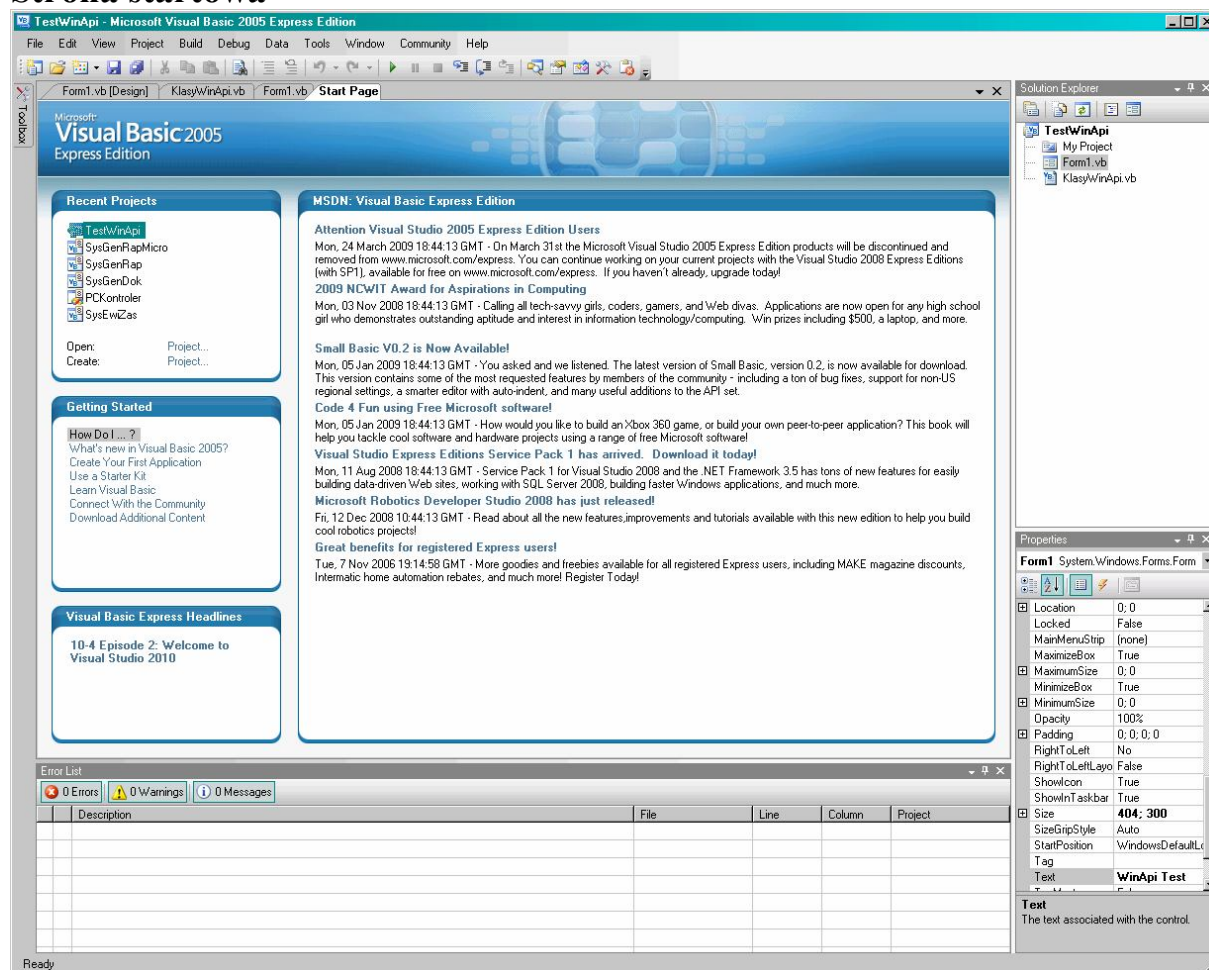


# IDE – (ang. Integrated Development Environment) czyli zintegrowane środowisko programistyczne

## Strona startowa



## Elementy strony startowej

**Panel Recent Project** -zawiera listę projektów, nad którymi programista ostatnio pracował.

Można zmienić domyślne ustawienia IDE i dopasować je do głównych zastosowań środowiska (Visual Basic, C#, C++, aplikacje sieciowe i tak dalej), .j. Aby dokonać takiej zmiany, należy otworzyć menu **Tools** i Wybrać opcję **Import and Export Settings Wizard**. W kreatorze, który się otworzy, można nie tylko importować i eksportować ustawienia, ale także przywrócić profil domyślny. Wybranie nowego profilu spowoduje automatyczne dostosowanie do niego strony startowej.

**Panel Getting Started** –to punkt wyjścia dla programistów poznających dopiero środowisko Visual Studio. Odnośniki widoczne na tym panelu zwykle dotyczą tworzenia nowych projektów, używania szablonów lub zestawów startowych, odpowiedzi na pytania lub nauki nowych rzeczy. Ten panel, stanowi doskonały punkt wyjścia do samodzielnych poszukiwań.

**Panel Visual Studio Headlines** -wyświetla ogłoszenia oraz nagłówki wiadomości związanych z Visual Studio. Te nagłówki prowadzą do wartościowych dla programistów informacji na temat aktualizacji produktów, zmian i tak dalej.

Widoczny na środku **panel z wiadomościami** jest specyficzny dla środowiska Visual Studio. Informacje związane są z nowymi książkami, wersjami beta, narzędziami, audycjami webcast

i innymi produktami. Dzięki temu programista może zapoznać się z nowościami nawet wtedy, jeśli nie ma czasu na przeglądanie wiadomości w sieci MSDN (Microsoft Developer Network).

Wiadomości są przesyłane do IDE poprzez kanał RSS (Rich Site Summary). Podstawowym zadaniem RSS jest udostępnianie list o strukturze XML zawierających podsumowania i odnośniki do kluczowych informacji z określonej strony. Dzięki tej technologii użytkownicy Visual Studio mogą otrzymywać opublikowane skróty najważniejszych wiadomości z określonego centrum rozwojowego MSDN.

Popularne kanał RSS dla strony startowej Visual Studio:

Opis	Adres URL
<b>Architektura i platforma .NET</b>	
Platforma .NET	<a href="http://msdn.microsoft.com/netframework/rss.xml">http://msdn.microsoft.com/netframework/rss.xml</a>
Architektura	<a href="http://msdn.microsoft.com/architecture/rss.xml">http://msdn.microsoft.com/architecture/rss.xml</a>
Wzorce i praktyki	<a href="http://msdn.microsoft.com/practices/rss.xml">http://msdn.microsoft.com/practices/rss.xml</a>
Urządzenia przenośne	<a href="http://msdn.microsoft.com/mobility/rss.xml">http://msdn.microsoft.com/mobility/rss.xml</a>
Bezpieczeństwo	<a href="http://msdn.microsoft.com/security/rss.xml">http://msdn.microsoft.com/security/rss.xml</a>
<b>Programowanie dla sieci</b>	
ASP.NET	
Silverlight	
Usługi sieciowe	
<b>Visual studio i języki</b>	
Visual Studio	<a href="http://msdn.microsoft.com/vstudio/rss.xml">http://msdn.microsoft.com/vstudio/rss.xml</a>
Team System	
Nowinki o C#	<a href="http://msdn.microsoft.com/vcsharp/rss.xml">http://msdn.microsoft.com/vcsharp/rss.xml</a>
Nowinki o Visual Basic	<a href="http://msdn.microsoft.com/vbasic/rss.xml">http://msdn.microsoft.com/vbasic/rss.xml</a>
<b>Programowanie za pomocą produktów i platform Microsoftu</b>	
Nowinki o C++	
Microsoft Office	
SQL Server	<a href="http://msdn.microsoft.com/sql/rss.xml">http://msdn.microsoft.com/sql/rss.xml</a>
BizTalk	
Windows Vista	

## **Konfigurowanie środowiska programistycznego**

Do **zarządzania ustawieniami środowiska** służy opcja *Import and Export Settings* z menu *Tools*. W czasie importowania można wybrać jeden z licznych zbiorów ustawień. Dostępnych jest kilka zestawów domyślnych (niestety nie dostępnych w wersji Express), w tym ustawienia dostosowane do języków i ról (na przykład programista aplikacji sieciowych lub tester).

Do **konfigurowania ustawień środowiska** służy opcja *Options* z menu *Tools*.

## **Pierwszy projekt**

Z menu *File* można utworzyć nowy projekt lub witrynę internetową. Projekty to po prostu szablony dla aplikacji dla systemu Windows, pakietu Office, urządzeń przenośnych i innych programów wykonywalnych.

Po utworzeniu projektu pod paskiem menu mamy:

**Standardowy pasek narzędzi** - pasek zapewnia szybki dostęp do wszystkich często używanych poleceń. Aby dostosowywanie pasek narzędzi należy wybrać opcję *Customize* z menu *Tool* lub kliknąć prawym przyciskiem myszy pasek środowiska IDE i wybrać tę samą opcję. Następnie wystarczy wybrać elementy (ikony) paska narzędzi i przenieść je lub usunąć. Można też użyć

zakładki *Commands*, aby dodać określone polecenia do wybranego paska narzędzi, jeśli zmiany okażą się niezadowolające, wystarczy użyć przycisku *Reset* z zakładki *Toolbars* i przywrócić ustawienia domyślne.

Po prawej stronie pod standardowym paskiem narzędzi mam:

„**Okno narzędzi**” (Toolbox) - zapewnia dostęp do wielu kontrolki i komponentów przydatnych w czasie tworzenia formularzy sieciowych i Windows. Ponadto zawiera prawie wszystkie elementy, które można przeciągnąć do okien projektowych przy budowaniu formularzy, schematów XML, klas i innych obiektów.

W centrum IDE mamy:

„**Graficzne okno projektowe**” –to obszar, na którym można tworzyć elementy takie jak formularze poprzez przeciąganie, upuszczanie, przenoszenie, zmianę rozmiaru i tak dalej. Visual Studio udostępnia wiele takich okien. Wspólnie umożliwiają one tworzenie elementów składających się na aplikację (i ich zrozumienie). Te elementy obejmują formularze Windows, formularze sieciowe, schematy klas czy szablony XML.

i

„**Edytor kodu**” –to główne narzędzie w czasie tworzenia kodu. Dodaje wcięcia i odstępy, tak aby kod był przejrzysty i czytelny. Udostępnia mechanizm *IntelliSense* i mechanizm uzupełniania instrukcji, dzięki czemu nie trzeba sprawdzać (lub zapamiętywać) wszystkich nazw bibliotek obiektów ani słów kluczowych. Grupuje kod w bloki, stosuje kolory dla słów kluczowych i komentarzy, wyróżnia błędy i nowy kod dodany po ostatniej kompilacji. Podsumowując, edytor kodu w Visual Studio udostępnia wiele funkcji pozwalających zwiększyć produktywność.

Po prawej stronie edytora kodu mamy:

Okno „**Solution Explorer**” (Struktury plików projektu) -umożliwia grupowanie plików składających się na aplikację i manipulowanie nimi. Rozwiązanie może zawierać kilka aplikacji. Aplikacją może być witryna internetowa, aplikacja dla systemu Windows, biblioteka klas czy aplikacja konsolowa. W tych kontenerach znajdują się pliki formularzy i klas zawierające kod, pliki XML i inne elementy.

Pod oknem „Solution Explorer” umieszczono okno:

Okno „**Properties**” (Okno właściwości) –Umożliwia kontrolowanie rozmiaru, wyglądu i innych własności kontrolki. Ponadto grupuje wspólne właściwości, ułatwiając dostęp do nich. Za pomocą tego okna można też powiązać zdarzenia dotyczące danej kontrolki z kodem aplikacji.

## **Zarządzanie wieloma oknami środowiska IDE**

**Przyczepianie** (ang. pinning) to proces przyklejania okna w danym miejscu. Nazwa związana jest z graficzną reprezentacją tej operacji — pinezką służącą do przyczepiania (widoczną na pasku tytułu okna narzędzi).

**Dokowanie** to proces umieszczania okna w jednym z „lepkich” miejsc w obrębie IDE. Zwykle oznacza to dokowanie okna do lewej, górnej, prawej lub dolnej krawędzi środowiska.

Aby wykonać dokowanie, najpierw trzeba sprawdzić, czy okno jest przyczepione (nie można dokować odłączonych okien). Jeśli tak jest, należy kliknąć i przytrzymać pasek tytułu, a następnie przeciągnąć okno tam, gdzie programista chce je zadokować. Visual Studio wyświetla przy tym kilka ikon ułatwiających dokowanie. Cztery ikony reprezentują krawędzie IDE: lewą, górną, prawą i dolną. Te ikony służą do umieszczania okna przy danej krawędzi środowiska. Ich użycie powoduje zadokowanie okna na całej wysokości lub szerokości IDE. Ikona w środku strzałek służy do dokowania wybranego okna względem innego okna.

Dostępne opcje uruchamiane z menu na pasku narzędzia okna:

- *Floating* -oznacza, że okno będzie „pływać” w miejscu określonym przez programistę „nad” środowiskiem IDE.
- *Dockable* -wyłącza ustawienie Floating.
- *TabbedDocument* –umożliwia umieszczenie okna w centrum środowiska IDE, aby działało podobnie jak okna projektowe czy edytor kodu.

Koniec cz.1

## Projektowanie formularza

**Formularz** to kontener na różne kontrolki.

**Kontrolki** to podstawowy mechanizm interakcji z formularzem.

Aby umieścić kontrolkę na formularzu, należy przeciągnąć na niego jej reprezentację z okna narzędzi. Niektóre kontrolki, tak zwane *komponenty*, nie mają graficznego interfejsu użytkownika. Jednym z nich jest zegar (kontrolka *Timer*). Komponent przeciągnięty na formularz zostaje umieszczony w odrębnym obszarze okna projektowego, tak zwanym *obszarze komponentów*. W obszarze komponentów można wybrać jeden z nich i uzyskać dostęp do jego właściwości poprzez okno *Properties*.

Rozmieszczanie kontroltek:

- **Z menu *Format*** –po wybraniu grupy kontroltek i użyciu poleceń menu *Format* do wyrównania ich względem siebie w pionie lub w poziomie, do ujednoczenia, zmniejszenia lub zwiększenia odstępów między nimi, umieszczenia kontroltek na środku formularza, a nawet zmiany atrybutów wyglądu kontroltek tak, aby miały taką samą wielkość.
- **Układ tabelaryczny** - to tabela nałożona na formularz. Rozmiar komórek tabeli (a jednocześnie odstępów między kropkami) można określić za pomocą wartości ***GridSize***, którą także można ustawić w oknie dialogowym *Options pozycja Windows Forms Designer* właściwość *LayoutMode*. Mniejszy rozmiar komórek oznacza węższe odstępów między kropkami, a co za tym idzie możliwość bardziej precyzyjnego rozmieszczania kontroltek.
- **Przyciąganie do linii** –na powierzchni formularza nie jest widoczna tabelka. W zamian okno projektowe wyświetla graficzne wskazówki w czasie poruszania kontrolki po formularzu. Właściwość *LayoutMode* ustawiamy na wartość ***SnapLines*** i ponownie otwieramy okno z formularza.

## Tworzenie kodu i zarządzanie nim

**Edytor kodu** to narzędzie środowiska Visual Studio przeznaczone do tworzenia plików tekstowych z kodem źródłowym. Edytor można uruchomić używając okna *Solution Explorer* - należy wybrać plik z kodem, plik tekstowy lub plik dowolnego innego typu, a następnie kliknąć plik prawym przyciskiem myszy i wybrać opcję *View Code*.

Okna edytora tekstu widoczne są jako zakładki w centralnym obszarze środowiska IDE. Jeśli otwartych jest kilka edytorów kodu, można uzyskać dostęp do nich poprzez te zakładki.

Okno edytora kodu **składa się** z trzech głównych obszarów:

- **panelu kodu** -to miejsce, gdzie wyświetlany jest dokument (kod programu)
- **marginesu ze wskazówkami** -to wąski, szary margines widoczny po lewej stronie edytora. Służy on do wyróżniania wierszy kodu z punktem przerwania lub zakładką
- **marginesu wyboru** -to wąski obszar pomiędzy marginesem ze wskazówkami a służącym do edycji obszarem panelu kodu. Ten margines udostępnia zaznaczanie wierszy, wyświetlanie numerów wierszy.

U góry edytora dostępne są dwie **listy rozwijane**, które umożliwiają szybkie poruszanie się po kodzie źródłowym poprzez wybór typu (lewa lista), a następnie wybór jego specyficznej składowej (właściwość, pole, funkcja i tak dalej; prawa lista). W Visual Basic te listy rozwijane nazywane są *Class Name* i *metod Name*. Wybór składowej powoduje bezpośrednie umieszczenie kursora w kodzie określonego typu.

### Przeszukiwanie dokumentów

Okno edytora tekstu udostępnia rozbudowane możliwości wyszukiwania i zamiany kodu.

Obsługiwane są trzy główne metody wyszukiwania:

- **Quick Find** -wyszukiwanie fragmentów tekstu w bieżącym dokumencie lub zbiorze otwartych dokumentów
- **Serach In Files** -wyszukuje tekst w plikach znajdujących się w dowolnym miejscu struktury katalogów projektu
- **Find Symbol** -przeznaczony głównie do wyszukiwania obiektów lub składowych o określonej nazwie.

### Doprecyzowanie wyszukiwania

Pod kontrolką *Look In* okna *Find and Replce* znajduje się grupa pól wyboru umożliwiających doprecyzowanie wyszukiwania. Opis opcji:

- **Match case** -powoduje wyszukiwanie z uwzględnieniem wielkości znaków tekstu wpisanego w liście rozwijanej *Find What*.
- **Match whole word** -powoduje wyszukiwanie jedynie całych łańcuchów znaków wpisanych w liście rozwijanej *Find What*.
- **Serach up** -powoduje przeszukiwanie dokumentów od końca do początku (w przeciwieństwie do domyślnego przeszukiwania od początku).
- **Serach hidden test** -(opcja domyślnie włączona) powoduje przeszukiwanie ukrytych obszarów kodu.
- **Use Wildcards** i **Use Regular expressions** -zmieniają sposób dopasowywania przez silnik wyszukiwający łańcuchów znaków wpisanych w polu *Find What*. Jeśli zaznaczone jest pole wyboru *Use*, dostępny jest mały przycisk widoczny po prawej stronie listy rozwijanej *Find What*. Ten przycisk pozwala wyświetlić narzędzie do tworzenia wyrażeń. Zawartość tego menu zmienia się w zależności od tego, czy programista wybrał symbole wieloznaczne (opcja *Wildcards*), czy wyrażenia regularne (opcja *Regular expressions*).

### Mechanizm IntelliSense

*IntelliSense* -to nazwa nadana grupie różnych narzędzi pomagających w pisaniu kodu i działających w oknie edytora tekstu. Na działanie IntelliSense składa się wiele odrębnych funkcji, które bezkonfliktowo współpracują ze sobą w czasie pisania kodu. Można je uruchomić także bezpośrednio z menu *Edit/IntelliSense*, menu kontekstowego lub przy użyciu kombinacji Ctrl+spacja.

- **CompleteWord** -czyli uzupełnianie słów to podstawowa funkcja IntelliSense zapewniająca oszczędność czasu. Kiedy programista wpisze wystarczającą liczbę znaków, aby mechanizm mógł rozpoznać wpisywane słowo, proponowane jest uzupełnienie wpisywanego tekstu. Wciśnięcie klawisza *Tab* lub *spacji* powoduje automatyczne wstawienie słowa w edytorze.
- **Quickinfo** -czyli okno z informacjami podręcznymi wyświetla kompletną deklarację danego fragmentu kodu oraz informacje pomocnicze z nim związane. Aby wyświetlić to okno, należy umieścić wskaźnik myszy na danym identyfikatorze. Pojawi się wtedy wyskakujące okno z informacjami dostępnymi na jego temat.
- **List Members** -okno działa identycznie jak uzupełnianie słów. W przypadku dowolnego typu lub przestrzeni nazw wyświetla ono przewijaną listę zawierając wszystkie zmienne i funkcje składowe danego typu.

- **Parametr Info** -okno z informacjami o parametrach, służy do interaktywnego wyświetlania wskazówek dotyczących parametrów potrzebnych w wywołaniu danej funkcji. Ta właściwość jest szczególnie przydatna do pisania wywołań funkcji z długimi listami parametrów lub mających wiele przeciążonych wersji. Okno z informacjami o parametrach jest wyświetlane zawsze, kiedy otworzymy nawias po nazwie funkcji.
- **Snippets** -(fragmenty kodu) to wbudowane wiersze kodu, które można wybrać i wstawić w edytorze kodu. Dostęp do każdego fragmentu kodu można uzyskać poprzez jego nazwę (alias). Fragmenty kodu służą do automatyzacji operacji, które polegają na wpisywaniu powtarzającego się kodu. Można utworzyć własne fragmenty kodu lub używać domyślnej biblioteki standardowych elementów kodu udostępnianej przez Visual Studio. Fragment kodu można wstawić klikając prawym przyciskiem myszy miejsce przeznaczone na ten kod, a następnie wybierając opcję *Insert Snippet* z menu podręcznego. Fragmenty kodu są pogrupowane.

## Okno Task List

Lista w oknie *Task List* to zintegrowana lista zadań do wykonania. Obejmuje ona wszystkie elementy, które z jakiegoś powodu wymagają uwagi lub kontroli. Aby wyświetlić to okno, należy otworzyć menu *View* i wybrać opcję *Task List*. Możemy wyróżnić trzy kategorie zadań:

- **Zadania związane z komentarzami** -są tworzone w kodzie źródłowym. Umieszczenie komentarza ze specjalnym literałem (znacznikiem) powoduje dodanie go przez IDE do listy zadań związanych z komentarzami. Dostępne są trzy takie znaczniki: HACK, TODO i UNDONE. Aby wyświetlić je w oknie zadań, należy zaznaczyć opcję *Comments* na liście rozwijanej w górnej części listy zadań.
- **Zadania związane ze skrótami** -to odnośniki do wierszy kodu. Te odnośniki można dodać za pomocą menu *Bookmarks*. W tym celu należy umieścić kursor w odpowiednim wierszu i wybrać opcję *Edit/Bookmarks/Add Task List Shortcut*. Zawartość tego wiersza kodu będzie widoczna jako opis zadania.
- **Zadania użytkownika** -są wpisywane bezpośrednio w oknie zadań. Obok listy rozwijanej z kategoriami zadań okna *Task List* widoczny jest przycisk *Create User Task*. Ten przycisk pozwala dodać nowy wpis do listy. Następnie można bezpośrednio wpisać tekst w kolumnie opisu w celu określenia tytułu zadania. Zadania użytkownika nie są związane z wierszem kodu aplikacji.

Koniec cz. 2

## Dodatkowe narzędzia środowiska IDE

### Okno Class View

Okno *Class View* -wyświetla przestrzeń nazw, typy, interfejsy i wyliczenia znajdujące się w projekcie. Okno składa się z czterech głównych komponentów graficznych:

- *Paska narzędzi* –udostępnia polecenia do zarządzania obiektami
- *Paska wyszukiwania* -umożliwiająca przeszukiwanie obiektów widocznych w panelu obiektów.
- *Panelu obiektów* –wyświetla drzewo obiektów zawartych w projekcie.
- *Panelu składowych* -wyświetla wszystkie właściwości, zdarzenia, stałe, zmienne, wyliczenia zdefiniowane w zaznaczonych obiektach.

### Okno Server Explorer

Okno *Server Explorer* pełni dwie funkcje:

- udostępnia różne usługi i zasoby systemowe znajdujące się na lokalnych oraz zdalnych komputerach (*Servers*),

- zapewnia dostęp do obiektów reprezentujących połączenie z danymi (*Data Connections*).

*W wersjach Express i Standard środowiska Visual Studio w oknie Server Explorer nie są widoczne serwery, a jedynie połączenia z danymi.*

### **Okno Object Browser**

Okno *Object Browser* przypomina funkcjonalnością, wyglądem i stylem okno *Class View*. Udostępnia hierarchiczny widok projektów, podzespołów, przestrzeni nazw, typów, wyliczeń i interfejsów.

### **Okno Dokument Outline**

Okno *Dokument Outline* udostępnia hierarchiczny widok elementów znajdujących się na formularzu Windows lub sieciowym. Narzędzie służy do zmiany obiektów nadrzędnych elementów formularza lub zmieniania ustawień kolejności kontrolki w oknie nadrzędnym. Pomaga zrozumieć dokładną strukturę logiczną formularza, która może być bardzo skomplikowana z graficznego punktu widzenia.

Koniec cz. 3