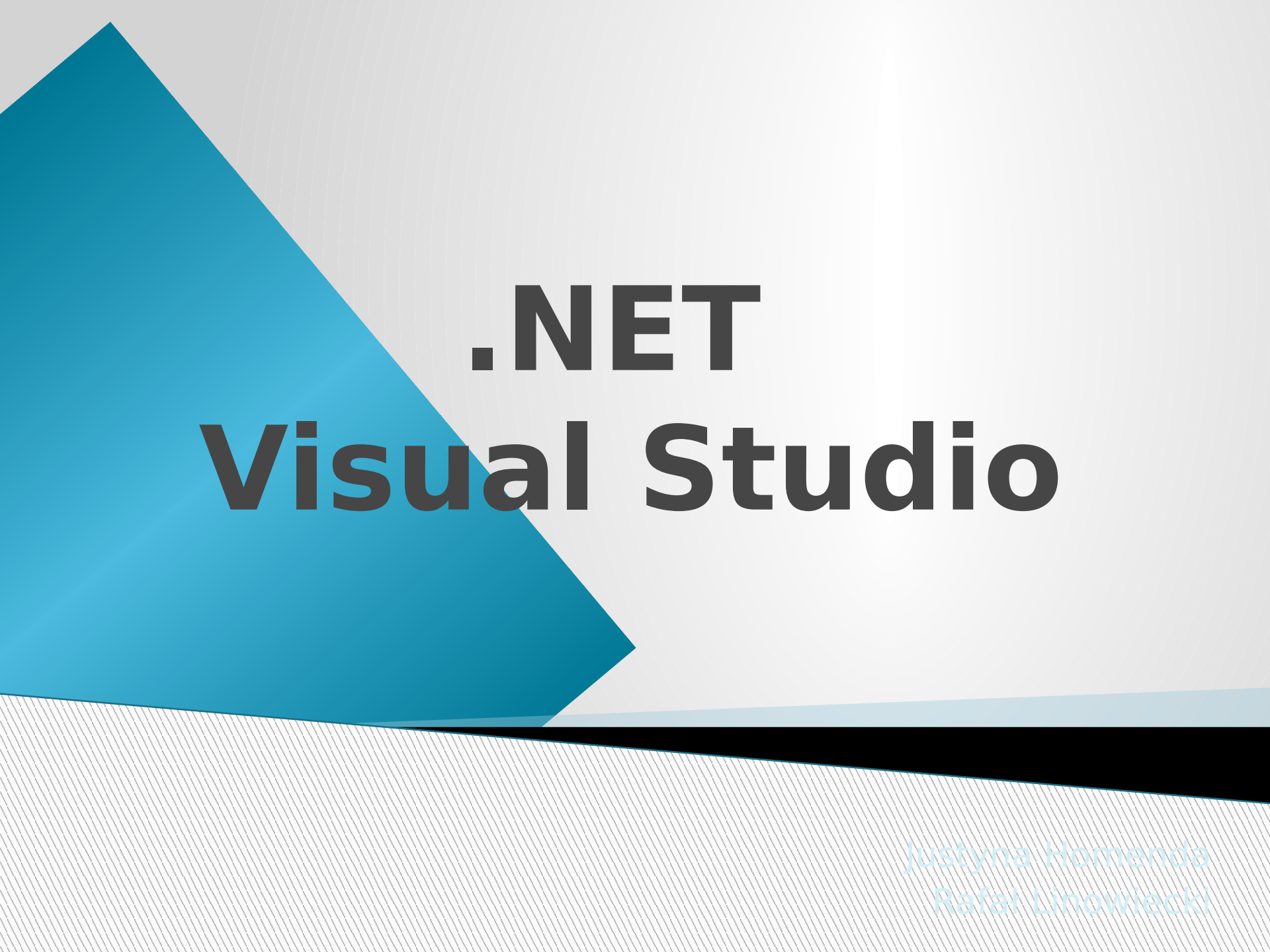




.NET Visual Studio

Justyna Homenda
Rafał Linowiecki



Microsoft

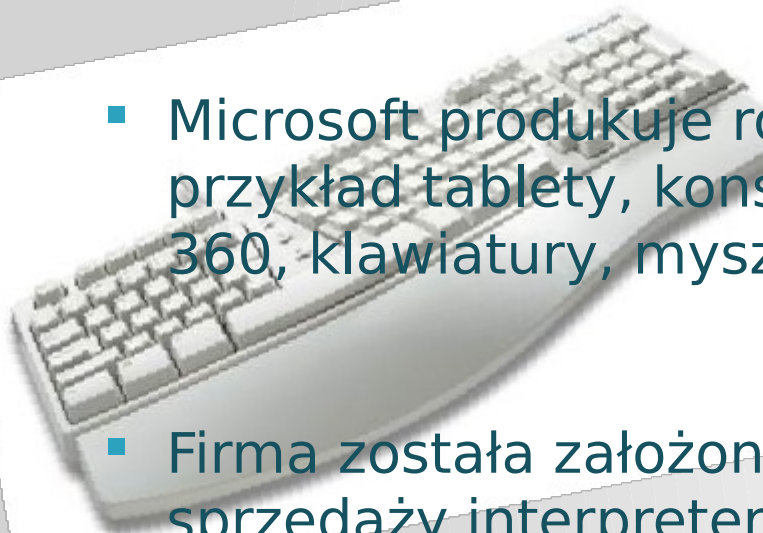
»» Informacje wstępne

Słowem wstępu... Microsoft

- jedna z największych na świecie firm z branży technologii informacyjnych
- szczególnie znana jako producent systemów operacyjnych **MS-DOS, Microsoft Windows** i oprogramowania biurowego **Microsoft Office**
- Spółka publiczna z siedzibą w Redmond w stanie Waszyngton, założona w 1975 roku przez **Billa Gatesa** i **Paula Allena**



- W 2012 roku firma posiadała 10 000 patentów
- Podstawowym produktem Microsoftu są systemy operacyjne dla serwerów i komputerów osobistych o nazwie Windows.
- Najnowszymi systemami operacyjnym Microsoft są **Windows 8** dla zwykłych zastosowań i system serwerowy **Windows Server 2012**.



- Microsoft produkuje również sprzęt komputerowy, na przykład tablety, konsolę Xbox oraz jej następcę Xbox 360, klawiatury, myszy itp.
- Firma została założona w celu wyprodukowania i sprzedaży interpretera **języka BASIC**
- Polski oddział Microsoftu został zarejestrowany w **1992** jako *Microsoft spółka z o.o*
- W pierwszych miesiącach istnienia polskiego oddziału firma sprzedawała głównie system operacyjny **MS-DOS**





Platforma .NET

»» Informacje ogólne

Wprowadzenie do .NET'a

.NET jest to platforma programistyczna, która wspomaga tworzenie, rozwój oraz testowanie aplikacji.



Obecnie pod nazwą .NET kryją się dwa główne środowiska:

- .NET Framework – który obejmuje wspólne środowisko uruchomieniowe (CLR) oraz bibliotekę klas .NET Framework. (framework –opisuje szkielet aplikacji, który zawiera poszczególne mechanizmy działania programu)
- Visual Studio – zintegrowane środowisko programistyczne (IDE) do tworzenia oprogramowania.

Trochę ciekawostek...



Firma Microsoft używała wcześniej terminu .NET do określenia szerokiej gamy produktów, np.

- .NET Enterprise **Servers** – czyli zgrupowanie technologii serwerowych (np. SQL Server) obecnie znane jako Windows Server System
- .NET My Services - środowisko kierowane dla budujących aplikacje i serwisy internetowe (chciano umożliwić użytkownikom dostęp do zasobów online)

Microsoft miał plany aby w przyszłości pojawiły się takie środowiska jak Windows .NET oraz Office .NET, jednakże gdy nie przyjęła się nazwa .NET My Services zrezygnowano z tego pomysłu.

Krótko o historii .Net

Przed pojawieniem się platformy .NET:

- programista, który korzystał ze środowisk programistycznych dostarczanych przez MS skazany był na Windows API (lub innych klas i funkcji opartych na API).
- wiele problemów z:
 - poprawnym zarządzaniem pamięci
 - obsługą błędów
 - brakiem mechanizmów kontroli
 - nieprzenośnością programów między różnymi procesorami lub niewykorzystywanie przez aplikacje wszystkich możliwości procesora
 - nieprzenośność aplikacji między wersjami systemu ze względu na różnice w zestawie i działaniu funkcji z zestawu API (także problem z przenaszalnością na różne języki programowania)



Powstanie inicjatywy .NET (**2000 rok**) – większość problemów znika, dzięki czemu programista może skupić się nad logiką aplikacji, czyli nad tym, co najważniejsze...

.NET Framework

Wprowadzone zmiany w porównaniu do Windows API ułatwiają tworzenie stabilnych, niezawodnych i wydajnych aplikacji dla systemu Windows, witryn internetowych pracujących na serwerze internetowych usług informacyjnych, a także usług internetowych XML Web Services.

Główne zalety środowiska .NET:

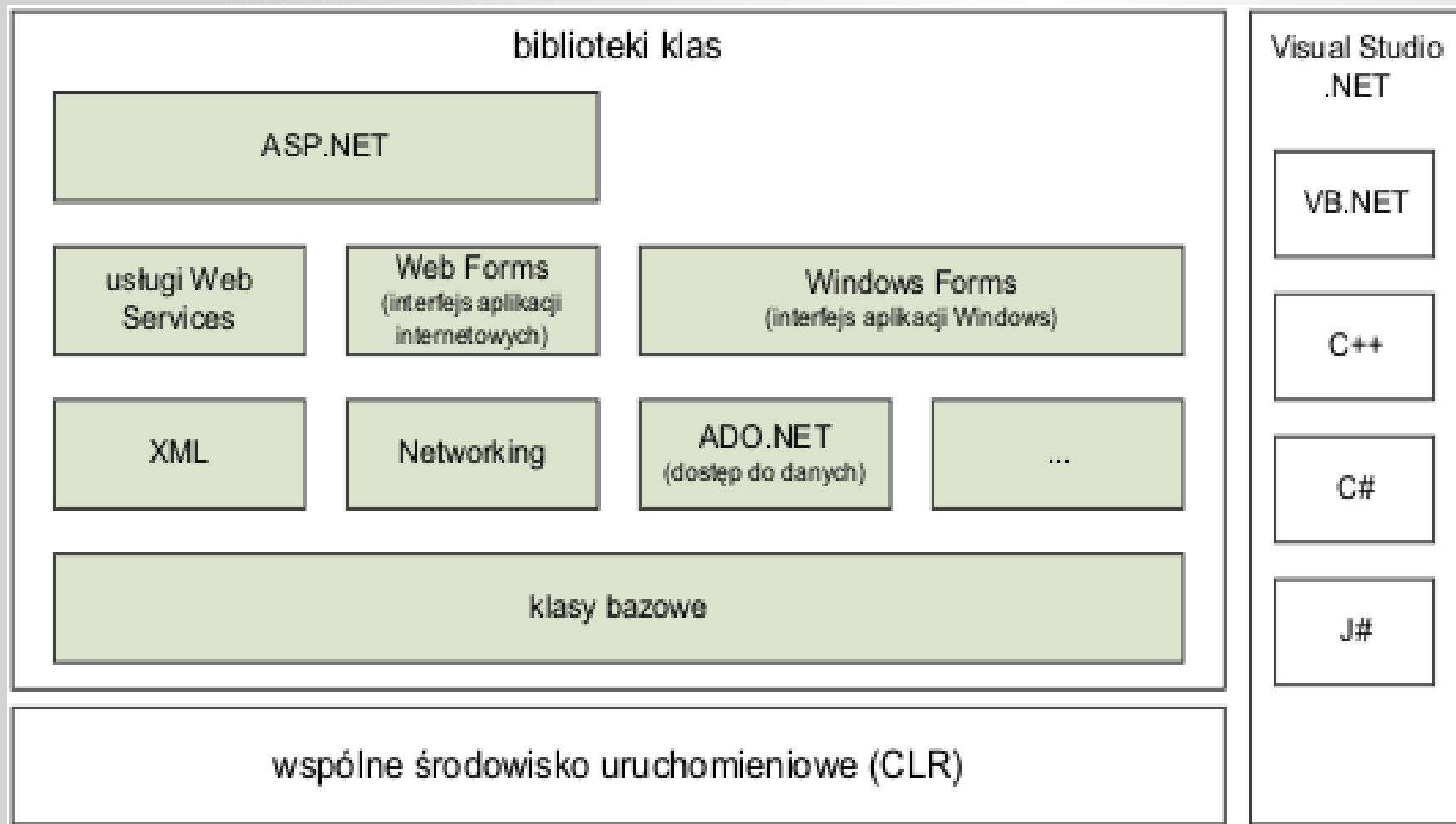
- zgodność z istniejącymi standardami (html, xml, soap)
- łatwa modyfikacja istniejących zbiorów klas – np. poprzez dziedziczenie dostępnych już klas i ich modyfikacja pod konkretny problem
- zarządzanie elementami systemu – kodem aplikacji, pamięcią oraz zabezpieczeniami
- taka sama funkcjonalność dla różnych języków programowania



Główne składniki platformy .NET

- wspólne środowisko uruchomieniowe (CLR)
- nowe, hierarchicznie zorganizowane biblioteki klas - ułatwiają obsługę graficznych interfejsów użytkownika, dostęp do baz danych i plików oraz komunikację sieciową. Biblioteką szczególnie zasługującą na uwagę jest **ASP.NET**, zastępująca dotychczasową technologię Active Server Pages (ASP). Biblioteka ta wykorzystywana jest do tworzenia dynamicznych witryn internetowych i stanowi podstawę do budowy aplikacji internetowych oraz usług XML Web Services.

Struktura platformy .NET



Platforma .NET

» Wspólne środowisko
uruchomieniowe - CLR

Wspólne środowisko uruchomieniowe - CLR

Common Language Runtime - to środowisko uruchomieniowe dla platformy .NET, przewidziane do pracy na wielu systemach operacyjnych i maszynach, jak MS Windows, Unix czy Macintosh.

Jest to maszyna wirtualna, która wykonuje kod wyrażony w Common Intermediate Language (CIL).



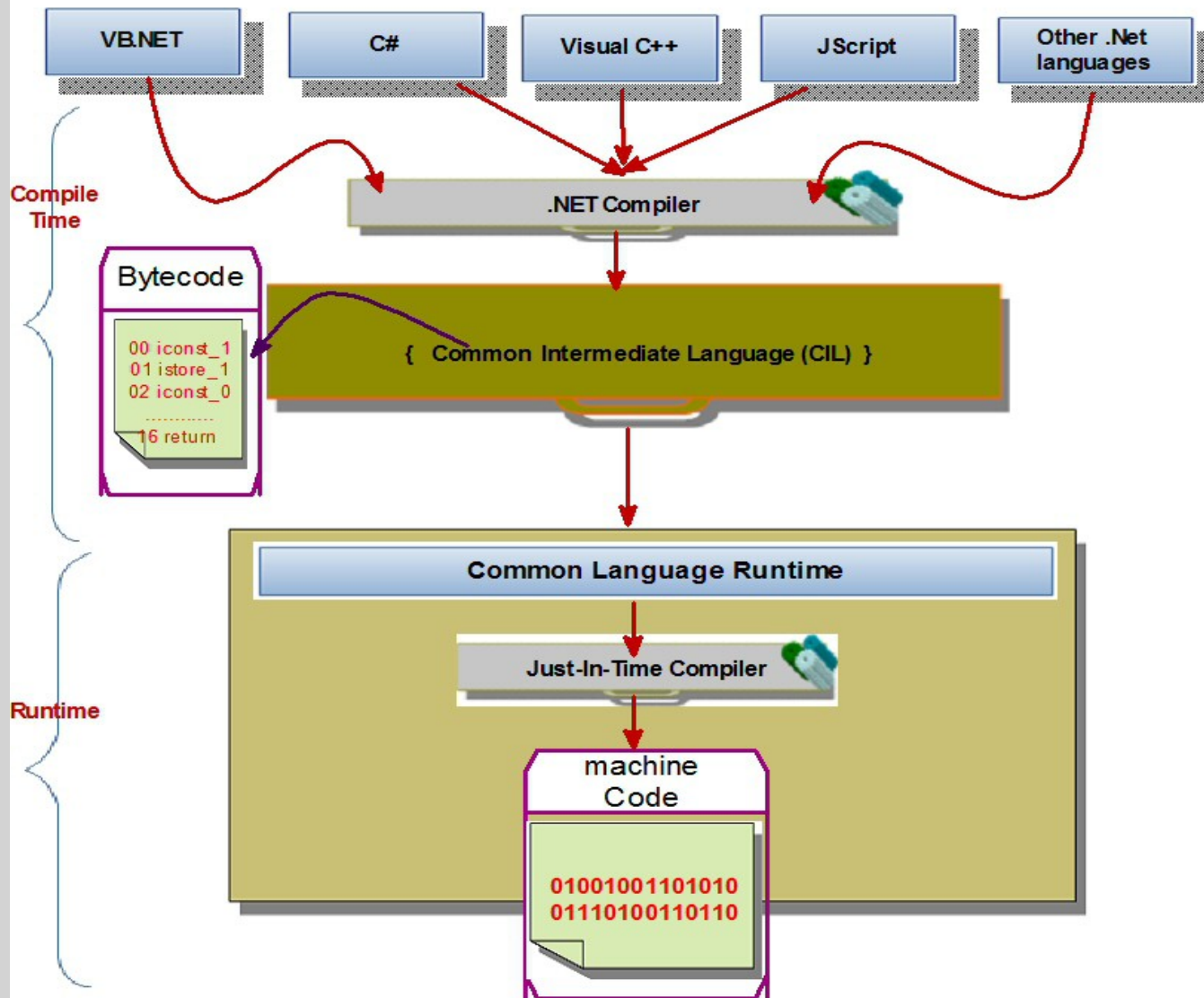
CLR - kompilowanie i wykonanie kodu

Środowisko CLR kompiluje i wykonuje zapisany w *standardowym języku pośrednim Microsoft (MSIL)* kod aplikacji zwany *kodem zarządzanym* (ang. *managed code*), zapewniając wszystkie podstawowe funkcje konieczne do działania aplikacji.

Podstawowym elementem CLR jest standardowy zestaw typów danych, wykorzystywanych przez wszystkie języki oparte na CLR, a także standardowy format metadanych, służących do opisu oprogramowania wykorzystującego te typy danych. CLR zapewnia także mechanizmy umożliwiające pakowanie kodu zarządzanego w jednostki zwane *podzespołami*.

.NET Framework Common Language Runtime

Design: Prabir Choudhury



Trochę ciekawostek...



W CLR wbudowane są także mechanizmy kontroli bezpieczeństwa wykonywania aplikacji — bezpieczeństwo oparte na uprawnieniach kodu (Code Access Security — CAS) oraz bezpieczeństwo oparte na rolach (Role-Based Security — RBS).

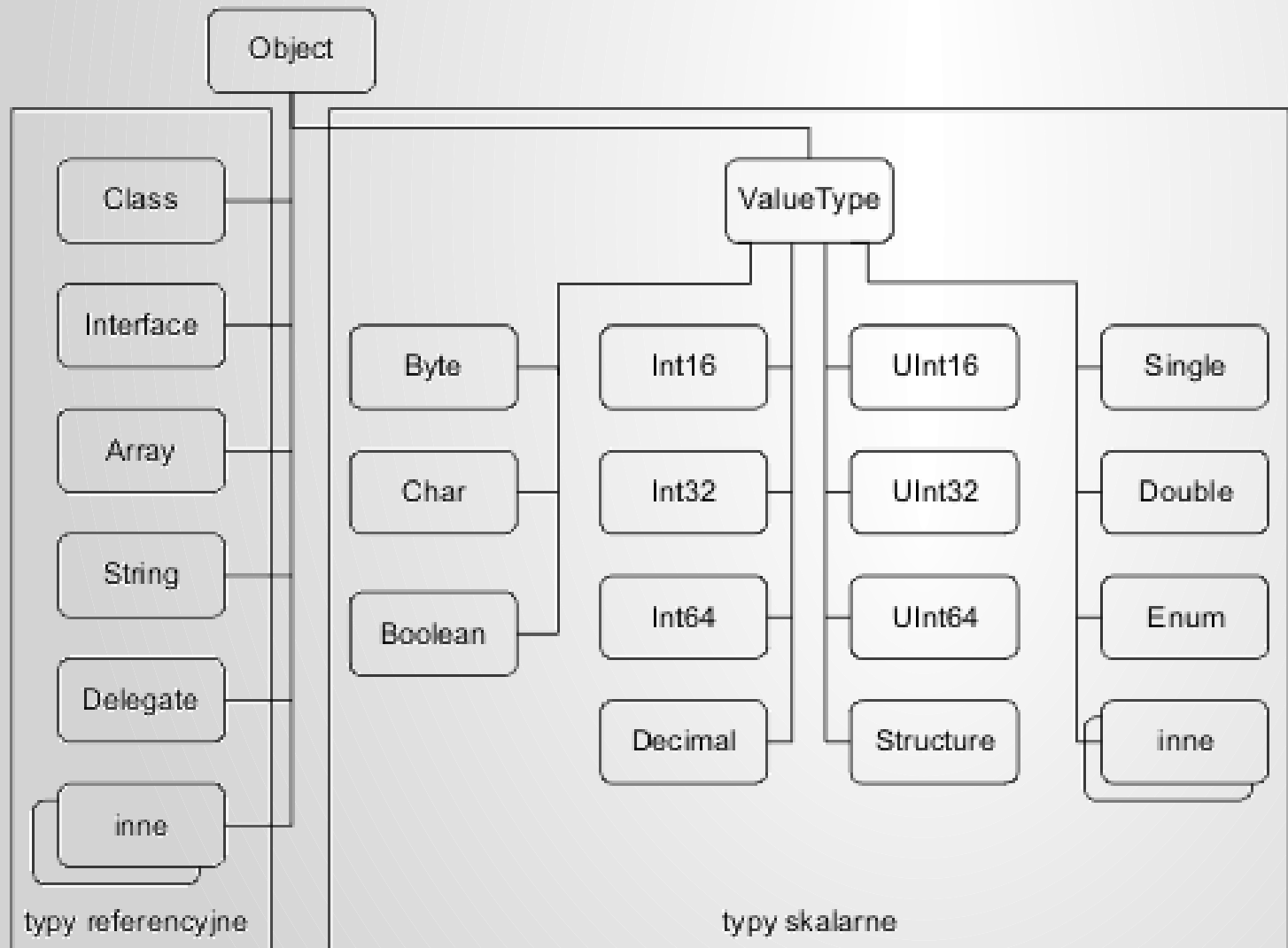
*CLR zawiera również moduł czyszczenia pamięci - **garbage collection**, który automatycznie usuwa zarządzane obiekty, dla których nie ma już referencji.*

Obsługa typów danych- CTS

Wspólny zestaw typów danych (*Common Type System*) - nie jest związany z żadną składnią lub słowami kluczowymi — definiuje jedynie zestaw typów danych, który może być wykorzystywany przez wiele języków. Każdy język zgodny z CLR może używać dowolnej składni, ale musi korzystać przynajmniej z części typów danych zdefiniowanych przez CTS.

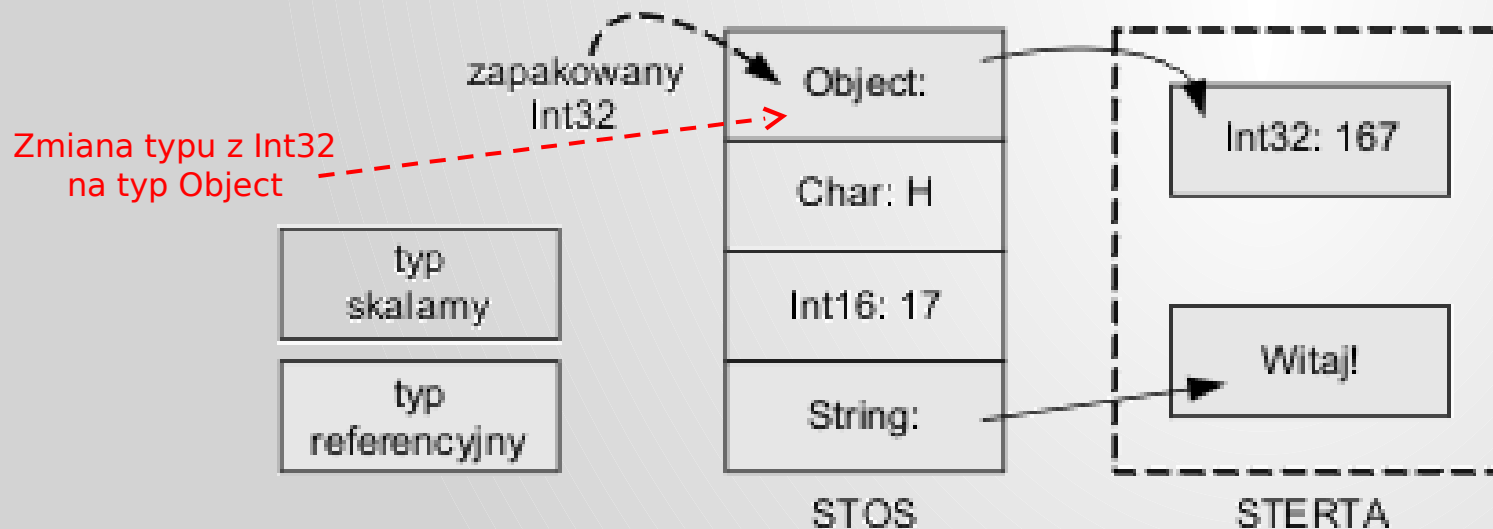
- ▶ Zestaw typów danych definiowany przez CTS należy do głównych składników CLR.
- ▶ Twórca języka może również zdefiniować własne typy jeżeli nie chce korzystać z typów już zdefiniowanych.

Najważniejsze typy zdefiniowane w CTS



Pakowanie - Boxing

Gdy zmienna typu skalarnego poddawana jest operacji *pakowania* (*boxing*), jej wartość jest umieszczana na sterce w odpowiednio zaalokowanym obszarze pamięci. Natomiast na stosie umieszczana jest referencja do tego obszaru.



Wspólna specyfikacja języka - CLS

Wspólna specyfikacja języka (*Common Language Specification*) - określa, jak duży podzbiór CTS musi zostać zaimplementowany, by język był zgodny z innymi językami wykorzystującymi CLS.

CLS wymaga na przykład, by język obsługiwał większość typów skalarnych — między innymi Boolean, Byte, Char, Decimal, Int16, Int32, Int64, Single, Double. Nie jest natomiast wymagana obsługa typów takich jak UInt16, UInt32, UInt64. CTS pozwala, by najniższy indeks tablicy był dowolną liczbą całkowitą, natomiast CLS wymaga, by najniższy indeks tablicy wynosił 0. CLS określa też inne ograniczenia ułatwiające efektywną współpracę różnych języków zgodnych z CLR.



Platforma .NET

» Biblioteki klas

Biblioteki klas - informacje ogólne

- ▶ Biblioteka klas, dostępna w pakiecie .NET Framework SDK, składa się z klas, interfejsów i stałych.
- ▶ Elementy biblioteki są pogrupowane w hierarchiczną strukturę przestrzeni nazw (ang. *namespace*).
- ▶ Biblioteki klas umożliwiają dostęp do usług systemowych i stanowią podstawę tworzenia aplikacji, komponentów i kontrolek .NET. Zadaniem przestrzeni nazw jest zawężenie zakresu, w którym obowiązują nazwy typów. Dzięki temu mogą istnieć dwie klasy o tej samej nazwie pod warunkiem, że są zdefiniowane w różnych przestrzeniach nazw. Aby przejść do kolejnego poziomu hierarchii, wystarczy postawić znak kropki (np. System.IO). Nazwy przestrzeni nazw dostarczanych przez Microsoft zawsze zaczynają się od etykiety System lub Microsoft.

Przestrzenie nazw w bibliotece klas

- Microsoft.CSharp — zawiera klasy wspierające kompilację oraz generowanie kodu w języku C#,
- Microsoft.JScript — zawiera klasy wspierające kompilację oraz generowanie kodu w języku JScript,
- Microsoft.VisualBasic — zawiera klasy wspierające kompilację oraz generowanie kodu w języku Visual Basic .NET,
- Microsoft.Vsa — zawiera interfejsy, umożliwiające integrację aplikacji ze skryptami, wykorzystującymi silniki skryptowe platformy .NET Framework oraz umożliwia kompilowanie i uruchamianie kodu w czasie pracy aplikacji,
- Microsoft.Win32 — zawiera dwa typy klas — klasy obsługujące zdarzenia wyzwalane przez system operacyjny oraz klasy pozwalające uzyskać dostęp do rejestru systemowego,
- System — ta przestrzeń nazw zawiera klasy podstawowe oraz klasy bazowe, które definiują najczęściej używane referencyjne i skalarne typy danych, zdarzenia, procedury obsługi zdarzeń, interfejsy, atrybuty i klasy przeznaczone do obsługi wyjątków.

System.IO

- ▶ Przestrzeń nazw System.IO umożliwia wszechstronny dostęp do systemu plików oraz zarządzanie nim.
- ▶ Obiekty typu *Reader* i *Writer* pozwalają na dostęp do plików binarnych, plików tekstowych i do strumieni.
- ▶ Obiekty **File**, **Directory** i **Path** umożliwiają zarządzanie systemem plików. Zdefiniowano również odpowiednie obiekty **Exception** do obsługi błędów dostępu do plików.

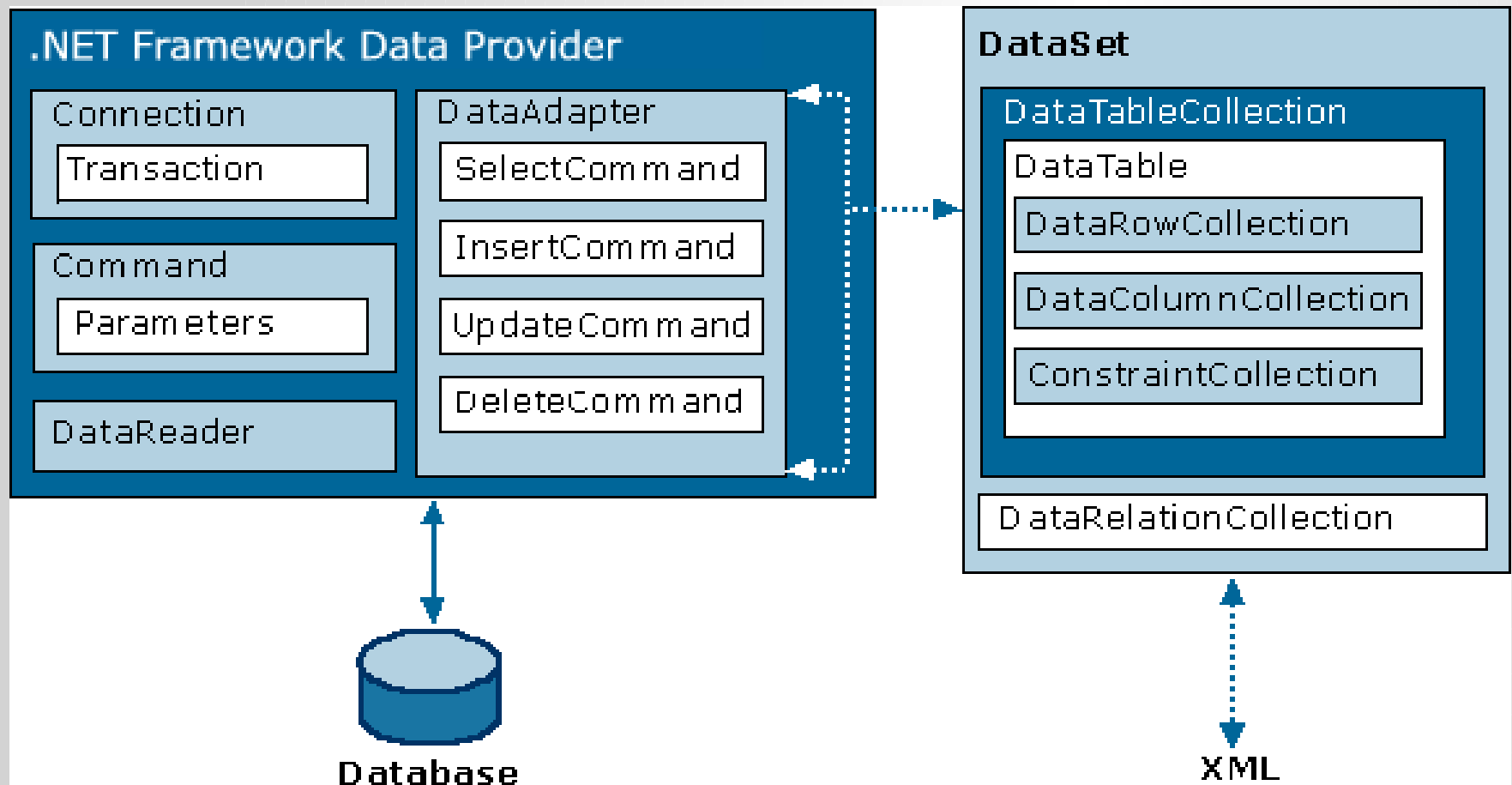
System.Runtime.Serialization

- ▶ Przestrzeń nazw System.Runtime.Serialization zawiera klasy pozwalające na serializację i deserializację obiektów.
- ▶ Serializacja - jest to proces konwersji obiektu (lub grupy powiązanych ze sobą obiektów) w ciąg bajtów, który może zostać zapisany lub przesłany do innego komputera.
- ▶ Deserializacja - to proces odwrotny, polegający na odtworzeniu obiektów na podstawie ciągu bajtów uzyskanego w procesie serializacji.

System.Data (ADO.NET)

- ▶ Do przestrzeni nazw System.Data należą głównie klasy, które są składnikami architektury ADO.NET. Architektura ta umożliwia tworzenie komponentów, pozwalających na wydajne zarządzanie danymi pochodzącymi z wielu różnych źródeł.
- ▶ Możliwa jest praca bez połączenia do serwera baz danych — pobranie danych z bazy, praca z tymi danymi bez utrzymywania połączenia z serwerem, a następnie synchronizacja wprowadzonych zmian.
- ▶ Architektura ADO.NET jest także wykorzystywana przez aplikacje Windows Forms i ASP.NET.

Architektura ADO.NET



Interfejsy

- ▶ Na platformie .NET Framework dostępne są dwa modele komunikacji — komunikacja z użytkownikiem oraz komunikacja z innymi aplikacjami.
- ▶ Komunikacja z użytkownikiem - może się odbywać za pomocą interfejsów takich jak Windows Forms, Web Forms lub konsola tekstowa.
- ▶ Komunikacja z aplikacjami - odbywa się za pośrednictwem interfejsu Web Services.

Windows Forms

- ▶ Interfejs Windows Forms (nazywany także WinForms) znajduje się w przestrzeni nazw `System.Windows.Forms`.
- ▶ Zawiera klasy przydatne przy tworzeniu aplikacji działających w systemie Windows, w pełni wykorzystujących bogate funkcje interfejsu użytkownika. Interfejs ten powstał na podstawie biblioteki *Windows Foundation Classes*, przygotowanej początkowo dla języka J++.
- ▶ Wszystkie języki programowania platformy .NET Framework — niezależnie od tego, z jakiej biblioteki korzystały dotychczas — wykorzystują ten sam sposób komunikacji z graficznym interfejsem użytkownika systemu Windows.
- ▶ Programista nie musi już korzystać bezpośrednio z funkcji interfejsu Windows API.

Web Forms

- ▶ Web Forms — interfejs użytkownika aplikacji internetowych ASP.NET — znajduje się w przestrzeni nazw `System.Web.UI`.
- ▶ Kontrolki są wyświetlane w aplikacji klienckiej (przeglądarce), ale ich kod uruchamiany jest na serwerze. Co prawda jest to związane z przesyłaniem sporej ilości danych, jednak dzięki temu możliwe jest upodobnienie działania interfejsu aplikacji internetowej do interfejsu zwykłej aplikacji Windows.
- ▶ Do projektowania interfejsu nie jest konieczna znajomość języka HTML — wystarczy przeciągnąć odpowiednie kontrolki na okno widoku projektu w graficznym środowisku programistycznym. Rozwiązuje to też problem zgodności kodu strony internetowej z różnymi przeglądarkami — w każdej przeglądarce kontrolka wyświetlana jest w najlepszy z możliwych sposobów.

Web Services

- ▶ Klasy, znajdujące się w przestrzeni nazw *System.Web.Services*, umożliwiają tworzenie usług internetowych XML Web Services, pozwalających na komunikowanie się pomiędzy aplikacjami pracującymi na różnych platformach, urządzeniach i pod kontrolą różnych systemów, z użyciem standardów takich jak HTTP, XML, XSD, SOAP i WSDL.



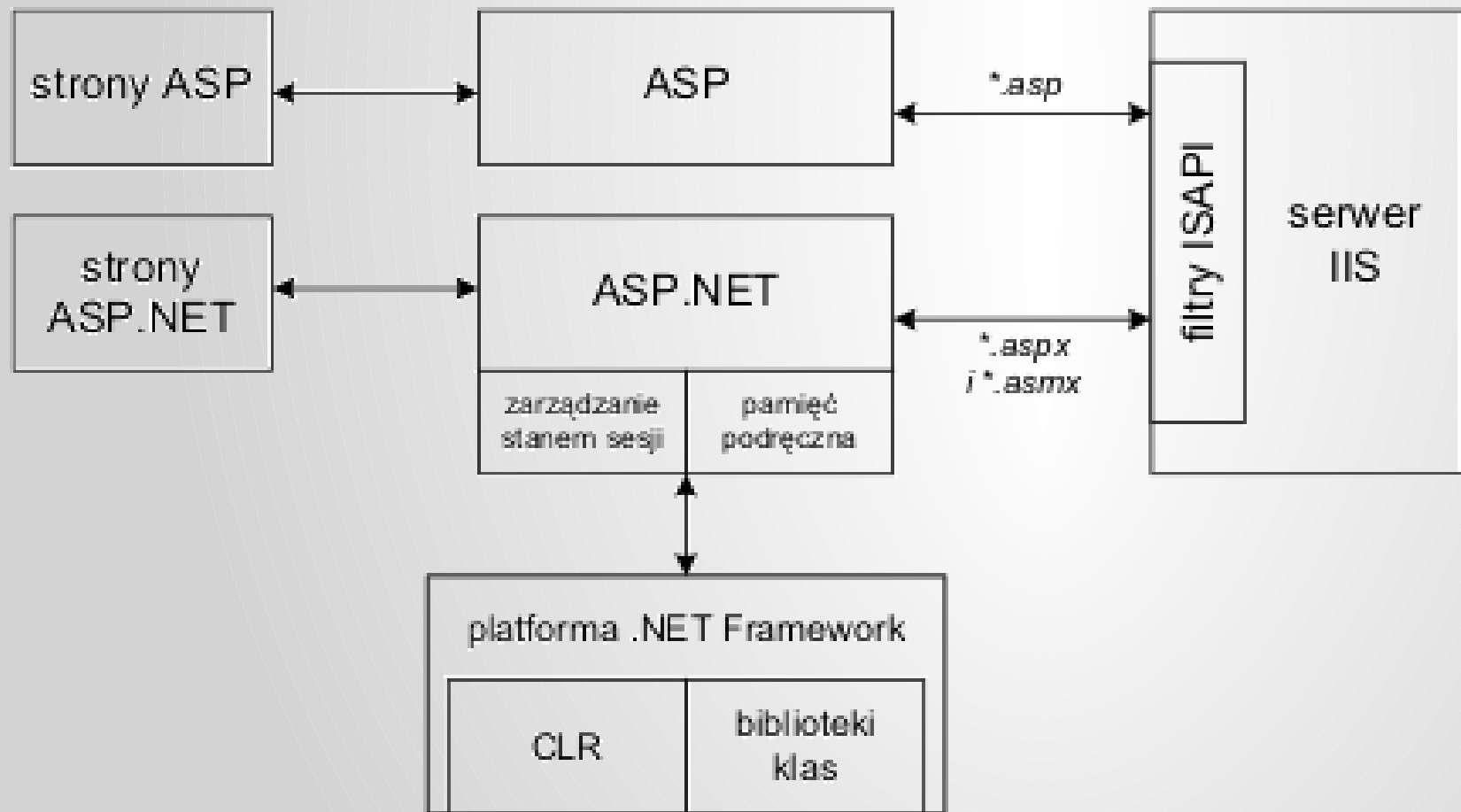
Platforma .NET

» ASP.NET

ASP.NET - coś dla webmasterów

- ▶ ASP.NET to technologia tworzenia dynamicznych stron internetowych, w pełni wykorzystująca funkcjonalność platformy .NET Framework oraz wspólnego środowiska uruchomieniowego CLR.
- ▶ Proponowana wcześniej przez Microsoft technologia tworzenia stron dynamicznych to ASP (Active Server Pages).

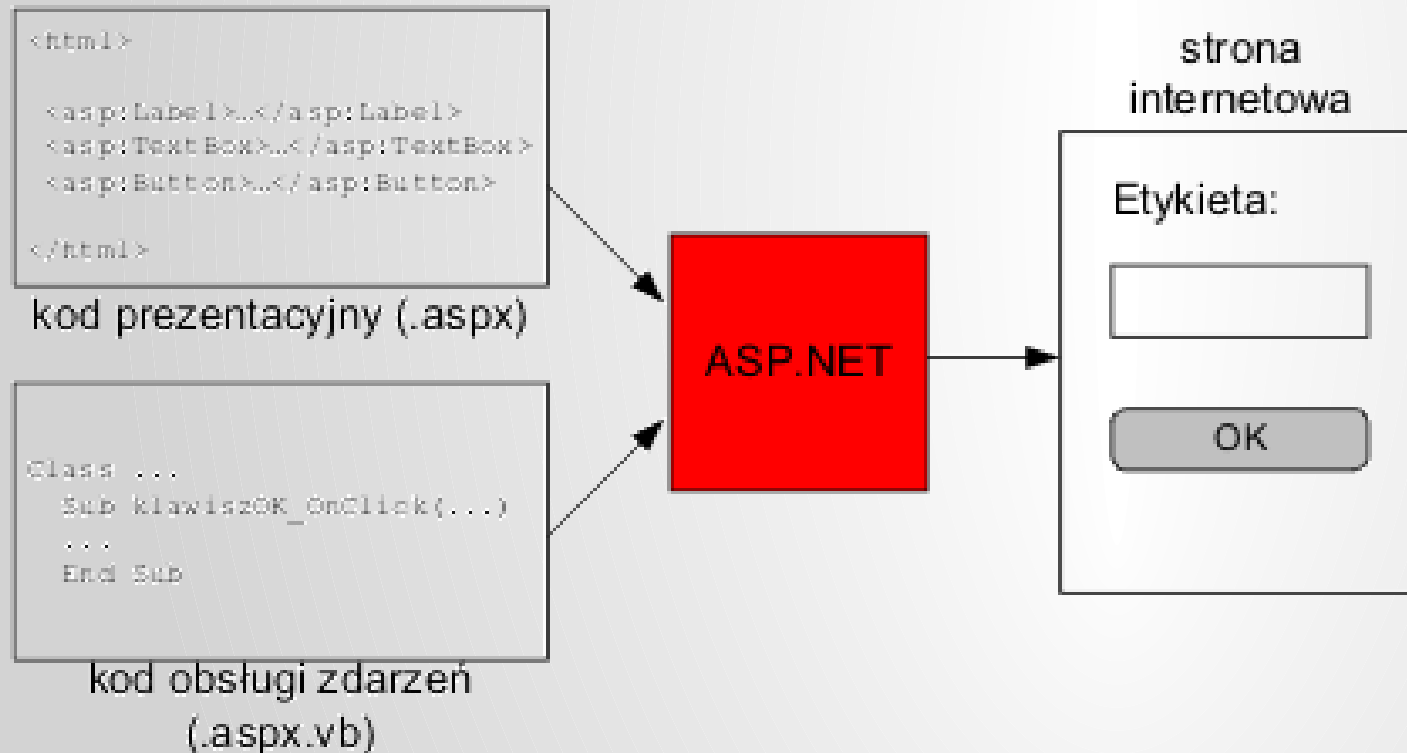
Architektura ASP.NET



Cel powstania ASP.NET

- ▶ Główną misją przyświecającą powstawaniu ASP.NET było uproszczenie procesu tworzenia aplikacji internetowych tak bardzo, jak to tylko możliwe. Osiągnięto to w ten sam sposób, w jaki Visual Basic zniósł bariery programowania dla Windows - poprzez implementację programowania zdarzeniowego. Programiści dodają do formularza kontrolki i piszą kod, który zostanie wykonany wtedy, gdy wystąpi określone zdarzenie związane z tymi kontrolkami (np. załadowanie lub opuszczenie strony, wpisanie tekstu do pola lub kliknięcie przycisku).

Kod schowany



ASP.NET pozwala na oddzielenie kodu obsługi zdarzeń od kodu prezentacji danych

Aplikacje internetowe

ASP.NET umożliwia tworzenie i wdrażanie aplikacji internetowych dwóch typów:

- **aplikacje Web Forms** - strony internetowe, generowane za pomocą skryptu, prezentujące dynamiczne treści albo stanowiące interfejs użytkownika lub formularz dla klientów takich jak przeglądarki
- **usługi internetowe XML Web Services** - udostępniają funkcjonalność innym aplikacjom i umożliwiają wymianę danych pomiędzy aplikacjami.



Platforma .NET

» .NET Compact Framework

.NET Compact Framework - co to takiego?

- ▶ .NET Compact Framework to uproszczona wersja .NET Framework, dostosowana do pracy na urządzeniach przenośnych i innych urządzeniach, działających pod kontrolą systemu operacyjnego Windows Embedded.
- ▶ Dzięki niej programiści mogą wykorzystywać jeden zestaw narzędzi oraz bibliotek API do tworzenia aplikacji dla całej gamy urządzeń - począwszy od komputerów kieszonkowych po stacje robocze oraz serwery.



.NET Compact Framework vs .NET Framework

- ▶ **usunięto niektóre API**
- ▶ **uproszczono pozostałe API**
- ▶ **zmieniono sposób przetwarzania komunikatów o błędach**

Dodano:

- ▶ **obsługa transmisji IrDA**
- ▶ **dostawca danych SQL CE – obsługa baz danych SQL Server CE Edition**



Platforma .NET

»» Bezpieczeństwo

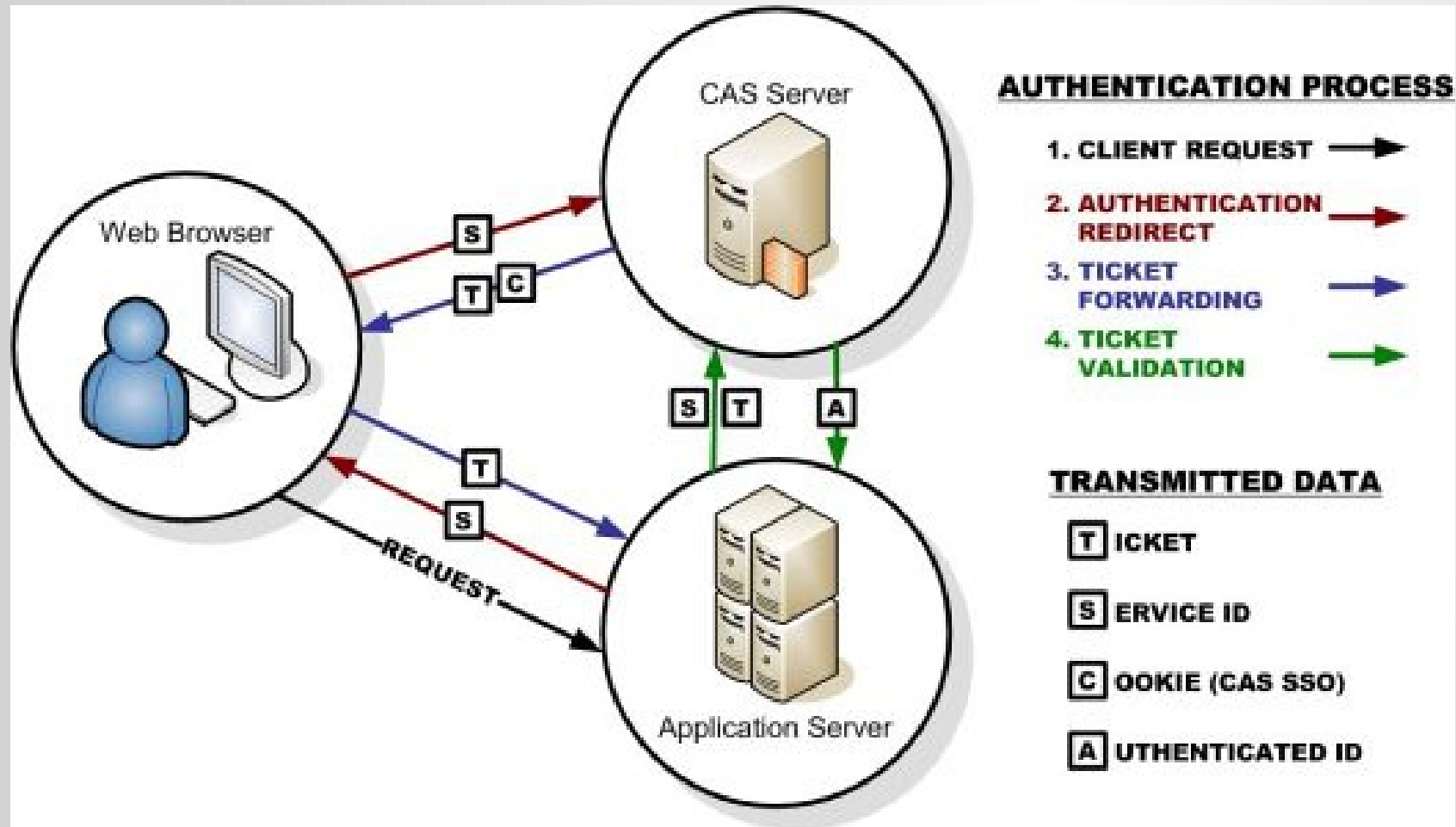
Bezpieczeństwo - informacje ogólne

- ▶ Podzespół określa zasięg zmiennych, wersję zawartego w nim oprogramowania oraz stanowi niepodzielną jednostkę wdrożeniową.
- ▶ Granice podzespołu określają także granice bezpieczeństwa. CLR definiuje dwa typy bezpieczeństwa implementowane w podzespołach:
 - *bezpieczeństwo oparte na uprawnieniach kodu*
 - *bezpieczeństwo oparte na rolach (role-based security — RBS).*

Bezpieczeństwo oparte na uprawnieniach kodu (CAS)

Zamiast pytać użytkownika, czy wykonać dany kod, czy nie, CAS (*code access security*) ustala prawa dostępu na podstawie zestawu uprawnień, których przydzielenia żąda dany podzespół oraz zestawu uprawnień udzielonego poprzez zasady bezpieczeństwa.

Działanie CAS



Rodzaje uprawnień CAS

- ▶ **UIPermission** — zezwolenie na dostęp do interfejsu użytkownika,
- ▶ **FileIOPermission** — zezwolenie na dostęp do plików oraz katalogów,
- ▶ **FileDialogPermission** — pozwolenie na dostęp jedynie do tych plików, które użytkownik wskazał za pomocą okna dialogowego,
- ▶ **PrintingPermission** — umożliwienie dostępu do drukarek,
- ▶ **EnvironmentPermission** — pozwolenie na dostęp do zmiennych środowiskowych,
- ▶ **RegistryPermission** — umożliwienie dostępu do rejestru systemowego komputera,
- ▶ **ReflectionPermission** — zezwolenie na dostęp do metadanych podzespołu,
- ▶ **SecurityPermission** — umożliwienie nadania większej grupy uprawnień, między innymi prawa uruchamiania kodu niezarządzanego,
- ▶ **WebPermission** — zezwolenie na nawiązywanie lub odbieranie połączeń internetowych.

Bezpieczeństwo oparte na rolach (RBS)

Bezpieczeństwo oparte na uprawnieniach kodu umożliwia uzależnienie dozwolonego zestawu operacji od nazwy podzespołu, jego twórcy, oraz miejsca pochodzenia. Nie zapewnia jednak kontroli uprawnień w zależności od tego, jaki użytkownik uruchomił dany podzespół. Zapewnienie tego rodzaju zabezpieczeń należy do funkcjonalności zwanej *bezpieczeństwem opartym na rolach* (*role-based security* — *RBS*).



Platforma .NET

» Języki programowania

Języki programowania zgodne z .NET

- ▶ Istotną nowością, jaka pojawiła się w platformie .NET, jest Common Language Infrastructure. Każdy język programowania, który spełni odpowiednie standardy (chodzi głównie o tzw. common object model), będzie miał dostęp do bogatej biblioteki .NET.
- ▶ Obecnie ponad 40 języków programowania jest zgodnych z .NET. Wiele z nich (np. Visual Basic, Delphi) musiało przejść istotne zmiany, by dostosować się do nowego środowiska. Dlatego też zwykle oznacza się języki dla platformy .NET przyrostkiem .NET (np. Delphi.NET, VB.NET itd.).

Podstawowe języki dostarczane przez Microsoft

- ▶ C#
- ▶ Visual Basic .NET
- ▶ C++/CLI (wcześniej Managed C++, wariant C++)
- ▶ J# (wariant języka Java opracowany przez Microsoft)
- ▶ JScript .NET (kompilowany wariant języka JScript)



Pozostałe ważniejsze języki programowania:

- ▶ COBOL
- ▶ Delphi (Delphi.NET – od wersji 8 środowiska)
- ▶ Eiffel
- ▶ Fortran
- ▶ Lisp
- ▶ Nemerle (opracowany przez wrocławskich naukowców)
- ▶ Perl
- ▶ Python
- ▶ Smalltalk





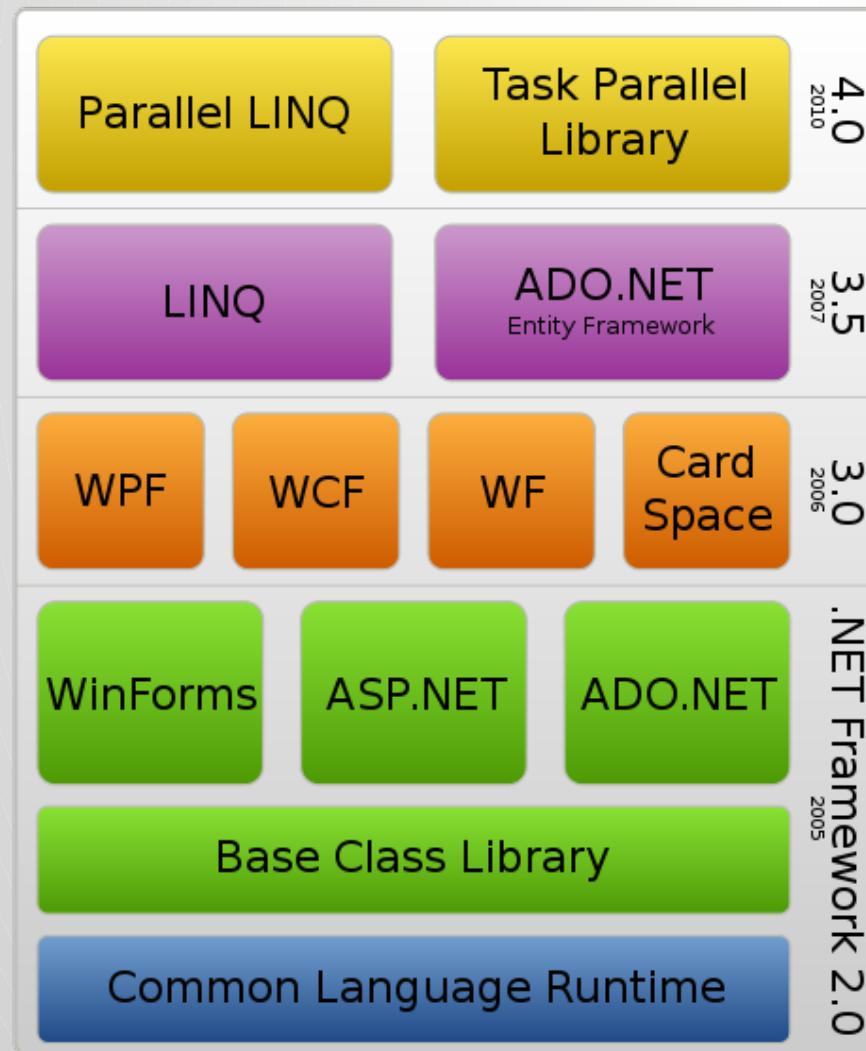
Platforma .NET

» Kolejne generacje

Kolejne edycje .NET Framework:

Generacja	Numer wersji	Data wydania	Wersja Visual Studio	Dostarczany z:
1.0	1.0.3705.0	13 Luty 2002	Visual Studio .NET	N/A
1.1	1.1.4322.573	24 Kwiecień 2003	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	7 Listopad 2005	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	6 Listopad 2006	Expression Blend	Windows Vista, Windows Server 2008
3.5	3.5.21022.8	19 Listopad 2007	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	12 Kwiecień 2010	Visual Studio 2010	N/A
4.5	4.5.50709.17929	15 Sierpień 2012	Visual Studio 2012	Windows 8, Windows Server 2012

Rozwój kolejnych generacji



The .NET Framework Stack



Microsoft .NET Framework 4.5

Każdy z Nas może pobrać za darmo najnowszą wersję platformy .NET 4.5, która jest kompatybilna ze wszystkimi poprzednimi wersjami:

[http://
www.microsoft.com/pl-pl/download/details.aspx?id=54539](http://www.microsoft.com/pl-pl/download/details.aspx?id=54539)



Microsoft®
Visual Studio®

Visual Studio

>> Informacje ogólne

Microsoft Visual Studio

- ▶ to zintegrowane środowisko programistyczne firmy Microsoft. Jest używane do tworzenia oprogramowania konsolowego oraz z graficznym interfejsem użytkownika, w tym np. aplikacje
 - Windows Forms
 - WPF
 - Web Sites
 - Web Applications



Natomiast aplikacje te mogą być pisane np. na platformy:

- Microsoft Windows
- Windows Mobile
- Windows CE.NET Framework
- Microsoft Silverlight

Zintegrowane środowisko programistyczne

Jest to aplikacja lub zespół aplikacji (środowisko) służących do tworzenia, jak i modyfikowania, testowania i konserwacji oprogramowania.

- ▶ Aplikacje takie charakteryzują się tym, że udostępniają złożoną, wieloraką funkcjonalność obejmującą:
 - edycję kodu źródłowego
 - kompilowanie kodu źródłowego
 - tworzenie zasobów programu (tzn. formatek / ekranów / okien dialogowych, menu, raportów, elementów graficznych takich jak ikony, obrazy itp.)
 - tworzenie baz danych



Różne koncepcje środowiska programistycznego

I. Środowisko jako osobny pakiet oprogramowania

- pakiet **Microsoft Visual Studio** (popularny na systemach rodziny Windows)
- narzędzia firmy **Borland**

Borland

II. Środowisko zintegrowane z systemem

(W takiej koncepcji, system operacyjny sam w sobie jest środowiskiem programistycznym, a wszystkie funkcje są z nim zintegrowane.)

Unix

UNIX

III. Rozwiązania pośrednie

- **Emacs** - tekstowy edytor, zintegrowany z **Uniksem**



- ▶ **Microsoft Visual Studio** służy do programowania konsolowego oraz z graficznym interfejsem.
- ▶ **Graficzny interfejs użytkownika** (ang. Graphical User Interface, GUI), często nazywany też środowiskiem graficznym.
 - **Środowisko graficzne**- ogólne określenie sposobu prezentacji informacji przez komputer oraz interakcji z użytkownikiem, polegające na rysowaniu i obsługiwaniu widżetów, podstawowego elementu graficznego interfejsu użytkownika (np. okno, pole edycji, suwak, przycisk).
- ▶ **Microsoft Visual Studio** pozwala na tworzenie samodzielnych aplikacji, a także aplikacji sieciowych, usług sieciowych oraz serwisów internetowych.
- ▶ Dawniej istniała możliwość zakupu osobnych pakietów dla poszczególnych języków programowania (np. Visual C++). Obecnie jedyną możliwością jest zakup całego produktu Visual Studio.



Microsoft®
Visual Studio®

Visual Studio

>> Historia

Kilka słów o historii Visual Studio

Visual Studio 97

- ▶ Pierwsza wersja Visual Studio łączyła w sobie kilka języków



Visual Studio 6.0

- ▶ Kolejna wersja została wprowadzona w 1998 roku i przetrwała bez zmian (oprócz poprawek) cztery lata, aż do wprowadzenia platformy .NET Framework.



Visual Studio .NET (2002) (Visual Studio 7.0)

- ▶ Najbardziej istotną zmianą było wprowadzenie platformy **.NET Framework**, dzięki której programy nie były kompilowane do kodu maszynowego, ale do języka pośredniego (CIL), który jest kompilowany do postaci ostatecznej podczas wykonywania programu.
- Jest to platforma programistyczna obejmująca środowisko uruchomieniowe oraz biblioteki klas dostarczające standardowej funkcjonalności dla aplikacji. Technologia ta nie jest związana z żadnym konkretnym językiem programowania, a programy mogą być pisane w jednym z wielu języków (C++/CLI, C#, F#, J#, Delphi 8 dla .NET, Visual Basic .NET).



- ▶ Wprowadzenie języka C# podobnego w składni do języka Java, lecz przeznaczonego wyłącznie do środowiska .NET. Język *Visual Basic* został zmieniony tak, aby pasował do platformy .NET.
- ▶ Języka C++ można było używać w dwóch wariantach:
 - C++ Managed obsługującego platformę .NET,
 - tradycyjnego C++ pozwalającego na kompilację bezpośrednio do kodu maszynowego

Visual Studio .NET 2003 (Visual Studio 7.1)

- ▶ użytkownicy poprzedniej wersji mogli ją uaktualnić do najnowszej wersji za niewielką dopłatą.
- ▶ Wersja ta wprowadzała uaktualnienie .NET Framework , oraz wsparcie dla projektowania aplikacji dla urządzeń przenośnych.



Visual Studio 2005 (Visual Studio 8.0)

- ▶ Visual Studio 2005 jest przeznaczona wyłącznie dla systemów **MS Windows 2000/XP/2003**. Najbardziej istotnymi zmianami w stosunku do poprzedniej wersji są:
 - zastosowanie nowszej wersji .NET Framework (2.0), co pociąga za sobą zmiany w językach programowania;
 - możliwość kompilacji dla procesorów 64-bitowych różnych typów;
 - było to ostatnie wydanie Visual Studio dla Windows 2000;
 - Pierwsza wersja z edycją Express Edition, dostępną za darmo;
 - zmiana Managed C++ na C++/CLI (w Visual C++).



Visual Studio 2008 (Visual Studio 9)

- ▶ Przygotowany z myślą o **Windows Vista i Office 2007**.
- ▶ W tej wersji zrezygnowano ze wsparcia dla **Windows 2000**
- ▶ Na witrynie Microsoftu dostępna jest wersja **Express Edition**
(przeznaczona dla hobbystów i małych firm, może być używana bezpłatnie bez ograniczeń czasowych)
- ▶ Zmianą jest znowu wykorzystanie nowej wersji **.NET Framework (3.5)**



► Visual Studio 2010 (Visual Studio 10)

- wykorzystanie .NET Framework 4.0
- znaczne zmiany w GUI
- wsparcie dla SQL 2008
- nowe opcje do testowania;
- narzędzia zintegrowanego środowiska programistycznego dla Windows 7 i Microsoft SharePoint 2010;
- nowe "drag and drop bindings" dla Silverlight i aplikacji Windows Presentation Foundation
- nowy język programowania F#
- wsparcie budowania aplikacji dla Windows Azure (platforma chmurowa firmy Microsoft. Udostępnia ona mechanizmy pozwalające przetwarzać dane, a także je składować)
- automatyczne znajdowanie popularnych błędów w kodowaniu.





► **Visual Studio 2012**

- Jest najbardziej rozbudowaną wersją produktu. Oprogramowanie zostało opracowane z myślą skutecznej działalności w środowisku, w którym dobre pomysły i szybkość są niezbędne.



Visual Studio 2010

>> Wersje oraz instalacja

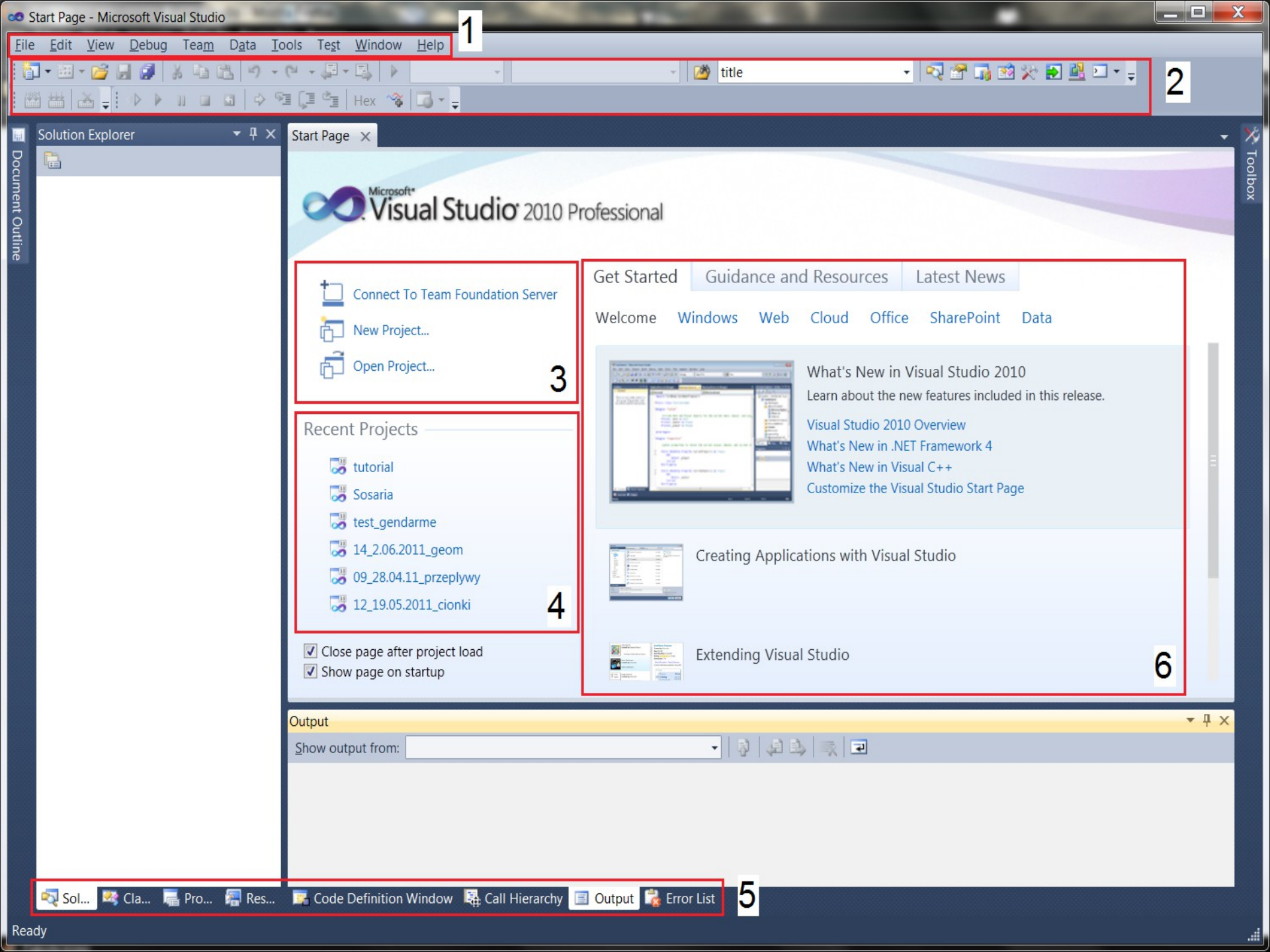
Visual Studio 2010 wersje

- ▶ Express – wersja najbardziej podstawowa
- ▶ Professional – wersja bardziej zaawansowana, upraszcza tworzenie, debuggowanie i wdrażanie aplikacji na wielu platformach
- ▶ Premium - to kompletny zestaw narzędzi, które upraszcza tworzenie aplikacji dla osób i zespołów, umożliwiając dostarczanie skalowalnych aplikacji wysokiej jakości.
- ▶ Ultimate - to kompleksowy zestaw narzędzi do zarządzania cyklem życia aplikacji dla zespołów zapewnienia jakości wyników, od projektu do wdrożenia. Jest to najbardziej zaawansowana wersja



Visual Studio 2010

»» Wprowadzenie do środowiska



1

2

3

4

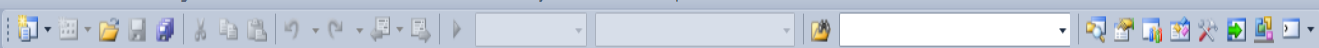
6

5

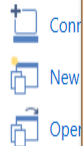
1. **Pasek menu** – znajdują się tutaj opcje edycji pliku i projektu;
2. **Paski narzędzi** – znajdują się tu nie tylko skróty do opcji edycji pliku, ale także i opcji projektu (tj. budowa, kompilacja i debugowanie programu – Brzmi strasznie? Nie ma czego się obawiać! Wszystkiego dowiemy się już za chwilę.);
3. **Praca z projektem** – skróty do tworzenia i otwierania projektów;
4. **Ostatnie projekty** – skróty do kilku ostatnio otwieranych projektów;
5. **Pasek zakładek** – w zależności od wybranej zakładki, mamy możliwość przyjrzenia się błędom w kodzie, wynikach działania programu czy budowy projektu;
6. **Wskazówki Microsoftu** – umożliwiają bliższe zapoznanie się ze środowiskiem.

Tworzenie nowego projektu

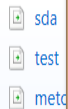
1. Klikamy **File | New | Project** (*Plik | Nowy | Projekt*) albo **New Project** w oknie głównym; Wybieramy **Win32 Console Application** (*aplikacja konsoli Win32*) z menu.
Uzupełniamy poniższe informacje: **Name** (*nazwa*) – nazwa projektu, np. nazwa_projektu;
2. **Location** (*lokalizacja*) – ścieżka dostępu do projektu; aby ją zmienić, należy kliknąć na **Browse** (*przeglądaj*) i wybrać odpowiedni folder; przeważnie pozostawiamy domyślną wartość;
3. **Solution name** (*nazwa rozwiązania*) – powinno być takie samo, jak nazwa projektu (pole wypełni się automatycznie).
4. Klikamy **OK**. Kolejne okno, które się pojawi, pełni rolę wyłącznie informacyjną. Klikamy **Next** (*następny*); Wyświetli się okno z opcjami projektu. Upewnijmy się, że jest zaznaczona opcja **Console Application** (*aplikacja konsolowa*). Następnie, pod **Additional options** (*dodatkowe opcje*) wybieramy **Empty project** (*pusty projekt*). Naciskamy przycisk **Finish** (*zakończ*).

Toolbox
General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.



Recent Projects

☒ Close pages
☒ Show pages

Output

Show output from

Code Definition Window

Output

Start Page

New Project

Recent Templates

Installed Templates

Visual C++

ATL
CLR
General
MFC
Test
Win32Other Languages
Other Project Types
Database
Modeling Projects
Test Projects

Online Templates

.NET Framework 4

Sort by: Default

Search Installed Templates



Win32 Console Application

Visual C++



MFC Application

Visual C++



Win32 Project

Visual C++



Empty Project

Visual C++



ATL Project

Visual C++



MFC DLL

Visual C++



Windows Forms Application

Visual C++



CLR Console Application

Visual C++



CLR Empty Project

Visual C++



Class Library

Visual C++



Custom Wizard

Visual C++

Type: Visual C++

A project for creating a Win32 console application

Name: nazwa_projektu

Location: c:\users\rafal linowiecki\documents\visual studio 2010\Projects

Browse...

Solution name: nazwa_projektu

☒ Create directory for solution☐ Add to source control

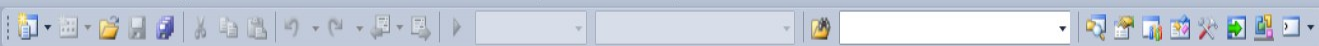
OK

Cancel

Solution Explorer



Server Explorer



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Connect To Team Foundation Server

New Project...

Open Project...

Recent Projects

sda

test

metoda_gradientu_prostego

☒ Close page after project load☒ Show page on startup

Output

Show output from:

Code Definition Window

Output

Win32 Application Wizard - nazwa_projektu

Application Settings

Overview

Application Settings

Application type:

☐ Windows application☒ Console application☐ DLL☐ Static library

Add common header files for:

☐ ATL☐ MFC

Additional options:

☒ Empty project☐ Export symbols☒ Precompiled header

< Previous

Next >

Finish

Cancel

Solution Explorer

Server Explorer

Rzut oka na zawartość projektu

External Dependencies

(zależności zewnętrzne) – ten katalog służy zaawansowanym programistom; nam nie będzie przydatny

Header Files (pliki nagłówkowe) – umieszczamy tu pliki z rozszerzeniem **.h**;

Resource Files (pliki zasobów) – zazwyczaj umieszcza się tu dodatkowe pliki, takie jak obrazki;

Source Files (pliki źródłowe) – umieszczamy tu pliki z rozszerzeniem np. **.cpp** (w przypadku C++).

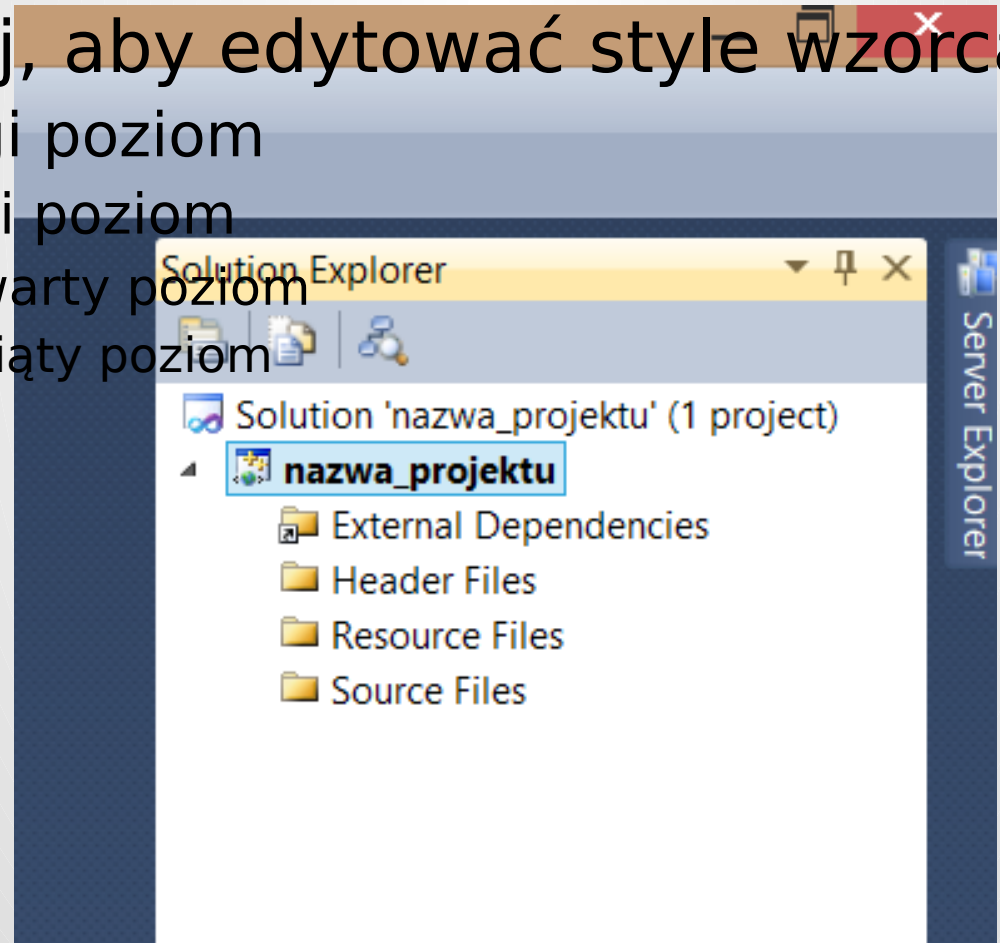
Kliknij, aby edytować style wzorca

Drugi poziom

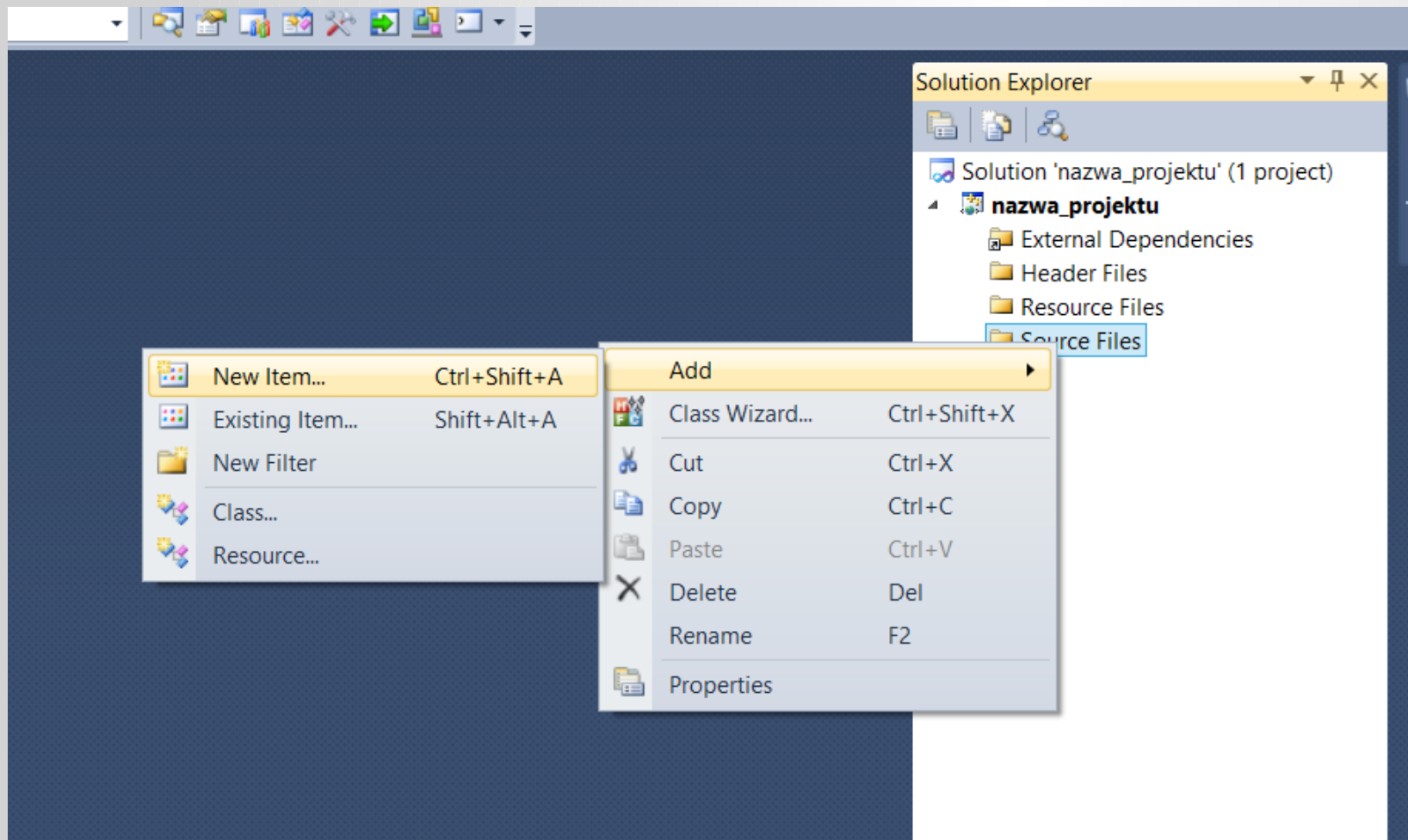
Trzeci poziom

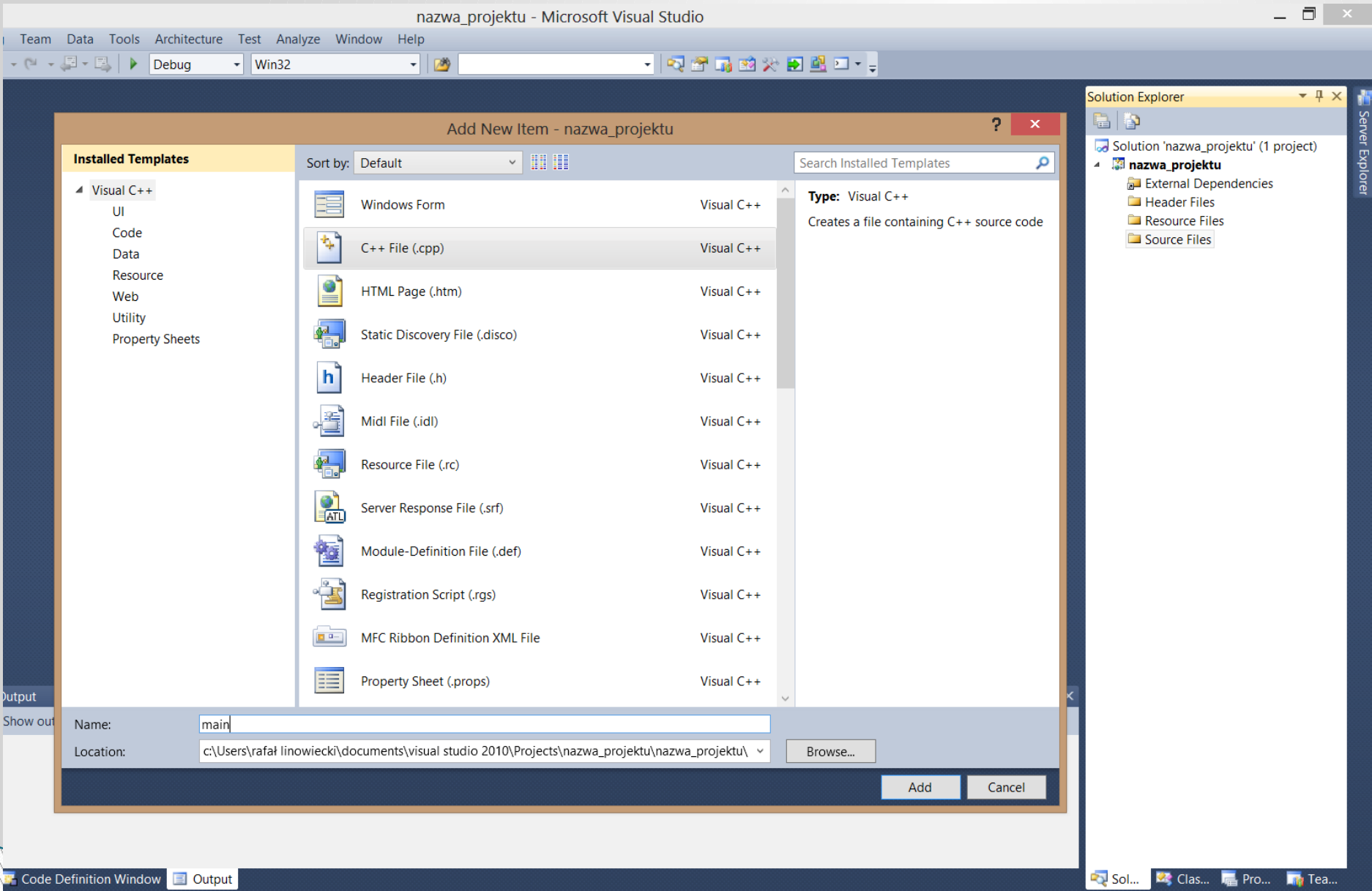
Czwarty poziom

Piąty poziom

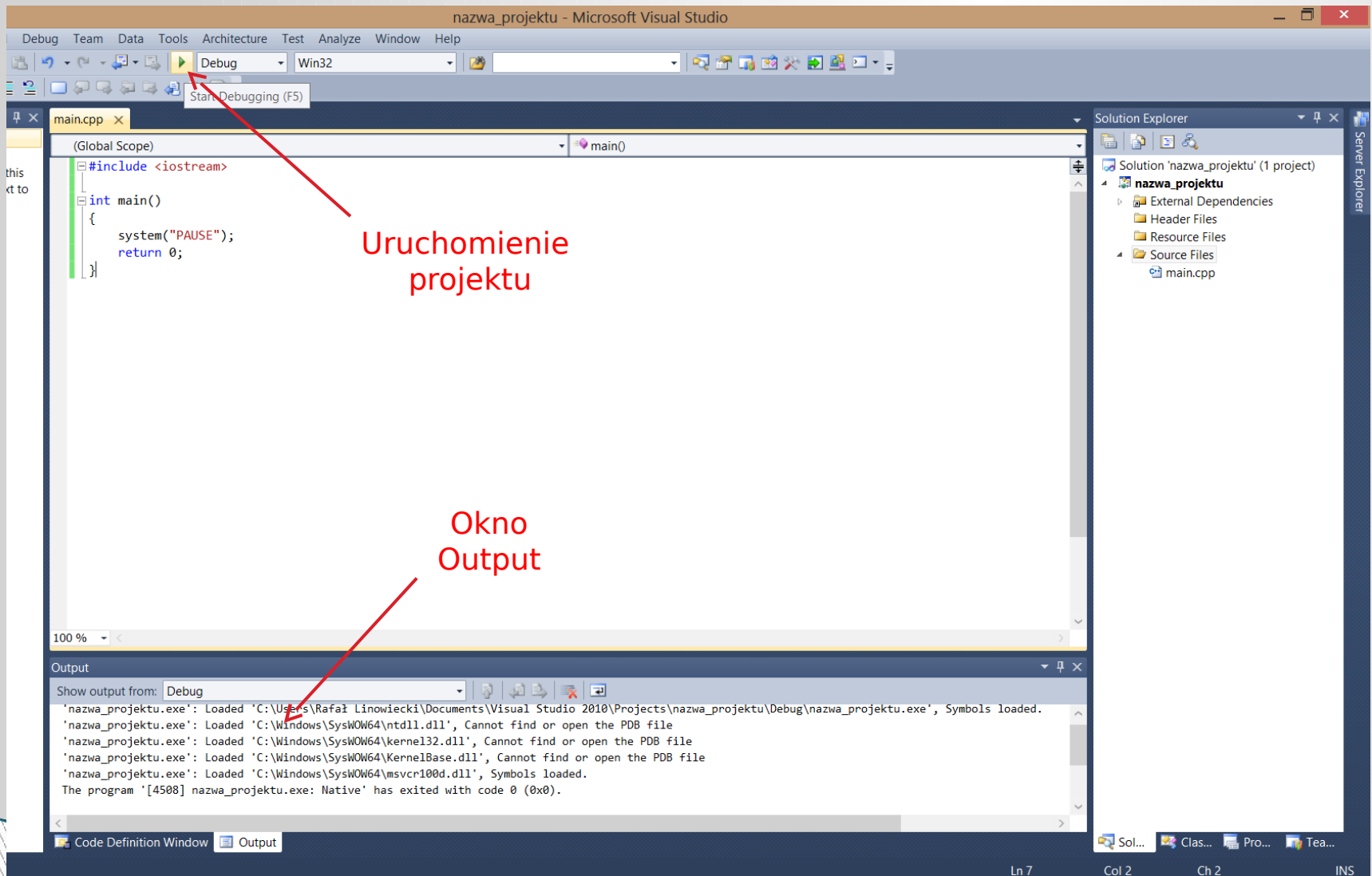


Dodawanie nowych plików do projektu





Budowa i uruchamianie projektu



Błędy

The screenshot shows the Microsoft Visual Studio IDE with a C++ project named 'nazwa_projektu'. The main window displays the code in 'main.cpp' with several syntax errors highlighted in red. The Error List window at the bottom shows 9 errors, with a red arrow pointing to the first error: 'error C2065: 'earwre': undeclared identifier'.

Code in main.cpp:

```
// #include <iostream>

int main()
{
    fdafa
    earwre
    324221
    4234
    system("PAUSE");
    return 0;
}
```

Error List:

Description	File	Line
error C2065: 'earwre': undeclared identifier	main.cpp	7
error C2143: syntax error: missing ';' before 'constant'	main.cpp	7
error C2143: syntax error: missing ';' before 'constant'	main.cpp	8
error C2146: syntax error: missing ';' before identifier 'system'	main.cpp	9
error C3861: 'system': identifier not found	main.cpp	9



Microsoft®
Visual Studio®

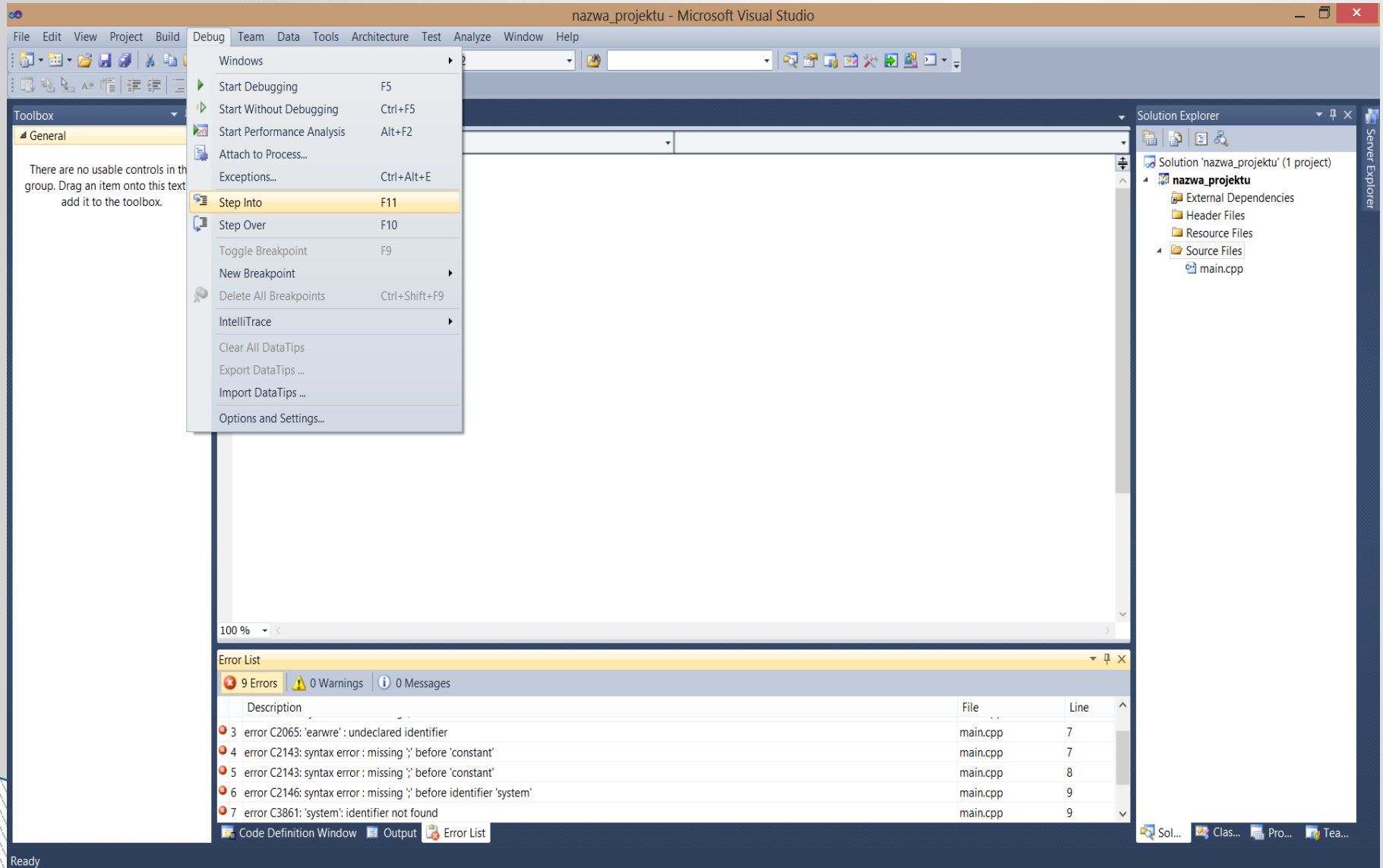
Visual Studio 2010

>> Debugger

Debugger – informacje ogólne

- ▶ Debugger jest narzędziem usprawniającym poszukiwanie i poprawianie błędów w naszym programie.
- ▶ Debugger umożliwia nam obserwowanie programu w trakcie jego działania.
- ▶ Podstawowymi narzędziami udostępnianymi przez debugger są pułapki i praca krokowa.
- ▶ Praca krokowa umożliwia wykonanie małego fragmentu programu na raz (wykonanie jednego kroku). Zazwyczaj jednym krokiem jest jedna linia programu.
- ▶ Pracę krokową umożliwiają polecenia Step Into i Step Over z menu Debug.

Praca krokowa



Step Into vs Step Over

wskaźnik
bieżącej
pozycji w
wykonywanym
programie

```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

int main() {
    Funkcja(1, 2);
    return 0;
};
```

```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

int main() {
    Funkcja(1, 2);
    return 0;
};
```

Wykonanie kroku
Step Over

```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

int main() {
    Funkcja(1, 2);
    return 0;
};
```

Wykonanie kroku
Step Into

Pułapka

- ▶ Pułapka jest miejscem, w którym wykonanie programu ma się zatrzymać.
- ▶ Pułapkę ustawiamy, klikając lewym przyciskiem myszy na lewym marginesie kodu. Czerwona kropka która pojawi się w efekcie kliknięcia symbolizuje pułapkę.

Pułapka

A screenshot of a code editor window. The title bar says '(Global Scope)'. The code is written in C# and consists of two methods: 'Funkcja' and 'main'. A red dot, representing a breakpoint, is set on the first line of the 'Funkcja' method. A green vertical line is visible in the left margin of the code editor.

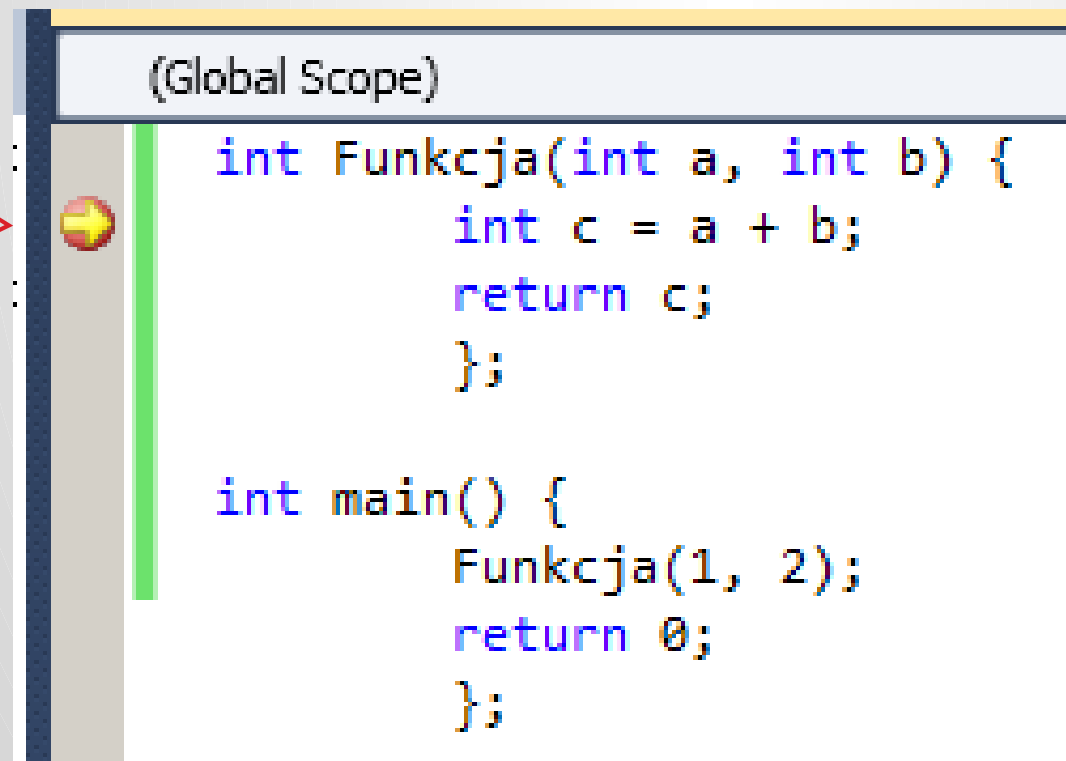
```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

int main() {
    Funkcja(1, 2);
    return 0;
};
```

- ▶ Gdy ustawimy pułapkę, możemy uruchomić program (Debug -> Start Debugging) i zaobserwujemy, że wykonanie zatrzyma się właśnie w miejscu pułapki.

Zatrzymanie wykonywania programu w miejscu pułapki podczas debuggowania



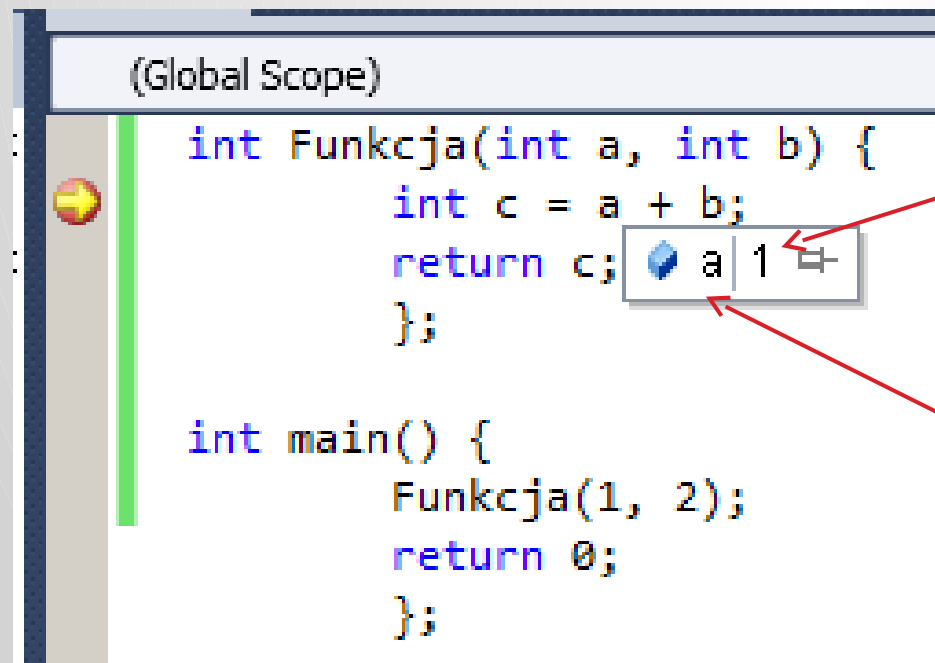
```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

int main() {
    Funkcja(1, 2);
    return 0;
};
```

Podglądanie wartości zmiennych

Pierwszą metodą jest najechanie kursorem na zmienną, której zawartość nas interesuje. Po chwili powinno pojawić się okienko z jej zawartością.



The screenshot shows a Visual Studio code editor window titled "(Global Scope)". The code contains two functions: `int Funkcja(int a, int b) { int c = a + b; return c; }` and `int main() { Funkcja(1, 2); return 0; };`. A yellow arrow cursor is hovering over the variable `a` in the `return c;` statement. A tooltip box has appeared, displaying `a 1`. A red arrow points from the text "wartość zmiennej" to the value `1` in the tooltip. Another red arrow points from the text "nazwa zmiennej" to the variable name `a` in the tooltip.

```
(Global Scope)

int Funkcja(int a, int b) {
    int c = a + b;
    return c;
};

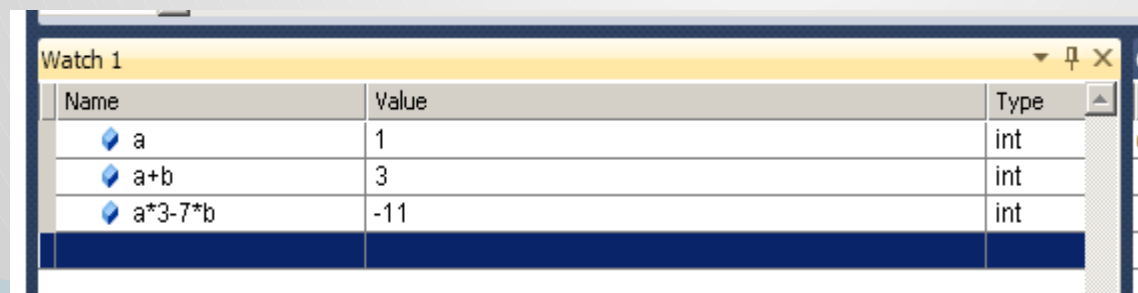
int main() {
    Funkcja(1, 2);
    return 0;
};
```

wartość
zmiennej

nazwa
zmiennej

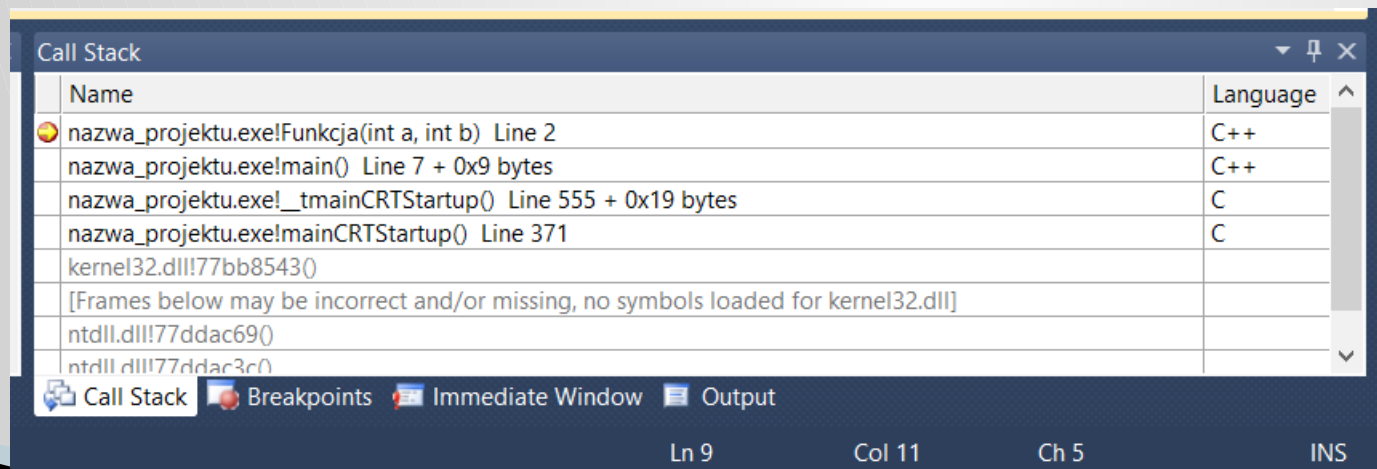
Drugą metodą jest skorzystanie z okien Locals i Watch. Powinny one pojawiać się automatycznie podczas pracy krokowej, lecz jeżeli nie są widoczne, możemy je włączyć poleceniami Debug -> Windows -> Locals i Debug -> Windows -> Watch -> Watch 1...4.

- ▶ Okno Locals zawiera wszystkie zmienne lokalne z bieżącej funkcji.
- ▶ W oknie Watch możemy wprowadzać dowolne zmienne (a nawet wyrażenia) których zawartość chcemy obejrzeć. Wystarczy dwukrotnie kliknąć na pustym polu w kolumnie Name i wpisać żądane wyrażenie.



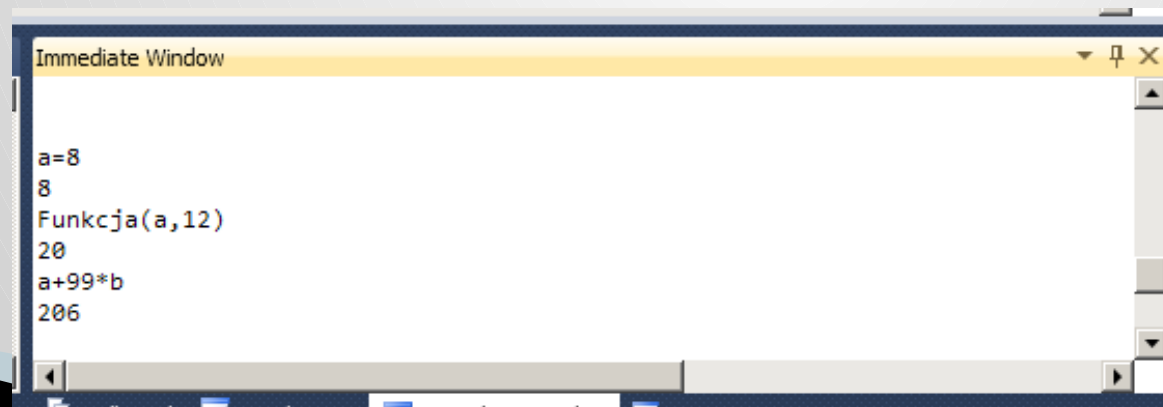
Okno Call Stack

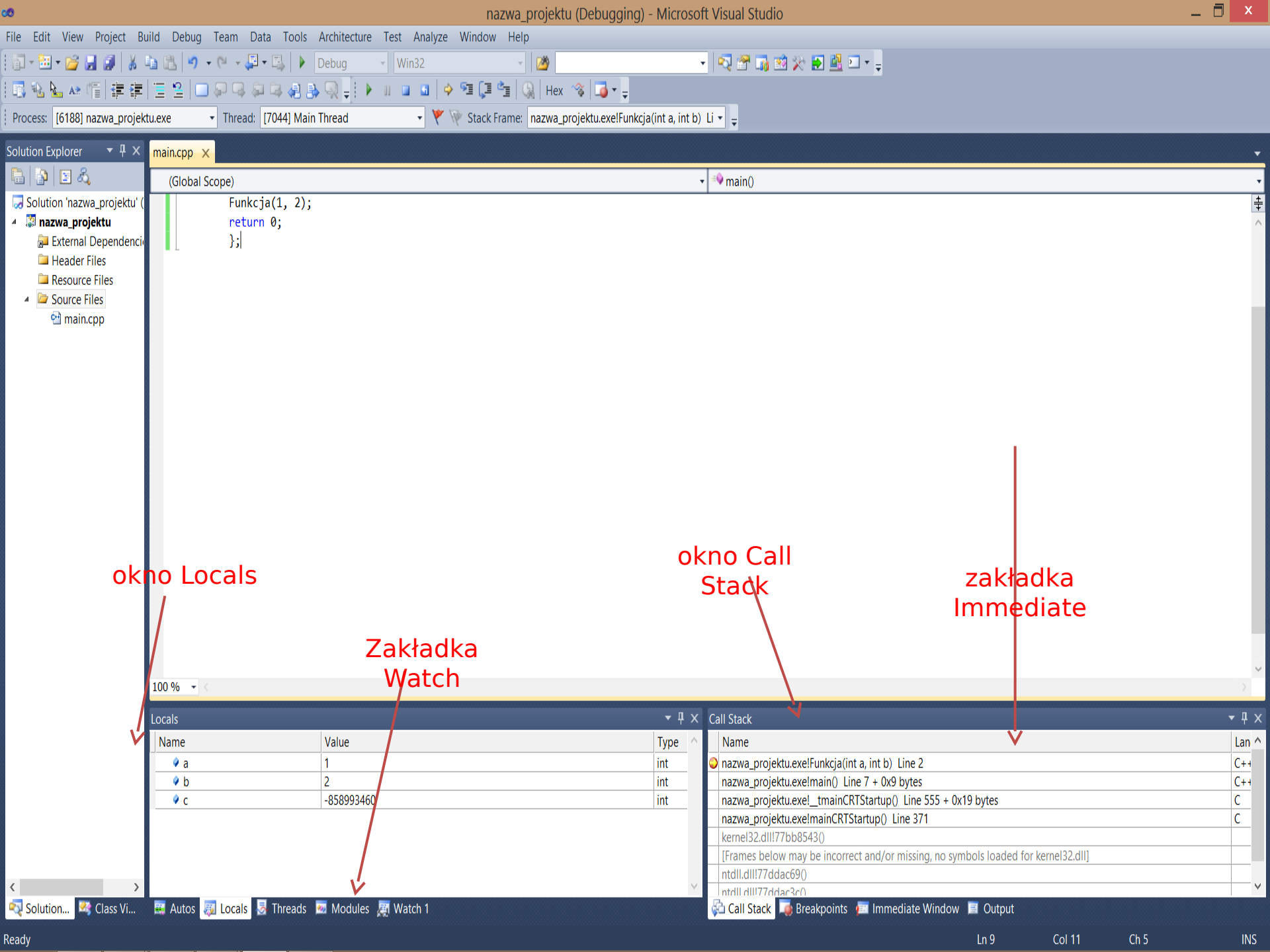
Okno C.S. przedstawia „drogę” którą doszliśmy do danego miejsca w programie. Pierwszy wpis przedstawia funkcję (wraz z parametrami wywołania) w której jesteśmy. Wpis poniżej mówi, w którym miejscu wywołano tę funkcję (z jakiej funkcji, jakie były parametry i w której linii). Kolejny wpis mówi z kolei, skąd wywołaliśmy funkcję na drugiej pozycji itp..



Okno Immediate

- ▶ Okno Immediate jest pewnym brudnopisem. Możemy w nim wpisywać wyrażenia, a środowisko obliczy ich wartość.
- ▶ Jeżeli chcemy dowiedzieć się, jaką wartość ma wyrażenie $a+b*99$ (gdzie a i b to zmienne z naszego programu) możemy w tym oknie wpisać $a+b*99$ i nacisnąć <Enter>. Wynik pojawi się w następnej linii.
- ▶ Możemy skorzystać z tego okna także do nadawania wartości zmiennym (pisząc np. $a = 8$) czy obliczania wartości funkcji z naszego programu, np. $\text{Funkcja}(a, 12)$.







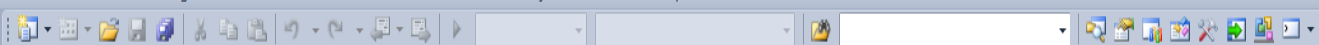
Microsoft®
Visual Studio®

Visual Studio 2010

>> Windows Forms

Windows Forms - powtórka

- ▶ Windows Forms – jest zbiorem elementów służących do tworzenia aplikacji okienkowych.
- ▶ Jednakże tworzenie aplikacji okienkowych to nie tylko tworzenie kodu, ale również projektowanie interfejsu użytkownika, który jest jednym z podstawowych elementów aplikacji okienkowej i decyduje w znacznym stopniu o jej atrakcyjności.



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Start Page



Recent

☒ Close
☒ Show

Output

Show out

New Project

Recent Templates

Installed Templates

- rest
- Win32
- Other Languages
 - Visual Basic
 - Visual C#
 - Windows
 - Web
 - Office
 - Cloud
 - Reporting
 - SharePoint
 - Silverlight
 - Test
 - WCF
 - Workflow
- Visual F#
- Other Project Types
- Database
- Modeling Projects
- Test Projects

Online Templates

.NET Framework 4

Sort by: Default

Search Installed Templates

	Win32 Console Application	Visual C++
	MFC Application	Visual C++
	Win32 Project	Visual C++
	Empty Project	Visual C++
	ATL Project	Visual C++
	MFC DLL	Visual C++
	Windows Forms Application	Visual C++
	CLR Console Application	Visual C++
	CLR Empty Project	Visual C++
	Class Library	Visual C++
	Custom Wizard	Visual C++

Type: Visual C++

A project for creating an application with a Windows user interface

Name: winform

Location: c:\users\rafal linowiecki\documents\visual studio 2010\Projects

Browse...

Solution name: winform

☒ Create directory for solution☐ Add to source control

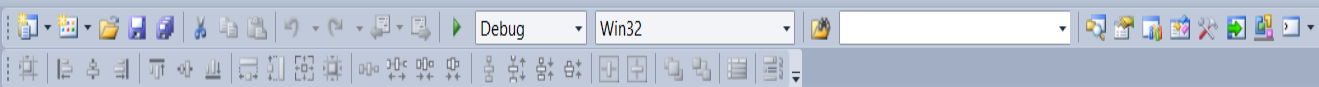
OK

Cancel

Solution Explorer



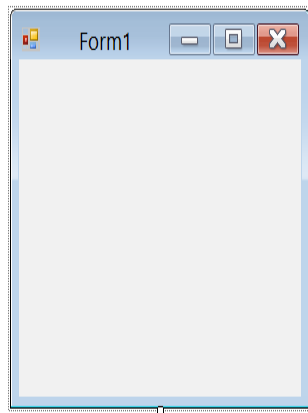
Server Explorer



Toolbox

- ▼ NameComboBox
- PictureBox
- ProgressBar
- RadioButton
- RichTextBox
- TextBox
- ToolTip
- TreeView
- WebBrowser
- Containers
 - Pointer
 - FlowLayoutPanel
 - GroupBox
 - Panel
 - SplitContainer
 - TabControl
 - TableLayoutPanel
- Menus & Toolbars
 - Pointer
 - ContextMenuStrip
 - MenuStrip
 - StatusStrip
 - ToolStrip
 - ToolStripContainer
- Data
 - Pointer
 - Chart
 - BindingNavigator
 - BindingSource
 - DataGridView
 - DataSet
- Components
 - Pointer
 - BackgroundWorker
 - DirectoryEntry
 - DirectorySearcher

Form1.h [Design]



Solution Explorer

- Solution 'winform' (1 project)
 - winform
 - External Dependencies
 - Header Files
 - Form1.h
 - Form1.resX
 - resource.h
 - stdafx.h
 - Resource Files
 - app.ico
 - app.rc
 - Source Files
 - AssemblyInfo.cpp
 - stdafx.cpp
 - winform.cpp
 - ReadMe.txt

Output

Show output from:



Microsoft®
Visual Studio®

Visual Studio 2010

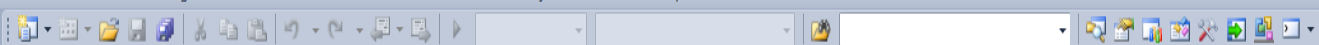


Windows Workflow
Foundation

Windows Workflow Foundation

- ▶ Windows Workflow Foundation – struktura która pozwala użytkownikom utworzyć systemowe bądź ludzkie przepływy pracy (workflows) w swoich aplikacjach dla systemów Windows.
- ▶ Składa się z przestrzeni nazw, wewnątrzprocesowego silnika przepływu pracy oraz kreatorów dla Visual Studio.
- ▶ WF może być użyty do prostych scenariuszy jak np. przedstawianie kontrolek UI bazując na danych wprowadzonych przez użytkownika czy kompleksowych korporacyjnych rozwiązań jak np. obsługa zamówień produktów czy zarządzanie magazynem.
- ▶ Windows Workflow Foundation wprowadza specyficzny model programowania, re-hostowalny i dostosowywalny silnik przepływu pracy oraz narzędzia do szybkiego budowania aplikacji modelu przepływu pracy na platformach Windows.

- ▶ Windows Workflow Foundation zapewnia spójny i znany z innych technologii .NET Framework 3.0 system tworzenia aplikacji, takich jak Windows Communication Foundation oraz Windows Presentation Foundation. API Windows Workflow Foundation umożliwia pełne wsparcie dla Visual Basic .NET oraz C#, wyspecjalizowany kompilator, debugowanie z wewnątrz przepływu pracy, graficzne projektowanie przepływu pracy oraz tworzenie przepływu danych przez pisanie kodu.
- ▶ Windows Workflow Foundation umożliwia, także rozszerzanie modelu do budowy dostosowywalnych aktywności (activities) które zawierają pełną funkcjonalność przepływu pracy, dla użytkownika końcowego do wielokrotnego użycia na przestrzeni wielu projektów.



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Start Page



Recent

☒ Close
☒ Show

Output

Show out

New Project

Recent Templates

Installed Templates

- Visual C++
- Other Languages
 - Visual Basic
 - Visual C#
 - Windows
 - Web
 - Office
 - Cloud
 - Reporting
- SharePoint
- Silverlight
- Test
- WCF
- Workflow
- Visual F#
- Other Project Types
- Database
- Modeling Projects
- Test Projects

Online Templates

.NET Framework 4

Sort by: Default

Search Installed Templates

	Silverlight Application	Visual C#
	Silverlight Class Library	Visual C#
	WCF Service Application	Visual C#
	ASP.NET Dynamic Data Entities Web Application	Visual C#
	Enable Windows Azure Tools	Visual C#
	Excel 2010 Workbook	Visual C#
	Outlook 2010 Add-in	Visual C#
	Word 2010 Document	Visual C#
	Activity Library	Visual C#
	WCF Workflow Service Application	Visual C#
	Crystal Reports application	Visual C#

Type: Visual C#

A WCF Workflow Service Application

Name: DeclarativeServiceLibrary2

Location: c:\users\rafal linowiecki\documents\visual studio 2010\Projects

Browse...

Solution name: DeclarativeServiceLibrary2

☒ Create directory for solution☐ Add to source control

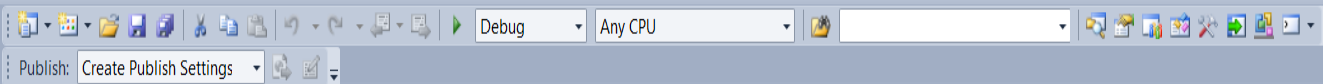
OK

Cancel

Solution Explorer



Server Explorer



Toolbox

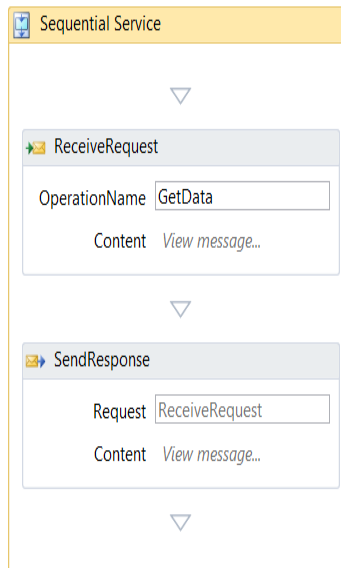
- Control Flow
 - Pointer
 - DoWhile
 - ForEach<T>
 - If
 - Parallel
 - ParallelForEach<T>
 - Pick
 - PickBranch
 - Sequence
 - Switch<T>
 - While
- Flowchart
 - Pointer
 - Flowchart
 - FlowDecision
 - FlowSwitch<T>
- Messaging**
 - Pointer
 - CorrelationScope
 - InitializeCorrelation
 - Receive
 - ReceiveAndSendReply
 - Send
 - SendAndReceiveReply
 - TransactedReceiveScope
- Runtime
- Primitives
- Transaction
- Collection
- Error Handling
- Migration
- General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Service1.xamlx

WorkflowService

Expand All Collapse All



Variables Imports

Output

Show output from:

Solution Explorer

- Solution 'DeclarativeServiceLibrary2' (1 pr)
- DeclarativeServiceLibrary2**
 - Properties
 - References
 - App_Data
 - Service1.xamlx
 - Web.config



Visual Studio 2010

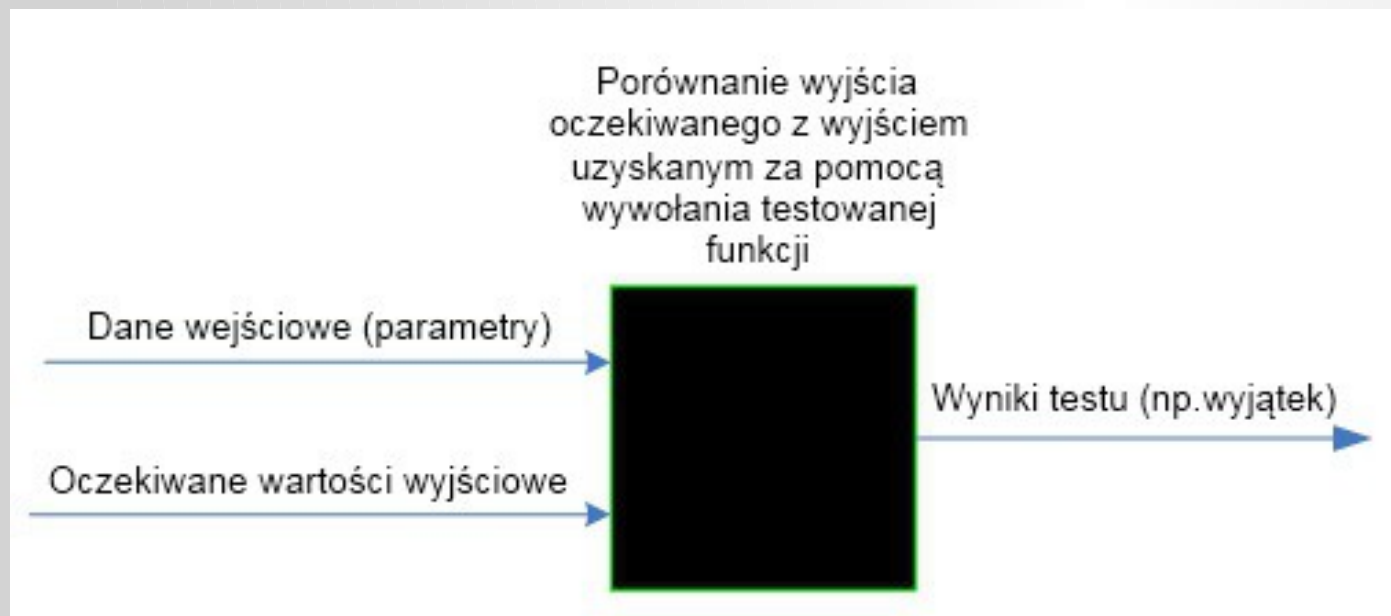
»» Testy jednostkowe

Testy jednostkowe

- ▶ Testy jednostkowe (ang. *unit test*) porównują oczekiwany wynik funkcji z rzeczywistym rezultatem. Na przykład sprawdzimy funkcję sumującą dwie liczby całkowite. W łatwy sposób można ustalić dane wejściowe oraz oczekiwany wynik:

Liczba A	Liczba B	Oczekiwany wynik
1	1	2
2	3	5
-5	5	0

- ▶ Testy jednostkowe często są nazywane testami czarnej skrzynki dlatego, że tester nie musi znać kodu zawartego w badanej funkcji. Dla testera ważny jest wyłącznie wynik wywołania funkcji – nie musi analizować kodu w niej zawartego.





Visual Studio 2010

»» Team Foundation Server

Team Foundation Server

Visual Studio Team Foundation Server 2010 jest platformą współpracy stanowiącą rdzeń rozwiązania do zarządzania cyklem życia aplikacji firmy Microsoft. Automatyzuje ona proces dostarczania oprogramowania i pozwala organizacjom na skuteczne zarządzanie projektami. Dzięki Team Foundation Server 2010 każdy członek zespołu może być bardziej skuteczny, efektywniej współpracować i dostarczać oprogramowanie lepszej jakości, tworząc jednocześnie bazę wiedzy i dzieląc się nią z innymi.

Korzyści stosowania TFS

- ▶ Usprawnienie przepływu danych w całym zespole
- ▶ Podgląd postępów prac w czasie rzeczywistym
- ▶ Możliwość kompleksowego śledzenia prac nad projektem oraz wykonywania audytu kodu źródłowego i wszystkich zmian
- ▶ Obsługa procesu Agile, Scrum, CMMI oraz swoich własnych metodyk

Niezbędne oprogramowanie



Microsoft
Visual Studio®
Team Foundation Server 2010



Microsoft
Visual Studio® 2010

Przeglądarka internetowa

Dodatkowo można połączyć się z systemem z poziomu



Microsoft
Project 2010



Microsoft
Excel.2010



Microsoft
SharePoint® Server 2010

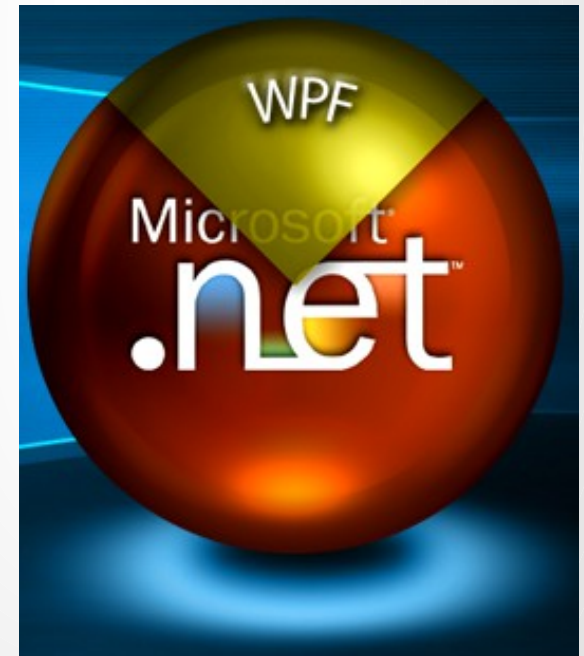


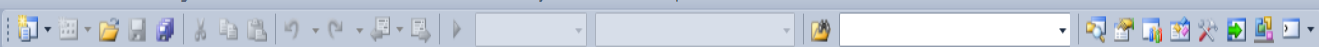
Visual Studio 2010

»» Windows Presentation
Foundation

Windows Presentation Foundation

- ▶ Windows Presentation Foundation (WPF, nazwa kodowa Avalon) – nazwa silnika graficznego i API bazującego na .NET 3, wchodzącego w skład WinFX.
- ▶ WPF integruje interfejs użytkownika, grafikę 2D i 3D, multimedia, dokumenty (nazwa kodowa Metro) oraz generowanie/rozpoznawanie mowy (do aplikacji sterowanych głosem).





Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Start Page



Recent Templates

Installed Templates

Visual C++

Other Languages

Visual Basic

Visual C#

Windows

Web

Office

Cloud

Reporting

SharePoint

Silverlight

Test

WCF

Workflow

Visual F#

Other Project Types

Database

Modeling Projects

Test Projects

Recent

Show

Output

Show output

Code Definition Window

Output

Error List

Immediate Window

New Project

.NET Framework 4

Sort by: Default

Search Installed Templates



Windows Forms Application

Visual C#



WPF Application

Visual C#



Console Application

Visual C#



ASP.NET Web Application

Visual C#



Class Library

Visual C#



ASP.NET MVC 2 Web Application

Visual C#



Silverlight Application

Visual C#



Silverlight Class Library

Visual C#



WCF Service Application

Visual C#



ASP.NET Dynamic Data Entities Web Application

Visual C#



Enable Windows Azure Tools

Visual C#

Type: Visual C#

Windows Presentation Foundation client application

Name:

WPF

Location:

c:\users\rafal linowiecki\documents\visual studio 2010\Projects

Browse...

Solution name:

WPF

☒ Create directory for solution☐ Add to source control

OK

Cancel

Solution Explorer



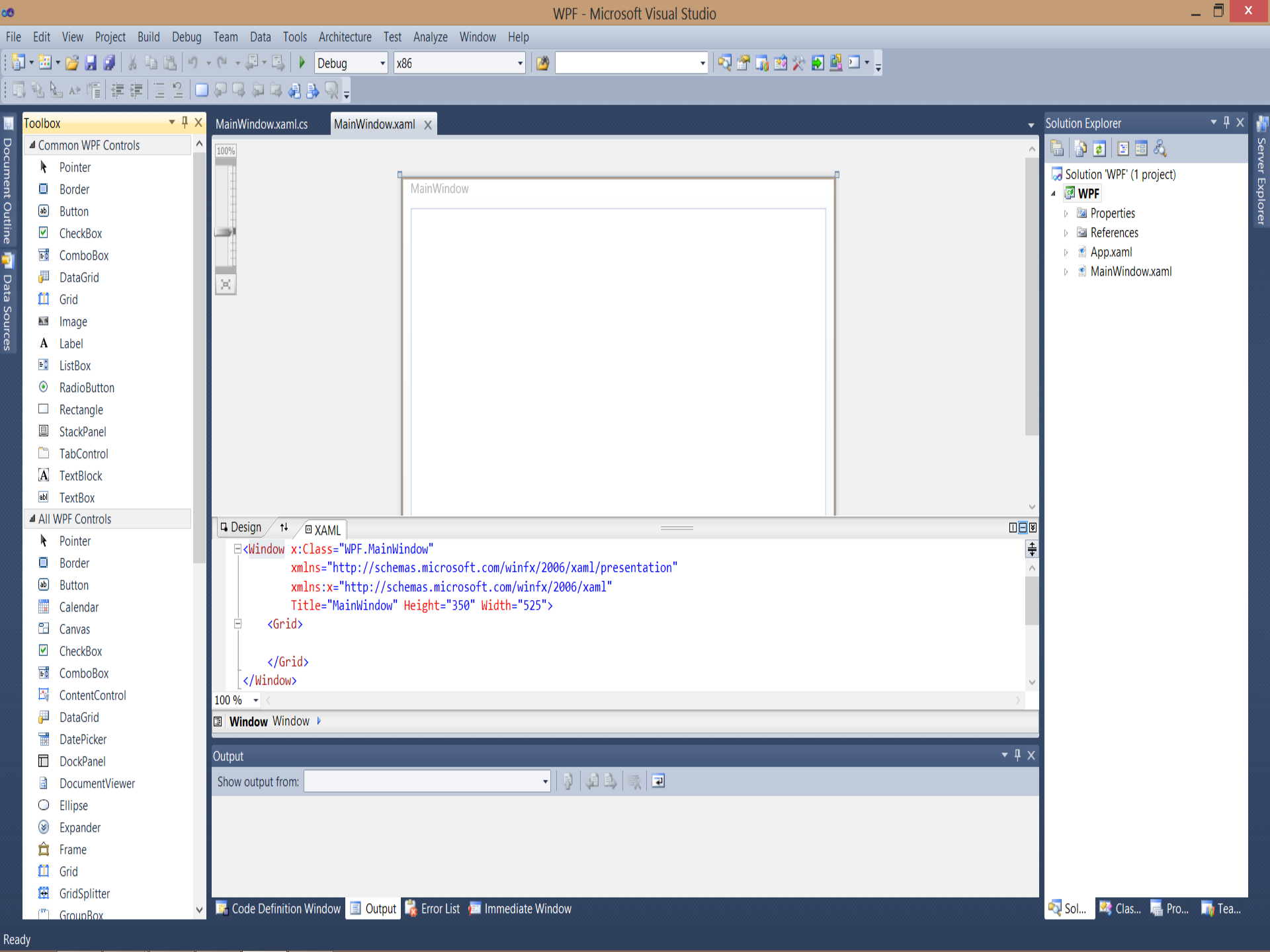
Server Explorer

Sol...

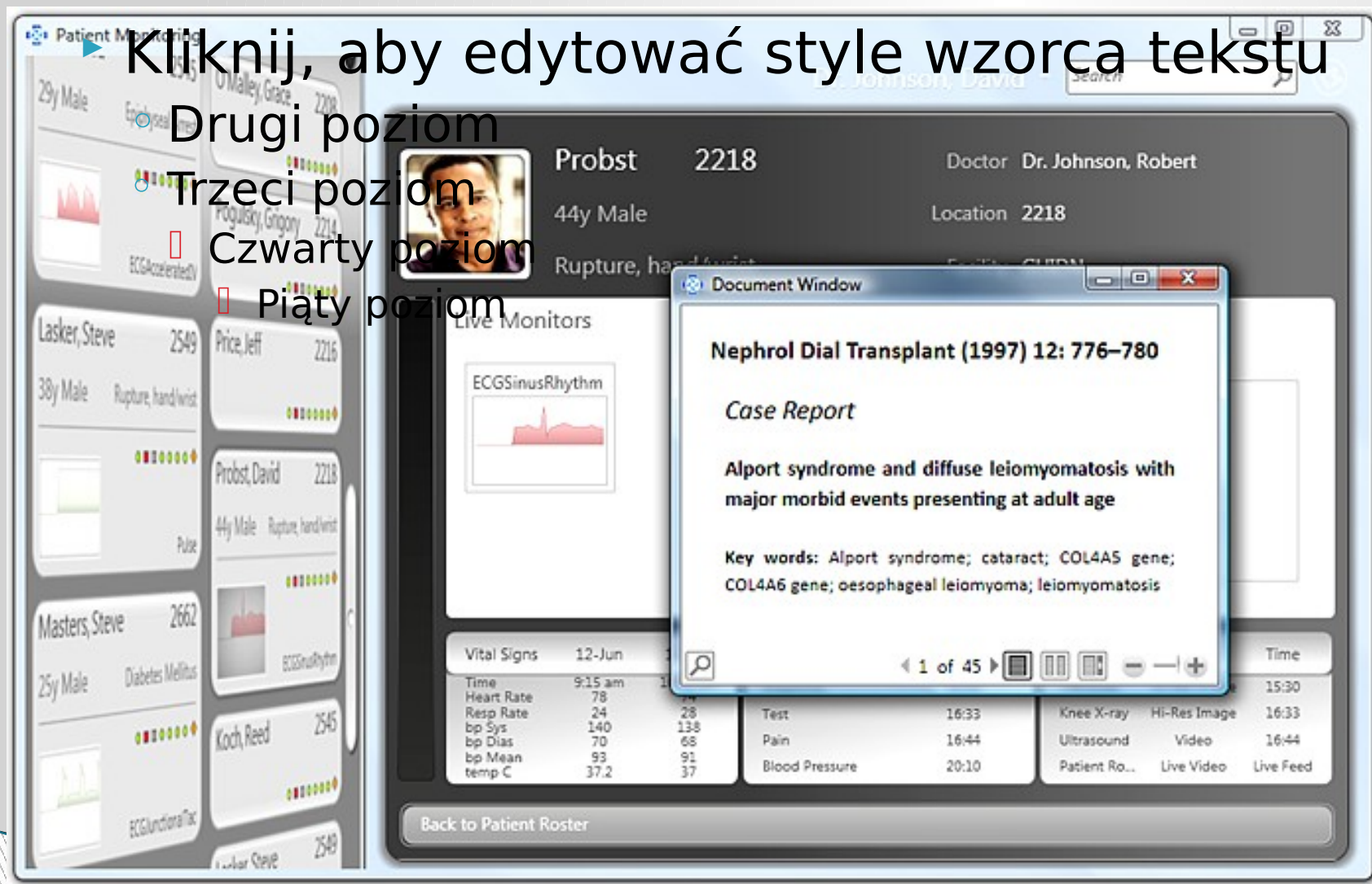
Clas...

Pro...

Tea...



Przykład aplikacji WPF



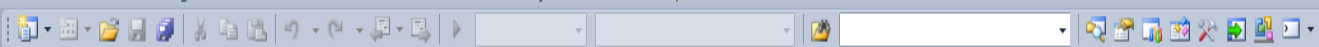


Visual Studio 2010

»» Refaktoryzacja

Refaktoryzacja kodu

- ▶ Jest to proces wprowadzania zmian w projekcie/programie, w wyniku którego zasadniczo nie zmienia się funkcjonalność.
- ▶ Celem refaktoryzacji jest więc nie wytwarzanie nowej funkcjonalności, ale utrzymywanie odpowiedniej, wysokiej jakości organizacji systemu.
- ▶ W ramach refaktoryzacji podejmowane są następujące działania:
 - modyfikowanie elementów systemu w celu wpasowania ich w przyjęte standardy i wzorce
 - poszukiwanie nowych standardów i wzorców, które pojawiły się w systemie w trakcie jego rozwoju i ich precyzyjne definiowanie (łącznie z wpasowywaniem istniejących elementów w te definicje).



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.



Recent

☒ Close
☒ Show

Output

Show out

Start Page

Recent Templates

Installed Templates

Visual C++

- ATL
- CLR
- General
- MFC
- Test
- Win32

Other Languages

- Visual Basic
- Visual C#
- Visual F#

Other Project Types

- Database
- Modeling Projects
- Test Projects

Online Templates

Name:

Refactor

Location:

c:\users\rafal linowiecki\documents\visual studio 2010\Projects

Browse...

Solution name:

Refactor

☒ Create directory for solution☐ Add to source control

OK

Cancel

New Project

.NET Framework 4

Sort by: Default

Search Installed Templates



Windows Forms Application

Visual C#



WPF Application

Visual C#



Console Application

Visual C#



ASP.NET Web Application

Visual C#



Class Library

Visual C#



ASP.NET MVC 2 Web Application

Visual C#



Silverlight Application

Visual C#



Silverlight Class Library

Visual C#



WCF Service Application

Visual C#



ASP.NET Dynamic Data Entities Web Application

Visual C#



Enable Windows Azure Tools

Visual C#

Type: Visual C#

A project for creating a command-line application

There are no us
group. Drag an i
add it to the toolbox.

add it to the toolbox.

```
using System.Linq;  
using System.Text;
```

```
namespace Refactor
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            string firstName = "Jan";
```

```
            string lastName = "Kowalski";
```

```
            Console.WriteLine(string.Format("FirstName:{0}", firstName));
```

```
            Console.WriteLine(string.Format("LastName:{0}", lastName));
```

```
            Console.ReadLine();
```

```
        }
```

```
    }
```

```
}
```

wyodrębnienie metody

100 %

Output

Show output from: Debug

```
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0.4.0.0__b77a5c561934e089\mscorlib.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\assembly\GAC_MSIL\Microsoft.VisualStudio.HostingProcess.Utilities\10.0.0.0__b03f5f7f11d50a3a\Microsoft.VisualStudio.HostingProcess.Utilities.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Windows.Forms\v4.0.4.0.0__b77a5c561934e089\System.Windows.Forms.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Drawing\v4.0.4.0.0__b03f5f7f11d50a3a\System.Drawing.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System\v4.0.4.0.0__b77a5c561934e089\System.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\assembly\GAC_MSIL\Microsoft.VisualStudio.HostingProcess.Utilities.Sync\10.0.0.0__b03f5f7f11d50a3a\Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll'  
'Refactor.vshost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\assembly\GAC_MSIL\Microsoft.VisualStudio.Debugger.Runtime\10.0.0.0__b03f5f7f11d50a3a\Microsoft.VisualStudio.Debugger.Runtime.dll'
```

Code Definition Window Output Error List Immediate Window

Solution Explorer

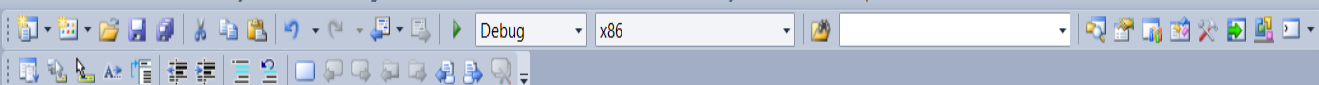
Solution 'Refactor' (1 project)

Refactor

Properties

References

Program.cs



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Program.cs*

Refactor.Program

Main(string[] args)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Refactor
```

```
{
    class Program
```

```
{
    static void Main(string[] args)
```

```
{
    string firstName =
```

```
    string lastName =
```

```
    Console.WriteLine(
```

```
    Console.WriteLine(
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

```
    Console.ReadLine()
```

nazwa
tworzonej
metody

Extract Method

New method name:

Nazwa_utworzonej_metody

Preview method signature:

private static void Nazwa_utworzonej_metody(string firstName, string lastName)

OK

Cancel

deklaracja
tworzonej
metody

Solution Explorer

Solution 'Refactor' (1 project)

Refactor

Properties

References

Program.cs

100 %

Output

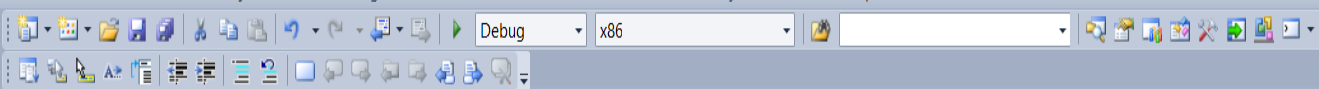
Show output from: Debug

Extract method

Extracting local variables information...

Code Definition Window Output Error List Immediate Window

Sol... Clas... Pro... Tea...



Toolbox

General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Program.cs*

Refactor.Program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Refactor
```

```
{
    class Program
```

```
    {
        static void Main(string[] args)
```

```
        {
            string firstName = "Jan";
```

```
            string lastName = "Kowalecki";
```

```
            Nazwa_utworzonej_metody(firstName, lastName);
```

```
        private static void Nazwa_utworzonej_metody(string firstName, string lastName)
```

```
        {
            Console.WriteLine(string.Format("FirstName:{0}", firstName));
```

```
            Console.WriteLine(string.Format("LastName:{0}", lastName));
```

```
            Console.ReadLine();
        }
    }
}
```

deklaracja nowo utworzonej metody

definicja nowo utworzonej metody

100 %

Output

Show output from: Refactor

Extract method

Extracting local variables information...

Generating new method code

Code Definition Window Output Error List Immediate Window

Solution Explorer

Solution 'Refactor' (1 project)

Refactor

Properties

References

Program.cs



Visual Studio 2012

»» Informacje wstępne

Visual Studio 2012

- ▶ Microsoft Visual Studio 2012 oferuje wiele nowych możliwości opracowywania aplikacji dla systemu Windows 8, sieci Web, środowiska SharePoint, urządzeń przenośnych oraz chmury.
- ▶ Zapewnia również narzędzia do zarządzania cyklem życia aplikacji, które pozwalają na wyeliminowanie barier w pracy zespołów i skrócenie całego cyklu w celu uzyskania dodatkowych korzyści.
- ▶ Ponadto zaprojektowany od nowa interfejs użytkownika upraszcza wykonywanie typowych operacji i poprawia wydajność.



Wersje VS 2012

- ▶ Ultimate - najlepszy zestaw narzędzi ułatwiających tworzenie aplikacji oraz inne działania w przedsiębiorstwie.
- ▶ Premium - dla elastycznych, dynamicznie dostosowujących się zespołów.
- ▶ Professional - dla pojedynczych programistów i niewielkich zespołów.
- ▶ Test Professional - lepsze możliwości dla testerów i innych uczestników projektu: całkowita integracja z procesem tworzenia oprogramowania.

Produkty VS Express 2012

- ▶ Za pomocą narzędzi Visual Studio Express można projektować wspaniałe aplikacje dla systemu Windows 8, Windows Phone i sieci Web. Największa zaleta? Narzędzia są całkowicie bezpłatne.



Porównanie funkcji dla wersji

Kategorie i funkcje Rozwiń wszystkie	Polecany film wideo	Ultimate z subskrypcją MSDN	Premium z subskrypcją MSDN	Test Professional z subskrypcją MSDN	Professional z subskrypcją MSDN	Professional
⊕ Debugowanie i diagnostyka		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Narzędzia do testowania		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Zintegrowane środowisko projektowania		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Obsługa platformy deweloperskiej		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Architektura i modelowanie		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Zarządzanie laboratorium		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Team Foundation Server		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Współpraca		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Oprogramowanie i usługi do użytku produkcyjnego		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Oprogramowanie do tworzenia i testowania aplikacji		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
⊕ Inne korzyści		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■

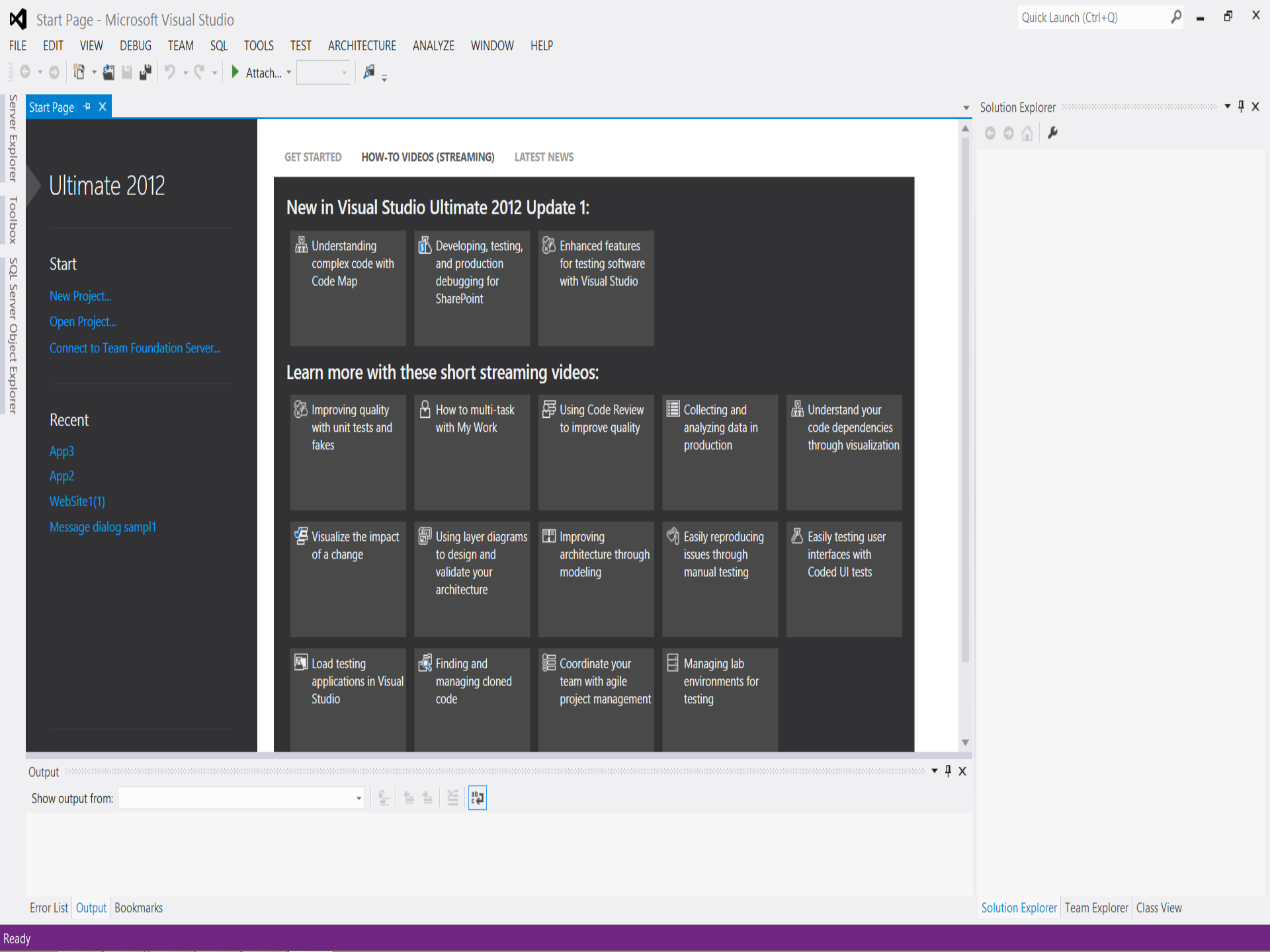


Visual Studio 2012

»» Co nowego?

Zupełnie nowy interfejs.

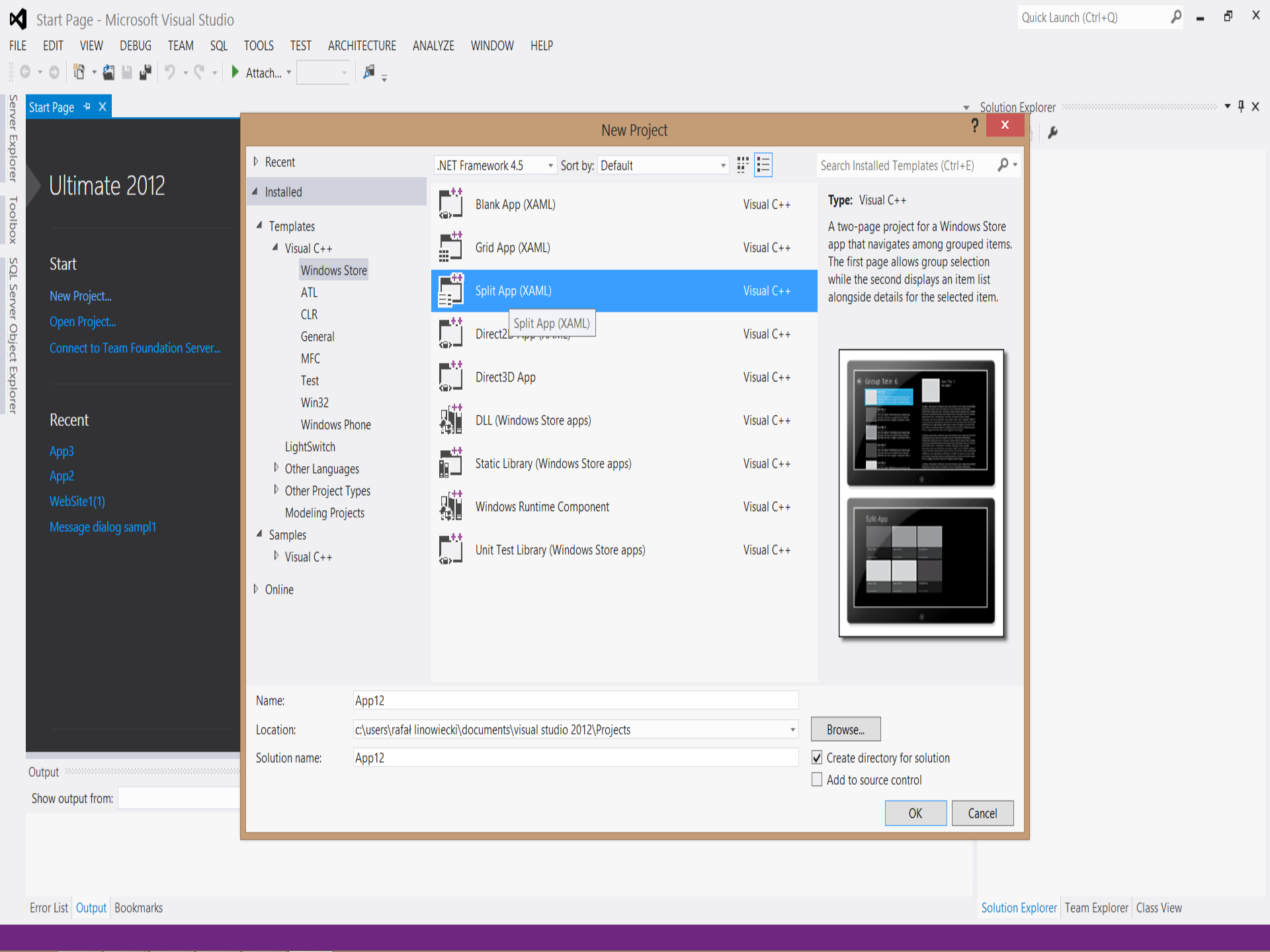
Już na pierwszy rzut oka zauważamy zupełnie inne środowisko IDE. Cały interfejs został zaprojektowany od nowa, aby usprawnić obieg pracy i zapewnić łatwy dostęp do narzędzi używanych na co dzień. Paski narzędzi i karty zostały uproszczone oraz dodano nowe, szybkie sposoby znajdowania kodu. Dzięki tym zmianom można wygodnie obsługiwać aplikację i wykonywane zadania.



Zgodność z systemem Windows 8

- ▶ Wraz z wprowadzeniem systemu Windows 8 rynek aplikacji uległ gruntownym zmianom. Visual Studio 2012 zapewnia nowe szablony, narzędzia do projektowania, testowania i debugowania, czyli wszystkie elementy potrzebne do błyskawicznego tworzenia atrakcyjnych aplikacji. Dodatkowo program Blend for Visual Studio zapewnia zestaw narzędzi wizualnych, które pozwalają w pełni wykorzystać zalety nowego interfejsu systemu Windows 8.
- ▶ Największą jednak zaletą są możliwości dostępne po ukończeniu prac nad aplikacją. W przeszłości przedstawienie gotowych produktów klientom nie zawsze było łatwe. Obecnie masz do dyspozycji Sklep Windows — ogólnodostępny kanał dystrybucji docierający do milionów użytkowników. Warunki są przejrzyste, a potencjał łatwy do zauważenia. Możesz więc opracować aplikację, rozpocząć sprzedaż i być może spędzić kilka następnych lat na plaży.

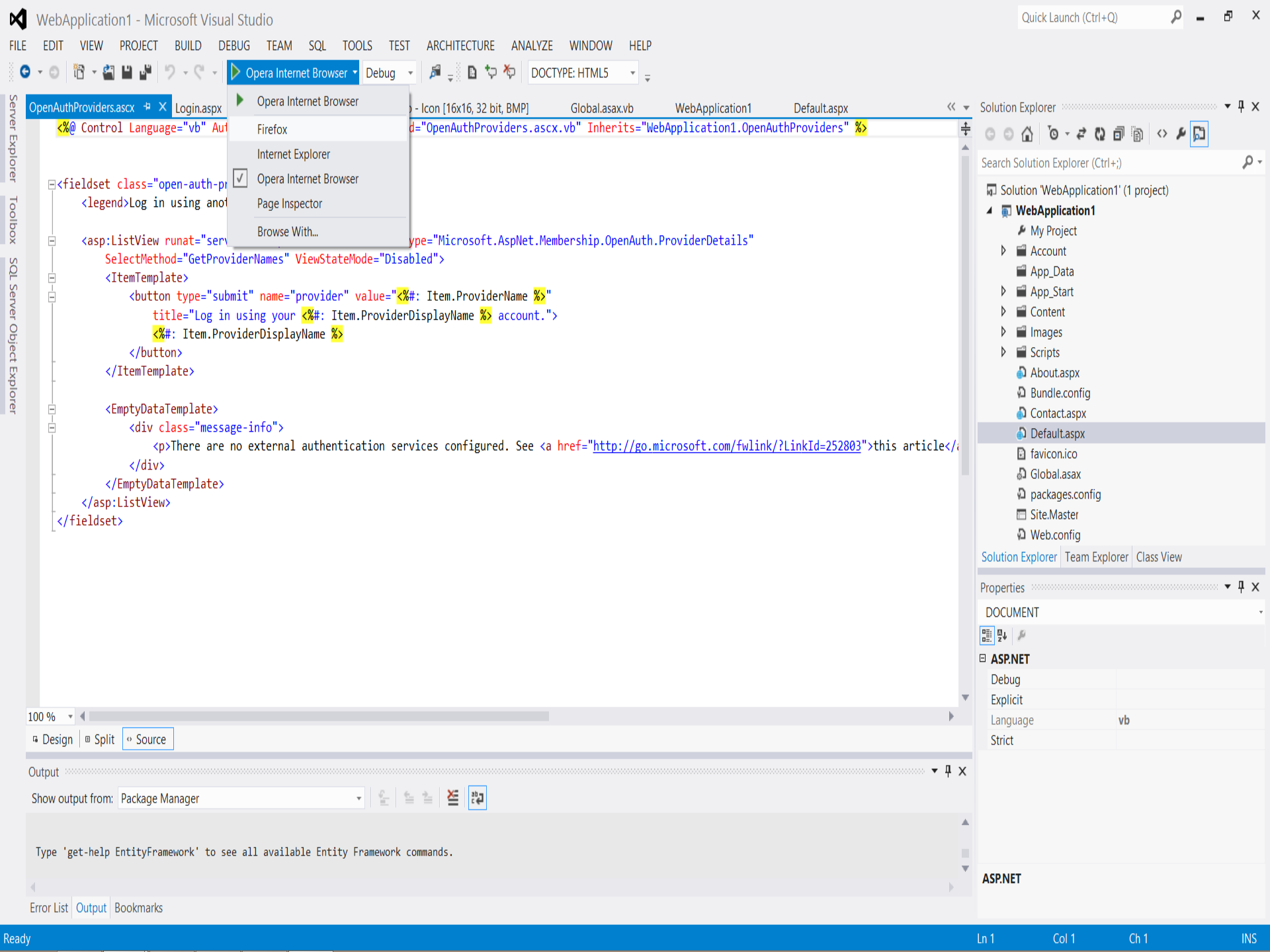




Coś dla webmasterów - czyli nowe sposoby tworzenia kodu dla aplikacji webowych.

- ▶ Visual Studio 2012 ułatwia również tworzenie aplikacji sieci Web dzięki nowym szablonom, lepszym narzędziom do publikowania oraz pełnej obsłudze nowych standardów, takich jak HTML5 i CSS3, jak również najnowszych zmian platformy ASP.NET. Debugowanie jest łatwiejsze ze względu na nowe narzędzie Inspektor stron, które pozwala na testowanie opracowywanego kodu strony bezpośrednio w środowisku IDE.
- ▶ Chcesz się zająć urządzeniami przenośnymi? Dzięki technologii ASP.NET możesz tworzyć aplikacje z elementami sterującymi, które są optymalizowane z myślą o telefonach, tabletach oraz innych urządzeniach z ekranami dotykowymi.





Visual Studio 2012 + SkyDrive.

W przeszłości konieczne było utrzymywanie własnego serwera. Skalowanie wymagało znacznych inwestycji w infrastrukturę. Teraz masz szybki dostęp do praktycznie nieograniczonej liczby serwerów w chmurze oraz możliwość dodawania większej ilości miejsca i mocy obliczeniowej w każdej chwili. Visual Studio oferuje doskonałe narzędzia ułatwiające przenoszenie aplikacji na platformę Windows Azure, w tym nowe szablony i opcje publikowania, buforowanie rozproszone oraz mniejsze wymagania dotyczące instalacji.



Team Foundation Server + ALM

- ▶ Do tej pory skupialiśmy się głównie na tworzeniu aplikacji. Wraz ze wzrostem złożoności aplikacji potrzebne jednak stają się narzędzia pozwalające zespołowi pracować szybciej i sprawniej. Dzięki programom Visual Studio i Team Foundation Server można stosować bardziej optymalne sposoby postępowania. Sposoby dostosowane do nowych wymogów, które nie zakłócają obecnych procedur. Narzędzia do śledzenia wymagań oraz informacji zwrotnych od uczestników projektu, klientów i członków zespołu znacznie ułatwiają współpracę w ramach całej firmy.

- ▶ Istnieje nawet możliwość zlecenia firmie Microsoft obsługi działań związanych z zarządzaniem cyklem życia aplikacji. Rozwiązanie Team Foundation Service zapewnia dostęp do funkcji ALM bez konieczności wdrożenia infrastruktury. W ten sposób nawet najmniejsze zespoły mogą uzyskać korzyści z mechanizmów kontroli wersji, oceny kodu oraz narzędzi do elastycznego planowania.
- ▶ Application Lifecycle Management (ALM, zarządzanie cyklem życia aplikacji) to ciągły proces zarządzania życiem wytwarzanego oprogramowania związany z 4 podstawowymi pojęciami: kod, proces, jakość, architektura.

What's new in .NET Framework 4.5?

Windows Presentation Foundation

Built-in Ribbon controls ★
 Databinding improvements
 Ability to add breakpoints to databindings
 Data source change aware views (Live Shaping) ★
 Validation improvements ⚡
 Improved legacy UI integration
 Dispatcher improvements ⚡
 Speed-up of large data sets 📈

Windows 8 support ★

Support for
Windows Runtime (WinRT)

.NET Profile
for Metro-style apps

Improved support for sharing
DLLs between .NET profiles

ASP.NET

ASP.NET Web API -
a new framework for REST endpoints ★

Support for implementing
WebSocket receivers ★

Built-in JavaScript + CSS
combining and minification ★

ASP.NET MVC 4

Async controllers ⚡
 Built-in mobile templates +
jQuery.Mobile support

Alternate views (e.g.
print version, mobile site)

Support for Recipes:
intelligent codegen ★

ASP.NET Web Pages 2

New site templates
 Versatile validation support
 Support for OAuth and OpenID ★
 Built-in map embedding tools;
supports Google, Bing and others

Did you know? ASP.NET Web Pages is yet another way to work the web besides Web Forms and MVC. WP sites use Razor and are typically developed with WebMatrix.

Web Forms

Strongly typed data binding
 MVC-like support for Models ★
 HTML encoded binding expressions

HTML5 support ★

Control support for new semantic elements
 Multifile support for FileUpload control
 Validator and UpdatePanel now support new HTML5 elements

Tooling

Markup editors improved (a lot)
 IIS Express used by default

Page Inspector ★

New templates and snippets

Asynchronous pipeline support ⚡
(Response, Request, HttpHandlers)

Performance improvements: Multicore JIT, 35 % faster startup, memory optimizations, assembly sharing between sites, pre-fetch support 📈

Request validation improvements:
AntiXSS built-in, validation usable per file

Windows Communication Foundation

New channels (UDP multicast, WebSockets, ...) ★
 Support for contract-first development ★
 Asynchronous operations ⚡
 Streaming improvements
 Simplified configuration, VS config validation
 Multiple auth modes for HTTP endpoints
 New, simple HttpClient class

Windows Workflow Foundation

C# Expressions
 State machine workflows are back! ★
 Workflow versioning ★
 Code-first activity design
 Faster execution 📈
 Designer usability improvements

Managed Extensibility Framework 2.0

Support for generic parts ★
 Debugging improvements
 Support for explicit and convention-based bindings between objects ★
 Support for binding POCOs: no more attribute requirements

ADO.NET

Sparse columns support improved (SQL Server)
 Passwords are now stored encrypted
 Asynchronous operations ⚡

SQL Express LocalDB

New light version of SQL Express for developer use. Supported in .NET 4.5, separate patch for 4.0 is coming.

SQL Server 2012 ("Denali") Support

High Availability support on connection string level
 Fast failover across multiple subnets
 Support for new spatial data types (polygons, arcs etc.) ★

Entity Framework 4.5

Enumeration support
 Migrations for schema changes ★
 Designer improvements

Spatial data type support ★
 Table-valued function support ★
 Multi-result sproc support

Multiple diagrams per model
 Code-first support ★
 Auto-compiled LINQ queries 📈

Base Class Library

Networking improvements (IPv6 enhanced, IDN, EAI etc.)
 Key interfaces (e.g. file IO) now support async ⚡
 New ArraySegment and ReadOnlyDictionary classes
 Support for CLR objects over 2 GB in size
 Resource file management performance improved 📈
 Unicode improvements (v6, console support)
 Background JIT on multicore platforms 📈

Task Parallel Library ⚡

Task thread controls improved:
Task.WaitAll/WaitAny, various timeout primitives available

TPL Dataflow: Tools for parallel data flow processing ★

C# 5.0

Support for async programming:
async and await keywords ⚡
 Methods can access call site info as parameters (CallerInfo)

Visual Basic 11

Iterator implementations (Yield)
 Async and Await equal to C# ⚡
 "Global" keyword for namespace referencing
 Call Hierarchy view available
 CallerInfo attributes (as in C#)

F# 3.0

Type providers ★
 Query expressions (LINQ) ★
 Auto-implemented properties

Visual C++ 11

C++11 standard support improved
 Auto-vectorization and parallelization of loops ⚡
 GPU-driven processing (C++ AMP) 📈
 Intellisense for C++/CLI

Legend:

⚡ = Asynchrony support
 📈 = Performance improvement
 ★ = Significant new feature

Microsoft
tech·days
Helsinki 2012

SANKO

Suomen Aktiivisten .NET-kehittäjien Kerho
www.facebook.com/sanko.net

offbeat
solutions
www.offbeat.fi

Information based on public sources available updated on 1st March 2012 (.NET 4.5 beta).

Composed by Jouni Heikkinen.
 Original with updates available @
www.heikkinen.net/hardcode/

Źródła

- ▶ „Zrozumieć platformę .NET” – Wydanie II David Chappell
- ▶ MS Visual Studio - <http://www.microsoft.com/visualstudio>
- ▶ MS MSDN - <http://msdn.microsoft.com/>
- ▶ Codeproject - <http://www.codeproject.com/>
- ▶ Wikipedia - <http://en.wikipedia.org>
- ▶ 4 Programmers - <http://4programmers.net>
- ▶ Barbara Czapiewska – opis .NET – wykład+laboratorium
- ▶ Andrzej Stefańczyk – Tworzenie aplikacji okienkowych - <http://ebooks.ms.strefa.pl/dane/c-sharp-2.pdf>
- ▶ Marek Sawerwain – Platforma .NET



Dziękujemy za uwagę!