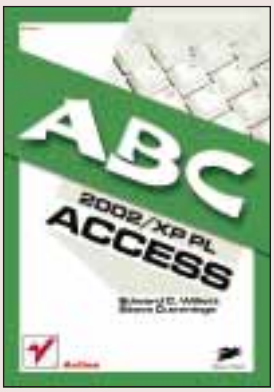
ABC Accessa 2002/XP PL



Autorzy: Edward C. Willett, Steve Cummings
Tłumaczenie: Marek Korbecki
Opracowanie wstępu i rozdziałów 1, 2: Daniel Chmielewski
ISBN: 83-7197-827-8
Tytuł oryginału: Office XP Bible
Format: B5, stron: 142

MS Access, program wchodzący w skład najnowszej wersji popularnego pakietu biurowego Office XP, to jedna z najpopularniejszych baz danych pracujących w systemach operacyjnych Microsoftu.

Książka ta, prezentująca najnowszą wersję popularnej bazy danych, jest przeznaczona zarówno dla początkujących, jak i bardziej zaawansowanych użytkowników programu. Dzięki niej można się dowiedzieć, jak samodzielnie przygotować bazę danych i jak z nią pracować. Dzięki książce samodzielnie utworzysz bazę danych do celów prywatnych lub służbowych. Może to być np. baza płyt kompaktowych, klientów czy kooperantów.

Książka składa się z siedmiu rozdziałów. W pierwszym dowiesz się, jak uruchamiać Accessa, poznasz podstawowe pojęcia związane z bazami danych oraz zapoznasz się z głównym oknem programu. Drugi rozdział omawia tworzenie podstawowych elementów bazy danych. Dowiesz się z niego, jak tworzyć własne tabele, formularze, raporty i kwerendy. W rozdziale tym opisano również sposoby „przekształcenia" arkuszy programu Excel na bazę danych Access. Rozdział trzeci przedstawia metody dostosowania Accessa do własnych potrzeb oraz zwiększenia wydajności bazy.

Czwarty omawia szczegółowo zagadnienia związane z tworzeniem tabel oraz wykorzystaniem kwerend, raportów i filtrów. Właśnie w nim znajdziesz szczegółowe informacje na temat tego, jak przeszukiwać bazę danych, aby znaleźć potrzebne informacje. W rozdziałach piątym i szóstym zajmiemy się językiem Visual Basic for Applications. Makra utworzone w tym języku mogą sprawić, że nasze bazy danych będą pracować w naprawdę profesjonalny sposób. Rozdział siódmy przedstawia zaawansowane zagadnienia związane z pracą z formularzami.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 5.

Sekrety projektu aplikacji bazy danych

W rozdziale:

- Planowanie własnych aplikacji Access
- Podejmowanie decyzji, kiedy korzystać z makr, VBA lub samego Accessa
- Tworzenie aplikacji użytkownika
- Dostosowywanie istniejących programów
- Sposoby, jak uzyskać więcej dzięki skromniejszemu kodowi

Skoro opanowaliśmy już podstawowe narzędzia konstrukcji baz danych (omówione w rozdziale 4.), czas przejść do bardziej zaawansowanych sposobów wykorzystania użyteczności Accessa. Prawdę mówiąc, Access to coś znacznie więcej niż program do obsługi baz danych — oferuje bowiem pełne środowisko programistyczne, umożliwiając tworzenie własnych aplikacji.

Ściśle rzecz biorąc, Access udostępnia dwa różne środowiska programowania aplikacji. W jednym możemy tworzyć w pełni działające aplikacje, korzystając z makr automatyzujących wydawanie poleceń dostępnych na paskach narzędziowych i w menu. Alternatywne środowisko umożliwia rezygnację z pracy z interfejsem Accessa i wykorzystanie VBA, oferującego pełną moc programowania. W niniejszym rozdziale omówimy plusy i minusy obu rozwiązań.

Witryna internetowa korporacji Microsoft udostępnia szereg artykułów szczegółowo omawiających zagadnienia programistyczne dotyczące wszystkich programów pakietu Office, w tym i Accessa. Archiwum to znajdziemy pod adresem *http://msdn.microsoft.com/office/*.

Jak rozumieć aplikacje Accessa?

Od strony koncepcyjnej *aplikacja* Accessa jest kompletnym tworem programistycznym, wykonującym szereg zadań związanych z zarządzaniem bazami danych, udostępniającym użytkownikowi wbudowane funkcje za pośrednictwem pasków narzędziowych i formularzy dostosowywanych do jego wymagań. Oczywiście, nie ma potrzeby rozważać, kiedy baza danych staje się aplikacją. Wystarczy powiedzieć, że aplikacje to rozwiązanie o dalece ambitniejszych celach i dostosowywanych do określonych wymagań w znacznie wyższym stopniu niż bazy danych, zawierające informacje, którymi zarządzamy za pośrednictwem standardowego interfejsu użytkownika oraz poleceń.

Kiedy rozpoczynamy tworzenie nowej aplikacji, w miejscu domyślnego okna bazy danych Accessa pojawia się formularz specjalnego rozdzielacza (przykład widoczny jest na rysunku 5.1). Pełniąc rolę centralnego panelu sterowania aplikacją, rozdzielacz udostępnia zestaw przycisków i innych kontrolek aktywujących polecenia lub odgałęzienia do innych formularzy, wyświetlających dane w celu podglądu, edycji lub kwerendowania. Jednocześnie standardowe elementy interfejsu Accessa zastępują paski narzędziowe i menu aplikacji, które mają również zdolność samoczynnego ukrywania się i wyświetlania adekwatnie do działań użytkownika.

Rysunek 5.1. *Główny panel przełączający aplikacji użytkownika jest elementem, który, rozpoczynając pracę nad aplikacją, widzimy jako pierwszy*

Tworząc aplikację, koniecznie powinniśmy pamiętać o umieszczaniu komentarzy dokładnie wyjaśniających kolejne zadania wykonywane przez makro czy skrypt VBA. Dzięki temu będziemy mieli pewność, że gdy przyjdzie nam lub komukolwiek innemu wprowadzić modyfikacje kodu poprawiające albo uaktualniające aplikację, nie będziemy mieli z tym problemu.

Choć aplikacje możemy budować na własny użytek, to jednak znacznie częściej tworzymy je w celu dystrybucji wśród innych użytkowników. Zapisując dane wraz z kontrolkami określonych zadań, umożliwiamy innym wykonanie pracy nawet przy znikomej znajomości Accessa.

Planowanie i implementowanie aplikacji

Proces planowania (opisany w rozdziale 4.) nabiera jeszcze większego znaczenia i wymaga baczniejszej uwagi, gdy dotyczy budowy kompletnej aplikacji. Projekt stanowiący fundament bazy danych — czyli tabele, pola, formularze, kwerendy i raporty — w dużym stopniu decyduje o końcowym sukcesie aplikacji. Należy się ponadto zastanowić, jakich poleceń zautomatyzowanych może oczekiwać użytkownik oraz jak je zaprezentować na rozdzielaczu, pasku menu czy narzędziowym.

Bardzo ważne podczas przygotowywania projektu jest rozważenie aplikacji jako serii zdarzeń, które mogą zachodzić w dowolnej kolejności. Przykładem zdarzenia może być kliknięcie przycisku myszą, otwarcie lub zamknięcie formularza. W aplikacjach Accessa, podobnie jak w programach VBA, takie zdarzenia wyzwalają określone funkcje. Chcąc umożliwić użytkownikowi wykonanie kwerendy, możemy tak skonstruować program, by kwerenda była uruchamiana za pomocą naciśnięcia określonego przycisku. Do wykonania kwerendy wywołanej kliknięciem przycisku aplikacja musi użyć makra lub kodu VBA.

Wybór właściwego narzędzia: makra Accessa a VBA

Pamiętajmy, że makra Accessa nie są tym samym, co oparte na VBA makra w programach Word, Excel i PowerPoint. Ponieważ do budowy aplikacji możemy użyć obu rodzajów narzędzi, ważnym krokiem jest podjecie decyzji o wyborze jednego z nich, lepiej spełniającego nasze potrzeby.

Różnice między makrem i VBA

Ujmując rzecz w skrócie, różnice między makrami i procedurami VBA przedstawiają się następująco.

- Makra Accessa automatyzują serie akcji, w większości korespondujących z poleceniami, które można wydać za pośrednictwem interfejsu użytkownika (wykorzystując menu lub przyciski na pasku narzędziowym). W ten sposób można otwierać tabele, uruchamiać kwerendy, wyświetlać formularze, drukować raporty i tak dalej.

Podstawy tworzenia makr w Accessie omówione zostały w rozdziale 3.

- Procedury VBA wykorzystywane w Accessie funkcjonują podobnie jak w innych aplikacjach pakietu Office. Jedyną charakterystyczną ich cechą jest to, że mogą czynić bezpośredni użytek z programowania obiektów należących do Accessa. VBA nie jest związany z interfejsem i dlatego jego wygląd i sposób współdziałania z bazami danych może znacznie odbiegać, zapewniając przy tym dalece większą moc, od sposobu, w jaki pracuje interfejs użytkownika.

Niezależnie od tych różnic granica między makrami i VBA jest niewyraźna. Makro może uruchamiać procedury VBA, a kod VBA może wykorzystywać metody obiektu `DoCmd` w celu wykonywania makr. Należy także zwrócić uwagę na zdolność Accessa do konwertowania makr na procedury VBA. Operację konwersji rozpoczyna wyselekcjonowanie obiektu makra w oknie bazy danych (jednak bez otwierania obiektu). Jeśli natomiast makro jest połączone z formularzem bądź raportem, należy wówczas otworzyć ów formularz lub raport w widoku projektu, a następnie wydać polecenie *Konwertuj makra* na język Visual Basic z menu *Narzędzia/Makro*.

Kiedy użyć makr, a kiedy VBA?

W jaki sposób wybrać między makrami i VBA? Ogólnie rzecz biorąc, makra są łatwiejsze w użyciu, ale VBA zapewnia o wiele większą użyteczność działania.

Oto rada: nie mając pewności, czy makra będą wystarczającym rozwiązaniem dla tworzonej aplikacji, należy zrezygnować z ich użycia. Nauka VBA jest niełatwa i ukończenie pierwszej aplikacji może potrwać dość długo, ale umiejętności, jakich nabędziemy, budując proste aplikacje, przydadzą się, kiedy przystąpimy do tworzenia bardziej skomplikowanych. Jeśli natomiast odkryjemy, że nasza „prosta" aplikacja wymaga więcej funkcji, niż z początku sądziliśmy, umiejętność kodowania VBA uchroni nas od konieczności pisania całej aplikacji od początku.

Prawdę mówiąc, Access nie wymusza na nas wyboru między makrami i VBA — w pojedynczej aplikacji można użyć obu rozwiązań jednocześnie. Poznając tajniki obydwu narzędzi, będziemy mogli wybrać z nich to, co najlepsze.

Korzyści wynikające ze stosowania makr

Czas na szczegóły. Makra są łatwiejsze w użyciu niż VBA, głównie z dwóch powodów. Po pierwsze, umiejętność posługiwania się interfejsem Accessa sprawia, że budowa aplikacji przy użyciu makr to jedynie rozszerzenie posiadanej wiedzy. Większość akcji wykonywanych przez makra to odpowiedniki regularnie wydawanych poleceń. Po drugie, Access pomaga nam w trakcie konstruowania makr, wyświetlając w oknie *Makro* pola argumentów niezbędnych do przeprowadzenia poszczególnych akcji. W przeciwieństwie do VBA, makra nie wymagają od nas nauki specjalizowanej składni.

Makra oprócz łatwości użycia oferują inne korzyści.

- Użytkownik może definiować skróty klawiaturowe dla całej aplikacji — w odróżnieniu od pojedynczych formularzy — wyłącznie za pomocą makr, omijając VBA.

O skrótach klawiaturowych mówi rozdział 3.

- Makra działają szybciej niż moduły VBA i w mniejszym stopniu obciążają pamięć. Tę zaletę w przypadku formularza czy raportu zauważymy jedynie wówczas, jeśli do obiektu nie został przypisany kod VBA (patrz punkt „Maksimum wydajności przy minimalnej ilości kodu VBA" w dalszej części rozdziału).

Zalety VBA

Kod VBA ma wiele zalet w porównaniu z makrami. Omawiamy je poniżej.

- **Łatwość konserwacji**. W miarę wzrostu stopnia skomplikowania aplikacji VBA czyni o wiele łatwiejszym śledzenie działania programu, gdyż umożliwia pracę z różnymi wzajemnie powiązanymi procedurami w ramach pojedynczego modułu. Ponieważ kod VBA dla formularza, raportu lub strony dostępu do danych jest częścią obiektu, towarzyszy temu obiektowi, jeśli przeniesiemy go do innej bazy danych.
- **Łatwość usuwania błędów**. Edytor języka Visual Basic wbudowany w Accessa udostępnia zaawansowane narzędzia do debuggingu. Nie oferuje natomiast podobnej funkcji dla makr.
- **Elastyczność**. Choć makra mogą dokonywać sprawdzania warunków i, odpowiednio do wyników, przeprowadzać różne akcje, struktura sterowania VBA jest znacznie bardziej rozbudowana. Dzięki możliwości definiowania argumentów dla procedur VBA w trakcie działania programu, aplikacja może elastyczniej niż makra reagować na pojawianie się określonych warunków.
- **Przydatność do wykonywania obliczeń**. VBA pozwala użytkownikowi zapisywać kod własnych funkcji wykonujących złożone, szczegółowe kalkulacje na liczbach lub tekście.
- **Łatwość obsługi błędów**. Makra w chwili wystąpienia błędu wyświetlają komunikat standardowy. Korzystając z VBA, możemy w takiej sytuacji wskazać rodzaj błędu — kod potrafi samodzielnie podejmować próby korygowania błędów, wyświetlając jednocześnie komunikat zawierający szczegółowe informacje.

VBA ma jeszcze jedną, ogromną zaletę: za pomocą języka Visual Basic aplikację bazy danych możemy napisać w dowolnym programie pakietu Office, jak również w odrębnym module Visual Basic. Po zaznajomieniu się z obiektami bazy danych, ich metodami i właściwościami, będziemy mogli pracować w dowolnym środowisku Visual Basic — a wybór środowiska spoza Accessa, o czym powiemy w kolejnym punkcie, może okazać się wyborem trafnym.

Kiedy nie tworzyć aplikacji baz danych w Accessie?

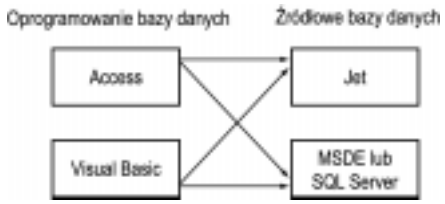
Ponieważ Access jest programem do obsługi baz danych, wchodzącym w skład pakietu Office, logicznym wydaje się, że to właśnie ten program będziemy wykorzystywać do tworzenia własnych aplikacji baz danych. Czasami jednak warto rozważyć inne opcje.

Powinniśmy bowiem pamiętać, że Access nie zarządza bazami danych bezpośrednio. Tym zajmuje się mechanizm obsługi bazy czyli oprogramowanie odpowiedzialne za przechowywanie i odczytywanie danych.

Mechanizmy obsługi baz danych, które można wykorzystać podczas pracy z Accessem, omówione zostały w rozdziale 3.

Rzeczywistą rolą Accessa jest pośredniczenie między użytkownikiem a mechanizmem obsługi bazy danych działającym w tle. Interfejs Accessa jest wyposażony w wiele komfortowych funkcji — takich jak projektowanie kwerend czy możliwość dostosowywania formularzy i raportów, z których każda znacznie upraszcza uzyskiwanie potrzebnych do pracy danych — a także w sterowanie sposobem ich wyświetlania. Jednak, bez względu na to, jak owe funkcje są przyjazne, faktem jest, że działają one w oddzieleniu od mechanizmu obsługi bazy, wykonującego właściwe zadania, związane z wyszukiwaniem i pozyskiwaniem żądanych informacji. Rysunek 5.2 w sposób schematyczny ilustruje tę rozdzielność ról.

Rysunek 5.2. *Ilustracja ta przedstawia przykłady połączeń interfejsów baz danych z mechanizmami ich obsługi*



Ponieważ interfejs użytkownika i mechanizm obsługi bazy danych są dwoma, odrębnymi modułami programowymi, pamiętać należy, że możliwe jest wykorzystywanie narzędzi innych niż Access, instruujących działania mechanizmu obsługującego. Dokładnie o to chodzi w tworzeniu aplikacji baz danych w VBA. Nawet jeśli kod aplikacji zapiszemy w Accessie, kod VBA będzie się komunikował z mechanizmem obsługującym nie za pomocą tego programu, a przy użyciu całkowicie niezależnego komponentu programowego, takiego jak ADO lub DAO.

Znaczenie akronimów ADO i DAO omówimy w rozdziale 6.

Programistyczne opcje tworzenia aplikacji bazy danych

Skoro więc Access nie jest jedynym narzędziem potrafiącym kierować pracą mechanizmu obsługi bazy danych, jakiego narzędzia powinniśmy używać do kodowania aplikacji? Opcje i wynikające z ich wykorzystania korzyści przedstawiają się następująco.

- **Access**. Aplikacje VBA tworzone w Accessie mogą czynić bezpośredni użytek z obiektów Accessa, takich jak formularze i raporty. Ponieważ formularze i raporty Accessa mają wbudowane funkcje baz danych, ich użycie może znacznie skrócić, w porównaniu do VBA, czas potrzebny na opracowanie aplikacji. Ponadto Access jest najlepszym rozwiązaniem, jeśli aplikacje tworzymy jedynie do użytku osobistego, gdy spodziewamy się częstego dodawania nowych funkcji w miarę pojawiania się nowych potrzeb.
- **Inna aplikacja Office, taka jak Word lub Excel**. W wielu aplikacjach użytkownika dostęp do bazy danych jest tylko jedną z funkcji. Jeśli w swych planach przewidujemy, że tworzona aplikacja powinna mieć możliwość intensywnego przetwarzania tekstu lub dokonywania zawiłych obliczeń, utworzenie jej w programie Word lub Excel umożliwi bezpośrednie wykorzystanie zasobów tych programów. Dzięki ADO lub podobnemu obiektowi kod VBA zachowuje możliwość odczytywania, wyświetlania

i manipulowania danymi. Aplikacja zaś będzie mniejsza, szybsza i mniej skomplikowana, niż gdybyśmy wykorzystali COM do automatyzacji pracy Accessa, wykorzystując program Word (lub Worda za pośrednictwem Accessa). Niewspomagane formularze i raporty VBA są związane z bazą danych, ale kod niezbędny do połączenia formularza z danymi nie jest wcale skomplikowany. Ponadto w celu połączenia formantów i formularzy z danymi możemy posłużyć się pakietem Office Developer lub innym, podobnym do niego narzędziem innego producenta. Nasze dzieło można przenieść bezpośrednio do innych aplikacji VBA, ponieważ nie jest ono formularzem specyficznym dla Accessa.

- **Visual Basic (nie VBA)**. Visual Basic (użyteczny przodek VBA) jest doskonałym narzędziem do programowania baz danych. Kod Visual Basic może nawiązywać połączenie z bazami danych przez ADO, podobnie jak VBA, a formularze są powiązane z bazą danych w podobny sposób jak formularze Accessa. Aplikacje utworzone w języku Visual Basic wyróżniają się w stosunku do innych, powstałych przy użyciu VBA w programie Access, Word czy Excel, dwiema istotnymi zaletami. Po pierwsze, działają o wiele szybciej, gdyż aplikacje Visual Basic są kompilowane, a nie interpretowane podczas uruchamiania. Po drugie, aplikacje te można bez problemu dystrybuować wśród innych użytkowników — nie muszą one bowiem korzystać z aplikacji nadrzędnej. Jedyną, dużą wadą jest to, że Visual Basic trzeba kupić.

Dlaczego Access zachowuje swoje znaczenie?

Nawet jeśli zrezygnujemy z Accessa jako środowiska programistycznego, wciąż pozostaje on wspaniałym narzędziem do tworzenia aplikacji baz danych. Korzystając z jego wygodnych w użyciu poleceń, możemy szybko budować tabele i kwerendy tworzące bazę danych. W Accessie można również zapisywać i testować kod VBA, przenosząc go do środowiska docelowego dopiero po ukończeniu, gdy potwierdzona zostanie jego funkcjonalność. Możliwe jest także szybkie prototypowanie formularzy aplikacji.

Nie można eksportować formularzy Accessa do aplikacji finalnych: są one bowiem niekompatybilne ze standardami VBA UserForms oraz formularzami stosowanymi w Visual Basic.

Magia tworzenia aplikacji: wykorzystanie kreatorów baz danych

Najprostszym sposobem zbudowania działającej aplikacji jest zlecenie tego zadania Accessowi. Kreatory dostarczane wraz z programem demonstrują siłę i zalety aplikacji, które są w pełni użyteczne. Zgromadzone w kolekcji dołączonej do Accessa kreatory

pozwalają tworzyć firmowe bazy danych, na przykład do zarządzania kontaktami czy stanu magazynu, a także przechowujące informacje osobiste, takie jak spisy kaset wideo czy dokumentację postępów w ćwiczeniach gimnastycznych.

Aby użyć kreatora, należy wydać polecenie *Nowy* z menu *Plik*. Na panelu *Nowy plik*, który się otworzy, trzeba odszukać i kliknąć opcję *Szablony ogólne*, zawartą w sekcji *Nowy z szablonu*. Chcąc utworzyć nową bazę danych Jet, powinniśmy wybrać szablon *Pusta baza danych*, a *Projekt (nowa baza danych)* wówczas, jeżeli chcielibyśmy zbudować projekt SQL Server.

Porównanie rozwiązań opartych na mechanizmach Jet i SQL Server znaleźć można w rozdziale 3.

Aby skonstruować bazę danych Jet, możemy również wykorzystać kreatory dostępne na karcie *Bazy danych* okna dialogowego *Szablony* (patrz rysunek 5.3).

Rysunek 5.3. *Access udostępnia kreatory pomagające użytkownikowi w tworzeniu typowych baz danych*

Uruchomiony kreator wyświetla serię paneli, na których dokonuje się wyboru pól włączanych do aplikacji oraz podstawowych opcji formatowania formularzy i raportów. W oparciu o nasze decyzje kreator tworzy obiekty niezbędne w konstruowanej aplikacji, w razie konieczności uzupełniając je procedurami Visual Basic. Wszystkie obiekty i kod są dostępne dla użytkownika, co pozwala przyjrzeć się, w jaki sposób Access łączy składniki aplikacji. Istniejące obiekty można, w razie potrzeby, dostosowywać oraz uzupełniać.

Jednak praktycznie rzecz biorąc, żaden, choćby najlepszy, kreator nie jest w stanie spełnić wszystkich naszych specyficznych potrzeb dotyczących zarządzania informacjami. Po wygenerowaniu kilku przykładowych aplikacji dojdziemy do wniosku, że powinniśmy utworzyć własną.

Tworzenie własnych aplikacji w Accessie

Proces konstruowania aplikacji w Accessie jest, w ogólnych zarysach, bardzo prosty. Rozpoczynamy go od przygotowania podstawowych elementów bazy danych, a następnie łączymy je, formując aplikację.

Projektowanie interfejsu użytkownika

Jednym z najważniejszych zadań należących do nas, projektantów aplikacji, jest przygotowanie interfejsu użytkownika, który będzie wyglądał zachęcająco, wydajnie pełniąc swoje funkcje. Na szczęście, swoją uwagę możemy skupić na estetyce i funkcjonalności aplikacji — mechanika budowy interfejsu jest bardzo prosta. Możliwości, jakie dają nam funkcje dostosowawcze „przeciągnij i upuść" pakietu Office, pozwalają bez większego wysiłku łączyć formularze, paski narzędziowe i menu.

Przygotowanie panelu przełączającego

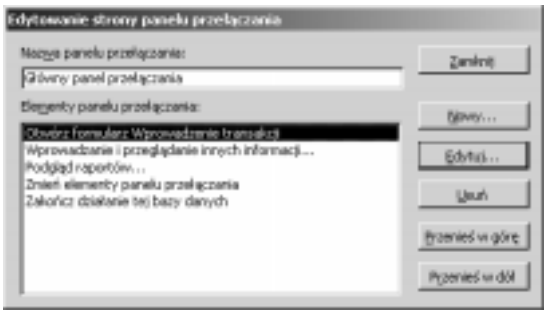
W rozdziale 7. omówimy podstawy tworzenia formularzy w Accessie. Opisane tam techniki planowania układu elementów można zastosować również podczas konstruowania paneli przełączających oraz innych, nie przeznaczonych do wyświetlania danych, a do aktywowania innych funkcji aplikacji.

Jedyną cechą specjalną tych formularzy jest to, że nie są one powiązane z żadnymi tabelami ani kwerendami w bazie danych. Tworząc formularz tego rodzaju, w oknie dialogowym *Nowy formularz* pole wyboru tabeli lub kwerendy dla formularza należy pozostawić puste. To pozwoli wykorzystać formularz jako podstawowy element tworzonego interfejsu użytkownika bez konieczności zajmowania się dołączaniem danych.

Korzystanie z Menedżera panelu przełączania

Aby uruchomić *Menedżera panelu przełączania*, należy użyć polecenia *Menedżer panelu przełączania* z menu *Narzędzia/Narzędzia bazy danych*. Jest to narzędzie tworzące nowe, w pełni funkcjonalne, formularze paneli przełączania i pozwalające na edycję istniejących. Aby utworzyć nowy panel przełączania, należy kliknąć przycisk *Nowa*, a następnie przycisk *Edytuj*, otwierający okno dialogowe *Edytowanie strony panelu przełączania* (patrz rysunek 5.4), umożliwiające dodawanie i aranżowanie układu poleceń na panelu. Powstające w ten sposób formularze są bardzo podobne do tych, które utworzyliśmy jako przykładowe za pomocą kreatorów Accessa. Mają one miły wygląd, doskonale funkcjonują, ale niewiele różnią się między sobą.

Do wszechstronnej edycji formularzy Accessa możemy użyć narzędzi projektowych, które omówimy w rozdziale 7.



Rysunek 5.4. *Menedżer panelu przełączania tworzy panele przełączające i pozwala edytować je w oknie dialogowym Edytowanie strony panelu przełączania*

Przygotowanie formularza startowego

Panel przełączania lub inny formularz, który ma być prezentowany użytkownikowi podczas startu aplikacji, to formularz uruchamiania. Wskazujemy go, wybierając z listy rozwijanej *Wyświetl formularz/stronę* w oknie dialogowym otwieranym poleceniem *Narzędzia/Uruchamianie*.

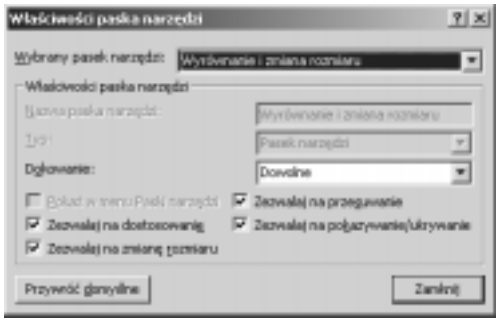
Dostosowywanie pasków poleceń aplikacji

W Accessie funkcjonują takie same, jak w pozostałych programach Office, techniki tworzenia własnych pasków narzędziowych, menu i przycisków. Jednak Access zapewnia nam znacznie szerszą kontrolę nad technicznymi ustawieniami tych elementów niż inne aplikacje pakietu Office.

Tworzenie nowych pasków menu, narzędziowych i menu skrótów

Kiedy konstruujemy własną aplikację, występujące w niej paski narzędziowe powinniśmy zbudować od podstaw, zamiast modyfikować istniejące.

Chociaż własne paski narzędziowe tworzy się podobnie jak w innych programach Office, to jednak tylko Access pozwala zmieniać ich właściwości bez zastosowania kodu VBA. W tym celu należy otworzyć kartę *Paski narzędzi* okna dialogowego *Dostosowywanie*, wywoływanego za pomocą polecenia *Narzędzia/Dostosuj* i kliknąć przycisk *Właściwości*. Pojawi się wówczas drugie okno dialogowe, *Właściwości paska narzędzi*. Tu możemy wybrać pasek z rozwijanej listy i zmodyfikować jego ustawienia (patrz rysunek 5.5).



Rysunek 5.5. *Właściwości pasków narzędziowych można przeglądać i modyfikować za pomocą tego okna dialogowego*

Okno dialogowe *Właściwości paska narzędzi* zapewnia możliwość podjęcia decyzji, czy paski zaimplementowane w naszej aplikacji będą mogły być dostosowywane, przesuwane, dokowane, ukrywane i tak dalej. Jeśli opracowujemy własny pasek narzędziowy, rezygnując z wykorzystania istniejącego, możemy również decydować o jego typie.

Przyłączanie własnych pasków narzędziowych do formularzy i raportów

Każdy z tworzonych pasków narzędziowych możemy zmienić na pasek menu, wybierając odpowiedni typ z listy rozwijanej w oknie dialogowym *Właściwości paska narzędzi*. Następnie możemy połączyć go z formularzem lub raportem albo wyznaczyć jako domyślny pasek menu dla całej bazy danych.

Aby przygotować pasek menu dla formularza bądź raportu, należy przypisać go jako właściwość *Pasek menu* na karcie *Właściwości* tego obiektu w widoku projektu. Od chwili gdy obiekt zostanie aktywowany w aplikacji, wybrany pasek menu pojawi się na pod jego górną krawędzią, tuż poniżej paska tytułowego, zastępując pasek globalny.

Mówiąc o globalnym pasku menu, należy mieć na myśli pasek przygotowany dla całej bazy danych za pośrednictwem okna dialogowego *Uruchamianie*, wywoływanego z menu *Narzędzia*. Pasek menu, który tu wybierzemy, jest wyświetlany (zamiast domyślnego paska Accessa) w całej bazie oprócz tych formularzy i raportów, dla których wybraliśmy inny pasek.

Regularne paski narzędziowe mogą być dołączane do formularzy i raportów również za pomocą właściwości *Pasek narzędzi* obiektu. Kiedy użytkownik aplikacji przełączy widok na ów obiekt, wraz z nim pojawi się wybrany pasek narzędziowy w zastępstwie domyślnego, przewidzianego przez Accessa dla obiektów tego rodzaju. Na przykład w przypadku formularza nie będzie widoczny pasek *Widok formularza*; zamiast niego wyświetlony zostanie pasek, który wybraliśmy.

Wybór paska narzędziowego jako *Podręczny*, dokonany w oknie dialogowym *Właściwości paska narzędzi*, powoduje skonwertowanie go do postaci menu skrótów, które reaguje na kliknięcia prawym klawiszem myszy. Podczas gdy okno dialogowe *Dostosowywanie* pozostaje otwarte, pasek pojawi się w menu *Niestandardowe* paska narzędzi *Menu skrótów*. Tu możemy uzupełnić pasek kolejnymi pozycjami menu.

Dodawanie poleceń do paska narzędzi

Podobnie jak Word pozwala przypisywać czcionki do przycisków paska narzędzi, tak Access umożliwia użytkownikowi wypełnianie pasków przyciskami służącymi do wyświetlania określonych tabel, kwerend, formularzy i raportów. Wystarczy jedno kliknięcie takiego przycisku, by wybrany obiekt bezzwłocznie pojawił się na ekranie. Tak jak w Wordzie, na paskach narzędzi można umieszczać także przyciski makr.

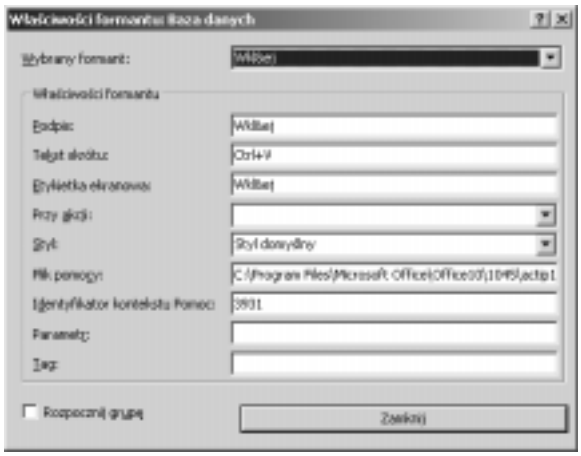
Po otwarciu okna dialogowego *Narzędzia/Dostosuj* wybieramy kartę *Polecenia* i odszukujemy elementy opisane jako *Wszystkie tabele*, *Wszystkie kwerendy* i tak dalej. W dalszej kolejności przeciągamy ikony wybranych obiektów na pasek narzędzi.

Otwieranie obiektu bazy danych za pomocą kliknięcia przycisku na pasku narzędzi jest równoznaczne z otwarciem okna *Bazy danych*. Nie mamy wpływu na to, który widok zostanie otwarty. Nie możemy również uruchamiać żadnych innych akcji. Aby uzyskać oczekiwane rezultaty, należy utworzyć odpowiednie makro i umieścić je na pasku narzędzi.

Wykorzystanie właściwości formantów paska narzędzi

Access zapewnia nam możliwość bezpośredniej regulacji ustawień pojedynczych formantów znajdujących się na paskach sterujących, w tym przycisków na paskach narzędziowych i menu, za pomocą okna dialogowego. W innych aplikacjach pakietu Office modyfikacja tych opcji wymaga zastosowania kodu VBA. Po otwarciu okna dialogowego *Dostosowywanie* należy kliknąć prawym klawiszem myszy wybrany element, przycisk lub pozycję menu, a następnie użyć polecenia *Właściwości*, umieszczonego u dołu wyświetlonego menu skrótów. Okno dialogowe *Właściwości formantu: Baza danych* pokazane jest na rysunku 5.6.

Rysunek 5.6. *Za pomocą okna dialogowego Właściwości formantu: Baza danych możemy określić specyficzne ustawienia przycisków narzędziowych i pozycji menu dla naszej aplikacji*



Access jest jedyną aplikacją pakietu Office, która nie wymaga użycia VBA do definiowania tekstu wyświetlanego obok formantu umieszczonego w menu jako jego pozycja. Tekst ten informuje zazwyczaj o skrótach klawiaturowych korespondujących z danym formantem. W Accessie napisy te redaguje się w polu *Tekst skrótu*. Podobnie pole *Etykietka ekranowa* pozwala zdefiniować treść napisu pojawiającego się na ekranie, gdy użytkownik ustawi wskaźnik myszy ponad formantem.

Przypisywanie funkcji VBA do przycisków na paskach narzędzi

Specjalny pasek narzędziowy o nazwie *Polecenia niestandardowe*, działający w połączeniu z oknem dialogowym *Właściwości formantu: Baza danych* (patrz rysunek 5.6), daje nam dostęp do dowolnych funkcji Visual Basic, wywoływanych jednym kliknięciem. Aby skonstruować taki pasek, powinniśmy zacząć od przygotowania potrzebnych funkcji, do czego służy edytor Visual Basic Editor.

Technika ta działa jedynie w odniesieniu do funkcji, a nie procedur `Sub`. Dlatego też powinniśmy upewnić się, że składnia jest poprawna.

Po przygotowaniu funkcji otwieramy okno dialogowe *Dostosowywanie*. Na samym szczycie listy *Polecenia* powinniśmy zobaczyć pozycję *Niestandardowe* (jeśli jej nie widać, należy upewnić się, że wybrana została kategoria *Plik*). Następnie przeciągamy tę pozycję na pasek narzędziowy. Klikamy ją prawym klawiszem myszy, po czym, wybierając polecenie *Właściwości* z menu skrótów, wywołujemy okna dialogowe *Właściwości formantu: Baza danych*.

Teraz w polu *Przy akcji* wpisujemy wyrażenie uruchamiające funkcję. Jeśli na przykład funkcja nazywa się `StayAlert`, wówczas powinniśmy wpisać `=StayAlert( )`.

Przygotowanie opcji startowych i zabezpieczanie interfejsu

Za pomocą okna dialogowego *Narzędzia/Uruchomienie* możemy sprawować pełną kontrolę nad tym, co użytkownik widzi po uruchomieniu aplikacji, a także do jakich narzędzi ma dostęp. Aby mieć pewność, że opracowany interfejs nie ulegnie żadnym modyfikacjom, należy dezaktywować wszystkie pola opcji w oknie dialogowym *Uruchamianie* i wybrać własny pasek menu, pasek menu skrótów oraz wyświetlany formularz lub stronę.

Usuwając na przykład oznaczenie pola opcji *Wyświetl okno bazy danych*, zapobiegniemy wyświetlaniu głównego okna Accessa — trudno bowiem skupić uwagę użytkownika, jeśli trzeba ją na siłę kierować ku oknu własnej aplikacji. Aby również zapobiec otwieraniu okna bazy danych przy użyciu klawiatury, należy wyłączyć opcję *Użyj specjalnych klawiszy programu Access*.

Jednak, mimo przedsięwziętych środków, nawet średnio doświadczony użytkownik będzie mógł uruchamiać aplikację, korzystając z Accessa. Mógłby, oprócz tego, wprowadzić zmiany na paskach narzędziowych, tak jak my, pracując w Accessie bez użycia funkcji dostosowywania. Użytkownik może również wyświetlać i ukrywać niestandardowe paski narzędziowe, pozycjonować je w dowolnym miejscu ekranu i swobodnie dostosowywać, dodając lub usuwając z nich przyciski a nawet całe paski narzędzi.

Aby uniemożliwić użytkownikowi dokonywanie opisanych zmian, powinniśmy wykonać trzy czynności.

1. Umieścić każdy pasek narzędzi aplikacji w miejscu, w którym mają one się znajdować. Następnie należy użyć okna dialogowego *Właściwości paska narzędzi* (pokazanego na rysunku 5.5) i za pomocą dostępnych tu opcji dopuścić lub zablokować możliwość przesuwania, skalowania czy dokowania paska.

2. Otworzyć okno dialogowe *Narzędzia/Uruchomienie* i dezaktywować pole wyboru opcji *Zezwalaj na zmiany pasków narzędzi/menu*.

   W ten sposób uniemożliwimy użytkownikom dostęp do funkcji dostosowywania oraz menu *Widok/Paski narzędzi*, co da nam pewność, że zawartość pasków narzędziowych nie ulegnie modyfikacji. To jednak nie uniemożliwia użytkownikom przesuwania, skalowania i dokowania pasków.

3. Jak dotąd, wszystko jest w porządku. Teraz musimy w jakiś sposób zablokować możliwość zmiany ustawień dokonanych w oknie dialogowym *Uruchamianie*. To wymaga zastosowania ochrony z wykorzystaniem hasła użytkownika.

Maksimum wydajności przy minimalnej ilości kodu VBA

Choć napisanie kodu VBA często jest konieczne, to czasami jednak można się bez niego obejść — istnieją nawet istotne powody ku temu, by minimalizować zastosowanie kodu w aplikacji Accessa. Jednym z nich jest, oczywiście, oszczędność czasu. (VBA ma duże wymagania w stosunku do programisty). Ponadto VBA obniża wydajność aplikacji, a na dodatek powiększa jej objętość.

Podczas konstruowania aplikacji większość czynności, jakie musimy wykonać, polega na tworzeniu prostych powiązań między elementami interfejsu użytkownika, takimi jak przyciski, oraz typowymi obiektami bazy danych czyli formularzami lub kwerendami. W większości przypadków chodzi zaś o to, by określony formant przenosił akcję do innego obiektu bazy. Do osiągnięcia tego celu wystarczy skromny formularz — VBA jest niepotrzebny. Kod VBA nie jest również konieczny do wykonywania masowych uaktualnień czy usuwania rekordów bazy danych. Wystarczy bowiem skonstruować kwerendę funkcjonalną i użyć jej jako obiektu bazy, wywoływanego z formularza.

Użycie obiektów niezwiązanych

Formularz lub raport niezwiązany to obiekt, do którego nie został przypisany żaden moduł VBA. Nie znaczy to wcale, że formularz jest niepełnowartościowy. W rozdziale 7. dowiemy się, że zachowuje on możliwość wyświetlania danych dzięki formantom związanym z polami bazy danych. Ponadto formanty można wiązać z hiperłączami, pozwalającymi otwierać inne obiekty bazy danych, bez żadnej pomocy ze strony VBA. Alternatywnym rozwiązaniem jest użycie makr do wykonywania akcji, takich jak ukrywanie lub wyświetlanie niestandardowych pasków narzędzi po otwarciu formularza lub w chwili zajścia określonego zdarzenia.

W porównaniu do obiektów zawierających moduły VBA formularze i raporty niezwiązane zajmują mniej miejsca na dysku i są szybciej ładowane do pamięci. Dopóki więc nie musimy korzystać z zaawansowanych funkcji VBA, rezygnacja z dołączania modułów do obiektów jest rozsądną decyzją.

To, czy do formularza bądź raportu dołączony został moduł, można sprawdzić, kontrolując zawartość pola właściwości *Ma moduł* w oknie właściwości obiektu. Właściwość tę znajdziemy u dołu karty *Inne*.

Do sprawdzania przypisań modułów do obiektów nie należy używać polecenia *Widok/Kod*. Powoduje ono bowiem utworzenie modułu, jeśli dotąd go nie było. Nawet pusty moduł może być przyczyną osłabienia wydajności aplikacji, marnotrawiąc przy okazji część pamięci.

Jeśli do obiektu przyłączony jest moduł, to pozostaje on w użyciu do chwili, gdy wartość właściwości *Ma moduł* przestawimy na *Nie*. Kiedy to uczynimy, moduł i cały zawarty w nim kod zostanie usunięty.

Wykorzystanie kwerend funkcjonalnych do zmniejszenia kodu VBA

Istnieje wiele sytuacji, w których należy wykonać określone operacje na grupie rekordów. Na przykład na początku każdego miesiąca rekordy wprowadzone w miesiącu poprzednim muszą znaleźć się w oddzielnej tabeli archiwizacyjnej. Zamiast jednak tworzyć kod VBA do wykonania tego zadania, powinniśmy użyć kwerendy funkcjonalnej.

Jak wskazuje określenie kwerendy, nie służy ona do pobierania danych, a jedynie do wykonywania na nich określonych akcji. Istnieją cztery typy kwerend funkcjonalnych: usuwające, uaktualniające, dołączające oraz tworzące tabele. Nazwy tych typów nie wymagają dodatkowych objaśnień.

W przykładowej sytuacji moglibyśmy posłużyć się kwerendą otwartą w widoku projektu, przygotowując ją jako kwerendę filtrującą rekordy na podstawie daty i wykorzystującą poprzedni miesiąc jako kryterium. Zamiast kodowania miesiąca powinniśmy użyć następującego wyrażenia, wpisując je w wierszu *Kryterium kwerendy*:

```
Month([PoleDanych])= If(Month(Now())=1,12,Month(Now())-1)
```

W powyższym wyrażeniu `PoleDanych` jest nazwą pola zawierającego daty, według których chcemy przefiltrować rekordy. Następnie moglibyśmy wybrać polecenie *Kwerenda tworząca tabelę* z menu *Kwerenda* i wpisać nazwę tworzonej tabeli archiwizacyjnej. W ten sposób powstałaby kwerenda gotowa do uruchomienia. Jej działanie polegałoby na odnalezieniu wszystkich rekordów wprowadzonych w ubiegłym miesiącu i wyeksportowanie ich do nowej tabeli.

Przygotowaną kwerendę można przypisać do formantu na panelu przełączania aplikacji lub podobnym formularzu. W tym celu należy otworzyć okno makra, wyznaczając akcję *OtwórzKwerendę* do uruchamiania kwerendy i wpisując jej nazwę jako właściwość *Nazwa kwerendy*. Akcję *OtwórzKwerendę* można poprzedzić akcją *UstawOstrzeżenia* i za jej pomocą wyłączyć wyświetlanie komunikatów ostrzegawczych, które w przeciwnym razie będą wyświetlane podczas uruchamiania kwerendy. Po zapisaniu makra można je przypisać, wykorzystując okno dialogowe *Właściwości*, do zdarzeń rozpoznawanych przez formularz, raport bądź jeden z ich formantów.

Pisanie kodu VBA w Accessie

Techniki, którymi dotąd zajmowaliśmy się w niniejszym rozdziale, sprawdzają się w wielu prostych aplikacjach: kod Visual Basic pozostaje w odwodzie na wypadek, gdyby jego użyteczność okazała się niezbędna. Choć główne elementy Visual Basic są wspólne dla ogółu programów Office, to wszystkie aplikacje VBA, w tym i Access, wykorzystują własne zestawy obiektów. Najważniejszymi obiektami Accessa są formularze, raporty oraz strony dostępu do danych, jak również obiekty pojedynczych formantów.

W rozdziale 7. omówione zostaną techniki pracy z obiektami baz danych w Access VBA.

Dystrybucja aplikacji

Po przygotowaniu bloków składowych aplikacji, dodaniu kodu łączącego i przetestowaniu wszystkich składników możemy zająć się upowszechnieniem swego dzieła. W kolejnych punktach omówimy zagadnienia związane z przygotowaniem gotowej aplikacji do dystrybucji.

Dzielenie bazy danych Jet w sieci

W większości firm typowa aplikacja bazy danych przeznaczona jest do używania przez wielu użytkowników, na odrębnych komputerach. Jeśli wyślemy swój projekt do więcej niż jednej osoby i jeśli grupa ma możliwość pracy sieciowej, najlepiej będzie, gdy nasza aplikacja zostanie przygotowana w dwóch częściach — jednej, przeznaczonej do zainstalowania w komputerze każdego użytkownika, a drugiej, instalowanej na serwerze. W takim przypadku każdy pracownik otrzyma odrębną kopię interfejsu aplikacji — wszystkie niestandardowe formularze i paski narzędziowe — jak również elementy funkcjonalne, takie jak kwerendy, makra i moduły. Właściwa baza danych — tabele i zawarte w nich dane — przechowywane są na serwerze sieciowym, gdzie pozostają dostępne dla wszystkich pracowników.

Taki podział aplikacji jest zasadny z kilku względów. Jednym jest to, że dane są zazwyczaj bezpieczniejsze, a tworzenie kopii zapasowych odbywa się regularnie. Ponadto ponieważ każdy pracuje z wykorzystaniem tej samej kopii bazy danych, nie musimy martwić się konserwacją i synchronizacją wielu zestawów danych.

Inną korzyścią podziału bazy danych jest to, że dystrybucja aplikacji staje się łatwiejsza. Ogromne tabele muszą być zapisane na serwerze tylko raz. Uaktualnianie pozostałej części aplikacji powinno być względnie łatwe, ponieważ interfejs danych, i to dowolnie dużych rozmiarów, jest jej relatywnie niewielką cząstką.

Access potrafi obsługiwać opisaną konfigurację. W przypadku projektów Access działających na podbudowie baz danych SQL Server, dane zawsze są oddzielone od plików projektu. Jako część projektu, oparta na mechanizmie Jet, aplikacja Accessa może się łączyć z tabelami przechowywanymi na dowolnym, dostępnym jej dysku twardym, serwerze lub witrynie internetowej. Aby połączyć aplikację z zewnętrznymi tabelami, należy użyć polecenia *Plik/Pobierz dane zewnętrzne/Połącz tabele*, otwierając w ten sposób kreatora, który pomaga użytkownikowi w odszukiwaniu i dołączaniu zewnętrznych tabel do bieżącej bazy danych. Chcąc natomiast umieścić tabele bieżącej bazy danych w innym pliku, należy posłużyć się narzędziem *Rozdzielacz baz danych*.

Zanim dokonamy podziału aplikacji, musimy zastanowić się, w jakim miejscu w sieci powinny znaleźć się tabele z danymi. Następnie, jeśli dysponujemy sprawnym łączem sieciowym, wydajemy polecenie *Narzędzia/Narzędzia bazy danych/Rozdzielacz baz danych*. Kreator *Rozdzielacz bazy danych* przeprowadzi nas przez całą procedurę, pytując o miejsce przeniesienia bazy danych. Wkrótce potem rozpocznie się proces oddzielania tabel od pozostałej części bazy (interfejsu aplikacji) i umieszczania ich w nowej bazie. Jeśli proces przebiegnie pomyślnie, użytkowa część aplikacji zostanie uaktualniona przez utworzenie łączy do tabel przeniesionych do nowej lokalizacji. W tym momencie aplikacja będzie się składała z dwóch oddzielnych plików bazy danych, które jednak będą funkcjonowały tak, jakby były jednym plikiem.

Replikacja bazy danych

Jeśli przyłączenie wszystkich użytkowników do tego samego serwera okaże się niemożliwe lub niepraktyczne, możemy skorzystać z funkcji replikacji bazy danych, jako sposobu na udostępnienie wszystkim tych samych danych. *Replikacja bazy danych* polega na wygenerowaniu dwóch lub więcej kopii tych samych informacji oraz ich zsynchronizowaniu, co ma zapewnić zgodność modyfikacji danych we wszystkich kopiach.

Jedna z replik pełni rolę wzorca projektowania. Jest to jedyna kopia, w której można wprowadzać zmiany projektu całej bazy danych. Każdy, dysponujący aplikacją, może jednak zmieniać, dodawać lub usuwać rekordy. Podczas każdej synchronizacji zmiany poczynione w jednej replice są uwzględniane w pozostałych. Jest to proces postępujący — między bazami przesyłane są tylko dane zmodyfikowane — dzięki czemu przebiega dość szybko.

Replikacja jest lepszym rozwiązaniem niż dzielenie bazy danych wtedy, gdy łącza sieciowe są powolne, a jej wydajność spada, kiedy na przykład wszyscy pracownicy działu sprzedaży jednocześnie podłączają swoje laptopy do sieci głównej. Jeśli w każdym laptopie zainstalujemy kopię całej bazy danych, wówczas każdy posługujący się takim komputerem pracownik będzie miał stały dostęp do względnie aktualnych danych, zachowując przy tym możliwość ich szybkiego odświeżenia w dowolnym momencie.

Polecenia związane z replikowaniem bazy zgromadzone są w podmenu *Narzędzia/Replikacja*. Po wydaniu polecenia *Utwórz replikę* uruchomiony zostaje proces replikowania, w trakcie którego nowa baza staje się *Wzorcem projektowania*. Wzorzec ten może być następnie wykorzystany podczas generowania replik dla kolejnych użytkowników.

Zabezpieczanie plików

Po pomyślnym przetestowaniu całej aplikacji bazy danych warto zachować ją pod postacią pliku *.mde*. W ten sposób spowodujemy usunięcie źródłowego kodu Visual Basic po jego skompilowaniu. Choć aplikacja będzie działała tak samo — a właściwie trochę lepiej, gdyż ograniczy swoje wymagania pamięciowe — nikt nie będzie mógł zobaczyć pracy, jaką włożyliśmy w jej zaprogramowanie ani też zmodyfikować składników, takich jak formularze czy kwerendy.

Aby zachować aplikację w pliku *.mde*, należy otworzyć menu *Narzędzia*, a następnie *Narzędzia bazy danych* i wydać polecenie *Utwórz plik MDE*. Oczywiście, trzeba również nadać nazwę nowemu plikowi *.mde*. Proces ten nie powoduje modyfikacji pliku oryginalnego, który jednak należałoby umieścić w bezpiecznym miejscu, skąd będziemy mogli go pobrać, gdy okaże się, że konieczne jest wprowadzenie ulepszeń.