

## 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

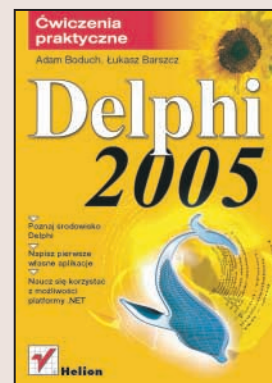
ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Delphi 2005. Ćwiczenia praktyczne

Autor: Adam Boduch  
ISBN: 83-7361-779-5  
Format: B5, stron: 148



- Poznaj środowisko Delphi
- Napisz pierwsze własne aplikacje
- Naucz się korzystać z możliwości platformy .NET

Środowisko programistyczne Delphi od dawna cieszy się popularnością wśród twórców oprogramowania. Od pierwszych wersji oferowało możliwość wizualnego tworzenia aplikacji za pomocą „klocków” – komponentów, łącząc to z prostotą języka Pascal. Teraz, w najnowszej wersji, Delphi pozwala na tworzenie aplikacji dla dynamicznie rozwijającej się platformy .NET. Dzięki temu znacznie poszerzył się zakres możliwych zastosowań flagowego produktu firmy Borland. Za jego pomocą możemy dziś tworzyć nie tylko aplikacje, ale również usługi sieciowe oraz dynamiczne strony WWW.

„Delphi 2005. Ćwiczenia praktyczne” to książka dla tych, którzy chcą poznać podstawy budowania aplikacji w najnowszej wersji środowiska Delphi. Przedstawia zasady korzystania z narzędzi oferowanych przez Delphi oraz sposoby zastosowania ich podczas tworzenia oprogramowania. Czytając tą książkę, poznasz język ObjectPascal, zasady programowania obiektowego oraz metody łączenia aplikacji z bazami danych. Nauczysz się także tworzyć witryny WWW w języku ASP.NET oraz usługi sieciowe.

- Podstawowe informacje o środowisku Delphi 2005 oraz platformie .NET
- Elementy języka Delphi
- Korzystanie z narzędzi oferowanych przez Delphi 2005
- Komponenty bazodanowe
- Tworzenie aplikacji opartych na bazach danych
- ASP.NET i Web Services

**Zrób pierwszy krok w programowaniu –  
poznaj i wykorzystaj możliwości, jakie oferuje Delphi 2005**



# Spis treści

<b>Wstęp</b> .....	<b>7</b>
<b>Rozdział 1. Zaczynamy</b> .....	<b>9</b>
Rzut okiem na środowisko .....	9
Projektant formularzy .....	11
Edytor kodu.....	12
Widok strukturalny .....	13
Inspektor obiektów.....	14
Jak to działa? .....	14
Podsumowanie.....	15
<b>Rozdział 2. Podstawy platformy .NET</b> .....	<b>17</b>
Czym jest .NET .....	17
.NET Framework .....	18
Podstawowa terminologia .....	18
Produkty dla programistów.....	19
Instalacja .....	19
Przegląd technologii .NET Framework .....	20
Środowisko Common Language Runtime (CLR).....	20
Podzespoły .....	21
Działanie CLR .....	23
Biblioteka klas .....	24
Common Language System (CLS) .....	24
Common Type System (CTS).....	24
Rodzaje aplikacji w Delphi 2005.....	25
Biblioteka VCL.....	25
Windows Forms .....	26
Web Forms.....	27
Podsumowanie.....	27
<b>Rozdział 3. Język Delphi</b> .....	<b>29</b>
Pierwszy program .....	29
Uruchamiamy program .....	32
Zapisywanie projektu.....	32
Otwieranie i zamykanie projektu .....	32

Piszemy kod programu .....	34
Aplikacje konsolowe .....	35
Wpisywanie tekstu w konsoli .....	37
Odczytywanie tekstu z konsoli.....	38
Pętle .....	39
Pętla for.....	39
Pętla repeat.....	41
Pętla while.....	41
Funkcje Break i Continue .....	42
Instrukcje warunkowe.....	43
Procedury.....	47
Funkcje .....	49
Moduły .....	52
Dołączanie modułu do aplikacji.....	53
Dostępność zmiennych .....	54
Tablice .....	56
Tablice dynamiczne .....	58
Rekordy .....	59
Klasy.....	61
Tworzenie własnych klas .....	62
Tworzenie instancji klasy.....	63
Konstruktory i destruktory .....	67
Sekcje klasy .....	68
Dziedziczenie .....	71
Podsumowanie.....	71
<b>Rozdział 4. Nowości w Delphi 2005 .....</b>	<b>73</b>
IDE .....	73
Historia.....	74
Widok strukturalny .....	74
Funkcja Help Insight.....	75
Wyniki wyszukiwania.....	75
Wyszukiwanie odwołań .....	76
Wyróżnianie błędów .....	76
Edycja synchroniczna .....	76
Edytor kodu.....	76
Debugowanie .....	78
Refaktoryzacja .....	79
Deklarowanie zmiennych.....	79
Deklarowanie pól .....	79
Wyodrębnianie metod.....	79
Wyodrębnianie łańcucha znaków .....	79
Zmiana nazw symboli .....	80
Importowanie przestrzeni nazw .....	80
Nowości języka Delphi.....	80
Pętla for..in..do.....	80
Wplatanie funkcji.....	81

Bazy danych .....	82
Obsługa baz danych ADO.NET .....	82
Ulepszenia narzędzi projektowych .....	82
Migracja danych.....	83
Ulepszenia eksploratora baz danych .....	83
Podsumowanie.....	83
<b>Rozdział 5. Bazy danych .....</b>	<b>85</b>
Czym są bazy danych? .....	85
Rodzaje baz danych .....	86
Bazy proste .....	86
Relacyjne bazy danych.....	86
Bazy danych typu klient-serwer.....	87
Borland Database Engine .....	87
Sterowniki bazy danych .....	88
Zbiory danych .....	88
Wykorzystanie BDE w Delphi .....	89
Poruszanie się po rekordach.....	90
Praca z komponentami.....	92
Modyfikacja zawartości .....	96
Komponenty dbExpress.....	98
Architektura klient-serwer .....	99
Narzędzia .....	100
Łączenie z serwerem.....	100
Tworzenie tabel.....	104
Tworzenie rekordów .....	105
Odczyt rekordów.....	106
Kasowanie danych .....	111
Uaktualnianie danych.....	111
Podsumowanie.....	112
<b>Rozdział 6. ASP.NET.....</b>	<b>113</b>
Dynamiczne strony WWW.....	113
ASP.NET .....	114
Narzędzia.....	114
Edytor.....	115
Serwer .....	115
Instalacja ASP.NET .....	116
ASP.NET w Delphi .....	118
Pierwszy projekt.....	119
Jak to działa?.....	120
Poruszanie się po interfejsie.....	121
Sesje .....	124
Podsumowanie.....	128
<b>Rozdział 7. Usługi sieciowe .....</b>	<b>129</b>
Wprowadzenie .....	129
Co to jest usługa sieciowa? .....	129
Działanie usług sieciowych.....	130
Komunikacja z usługami.....	130

---

Opis usługi Web .....	131
Udostępnianie usługi .....	131
Używanie usług sieciowych .....	131
Wykorzystujemy wyszukiwarkę google.com .....	132
Opcje interfejsu API .....	132
Interfejs aplikacji .....	133
Ładowanie usługi sieciowej .....	133
Korzystanie z usługi Web .....	134
Tworzenie usług sieciowych.....	137
Pisanie kodu .....	138
Podgląd usługi sieciowej.....	139
Usługa Web na stronie ASP.NET.....	142
Bezpieczeństwo usług sieciowych.....	143
Podsumowanie.....	145

# Rozdział 3.

---

# Język Delphi

Delphi (wcześniej Object Pascal) to język programowania wykorzystywany w środowisku o tej samej nazwie. W niniejszym rozdziale omówiono podstawowe elementy tego języka, charakterystyczne dla większości języków programowania.

## Pierwszy program

Na początek zbudujmy prosty program. Celem naszego pierwszego ćwiczenia będzie umieszczenie napisu w oknie programu. Za wyświetlanie takiego prostego napisu odpowiada komponent `TLabel`.

### Ćwiczenie 3.1.

---

*Aby umieścić komponent `TLabel` w oknie naszego programu, wykonaj następujące czynności:*

1. Uruchom program Delphi.
2. Z menu *File* wybierz polecenie *New*, a następnie *VCL Application — Delphi for .NET*.
3. Na palecie narzędzi odnajdź komponent `TLabel` (rysunek 3.1).

#### Rysunek 3.1.

*Przycisk `TLabel`  
na Palecie narzędzi*



4. Kliknij go.
5. Kliknij formularz.
6. Komponent powinien się pojawić w miejscu kliknięcia (rysunek 3.2).

**Rysunek 3.2.**

Komponent TLabel  
na formularzu



Na formularzu powinien widnieć napis Label1.

---

Napis można zmienić za pomocą *Inspektora obiektów*.

**Ćwiczenie 3.2.**

Aby zmienić tekst etykiety:

1. Kliknij jednokrotnie obiekt (komponent) TLabel — powinien zostać zaznaczony. Na jego obrzeżach pojawiają się symbole zaznaczenia (rysunek 3.2).
2. Odszukaj w Inspektorze obiektów pozycję Caption. Określa ona tekst wyświetlany w komponencie.
3. Kliknij jednokrotnie pozycję Caption — powinna zostać zaznaczona (rysunek 3.3).

**Rysunek 3.3.**

Właściwość Caption  
komponentu



Lista podzielona jest na dwie kolumny. Pozycje po lewej określają nazwę *właściwości* komponentu, natomiast w kolumnie po prawej znajdują się *wartości* owej właściwości. Po zaznaczeniu właściwości nadaj jej wartość — po prostu wpisz np. słowo Witaj! i naciśnij *Enter*.

---

W tym momencie zmieniłeś napis na etykiecie. Nie sądzisz jednak, że jest on trochę za mały w stosunku do formularza? Nie ma problemu — można go łatwo powiększyć; temu właśnie poświęcimy nasze kolejne ćwiczenie.

**Ćwiczenie 3.3.**

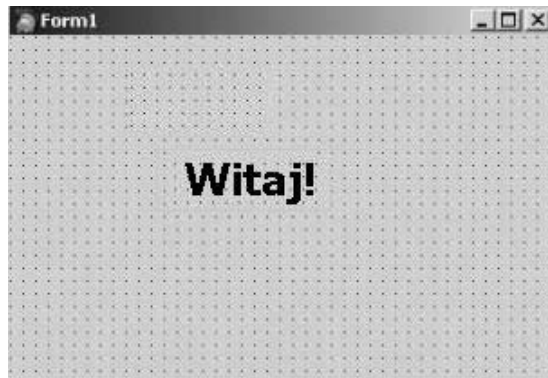
Aby powiększyć napis w etykiecie TLabel:

1. Ponownie zaznacz komponent TLabel, a w Inspektorze obiektów odszukaj pozycję Font (znajdującą się w sekcji *Visual*). Określa ona, jaką czcionką jest wyświetlany nasz napis.
2. Po lewej stronie właściwości Font — w odróżnieniu od Caption — znajduje się ikona ze znakiem plusa. Symbol ten służy do rozwijania listy następnych właściwości związanych z czcionką. Kliknij zatem tę ikonkę, aby wyświetlić kolejne dane.
3. Odnajdź właściwość Size i nadaj jej wartość 20 (postępuj tak, jak w ćwiczeniu 3.2).
4. Na formularzu powinieneś zobaczyć podgląd napisu, czyli jego wersję napisaną czcionką o rozmiarze 20 punktów.

5. Wypadałoby jeszcze zmienić krój czcionki. W tym celu posłuż się właściwością Name — zaznacz ją.
6. Po zaznaczeniu po prawej stronie powinien zostać wyświetlony przycisk ze strzałką. Po jego naciśnięciu ukaże się lista rozwijalna z dostępnymi w systemie czcionkami. Wybierz np. Arial Black.

Nasz napis wygląda już o wiele lepiej — patrz rysunek 3.4.

**Rysunek 3.4.**  
Etykieta  
na formularzu



Zaznaczając komponent jednym kliknięciem myszy, możesz go przesuwać, rozciągać, chwytając za rogi itp.

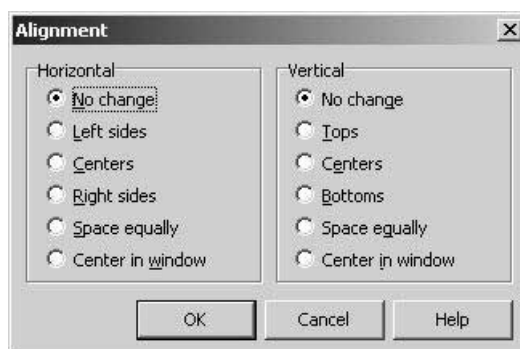
Możesz także to zrobić za pomocą dostępnych w Delphi funkcji. W tym celu wykonaj poniższe ćwiczenie.

#### Ćwiczenie 3.4.

*Aby przenieść komponent:*

1. Zaznacz go.
2. Kliknij go prawym przyciskiem myszy i z menu *Position* wybierz polecenie *Align*. Zostanie otwarte okienko (rysunek 3.5) służące do wyrównywania położenia obiektu.

**Rysunek 3.5.**  
Okno do ustalania  
położenia





3. Okienko składa się z dwóch części — *Horizontal* (położenie w poziomie) oraz *Vertical* (położenie w pionie).
4. W obu częściach zaznacz pozycję *Center in Window* i naciśnij *OK*.

Nasz napis został umieszczony dokładnie w środku formularza.

---

## Uruchamiamy program

### Ćwiczenie 3.5.

---

*Aby uruchomić program:*

1. Wystarczy, że z menu *Run* wybierzesz polecenie *Run*. Nasz projekt zostanie wówczas skompilowany, po czym zostanie uruchomiony program. Tę samą czynność możesz wykonać, korzystając z przycisku — skrótu na pasku narzędziowym Delphi (rysunek 3.6).

**Rysunek 3.6.**  
*Uruchamianie programu*



Można także używać skrótu klawiaturowego — do uruchamiania służy klawisz *F9*. Jest to chyba najwygodniejszy sposób.

## Zapisywanie projektu

### Ćwiczenie 3.6.

---

*Aby zapisać projekt:*

1. Z menu *File* wybierz polecenie *Save All*.
  2. Zostanie otwarte okno, w którym należy wskazać katalog do zapisu oraz wpisać nazwę pliku. Utwórz więc na dysku jakiś katalog — np. *Pierwszy program* — i przejdź do niego.
  3. Zostaniesz poproszony o podanie nazwy formularza — wpisz np. *MainFrm*. Następnie należy podać nazwę projektu, czyli np. *Pierwszy*.
- 

## Otwieranie i zamykanie projektu

Kolejne ćwiczenie polegać będzie na otwarciu oraz zamknięciu projektu, a następnie na ponownym jego uruchomieniu.

### Ćwiczenie 3.7.

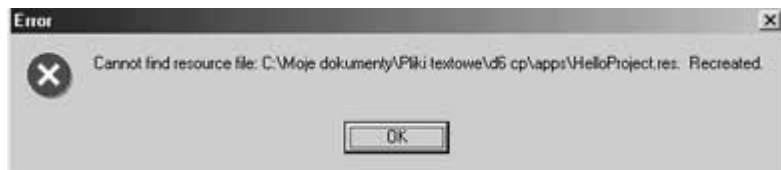
Aby otworzyć istniejący projekt:

1. Z menu *File* wybierz polecenie *Close All*.
2. Projekt, nad którym pracowałeś, zostanie zamknięty. Jeżeli od czasu ostatniego zapisu nastąpiły jakieś zmiany, zostaniesz zapytany, czy chcesz ponownie zapisać projekt. Jeżeli wszystko się powiedzie, z ekranu zniknie okno formularza, a Inspektor obiektów stanie się nieaktywny.
3. Przejdź do katalogu z programem, gdzie zapisałeś swój projekt. Usuń z niego pliki o rozszerzeniach *\*.cfg*, *\*.dof*, *\*.res*, *\*.bdsproj* i ewentualnie wszystkie pliki z tyldą (~) w rozszerzeniu. W katalogu powinny pozostać jedynie trzy pliki.
4. Otwórz ponownie ten projekt. W tym celu z menu *File* wybierz polecenie *Open Project*. Wskaż ścieżkę do pliku *\*.dpr*. Nasz projekt zostanie teraz otwarty.

Program Delphi wyświetli teraz informację, że w katalogu z programem nie znaleziono pliku *\*.res* (rysunek 3.7). To nic! Zignoruj ten komunikat, naciskając przycisk *OK* — Delphi otworzy projekt nawet bez tego pliku!

**Rysunek 3.7.**

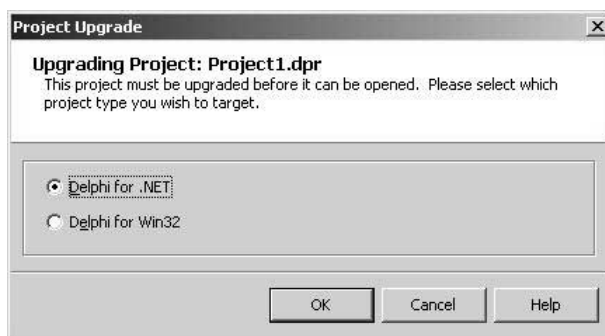
*Ostrzeżenie*



Jeżeli usuniesz plik *\*.bdsproj*, projekt zostanie otwarty także bez tego pliku, ale najpierw zostaniesz zapytany, czy była to aplikacja dla .NET czy Windows 32 (rysunek 3.8). Okno to pojawi się także w przypadku próby otwarcia projektu utworzonego w poprzednich wersjach Delphi.

**Rysunek 3.8.**

*Typ aplikacji*



Od tej pory możesz ponownie pracować nad swoim projektem.



Chcąc skompilować program, niekoniecznie musisz go uruchamiać. Do samej kompilacji służy polecenie *Compile* z menu *Project* (skrót klawiaturowy — *Ctrl + F9*).

# Piszemy kod programu

Tworzenie programów nie polega wyłącznie na układaniu komponentów na formularzu. Aby program był bardziej funkcjonalny, należy także napisać odpowiednie polecenia (stworzyć kod źródłowy).

## Ćwiczenie 3.8.

*Aby wpisać instrukcje języka programowania:*

1. Na formularzu umieść komponent `TButton` (przycisk).
2. Oprócz przycisku umieść także komponent `TLabel`. W naszym programie po naciśnięciu przycisku zmieniany będzie tekst widniejący w etykiecie.
3. Przejdź do zakładki *Events* (zdarzenia) Inspektora obiektów i znajdź `OnClick`.
4. Kliknij to pole dwukrotnie (dla zdarzenia `OnClick` można wykonać to samo, klikając dwukrotnie komponent). Zostaniesz przeniesiony do *Edytora kodu* i ujrzysz taki fragment:

```
procedure TForm1.Button1Click(Sender: TObject);
begin

end;
```

5. Interesujący nas kod źródłowy wpisz pomiędzy słowa `begin` i `end`:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Label1.Caption := 'Nowy tekst...';
end;
```

Uruchom teraz program klawiszem *F9* i sprawdź jego działanie. Po naciśnięciu przycisku tekst na etykiecie powinien zostać zmieniony. W tym kodzie zmieniłem po prostu wartość właściwości `Caption` etykiety (`Label`). `Label1` oznacza nazwę komponentu; można ją łatwo zmienić.

## Ćwiczenie 3.9.

*Aby zmienić nazwę komponentu:*

1. Za pomocą klawisza *F12* możesz przełączać się z edytora kodu do formularza i odwrotnie. Zrób to teraz; powinien zostać wyświetlony formularz.
2. Zaznacz komponent `TLabel`.
3. W Inspektorze obiektów odnajdź właściwość `Name` i zmień ją. Wartość `Label1` zastąp np. słowem *Etykieta*.
4. Spróbuj ponownie uruchomić program klawiszem *F9*. Nie udało się? Został zapewne wyświetlony edytor kodu i podświetlony został wiersz, w którym znajduje się błąd — tak, jak na rysunku 3.9.

**Rysunek 3.9.**

Błąd podświetlony przez edytor Delphi

```
27 procedure TForm1.Button1Click(Sender: TObject);  
28 begin  
29   label1.Caption:='Nowy tekst...';  
30 end;  
31  
32 end.
```

Kompilator informuje Cię, że w podświetlonym wierszu gdzieś jest błąd? Gdzie? Na dole pojawiło się okienko z napisem [Error] Unit1.pas(29): E2003 Undeclared identifier: 'label1'. Kompilator informuje Cię w ten sposób, że w wierszu 29. występuje nieznane słowo label1. Ciąg label1 oznaczał bowiem nazwę komponentu, a tę nazwę właśnie zmieniliśmy na Etykieta. Teraz, żeby wszystko działało, należy zamiast słowa label1 wpisać Etykieta.

```
Etykieta.Caption := 'Zmieniłem tekst...';
```

Kolejne ćwiczenie będzie polegać na zmianie nazwy przycisku oraz formularza.

**Ćwiczenie 3.10.**

*Aby zmienić nazwę formularza:*

1. Zaznacz komponent Button, klikając go jednokrotnie.
2. Tak, jak w poprzednim przypadku, odszukaj właściwość Name i zmień ją na Przycisk. Zwróć uwagę, że również właściwość Caption zmieniła się i ma teraz wartość Przycisk.
3. Jeśli klikniesz jednokrotnie w obrębie formularza, usuniesz zaznaczenie wszystkich komponentów, a zaznaczony zostanie sam formularz. Również właściwości formularza mogą ulec zmianie.
4. Zmienimy właściwości Caption oraz Name formularza. Odszukaj je. W pierwszym przypadku (Caption) wpisz Mój Drugi Program. Zwróć uwagę na odzwierciedlenie tej modyfikacji w formularzu. Napis na pasku tytułowym formularza uległ zmianie.
5. Zmień właściwość Name, wpisując MainForm.

## Aplikacje konsolowe

Przez moment nie będziemy używać formularza ani żadnych komponentów, lecz korzystać z konsoli.



Konsola, inaczej zwana wierszem poleceń lub — w systemach Windows — trybem DOS, jest interfejsem, w którym polecenia są wpisywane z klawiatury, a prezentacja wyników przez komputer odbywa się w trybie tekstowym.

Na początek stworzymy projekt, który nie będzie zawierał formularza.

**Ćwiczenie 3.11.**

*Aby utworzyć aplikację konsolową:*

1. Z menu *File* wybierz polecenie *New*, a następnie *Other*.
2. Z drzewa po lewej stronie wybierz *Delphi Projects*.
3. W oknie po prawej stronie wybierz *Console Application*.

Po wykonaniu tych czynności Delphi utworzy nowy projekt; zawartość edytora kodu będzie wyglądać tak, jak to przedstawiono na rysunku 3.10.

**Rysunek 3.10.**

*Edytor kodu*

```
1  program Project2;  
2  
3  {$APPTYPE CONSOLE}  
4  
5  uses  
6     SysUtils;  
7  
8  begin  
9     { TODO -oUser -cConsole Main : Insert code here }  
10  end.  
11
```

Przed blokiem `begin` znajduje się dyrektywa `{$APPTYPE CONSOLE}`. Dzięki tej instrukcji program zostanie uruchomiony jako konsola. Słowo `end.` (z kropką na końcu) oznacza koniec programu. Instrukcje do wykonania programu zawarte są zawsze pomiędzy słowami `begin` i `end.`

Zmodyfikujemy nasz program tak, aby rzeczywiście wykonywał jakieś operacje.

**Ćwiczenie 3.12.**

*Aby utworzyć prostą aplikację konsolową:*

1. Zmodyfikuj kod programu do następującej postaci:

```
program console;  
uses  
    SysUtils;  
{$APPTYPE CONSOLE}  
  
begin  
    Readln;  
end.
```

W wyniku użycia polecenia `Readln` program czeka, aż użytkownik wciśnie klawisz *Enter*.

2. Uruchom program, aby sprawdzić jego działanie.

## Wpisywanie tekstu w konsoli

Dotychczas nasz program nie wykonuje żadnych operacji — tzn. uruchamia się (to już coś!) i czeka, aż naciśniesz klawisz *Enter*. Zmodyfikujmy go tak, aby wyświetlał jakiś tekst.

### Ćwiczenie 3.13.

---

*Aby wyświetlić tekst w konsoli:*

1. Zmodyfikuj program do takiej postaci:

```
program console;
uses
  SysUtils;
{$APPTYPE CONSOLE}
begin
  WriteLn('Cześć! Jestem aplikacją konsolową! Zamknij mnie klawiszem Enter!');
  ReadLn;
end.
```

2. Uruchom aplikację.

Aplikacja zostanie teraz uruchomiona i wyświetlony zostanie tekst: Cześć! Jestem aplikacją konsolową! Zamknij mnie klawiszem Enter!.

---

Jak już się zdążyłeś zorientować, do wpisywania tekstu w konsoli służy właśnie polecenie `WriteLn`. Można w tym celu skorzystać także z innego polecenia — `Write`. Różnica pomiędzy nimi polega na tym, iż to pierwsze polecenie wpisuje na ekranie tekst i na końcu dodaje znak nowego wiersza. Drugie z poleceń (`Write`) jedynie wpisuje tekst, ale nie dodaje znaku nowego wiersza na końcu.

### Ćwiczenie 3.14.

---

*Aby zaobserwować różnice pomiędzy poleceniami `Write` oraz `WriteLn`:*

1. Kod programu zmodyfikuj do takiej postaci:

```
program console;
uses
  SysUtils;
{$APPTYPE CONSOLE}
begin
  Write('Ten tekst nie jest zakończony znakiem nowego wiersza');
  WriteLn('Ten tekst natomiast jest zakończony znakiem nowego wiersza');
  Write('A ten?');
  ReadLn;
end.
```

2. Teraz uruchom program, aby przekonać się, jak zadziała.
-