helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

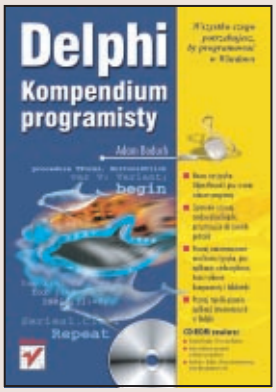
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Delphi. Kompendium programisty



Autor: Adam Boduch
ISBN: 83-7197-977-0
Format: B5, stron: 328

Środowisko Delphi firmy Borland to doskonały wybór dla programistów aplikacji Windows, zarówno tych doświadczonych, jak i zupełnie początkujących. Język ObjectPascal, wywodzący się z Pascala jest łatwy do nauczenia się, gwarantuje przy tym nabycie właściwych nawyków programistycznych. Wygodne środowisko programistyczne, bogata biblioteka gotowych modułów i ogromna liczba stworzonych aplikacji, należą do niezaprzeczalnych atutów Delphi.

Książka „Delphi. Kompendium programisty", to przystępnie napisany podręcznik, omawiający zarówno język ObjectPascal, jak i środowisko Delphi. Jest on przeznaczony zarówno dla osób, które dopiero zaczynają przygodę z programowaniem, jak i dla doświadczonych programistów używających wcześniej innych języków programowania, a teraz pragnących skorzystać z Delphi.

Dzięki tej książce dowiesz się:

- Jak pisać programy w języku ObjectPascal
- Czym są biblioteki VCL i jak ich używać
- Jak korzystać ze zintegrowanego środowiska programistycznego IDE Delphi
- Jak tworzyć aplikacje wielowątkowe
- Jak tworzyć własne komponenty i biblioteki DLL
- Jak używać Windows API
- Jak tworzyć w Delphi aplikacje internetowe

„Delphi. Kompendium programisty" to kompendium w pełnym tego słowa znaczeniu: kompletne i wyczerpujące opracowanie. Po przeczytaniu tej książki będziesz wielokrotnie do niej wracał.

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

Spis treści

Rozdział 1. Wstęp do programowania w Delphi 9
Ogólnie o programowaniu 9
Czym jest Delphi? 10
Proces instalacji 11
Porównanie różnych dystrybucji Delphi 6 12
Porównanie do poprzednich wersji 12
Rzut oka na środowisko 12
Paleta komponentów 13
Projektant formularzy 14
Inspektor Obiektów 14
Edytor kodu 15
Piszemy pierwszy program 16
Tworzenie projektu 16
Edycja kodu źródłowego 16
Zapisywanie projektu 17
Podsumowanie 18

Rozdział 2. Object Pascal 19
Podstawowa składnia 19
Trochę o modułach 21
Komentarze 22
Zmienne 23
Typy zmiennych 24
Stałe 25
Instrukcje warunkowe 25
Instrukcja if 26
Instrukcja else 27
Instrukcja case 28
Operatory 29
Typy 30
Rzutowanie 32
Konwersja 33

Tablice ... 34
Tablice statyczne ... 34
Tablice dynamiczne ... 35
Tablice dwuwymiarowe ... 36
Procedury ... 36
Funkcje ... 39
Moduły ... 41
Initialization oraz finalization ... 43
Rekordy ... 44
Instrukcja wiążąca with ... 46
Operacje matematyczne ... 46
Pętle ... 47
Pętla for ... 47
Pętla repeat ... 49
Pętla while ... 50
Polecenia Continue i Break ... 51
Polecenie Continue ... 51
Polecenie break ... 52
Etykiety ... 52
Funkcje Pred i Succ ... 53
Pliki dołączane ... 53
Wskaźniki ... 54
Podsumowanie ... 56

Rozdział 3. VCL ... 57

Biblioteka wizualna ... 57
Rzut oka na kod formularza ... 58
Klasy ... 61
Podstawowe informacje dotyczące klas ... 62
Dziedziczenie ... 62
Odwołanie do klasy ... 63
Dostęp do klasy ... 64
Przedefiniowanie metod ... 67
Typy metod ... 68
Konstruktory i destruktory ... 68
Przykładowa gra ... 69
Parametr Sender procedury ... 84
Operatory is i as ... 85
Sposoby na zmniejszenie pliku wykonywalnego ... 86
Zbiory ... 86
Typ Variant ... 87
Wskaźnik Self ... 88
Podsumowanie ... 89

Rozdział 4. IDE ... 91

Pasek narzędziowy ... 91
Repozytorium ... 92
Projektant formularzy ... 93
Menu projektanta formularzy ... 94
Ustawienia pulpitu ... 95
Opcje projektu ... 95
Strona Forms ... 95
Strona Applications ... 96

Strona Compiler ... 96
Strona Linker ... 97
Strona Directories/Conditionals ... 97
Strona Version Info ... 97
Strona Packages ... 97
Kilka formularzy w jednym projekcie ... 98
Aplikacje wielodokumentowe MDI ... 99
Projektowanie głównego formularza ... 100
Tworzenie menu ... 100
Projektowanie okna Child ... 102
Pisanie kodu ... 103
Porównanie do poprzednich wersji Delphi ... 109
Nowe komponenty w wersji Personal Edition ... 109
Test — porównanie różnych wersji Delphi ... 109
Zmiany w zakresie IDE ... 110
Podsumowanie ... 111

Rozdział 5. Multimedia ... 113

Klasa TCanvas ... 113
Zaczynamy ... 113
Obiekt TFont ... 114
Główne metody klasy Tcanvas ... 117
Pióra ... 118
Pędzle ... 119
Bitmapy ... 120
Pliki JPG ... 121
Rysowanie tekstu ... 122
Uchwyty ... 123
Odtwarzanie dźwięków ... 124
Odtwarzanie za pomocą polecenia PlaySound ... 124
Komponent TMediaPlayer ... 125
Odtwarzanie dźwięku ... 126
Zasoby ... 128
Edytor zasobów ... 128
Kursory ... 130
Ikony ... 131
Bitmapy w zasobach ... 132
Ręczne tworzenie zasobów ... 133
Ładowanie plików JPG ... 133
Ładowanie plików EXE ... 135
Odtwarzanie dzwięku z zasobu ... 136
Animacje tekstowe w Delphi ... 136
Podsumowanie ... 142

Rozdział 6. Operacje na plikach ... 143

Pliki tekstowe ... 143
Otwieranie, zamykanie, tworzenie ... 143
Edycja ... 145
Pliki typowane ... 146
Tworzenie plików typowanych ... 146
Edycja ... 147
Pliki strumieniowe ... 152
Przykład zastosowania strumieni — wyciąganie tagu z pliku MP3 ... 155

Pliki amorficzne ... 160
Kopiowanie, przenoszenie i kasowanie plików ... 162
Wyszukiwanie plików ... 164
Rekurencja ... 166
Podsumowanie ... 172

Rozdział 7. Programowanie zaawansowane ... 173
Wyjątki ... 173
Słowa kluczowe try i except ... 174
Słowo kluczowe finally ... 175
Słowo kluczowe raise ... 176
Selektywna obsługa wyjątków ... 178
Przechwytywanie wyjątków ... 178
Pliki INI ... 180
Budowa ... 180
Tworzenie i otwieranie ... 181
Edycja ... 182
Usuwanie kluczy ... 185
Rejestr Windows ... 186
Budowa kluczy ... 187
Tworzenie i otwieranie ... 188
Edycja rejestru i dodawanie własnych kluczy ... 188
Usuwanie danych ... 191
Przykład programu: rejestracja rozszerzenia ... 193
Pozostałe funkcje do operowania na rejestrze ... 201
Komunikaty ... 201
Obsługa komunikatów ... 202
Wysyłanie komunikatów ... 204
Wysyłanie własnych komunikatów i komunikacja między aplikacjami ... 205
Podsumowanie ... 208

Rozdział 8. Aplikacje wielowątkowe ... 209
Nieco o aplikacjach wielowątkowych ... 209
Klasa TThread ... 210
Priorytety dla wątków ... 212
Współdziałanie wątków ... 212
Przykładowy program — wyszukiwanie wielowątkowe ... 215
Podsumowanie ... 223

Rozdział 9. Biblioteki DLL ... 225
Słowo o bibliotekach DLL ... 225
Tworzenie bibliotek DLL ... 226
Eksportowanie procedur i funkcji ... 228
Umieszczanie formularzy w bibliotekach DLL ... 228
Ładowanie procedur z bibliotek ... 230
Ładowanie statyczne ... 230
Ładowanie dynamiczne ... 232
Procedura inicjująco-kończąca ... 234
Umieszczanie zasobów w bibliotece ... 238
Komentarz biblioteki ... 241
Podsumowanie ... 242

Rozdział 10. Pisanie komponentów ... 243
Własne komponenty w Delphi ... 243
Tworzenie własnego komponentu ... 244
Edycja kodu ... 244
Instalacja komponentu ... 245
Przykładowy komponent — TURLabel ... 247
Klasa bazowa dla komponentu ... 247
Obsługa komunikatów ... 247
Metody komponentu ... 248
Ikona dla komponentu ... 251
Demo, czyli wykorzystanie komponentu w aplikacjach ... 251
Komponenty graficzne ... 252
Ogólny zarys klasy dla komponentu TFly ... 253
Kod źródłowy komponentu ... 255
Typy właściwości ... 260
Klauzule default i nodefault ... 262
Pakiety komponentów ... 263
Podsumowanie ... 265

Rozdział 11. Wstęp do programowania w WinAPI ... 267
Podstawowe informacje dotyczące WinAPI ... 267
Pierwszy program w API ... 268
Tworzenie formularza ... 268
Funkcja okienkowa ... 269
Rejestracja klasy ... 270
Tworzenie formy ... 271
Komunikaty i uchwyty ... 273
A co z łańcuchami danych? ... 273
Umieszczenie komponentów w formie ... 275
Obsługa komunikatów dla komponentów ... 278
Tworzenie bardziej zaawansowanych komponentów ... 281
Czcionki ... 283
Obsługa zdarzeń — przypomnienie ... 283
Podsumowanie ... 286

Rozdział 12. Programy internetowe w Delphi ... 287
Z czego korzystać? ... 287
Instalacja pakietu ICS ... 288
Korzystanie z pakietu ICS ... 288
Wykorzystanie protokołu HTTP ... 288
Przesyłanie danych metodą POST ... 291
Wysyłanie poczty poprzez aplikacje ... 295
Przeglądarka internetowa ... 296
Wykorzystanie gniazdek ... 299
Łączenie z komputerem ... 299
Przesyłanie informacji ... 300
Odbieranie informacji ... 301
Piszemy prosty komunikator ... 301
Podsumowanie ... 308

Skorowidz ... 309

1.

Wstęp do programowania w Delphi

W tym rozdziale:

- Opowiem Ci o programowaniu, a w szczególności o Delphi
- Zapoznasz się z interfejsem Delphi
- Napiszesz swój pierwszy program

Ogólnie o programowaniu

Wiele lat temu programista uważany był za „czarodzieja", a sama sztuka pisania programów — za „czarną magię", która znana była tylko nielicznym. Nic dziwnego. Kiedyś dla programisty dostępny był jedynie język niskiego poziomu — Asembler. Programowanie w takim języku polegało na wpisywaniu szeregu poleceń, dzięki którym można było osiągnąć zamierzony efekt. Zaletą programowania w Asemblerze jest duża możliwość manipulowania komputerem — programista ma duże pole manewru. Wadą tego języka jest jednak mało czytelny kod. Asembler nadal ma swoich zwolenników, lecz w dużym stopniu został zastąpiony przez języki wysokiego poziomu, które także oferują wstawianie kodu Asemblera (Delphi, C++). Programy napisane w Asemblerze charakteryzują się bardzo małym rozmiarem pliku (może to być nawet rozmiar poniżej 1 kB!). Od tamtego czasu jednak wiele rzeczy się zmieniło — powstało wiele nowych języków programowania. Wśród nich bardzo popularny jest C++. Jest to język wysokiego poziomu, który jest bardzo potężnym narzędziem — oferuje wiele możliwości. Istnieje wiele kompilatorów języka C++, wśród których najbardziej znane to Microsoft Visual Studio oraz Borland C++ Builder.



Uwaga

Kompilator jest to program, który przekształca (kompiluje) tzw. kod źródłowy do postaci kodu maszynowego, czyli takiego, który zostanie zrozumiany przez komputer.

Kod źródłowy jest to system komend (język), którym posługuje się programista do zaprogramowania pewnych czynności, które mają być wykonane przez komputer. Kod źródłowy musi być zrozumiały dla kompilatora, aby ten mógł przekształcić go na plik wynikowy (aplikację z rozszerzeniem *.exe*).

Ostatnio, bo wcale nie tak dawno, powstał na przykład język skryptowy PHP do tworzenia aplikacji internetowych. Język ten w większości opiera się na C++. Także Perl, który jest również językiem wykonywanym po stronie serwera, zawiera wiele elementów C++. To przemawia samo za siebie — język C++ jest niezwykle popularny, ale o wiele trudniejszy od Delphi. Chcąc programować w C++, masz duży wybór kompilatorów. Jednakże chcąc zapewnić sobie wygodę pracy trzeba zapłacić i to niemało za środowiska typu C++ Builder lub Microsoft Visual C++, które tak jak Delphi są środowiskami wizualnymi.

Sporą popularność zdobył sobie powstały w latach 80. język Pascal. Charakteryzuje się on prostotą obsługi (w porównaniu np. do C++) i szybkim kompilatorem. Jednak język Pascal (i najpopularniejszy jego kompilator — Turbo Pascal) działał w środowisku DOS-a, czyli inaczej mówiąc — napisane w nim programy uruchamiane być mogły w systemie DOS, co w obecnych czasach, gdy na większości komputerów królują systemy Windows, staje się już mało efektywne. Firma Borland postanowiła to zmienić i na podstawie języka Pascal stworzyć nowe narzędzie, które umożliwiałoby pisanie aplikacji na system Microsoft Windows. I tak powstało środowisko, które jest tematem niniejszej książki… Delphi… Trzeba jednak w tym miejscu wyjaśnić jedną rzecz — Delphi nie jest językiem programowania. Wiem, że wielu ludzi na pytanie: „w czym programujesz?", odpowiada — „w Delphi". Delphi jest jedynie wizualnym środowiskiem, ale korzysta z języka Pascal, bardzo często nazywanego językiem *Object Pascal*.

Czym jest Delphi?

Delphi jest narzędziem do szybkiego tworzenia aplikacji (RAD). Jest to środowisko wizualne. A co to oznacza? Uruchom dowolny program działający pod Windows (np. Outlook Express). Widzisz tam rozmaite listy rozwijalne, przyciski i kontrolki. W Delphi możesz to wszystko zrobić jednym kliknięciem myszy. Po prostu tworzenie programu od strony wizualnej jest dziecinnie proste. Nie myśl jednak, że tworzenie programów opiera się tylko na klikaniu myszką. Nie, nie, takie proste to to nie jest. Oczywiście trzeba także pisać różne instrukcje, aby osiągnąć zamierzony efekt. Jak już mówiłem — Delphi jest oparte na Pascalu, więc znając ten język nie będziesz miał większych problemów, aby przenieść się do tego środowiska. Od razu jednak chciałbym powiedzieć, że nie ma kilku kompilatorów Delphi. Delphi jest tylko jedno i trzeba za nie zapłacić. Nie martw się jednak. Istnieje kilka dystrybucji tego narzędzia, które różnią się od siebie ilością dostępnych opcji oraz oczywiście ceną. W obecnej wersji opatrzonej numerkiem 6 najprostsza dystrybucja to Personal Edition, jest ona darmowa, dostępna w Internecie. Można ją ściągnąć ze strony producenta (*www.borland.pl*), zainstalować (wersja instalacyjna

zajmuje ok. 140 MB) i korzystać do woli. Jest jednak pewne ograniczenie. Programy napisane w Delphi 6 PE (Personal Edition) nie mogą być sprzedawane, czyli nie możesz czerpać z nich materialnych korzyści. A i tak Borland poszedł nam na rękę, gdyż w poprzednich wersjach nie było darmowej dystrybucji, dostępnej bez ograniczeń. Ta książka pisania jest na podstawie Delphi 6 PE i wszystkie kody źródłowe będą działać bez żadnych problemów, jeżeli będą kompilowane właśnie w tej wersji.

Po uruchomieniu Delphi nie będziesz mógł korzystać z polskich liter. Aby to umożliwić, będziesz musiał stworzyć nowy klucz w rejestrze Windows. Aby tego dokonać, postępuj według poniższych instrukcji:

1. Z menu *Start* wybierz *Uruchom*.
2. W polu tekstowym wpisz `regedit`.
3. W otwartym edytorze rejestru odnajdź klucz *HKEY_CURRENT_USER\SOFTWARE\Borland\Delphi\6.0\Editor\Options*.
4. Z menu *Edycja* wybierz *Nowy/Wartość ciągu*.
5. Po prawej stronie pojawiła się nowa wartość, nadaj jej nazwę: `NoCtrlAltKeys`.
6. Kliknij na wartości dwukrotnie i w polu tekstowym wpisz cyfrę `1`.

Po zamknięciu edytora rejestru i ponownym uruchomieniu Delphi powinieneś już móc korzystać z polskich znaków.

Proces instalacji

Proces ten jest nieco skomplikowany — szczególnie w wersji Personal (w pozostałych wersjach jest na pewno prostszy). Niech zapomną o Delphi te osoby, które nie mają połączenia z Internetem lub dostępu do niego. Wersja Personal Edition wymaga rejestracji każdej kopii programu przez Internet. Nim to zrobisz, musisz założyć swoje konto, w którym podasz hasło, e-mail i wypełnisz długą ankietę. Nie jest to co prawda przyjemne, ale konieczne. Jeżeli masz już rozpakowany program, przystąp do instalacji. Podczas niej będziesz musiał podać dwa klucze rejestracyjne, które dostaniesz po wypełnieniu ankiety na stronie *http://www.borland.com./delphi/personal/del6personal_keyanddown_steps.html* (po tym będziesz miał możliwość ściągnięcia programu i dostaniesz klucz rejestracyjny — np. r444-h6yqe-rwwuy oraz qk9-5hu). Dzięki temu proces instalacji będzie mógł być kontynuowany. Nie zapomnij podczas podawania swoich danych wpisać poprawnego adresu e-mail, gdyż to na niego wysłany zostanie klucz rejestracyjny. Nie zapomnij także swojego hasła i loginu!

Po zainstalowaniu programu, w *Menu Start* pojawi się skrót do niego. Pokaże się także okienko rejestracji programu. Na początku musisz wybrać, czy masz już założone konto na stronie Borlanda (już masz, gdyż wypełniłeś ankietę), po tym program będzie się próbował połączyć przez Internet z serwerem Borlanda — jeżeli to się powiedzie, będziesz mógł kontynuować pracę z programem. Oczywiście ten proces musisz przeprowadzić tylko raz. Potem możesz korzystać z programu bez żadnych ograniczeń.

Porównanie różnych dystrybucji Delphi 6

W obecnej wersji produkt ten jest rozpowszechniany w trzech dystrybucjach: Personal, Professional, Enterprise. Ta ostatnia jest najbardziej rozbudowana i jak się można domyślić — najdroższa. Nie ma się jednak co przejmować — nawet w najprostszej napiszesz i skompilujesz swój program. Wersja Personal posiada także przykłady, które znajdują się w katalogu *Demos* oraz dokumentację elektroniczną (help). Użytkownik Delphi 6 Personal nie może sprzedawać swoich programów. Wersja Personal nie posiada także komponentów obsługi baz danych. Tylko wersja Enterprise posiada obsługę takich baz danych jak: Paradox, dBase, FoxPro, Access, MySQL, InterBase, Oracle, DB2, BDE. Skoro mowa o komponentach, to w wersji Personal jest ich ponad 85, w wersji Professional ponad 255, a w wersji Enterprise — ponad 300. W wersji Enterprise dostępne są nowe usługi wprowadzone w Delphi 6 — BizSnap i WebSnap.

Porównanie do poprzednich wersji

Jeżeli miałeś już do czynienia z poprzednimi wersjami Delphi, pierwsze, co zwróci Twoją uwagę, to nowe okienko nad *Inspektorem Obiektów*. Nazywa się ono *Object TreeView*. Jest to proste udogodnienie, dające podgląd na formularz i wszystkie umieszczone na nim komponenty. Wszystko jest przedstawione w formie drzewa — klikając konkretny obiekt aktywujemy dany komponent. Ja jednak jestem przyzwyczajony do wersji 5 i nie używam tego wynalazku — po prostu zamykam to okno i rozciągam na całą wysokość ekranu *Inspektora Obiektów*.

Sporo jest zmian graficznych — teraz wszystko wygląda ładniej. Np. po naciśnięciu *Ctrl+spacja* pojawia się, tak jak w poprzednich wersjach, okno z listą możliwych do zastosowania procedur i funkcji. Teraz wszystko jest ładnie pokolorowane — inny kolor mają funkcje, inny procedury, a jeszcze inny zmienne. Wada, jaką przy tej okazji zauważyłem, jest taka, że po wpisaniu np. litery „B" program analizuje wszystkie właściwości, funkcje i następnie pokazuje tylko te na literę „B", które można właśnie wykorzystać. Mnie nie spodobało się, że ta lista jest skracana i widnieją tylko te właściwości na literę „B". W poprzednich wersjach następował w takim przypadku tylko skok do tych metod na literę „B" i nic więcej.

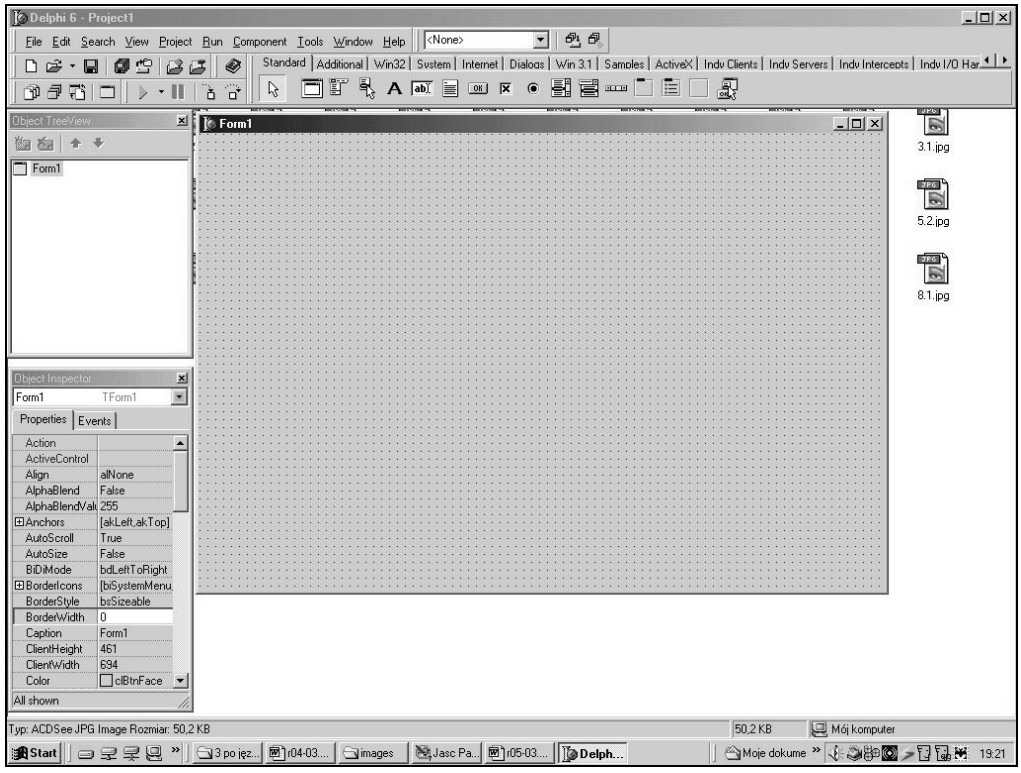
Co do komponentów, to dodanych zostało parę, ale w wersji podstawowej (Personal) nie jest ich za dużo. Nowe komponenty to: `TLabeledEdit` (`Label + Edit`), `TColorBox` (komponent `ComboBox` z możliwością umieszczania kolorowych kwadracików obok pozycji), `TComboBoxEx` (możliwość umieszczania ikon w pozycjach `ComboBox`), `TShellTreeView` (wyświetla drzewo katalogów), `TShellComboBox` (lista rozwijalna katalogów), `TShellListView` (`ListView`, w którym wyświetlony jest podgląd danego katalogu).

Jest troszeczkę zmian, ale nie są one aż tak widoczne, a tym bardziej przydatne. Środowiskiem bardziej szczegółowo zajmiemy się w kolejnych rozdziałach.

Rzut oka na środowisko

Po zainstalowaniu Delphi, w *Menu Start* powinieneś mieć do niego skrót.

Jak wyglada Delphi po uruchomieniu, przedstawiono na rysunku 1.1. Niech Cię na początku nie przeraża ilość ikonek, przycisków, pozycji, opcji — wszystko z czasem stanie się łatwe, uwierz mi. Jak powiedziałem wcześniej, Delphi to środowisko wizualne, a więc także podczas pracy z programem masz podgląd na swój projekt i wiesz, jak będzie on wyglądał po uruchomieniu.



Rysunek 1.1. *Wygląd Delphi*

Paleta komponentów

Na górze masz okienko z różnymi zakładkami. Na tych zakładkach znajdują się różne przyciski. Te przyciski nazywają się *komponentami*.



Uwaga

Komponent — inaczej obiekt. Programowanie w Delphi jest o tyle łatwe, że mamy już gotowe „klocki". Te klocki to właśnie komponenty. Wykonują one za nas różne czynności, dzięki czemu tworzenie aplikacji trwa szybciej.

Firma Borland udostępniła w swoich produktach tzw. *wizualną bibliotekę komponentów*, zwaną *VCL* (*Visual Component Library*). Także wszystkie komponenty (a jest ich w wersji Personal Edition 85) to VCL. Są to różne kontrolki, pola edycyjne — każdy z tych komponentów ma swoją funkcję. Jeżeli spojrzysz na paletę komponentów, zauważysz pewnie obrazki, z którymi nie raz się spotkałeś. Są to, wykorzystywane w prawie każdej aplikacji pod Windows, przyciski, menu itp. Dobrze, na razie nie zawracaj sobie tym głowy — po pewnym czasie wszystko stanie się prostsze i bardziej przejrzyste.

Projektant formularzy

To duże, czyste okno pośrodku Twojego ekranu to właśnie projektant formularzy, zwany potocznie formą lub formatką (od słowa formularz). To jest Twój program. Tak, tak — to jest podgląd Twojej aplikacji. Jest to okno, które możesz przenosić, rozciągać itp. A teraz naciśnij klawisz *F9*. Po tym Delphi uruchomi Twój program. Na razie nawet możesz o tym nie wiedzieć, gdyż z formularza zniknie tylko pomocnicza siateczka. To jest Twój pierwszy program. Co prawda na razie nic on nie robi, bo jest tylko czystym oknem, ale niedługo to się zmieni… Możesz już zamknąć ten program, tak jak zamykasz zwykłe okno — wciskając krzyżyk w prawym górnym rogu. Teraz nasz program unowocześnimy, dodając do niego jakiś komponent — na przykład przycisk. Na palecie komponentów odszukaj ikonę przycisku (ang. *Button*). Kiedy kursor znajdzie się nad tą ikoną, naciśnij lewy przycisk myszy. Teraz najedź kursorem nad Twój formularz i w obojętnie którym miejscu wciśnij ponownie lewy klawisz myszki. Zobaczysz na formularzu nowy komponent — komponent `Button`. Moje gratulacje! Właśnie umieściłeś swój pierwszy obiekt.

Teraz nadszedł czas na ponowne uruchomienie programu klawiszem *F9*. Jak widzisz, program ponownie się uruchomił, tyle że w oknie znajduje się nowy obiekt. Możesz sobie w niego poklikać — i tak się nic nie stanie, gdyż nie zaprogramowaliśmy obsługi jego zdarzenia. Już Ci się to spodobało? A będzie jeszcze lepiej…

Ponownie zamknij program i powróć do jego edycji. Taką kontrolkę możesz dowolnie przesuwać w obrębie formularza, rozciągać chwytając za rogi tego obiektu. Możesz z nim zrobić wszystko tak, aby dopasować go do wyglądu Twojego programu. Usuwanie następuje oczywiście po naciśnięciu klawisza *Delete*. Możesz także taki obiekt wycinać i wklejać. Po zaznaczeniu go naciśnij kombinację klawiszy *Ctrl+X* (wycinanie). Komponent zniknął. Teraz, aby ponownie umieścić go na formularzu, naciśnij kombinację *Ctrl+V* (wklej). W ten sposób możesz na formularzu postępować z dowolnymi komponentami. Jeżeli chcesz umieścić jeszcze jeden komponent `TButton`, postępuj tak jak wcześniej. Zauważ, że tym razem etykieta na komponencie zmieniła się i napis na obiekcie to `Button2`. Ten napis oczywiście można zmieniać, ale tym zajmiemy się nieco później.

Inspektor Obiektów

Inspektor Obiektów jest tym oknem po lewej stronie ekranu. Jak widzisz, zawiera ono szereg pozycji. Te pozycje to *właściwości*.



Uwaga

Właściwości określają zachowanie i wygląd komponentu. Za pomocą właściwości można na przykład ustalić rozmiar i krój używanej przez komponent czcionki.

Kliknij raz lewym przyciskiem w obrębie formularza. Spowoduje to odznaczenie wszelkich komponentów, a forma stanie się aktywna. W *Inspektorze Obiektów* wyświetlą się właściwości formularza głównego programu. Jak widzisz, ta lista jest dość długa, co oznacza, iż będziemy mogli w większym stopniu określić wygląd naszego programu. Możemy na przykład określić stan okna po uruchomieniu. Po uruchomieniu programu okno może być zminimalizowane, zmaksymalizowane lub pozostać normalne (domyślna wartość). Odnajdź więc właściwość `WindowState` (powinna być na samym dole). W lewej części masz wypisane nazwy właściwości, a po prawej ich wartości. Zaznacz pozycję `WindowState` klikając ją raz. Po prawej stronie pojawi się strzałka, po której naciśnięciu rozwinie się lista z możliwymi do zastosowania opcjami. Wybierz opcję `wsMinimized`. Od tej pory po uruchomieniu programu będzie on domyślnie minimalizowany do paska stanu. Sprawdź to, ponownie uruchamiając program klawiszem *F9*.

Kolejny przykład. W *Inspektorze Obiektów* znajduje się także właściwość `Caption`, odnajdź ją. Określa ona tekst, który wpisany jest na belce tytułowej programu. Po zaznaczeniu tej właściwości, wpisz w polu edycyjnym, znajdującym się po prawej stronie, jakiś tekst, który będzie widniał na pasku tytułu programu. Wpisz np. `Mój pierwszy program` i potwierdź to naciskając klawisz *Enter*. Jak widzisz, wszelkie zmiany zmieniają wygląd formularza. To jest duża zaleta Delphi — nie musisz uruchamiać programu, aby sprawdzić efekt wprowadzonych zmian.

Inspektor Obiektów posiada jeszcze jedną zakładkę — *Events* (zdarzenia), która zawiera listę zdarzeń.



Uwaga

Zdarzenie jest to czynność, która będzie występować po zajściu określonych wydarzeń (np. po kliknięciu myszką w obszarze programu, poruszeniu myszką, wciśnięciu klawisza).

Jeżeli klikniesz na tę zakładkę, ujrzysz listę możliwych do wykorzystania zdarzeń. Np. *OnActivate* (występuje podczas pokazania się okna). Nie będziemy się tym teraz zajmować, gdyż aby zaprogramować czynności, które mają być wykonane po zajściu określonego zdarzenia, należy znać chociażby podstawy języka programowania.

Edytor kodu

Programowanie w Delphi nie opiera się tylko na budowaniu aplikacji z klocków — trzeba także się trochę wysilić i napisać kod, który będzie zrozumiały dla kompilatora. Cały kod wpisuje się w edytorze kodu. Jeżeli masz otwarty projekt w Delphi, wciśnij *F12* — ujrzysz kod źródłowy swojej aplikacji, a dokładniej — kod źródłowy formularza (w Delphi możesz stworzyć kilka formularzy, a zarazem kilka plików źródłowych). Na razie się nie przejmuj tekstem, który już tam widnieje, teraz nie jest to ważne. Wszystkiego dowiesz się w kolejnym rozdziale. Na razie musisz tylko wiedzieć, że coś takiego istnieje.

Piszemy pierwszy program

Nadszedł już czas, aby napisać pierwszy poważniejszy program. Co prawda w poprzednich częściach tego rozdziału uruchamiałeś już swój program, ale czy to można nazwać pisaniem? Przecież nie napisałeś ani jednej linijki kodu! Będzie to najprostszy programik wykorzystujący VCL, ale w końcu od tego się zaczyna, prawda? Będzie on po naciśnięciu przycisku wyświetlał okno informacyjne.

Tworzenie projektu

Z menu *File* wybierz opcję *New Application*. Najpierw Delphi spyta Cię, czy nie zachować przypadkiem projektu, nad którym teraz pracujesz. Naciśnij klawisz *No*, gdyż na razie nie chcesz zapisywać swojego kodu. Na ekranie ponownie ujrzysz czysty formularz, bez żadnych komponentów. Na początek trzeba zmodyfikować pewne właściwości formularza. Odnajdź w *Inspektorze Obiektów* właściwość `Name`. Właściwość ta określa nazwę danego obiektu. Nazwa w żadnym wypadku nie może się powtarzać. Wpisz tutaj np. `MainForm` (główna forma). Teraz we właściwości `Caption` wpisz tytuł okna — np. `Program Powitanie`. Dobrze, mamy już nazwę oraz tytuł okna. Nie musimy jednak mieć aż tak dużego formularza. Można go trochę zmniejszyć. Jest to proste. Wystarczy, że chwycisz za boki formy i odpowiednio je zmniejszysz. Równie dobrze można by było w *Inspektorze Obiektów*, we właściwościach `Width i Height`, wpisać odpowiednie wartości. Na formularzu umieść teraz przycisk. Na wszelki wypadek jeszcze raz objaśnię jak to się robi. Kliknij na palecie komponentów w miejscu obrazka z przyciskiem. Następnie kursor myszy ustaw gdzieś nad formularzem. W miejscu, w którym ma znajdować się komponent, naciśnij lewy przycisk myszy. Przycisk zostaje umieszczony. Teraz, gdy komponent jest zaznaczony, przyjrzyjmy się jego właściwościom w *Inspektorze Obiektów*. Odszukaj właściwość `Name` i nadaj komponentowi jakąś sensowną nazwę — np. `myButton`. Teraz we właściwości `Caption` nadaj jakąś etykietę dla przycisku. Ja wpisałem tam `Wciśnij mnie`. Jak już pewnie zauważyłeś — dla formularza i dla komponentu `Button` istnieją różne właściwości, które można swobodnie edytować.

Edycja kodu źródłowego

Mamy na formularzu umieszczony przycisk, ale co dalej? Należałoby zaprogramować zdarzenie kliknięcia w obrębie przycisku. Nie będzie to trudne. Podwójnie kliknij przycisk. Delphi przeniesie Cię do edytora kodu i ustawi kursor w środku tzw. *procedury*. Tym zajmiemy się później — teraz kod wygląda tak:

```
procedure TMainForm.myButtonOnClick(Sender: TObject);
begin
end;
```

To, co ma być wykonane po wciśnięciu przycisku, należy wpisać pomiędzy słowa `begin` i `end`. Wpisz tam jedną instrukcję:

```
ShowMessage('Witaj! Jestem Twoim pierwszym programem!');
```

Czyli cały kod powinien wyglądać tak:

```
procedure TMainForm.myButtonOnClick(Sender: TObject);
begin
  ShowMessage('Witaj! Jestem Twoim pierwszym programem!');
end;
```

Poznałeś właśnie pierwsze polecenie Delphi: `ShowMessage()`. To polecenie powoduje wyświetlenie okienka informacyjnego Windows. Tekst, który ma być zawarty w tym oknie, wpisujemy w apostrofach.

Teraz możesz już uruchomić program klawiszem *F9*. Delphi skompiluje projekt oraz wyświetli okno programu. Jeżeli kompilacja się nie powiedzie i Delphi wyświetli jakieś błędy, popraw kod. W takim przypadku kompilator podświetli linię, w której znajduje się błąd. Na tym etapie żaden błąd nie powinien się pojawić, lecz jeżeli tak się stanie, porównaj kod z moim i w razie czego popraw. Zwróć także uwagę na średnik na końcu każdej linii (oprócz `begin`). Jeżeli jednak program się uruchomił, naciśnij przycisk — kod wpisany w programie powinien zadziałać, czego rezultatem będzie pojawienie się okienka z tekstem `Witaj! Jestem Twoim pierwszym programem!` Gratulacje! Napisałeś właśnie swój pierwszy program.

Zapisywanie projektu

Wypadałoby projekt zapisać i zachować jako pamiątkę w katalogu *Mój pierwszy program*. Ja mam u siebie na dysku katalog *Delphi/programy* i to właśnie w nim trzymam swoje projekty. Tobie radzę zrobić tak samo. Także w katalogu *Delphi/programy* stwórz katalog *Pierwszy*. Teraz w Delphi wybierz polecenie *File/Save All*. Spowoduje to zapisanie całego projektu. Najpierw będziesz musiał podać katalog, w którym Delphi ma zapisać pliki. Teraz w okienku wpisz nazwę pliku — formularza. Ja wpisałem `MainForm`. Później będziesz musiał podać nazwę projektu. Ja wpisałem `Pierwszy`. Następnie przejdź do katalogu, gdzie projekt został zapisany. Jak widzisz, powstało tam kilka plików. W Delphi wybierz opcję *Project/Build All*. Po wybraniu tej opcji Delphi skompiluje projekt do postaci pliku wykonywalnego EXE. Zajrzyj teraz do katalogu z programem — po kompilacji w katalogu powstało kilka kolejnych plików. W tym plik EXE, który jest naszą aplikacją. Tak więc Delphi stworzyło zewnętrzny plik wykonywalny, który możesz dać np. koledze. Oczywiście nie musisz dołączać pozostałych plików, które znajdują się w katalogu.

- **.pas* — jest to plik zawierający kod źródłowy formularza.
- **.dmf* — w tym pliku przechowywane są dane dotyczące formularza (np. jakie komponenty są umieszczone na formularzu, jaka jest ich pozycja).
- **.dcu* — jest to skompilowany plik formularza.
- **.dpr* — plik główny programu.
- **.cfg* — zawiera opcje Delphi, a konkretnie — ustawienia kompilatora.
- **.dof* — opcje projektu.
- **.res* — są to tzw. zasoby. O zasobach będziemy mówili w dalszej części książki.

Plik *.*res* to tak zwane zasoby, czyli np. ikona, która ozdabia plik wykonywalny EXE. Jeżeli chcesz dać koledze kod źródłowy programu, to wystarczą pliki PAS, DMF, DPR i ewentualnie RES. Pozostałe pliki możesz usunąć — Delphi utworzy je ponownie podczas kompilacji. W zależności od ustawień, Delphi podczas zapisywania może tworzyć także kopie zapasowe plików, których rozszerzenie jest poprzedzone znakiem ~. Ja jednak tę opcję wyłączam. Żeby wyłączyć tworzenie plików zapasowych w Delphi, z menu *Tools* wybierasz *Editor Options*, a następnie klikasz na zakładkę *Display*. Na niej odznaczasz pozycje *Create Backup Files*.

Bardzo widoczna wada Delphi to rozmiary plików wykonywalnych. Zobacz rozmiar pliku EXE — zwykła forma, przycisk i kilka linii kodu źródłowego zajmuje ponad 200 kB. Przyznasz, że nie jest to mało. Wraz z każdą kolejną wersją, rozmiar pliku wykonywalnego rośnie. Są jednak sposoby, aby temu zapobiec. Istnieją bowiem programy kompresujące, po których uruchomieniu rozmiar pliku EXE zostanie zmniejszony. Najpopularniejszym jest chyba UPX, który potrafi zmniejszyć objętość oryginalnego pliku nawet o 60%.

Podsumowanie

W tym rozdziale nauczyłeś się posługiwać narzędziem, jakim jest Delphi. Poznałeś trochę to środowisko, lecz nie poznałeś samego języka, którym Delphi się posługuje, czyli Pascala (programiści używają w stosunku do niego innej nazwy — *Object Pascal*). Zapraszam do przeczytania kolejnego rozdziału, w którym zapoznasz się z tym językiem. Bo programowanie w Delphi nie opiera się tylko na budowaniu…