

## Microsoft Visual Basic kontra Borland Delphi **W wojnie o szybkość**

Jacek Szarapa

**Visual Basic Microsoftu i Delphi Borlanda to na pozór różne narzędzia wykorzystujące inne języki programowania. Oba powstały jednak w tym samym celu, by wspomagać pracę programistów.**

Visual Basic jest pierwszym wizualnym narzędziem, które powstało w celu ułatwienia pracy programistom tworzącym oprogramowanie działające w systemie Windows. Narzędzie to było na tyle proste w użyciu, że przez pewien czas stanowiło podstawowe wyposażenie początkującego programisty. Wkrótce jednak okazało się, że Visual Basic nie jest pakietem doskonałym. Pojawiła się pierwsza wersja Delphi wprowadzona na rynek przez firmę Borland. Narzędzie to dysponowało już znacznie większymi możliwościami. A przede wszystkim było pierwszym narzędziem wizualnym.

W niedługim czasie ukazały się także różne wersje języka C, umożliwiające programowanie w systemie Windows. Te jednak nie zdobyły tak dużej popularności jak *Delphi* i *Visual Basic*, głównie ze względu na swoją złożoność. Mimo to język C pozostał zdecydowanie najlepszym językiem programowania; wysoko cenionym przy bardziej zaawansowanych projektach. *Delphi* zawdzięcza swoją popularność podobieństwu z zasadami programowania w języku Pascal - który to od dłuższego czasu dominuje w większości szkół o profilu informatycznym. *Visual Basic*, natomiast bazuje na języku *Basic* powszechnie stosowanym w komputerach 8-bitowych, nie tak dawno wypartych przez komputery klasy PC.

### **Obrazowanie tylko podstawowe**

Wydaje się, że mimo, iż *Visual Basic* jest stosunkowo szybkim narzędziem, to w porównaniu z *Delphi* pozostaje kilka kroków w tyle. Przede wszystkim dlatego, że programowanie przy użyciu tego narzędzia nie jest niestety zorientowane obiektowo w pełnym tego słowa znaczeniu. Obsługa obiektów oparta jest na modelu składnikowym COM. Zapewnia on implementacje obiektów zgodnych z tym modelem, polimorfizm oraz hermetyzację danych. Dziedziczenie ogranicza się tylko do implementacji interfejsów klasy bazowej.

*Visual Basic* jest kompilatorem, który generuje kod wynikowy do tak zwanego „pseudo - kodu”. Oznacza to, że taki kod jest zamieniany na postać zrozumiałą dla systemu, dopiero w momencie uruchomienia programu. Nie trzeba chyba nikogo przekonywać, że taki sposób działania wprowadza dodatkowe obciążenie procesora i w efekcie spowolnienie działania programu. Metoda ta ma niekiedy także swoje zalety - na przykład mniejszy kod wynikowy

oraz szybsze wykonywanie złożonych operacji matematycznych. *Visual Basic 5.0* w wersji *Professional* dysponuje już możliwością pełnej kompilacji do kodu zrozumiałego dla systemu, co pozwala przyspieszyć działanie programu prawie 20-krotnie (według danych Microsoft'u).

Tworzenie aplikacji w pakiecie Microsoft'u ogranicza się do określenia podstawowych parametrów elementów umieszczanych na formie oraz przypisania prostych fragmentów kodu do zdarzeń obsługiwanych przez te elementy. Kod pisze się dość szybko przede wszystkim dlatego, że *Visual Basic* cały czas śledzi poczynania programisty, wyświetlając opisy i dopuszczalne parametry edytowanych właśnie elementów i komend. Narzędzie to automatycznie uzupełnia wpisywane nazwy parametrów, tak więc w większości przypadków wystarczy wpisać tylko kilka pierwszych znaków nazwy a reszta zostanie dodana przez VB. Ważniejsze elementy składni samego kodu są kolorowane, co pozwala na szybszą orientację w kodzie.

Sam język jest dość wygodny, mimo iż nie posiada oszałamiających możliwości. Wygodę obsługi programistom Microsoftu udało się uzyskać dzięki umożliwieniu wykonywania większości operacji przy użyciu wewnątrz systemowych bibliotek za pośrednictwem programowego interfejsu API, do którego dostęp jest bardzo prosty. Takie uproszczenie języka spowodowało jednak spadek jego funkcjonalności w bardziej zaawansowanych projektach. W przeciwieństwie do Delphi ograniczone są tu do minimum operacje wykonywane na bitach oraz działania ingerujące bezpośrednio w zawartość pamięci.

*Visual Basic* dysponuje bogatym zestawem narzędzi obsługi baz danych, zarówno z poziomu samego kompilatora jak i programu. Microsoft zawarł również w pakiecie 32-bitowy mechanizm obsługi baz danych „*Jet 3.0*”. Jednak bezpośrednia komunikacja z samymi bazami danych odbywa się za pomocą DAO (*Data Access Object*). Ten sam mechanizm został wykorzystany także w aplikacji *Access95*. Dzięki temu, że Microsoft dąży do tego by z *Visual Basic*'a utworzyć narzędzie korporacyjne, można za jego pomocą tworzyć aplikacje w środowisku klient/serwer. Dostępne są na przykład elementy RDO (*Remote Data Objects*) umożliwiające dostęp do baz danych zgodnych z ODBC.

Duży nacisk położono także na komunikację z sieciami lokalnymi, a także rozległymi. Użytkownik dostaje tu do dyspozycji całą masę elementów umożliwiających wykonywanie prostych, a także złożonych operacji w obrębie sieci. Słabym punktem programu napisanego w *Visual Basic* jest problem z jego dystrybucją. Program taki do poprawnego działania wymaga dołączenia dużej ilości zewnętrznych bibliotek DLL, co powoduje zwiększenie rozmiaru programu o co najmniej 3,5 MB, przy dość rozbudowanym projekcie. Microsoft uznał, że większość funkcji można by umieścić w bibliotekach, dzięki czemu różne programy mogły by korzystać z tych samych bibliotek. Takie rozwiązanie zmniejszałoby ilość miejsca wymaganego na twardym dysku przez

kolejne instalowane aplikacje. Paradoksalnie właśnie Borland storpedował pomysł Microsoftu, gdyż oprogramowanie pisane w Delphi nie wykorzystuje zewnętrznych bibliotek. Rosnąca popularność Delphi oznacza zatem kłopoty w dystrybucji aplikacji pisanych przy pomocy Visual Basica. Efektem tego jest sytuacja, kiedy na rynku aplikacje tworzone za pomocą Visual Basica stanowią mniejszość, co nie oznacza bynajmniej nagłego przyrostu miejsca na dyskach użytkowników. Biblioteki bowiem i tak, i tak będą musiały pozostać. Niekiedy jednak ich liczbę można znacznie ograniczyć. W pakiecie Visual Basic zawarto program pomagający przygotowanie odpowiedniego zestawu plików. Jest w nim także dostępny, wraz z kodem źródłowym, prosty instalator aplikacji.

Visual Basic dysponuje bardzo wygodnym interfejsem, bardzo dobrymi narzędziami do zarządzania projektami oraz samym kodem. Dostępna jest także obszerna biblioteka kreatorów - które w dużym stopniu przyspieszają proces budowania aplikacji. Narzędzie to także w pełni wykorzystuje technologię ActiveX, która jest poza bibliotekami DLL, jedynym sposobem na dodanie nowych elementów wizualnych. Pakiet zawiera również kreator umożliwiający samodzielne tworzenie elementów ActiveX.

### **Wszystkie strony kompilacji**

Delphi można nazwać kompilatorem przez duże K. Obsługiwany przez niego język jest w przeciwieństwie do Visual Basica zorientowany obiektowo. Delphi poszczycić się zarówno kilkakrotnie większą szybkością działania jak i małym rozmiarem generowanego kodu wynikowego.

Nie wszystko jednak w tym narzędziu jest doskonałe. Główną wadą programu jest dość niewygodny interfejs, który ani trochę nie ułatwia poruszania się pod elementach kodu. Użytkownicy nie zawsze wiedzą o tym, że główny interfejs można modyfikować i dodawać do niego pomocnicze funkcje poprzez wykorzystanie dodatkowych nakładek. W większości jednak przypadków, za rozszerzenie interfejsu trzeba dodatkowo zapłacić.

Proces tworzenia aplikacji ogranicza się, tak jak w Visual Basic'u do umieszczenia odpowiednich elementów na formie, ustawienia ich podstawowych parametrów oraz zdefiniowania reakcji programu na działania użytkownika. Wprowadzany kod nie jest kontrolowany na bieżąco. Pełna kontrola poprawności składni następuje dopiero na etapie kompilacji. Ewentualne inne błędy w aplikacji nie powodują załamania pracy napisanego programu. Taką odporność aplikacje zawdzięczają bardzo dobremu mechanizmowi obsługi błędów. Pisanie samego kodu podobnie jak w przypadku Visual Basic'a jest dość proste ze względu na pojawiające się opisy wprowadzanych elementów oraz ich dopuszczalnych parametrów. Orientowanie się w kodzie ułatwia jego kolorowanie. Podczas tworzenia kodu programu można posłużyć się zestawem kreatorów, wydajnie ułatwiających proces tworzenia pewnych elementów kodu. Używany język jest bardzo

rozbudowany, a jego funkcjonalność jest dodatkowo zwiększona wbudowaną obsługą API. Delphi zawiera również bogaty zestaw narzędzi obsługi baz danych. Komunikacja z nimi odbywa się za pomocą sterowników ODBC. Wersja klient/serwer umożliwia także budowanie aplikacji działających na rozległych bazach danych. Wersja ta posiada także zestaw elementów umożliwiających komunikację z sieciami lokalnymi oraz rozległymi. Aplikacje stworzone za pomocą tego narzędzia mogą być szybko rozprowadzane dzięki pakietowi InstallShield dodanemu do Delphi.

Kompilator Borlanda może być łatwo rozszerzany o nowe elementy wizualne. Dotyczy to zarówno elementów ActiveX jaki i wewnętrznych obiektów tzw. komponentów. Przewaga drugiego rozwiązania jest szczególnie widoczna w momencie przygotowywania pakietu dystrybucyjnego utworzonej aplikacji. Komponenty są wkompilowywane w sam program, natomiast elementy ActiveX muszą zostać dołączone do pakietu w postaci dodatkowych plików. Delphi umożliwia tworzenie własnych elementów ActiveX. Sam proces jest bardzo prosty, gdyż większość czynności wykonuje za programistę kompilator, tworząc zarys kodu za pomocą kreatora i uaktualniając go na bieżąco w czasie tworzenia. Delphi obsługuje także technologię OLE, oraz umożliwia korzystanie z modelu COM.

### **Co dalej, kompilatorze?**

Zarówno Visual Basic jak i Delphi są ciągle rozbudowywane o nowe możliwości i mogą poszczycić się implementacją najnowszych technologii. W ciągu najbliższych kilku miesięcy producenci powyższych narzędzi planują pokazać światu najnowsze wersje swoich produktów. W przygotowywanej właśnie najnowszej wersji pakietu firmy Visual Basic, duży nacisk położono na dopracowanie i rozszerzenie elementów obsługi baz danych. Testowa wersja 6.0 posiada zaimplementowane technologie ActiveX Data oraz OLE, co umożliwi w łatwy sposób łączenie źródeł relacyjnych i nierelacyjnych. Wersja ta posiada także narzędzia, które pozwalają na tworzenie i modyfikowanie baz danych takich jak Oracle i AS/400. Uproszczono w niej również proces łączenia baz danych z elementami wizualnymi.

Za pomocą Visual Basic 6.0 będzie można przygotować komponenty działające po stronie serwera. Microsoft wprowadził do tej wersji także nowy element zwany „WebClasses”. Są to obiekty typu COM działające ze strony serwera, które mogą być tworzone za pomocą Visual Basic'a i używane w środowiskach Web'u. Ich działanie ogranicza się jednak do platformy i serwerów Web'u, dodatkowo obsługujących Active Server Pages (ASP), działających w środowisku Windows. Limit ten stawia jednak pod znakiem zapytania użyteczność tej innowacji przy wykorzystaniu różnorodnych serwerów działających w sieci. W przygotowywanej obecnie wersji dostępne jest także nowe narzędzie, umożliwiające tworzenie stron DHTML, które mogą być

wykorzystane do tworzenia nakładek przeglądarki Internet Explorer .

Nowe wydanie Delphi - jak przynajmniej twierdzi Borland - umożliwi bezproblemowe tworzenie wielowarstwowych aplikacji opartych na Windows, Javie i ActiveForm. W dużym uproszczeniu ActiveForm pozwala na umieszczanie i obsługiwanie aplikacji, utworzonych za pomocą Delphi, przez internetowe przeglądarki. Delphi obsługuje między innymi technologie COM, DCOM (Distributed Component Object Model) , CORBA, MTS (Microsoft Transaction Server) HTTP. Rozszerzono obsługę baz danych. Delphi umożliwia dostęp do baz Oracle, DB2, serwerów SQL i innych. Zaimplementowano również obsługę Microsoft OLE DB, dzięki czemu możliwa jest integracja relacyjnych i nierelacyjnych danych w aplikacjach. Dołączono także obsługę Oracle8 z nowymi obiektami relacyjnymi. Rozbudowany w nowej, przygotowywanej wersji interfejs pozwala na lepsze i szybsze zarządzanie kodem źródłowym tworzonej aplikacji.

### **Nie do końca przegrani**

Visual Basic i Delphi to tak naprawdę dwie różne strategie podejścia do tworzenia oprogramowania. Różnice są widoczne już na poziomie języka. Delphi, jako pakiet oparty na Pascalu dysponuje dopracowanym językiem programowania, spójnym, bez zbędnej redundancji w składni. Twórcy Visual Basic postawili sobie za zadanie stworzenie pakietu który szybko może być opanowany przez każdego. Większość zapisywanych w nim instrukcji odpowiada sposobom wyrażania zdań w języku naturalnym. Z drugiej strony, Pascal jest językiem znajdującym wyznawców wśród absolwentów szkół informatycznych. Oni właśnie Visual Basicu muszą uczyć się od początku.

Visual Basic korzysta z technologii COM, co powoduje że składniki mogą być pisane w dowolnym języku, ale przez to ich łączenie z narzędziem Microsoftu może niektórym wydawać się dziwne. Komponenty Delphi są integralną częścią pakietu - dostosowane do metodologii obiektowej pochodzącej jeszcze z Pascala 7.0 znacznie lepiej wpasowują się w strukturę programu. Równocześnie ze względu na fakt, że Delphi dysponuje dużą rzeszą wielbicieli, można łatwo znaleźć gotowy, a co więcej zwykle darmowy komponent realizujący daną funkcję. Natomiast składniki ActiveX są zwykle znacznie bardziej rozbudowane. Mało jest składników dostępnych za darmo.

## W wojnie o szybkość

	Borland Delphi 3.0	MS Visual Basic 5.0
<b>Ogólne</b>		
Typ narzędzia	Kompilator	do wersji 5.0 interpreter, od wersji 5.0 kompilator
Szybkość działania	Bardzo duża	Duża
Dostępna pomoc z poziomu programu	Dobra	Bardzo Dobra
Stabilność narzędzia	Dobra	Dobra
Cena pakietu w wersji podstawowej	Przystępna	Wysoka
<b>Ogólna ocena</b>	<b>Bardzo dobry</b>	<b>Słaby</b>
<b>Interfejs programu</b>		
Wygoda pracy	Dobra	Bardzo Dobra
Możliwość dostosowania do własnych potrzeb	Bardzo Dobra	Bardzo Dobra
Zarządzanie projektem	Dobre	Bardzo Dobre
Zarządzanie kodem	Słabe	Bardzo Dobre
<b>Ogólna ocena</b>	<b>Dobry</b>	<b>Bardzo dobry</b>
<b>Kod źródłowy</b>		
Możliwość przenoszenia kodu pomiędzy różnymi wersjami pakietu	Dobra	Bardzo Dobra
Podpowiedzi w trakcie pisania kodu (CodeInSight)	Tak	Tak
Kolorowanie składni	Tak	Tak
Szybkość tworzenia kodu	Duża	Duża
Obiektowość	Pełna	Zgodna z modelem COM
<b>Ogólna ocena</b>	<b>Bardzo dobry</b>	<b>Dobry</b>
<b>Generowany kod wynikowy</b>		
Obsługiwane systemy	Windows95/98/NT	Windows95/98/NT
Szybkość działania	Bardzo Duża	Mała
Rozmiar kodu wraz z wymaganymi elementami zewnętrznymi	Mały	Bardzo Duży
Stabilność samodzielnie działającej aplikacji	Duża	Mała

Reakcja na błędy w pisanym programie	Nie załamują działanie programu	Załamują działanie programu.
<b>Ogólna ocena</b>	<b>Bardzo dobry</b>	<b>Słaby</b>
<b>Dystrybucja utworzonych aplikacji</b>		
Pomoc w przygotowaniu pakietu dystrybucyjnego aplikacji	Dobra	Dobra
<b>Ogólna ocena</b>	<b>Dobry</b>	<b>Dobry</b>
<b>Ogólna ocena pakietu</b>	<b>Bardzo dobry</b>	<b>Dobry</b>

**Jacek Szarapa** jest autorem shareware'owej aplikacji Tiger 98, edytora skryptu html.