

Gliwice, Maj 2004

Programowanie w środowiskach graficznych

Biblioteka VCL

Marek Mittmann

Plan wykładu

- Wprowadzenie
- Podstawowe klasy
- Grafika w VCL
- Dostęp do baz danych

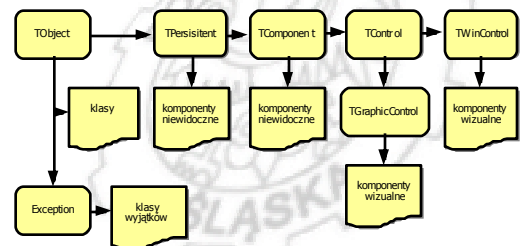
2

Podstawowe koncepcje VCL

- Formularze
- Komponenty
- Właściwości
- Metody
- Programowanie sterowane zdarzeniami

3

Hierarchia klas VCL



4

Podstawowe klasy

- TObject
- TComponent
- TGraphicControl, TWinControl
- TScreen
- TApplication
- TForm
- Exception

5

Klasa TScreen

- Lista wszystkich okien aplikacji
- Aktywny formularz, aktywna kontrolka
- Informacje o rozmiarach ekranu
- Informacje o dostępnych czcionkach i kursorach

6

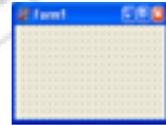
Klasa TApplication

- Obsługa pętli komunikatów
- Dostarczanie pomocy kontekstowej
- Przetwarzanie skrótów klawiszowych
- Obsługa wyjątków
- Zarządzanie oknem głównym aplikacji

7

Klasa TForm

- Baza dla różnych rodzajów okien
 - Okno główne aplikacji
 - Okno potomne MDI
 - Dialog
- Umieszczanie komponentów
- Projektowanie wizualne
- Dziedziczenie wizualne



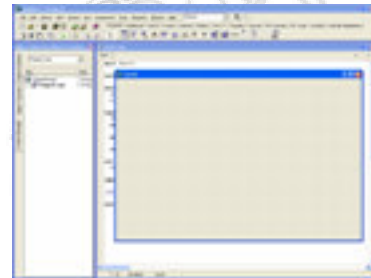
8

Rodzaje komponentów

- Niewidoczne
 - Nie wyświetlają nic na ekranie
 - Wywiedzione z klasy *TComponent*
- Komponenty wizualne wykorzystujące kontrolki Windows
 - Pochodne klasy *TWinControl*
- Komponenty wizualne implementowane w całości wewnątrz VCL
 - Dziedziczą po klasie *TGraphicControl*

9

Czas na demonstrację



10

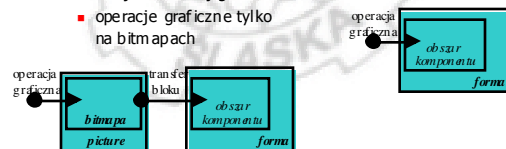
Grafika w VCL

- Klasa *TCanvas*
 - Zmiana pióra, pędzla i czcionki
 - Podstawowe funkcje graficzne
 - Tablica pikseli
 - Wyświetlanie plików graficznych
 - Dostęp do DC, używanie funkcji GDI
- Metoda *Paint*
- Tryb *OwnerDraw* w kontrolkach

11

TImage i TPaintBox

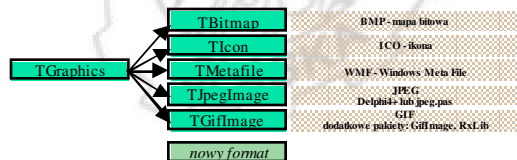
- TImage
 - dwufazowe rysowanie - nie widać
 - wolniejszy - brak akceleracji sprzętowej
 - wczytanie dowolnej zarejestrowanej grafiki
 - operacje graficzne tylko na bitmapach
- TPaintBox
 - szybszy niż *TImage*
 - konieczność odświeżania - znikanie za wartości
 - widoczne rysowanie przy dużej liczbie operacji



12

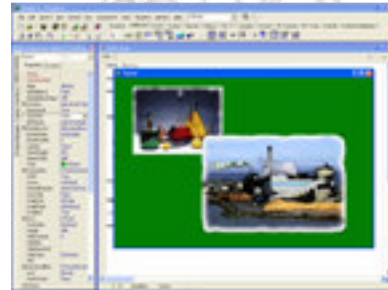
Klasa TPicture

- Zarządzanie formatami graficznymi
- Wczytywanie, przechowywanie i wyświetlanie plików graficznych w dowolnym zarejestrowanym formacie



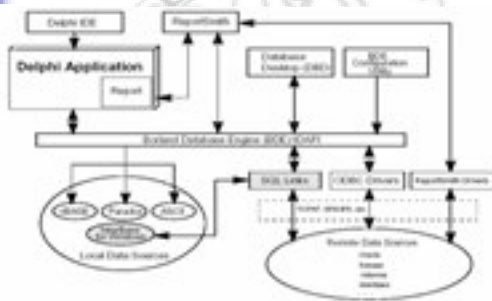
13

Czas na demonstrację



14

Dostęp do baz danych



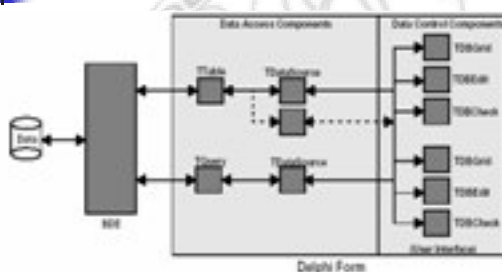
15

Obsługiwane źródła danych

- Paradox, dBase – bazy plikowe
- Pliki ASCII zawierające tabele utworzone w *Database Desktop*
- Lokalny serwer *InterBase*
- Serwery SQL: *Oracle, Sybase, Microsoft SQL Server, Informix, InterBase*
- Źródła danych ODBC

16

Komponenty bazodanowe



17

Komponenty dostępowe

- **TDataSource** – wiąże *TTable, TQuery, TStoredProc* z komponentami wizualnymi
- **TTable** – pobiera i udostępnia dane z tabeli
- **TQuery** – pobiera i udostępnia dane wybrane za pomocą zapytania SQL
- **TStoredProc** – dostęp do procedur składowanych
- **TDatabase** – ustanawia i utrzymuje połączenie z bazą danych
- **TBatchMove** – kopiowanie struktur i zawartości tabel
- **TReport** – tworzenie, podgląd i wydruk raportów

18

Kontrolki wyświetlające dane

- TDBGrid
- TDBNavigator
- TDBText, TDBEdit
- TDBCkCheck Box, TDBRadioGroup
- TDBListBox, TDBComboBox
- TDBImage
- TDBLookupList, TDBLookupCombo

19

Klasa DataSet

- Otwieranie i zamykanie
 - Właściwość *Active*, metody *Open* i *Close*
- Dostęp do pól rekordu
 - *FieldValues*, *FieldByName*, *Fields*
- Nawigacja
 - Metody *First*, *Last*, *Prior*, *Next*, *MoveBy*
 - Właściwości *EOF* i *BOF*

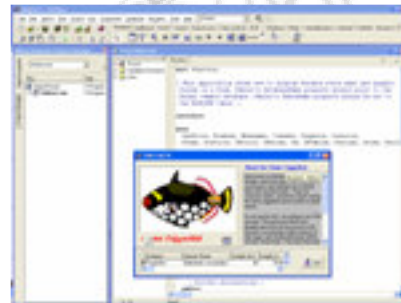
20

Klasa DataSet

- Stany
 - *Inactive*, *Browse*, *Edit*, *Insert*, *SetKey*, *CalcFields*
- Edycja
 - *Edit*, *Append*, *Insert* – rozpoczęcie edycji
 - *Post*, *Cancel* – zatwierdzenie/anulowanie
 - *Delete* – usunięcie rekordu
- Zdarzenia

21

Czas na demonstrację



22

Dziękuję za uwagę

