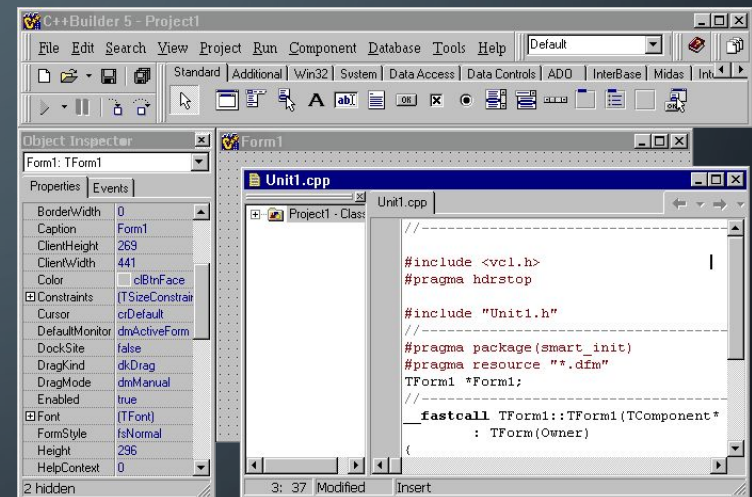
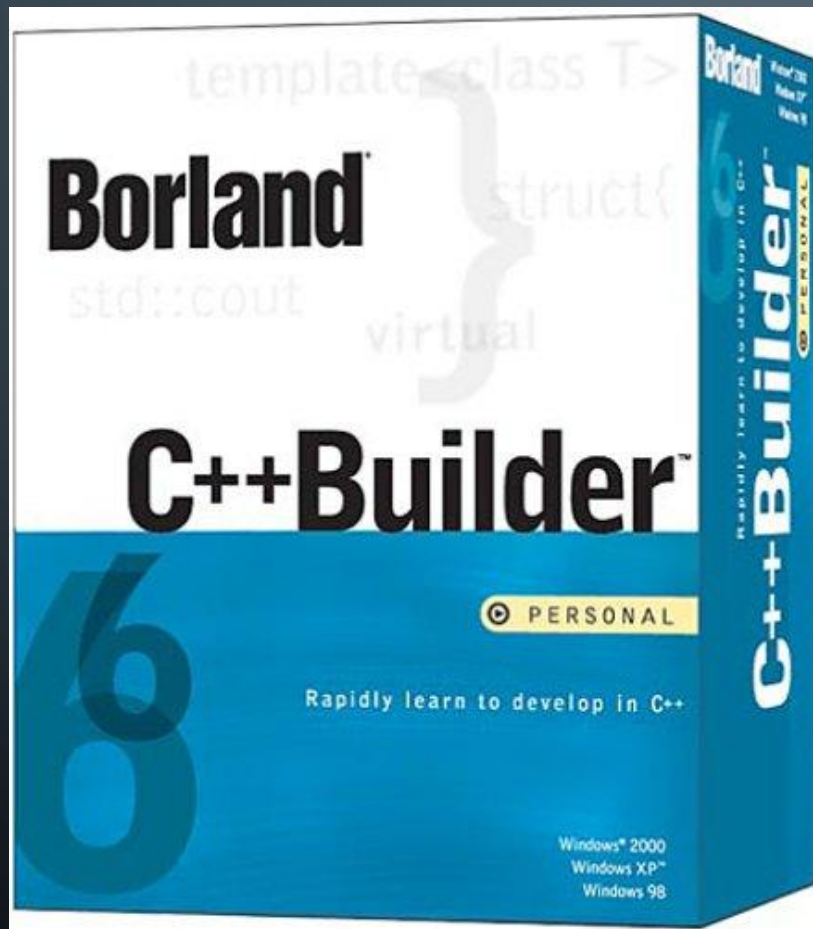


C++ Builder



Презентацію виконав
Студент групи БЧКІ
1-14
БЄЛОВ ІПЛЯ

ПРОГРАМУВАННЯ ДЛЯ WINDOWS. C++ BUILDER

Переваги програмування для Windows

- Підтримка віртуальної пам'яті до 4 Гбайт
- Багатозадачність
- Стандартний графічний інтерфейс
- Незалежність програм від апаратури
- Обмін даними між програмами

ПРИНЦИПИ ПРОГРАМУВАННЯ ДЛЯ WINDOWS В C++ BUILDER

Принцип управління подіями.

Додаток обробляє повідомлення про виниклі події. При цьому виконуються функції - обробники подій. Візуальне компо-нентне проектування

Програма будується візуально із стандартних компонентів. Властивості компонентів налаштовуються при розробці програми.

БІБЛІОТЕКА КОМПОНЕНТІВ C++ BUILDER

C++ Builder включає ту ж бібліотеку візуальних компонентів VCL, що і Delphi.

Всі класи VCL є спадкоємцями TObject. Так само як в Delphi, виділяють компоненти - спадкоємці TComponent, елементи управління (візуальні компоненти) - спадкоємці Tcontrol

На відміну від Delphi конструктори і деструктори мають імена як в C ++ - <тип> і ~ <тип>

ВІДМІННОСТІ ОБ'ЄКТНОЇ МОДЕЛІ C++ BUILDER ВІД СТАНДАРТА

- Опис адрес методів різних класів. Використовується для підключення обробників подій

- тип (`__closure ім'я`) (параметри)

```
void __fastcall( __closure *p) (TObject *Sender);  
p = &Form1->Button2Click;  
p(Sender);
```

- Властивості. Властивості використовуються для доступу до полів класів через методи. Семантика властивостей повністю аналогічна властивостям Delphi

- `__property` ТИП ім'я = {read = МЕТОД ЧТАННЯ, write = МЕТОД ЗАПИСУ}

```
class TNumber  
{ private: int Num;  
          int GetNum() { return Num; }  
          void SetNum( int N) {Num = N; }  
public:  
  __property int Number = {read = GetNum, write = SetNum};  
};
```

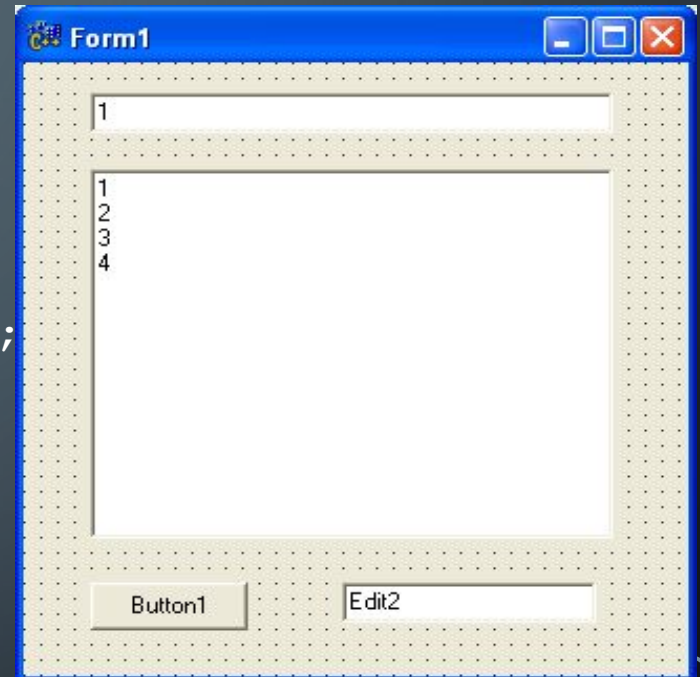
VCL-СУМІСНІ КЛАСИ

- Успадковуються від TObject
- Можуть містити секцію `__published` для властивостей, видимих в інспекторі об'єктів
- Чи не дозволяється використовувати множинне спадкування
- Об'єкти повинні створюватися динамічно
- Класи повинні мати деструктор
- Не створюється конструктор копіювання
- Є підтримка засобів Delphi - динамічних методів, DLL, пакетів.
- Є сумісні типи даних - безлічі (Set), Currency, AnsiString,

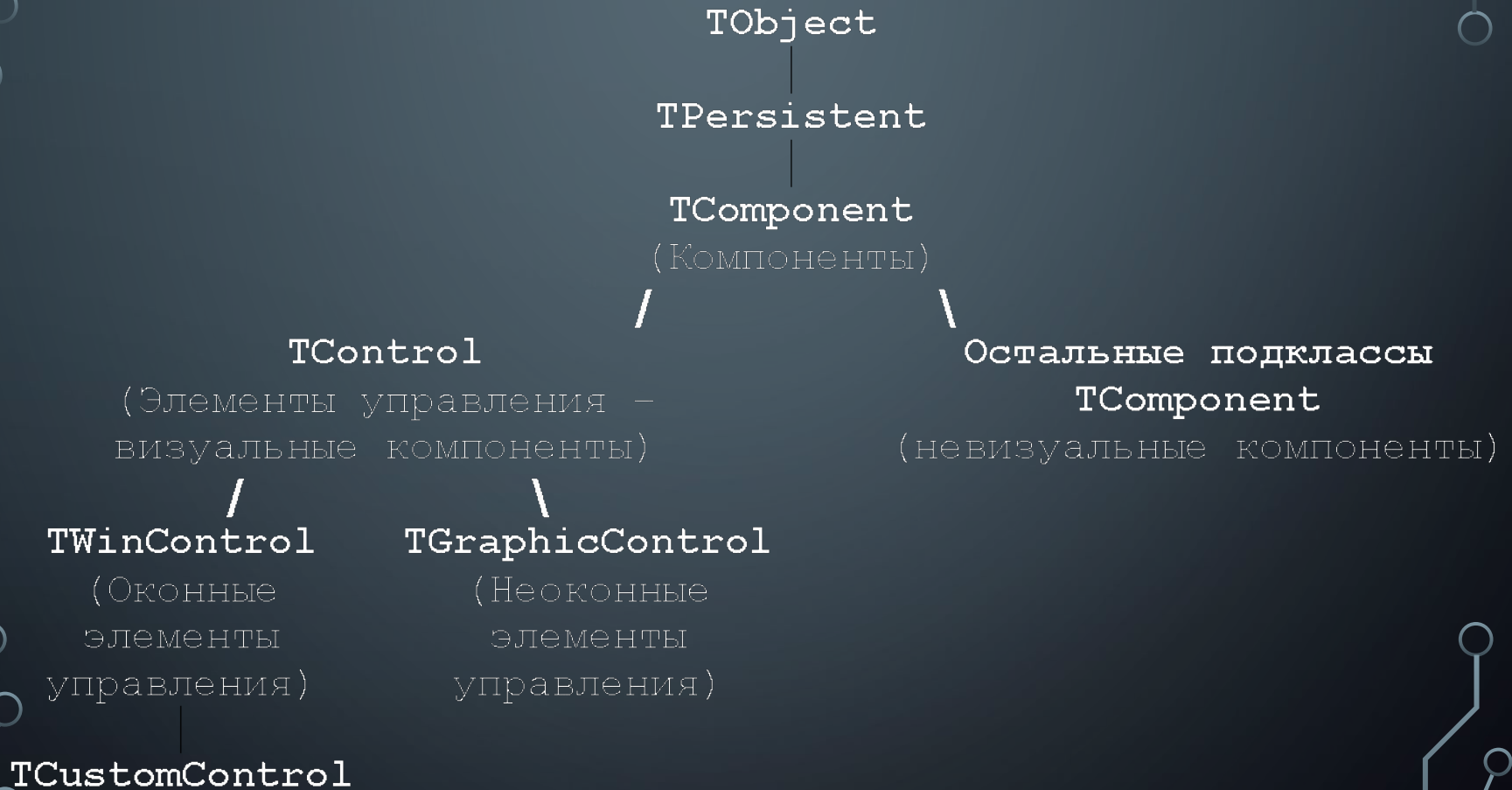
ПРИКЛАД ПРОГРАМИ

```
void __fastcall TForm1::Edit1KeyDown
(Object *Sender, WORD &Key,
TShiftState Shift)
{
    if (Key == VK_RETURN)
    { Mem1->Lines->Append(Edit1->Text);
      Edit1->SelectAll();
    }
}

void __fastcall TForm1::Button1Click
(Object *Sender)
{
    int s = 0;
    for (int i = 0; i < Mem1->Lines->Count; i++)
        s = s + StrToIntDef
        ( Mem1->Lines->Strings[i], 0);
    Edit2->Text = s;
}
```



ІЄРАРХІЯ VCL



ОСНОВНІ КЛАСИ VCL

Кожен наступний клас успадковує поля, методи і властивості предка.

`TObject` - предок всіх класів. Містить метод `Free ()` для знищення об'єкта, якщо він був створений, властивості `ClassName` - ім'я класу.

`TPersistent` додає методи, необхідні для розміщення компонента в пам'яті або в файлі. Містить зокрема метод `void Assign (TPersistent * Source)`, що копіює всі дані з об'єкта `Source` в поточний об'єкт.

`TComponent` - предок компонентів - об'єктів, властивості яких можна змінювати при розробці програми з середовища `C++ Builder`. Містить властивості `__property AnsiString Name` - ім'я компонента (середина дає стандартні імена типу `Label1`, `Edit2`, але їх можна змінити) `__property int Tag` - число, яке програміст може використовувати на свій розсуд.

ЗАГАЛЬНІ ВЛАСТИВОСТІ ЕЛЕМЕНТІВ УПРАВЛІННЯ

1. Розмір і розташування на екрані

Left, Top - координати лівого верхнього кута компонента в пікселях щодо координат компонента-батька.

Height, Width - висота і ширина компонента.

2. Вирівнювання і масштабування Логічне властивість

AutoSize (підлаштовується Чи розмір елемента підлаштовується під його вміст)
Властивість Align - вирівнювання (значення - alNone, alTop, alBottom, alLeft, alRight, alClient)

Властивість Anchors - безліч прив'язок елемента до країв батька. Свойство Constraints содержит значення об'єкта из 4-х свойств – MinWidth, MaxWidth, MinHeight,

Max Height - задають мінімальні і максимальні розміри компонента.

3. Зовнішній вигляд компонента

Color - колір, Cursor - покажчик миші, Font - шрифт для тексту.

Caption - задає заголовок компонента, Text - редагований текст.

Visible - видимість компонента, Enabled – активність.

Властивість PopupMenu містить покажчик на контекстне меню.

ОСНОВНІ ПОДІЇ

Елемент управління може обробляти наступні події

OnClick - на елементі натиснута і відпущена ліва кнопка миші

OnDbClick - Проведений подвійне клацання мишею.

OnMouseDown - була натиснута клавіша миші. Параметр **Button** містить інформацію про те, яка кнопка була натиснута **mbLeft**, **mbRight** або **mbMiddle**. **X**, **Y** - координати покажчика щодо лівого верхнього кута області компонента. **Shift** - стан **Shift**, **Ctrl**, **Alt**.

OnMouseUp - була відпущена клавіша миші.

OnMouseMove - миша переміщується треба об'єктом.

Віконні елементи управління успадковуються від **TWinControl**.

Віконний елемент може бути батьківським для інших елементів (св-во **Controls**).

Віконні елементи обробляють подія від клавіатури **OnKeyDown** параметр **WORD Key;**
TshiftState State

OnKeyPress параметр **char Key;**

OnEnter - виникає, коли елемент стає активним - отримує фокус введення (миша або TAB).

OnExit - коли елемент втрачає фокус введення

СТАНДАРТНІ КОМПОНЕНТИ

TLabel, TEdit

- Мітка TLabel. Текст - властивість Caption.
- Поле введення TEdit дозволяє ввести один рядок тексту. Текст - властивість Text.
- Властивості: bool AutoSelect - якщо властивість має значення true, то текст виділяється при активізації редактора.
- Int MaxLength - максимальне число символів для редагування.
- bool ReadOnly- чи можна змінювати текст.
- Кнопка TButtonТекст кнопки задається властивістю Caption. При натисканні викликає обробник події OnClick. Виклик методу Click () еквівалентний клацанню по кнопці.
- Прапорець TCheckBox, перемикачі RadioButton.
- Підпис задається у властивості Caption. Головна властивість - Checked. Ця властивість приймає значення true, якщо прапорець включений, і false, якщо він скинутий. Щоб реагувати на зміну стану прапорця, треба створити обробник події OnClick

СТАНДАРТНИЙ КОМПОНЕНТ ТМЕМО

- Багаторядковий редактор Memo.
- Головна властивість - Lines (Рядки), що має тип TStrings *. У ньому зберігається список рядків, введених користувачем.
- Ці рядки можна обробляти всіма методами, доступними в класі TStrings, наприклад зберігати у файлі:
- Memo1 -> Lines-> SaveToFile ("C: \ \ Memo.TXT");
- Наявність у текстовій області смуг прокрутки задається у властивості ScrollBars.
- Метод Clear () - очищення.
- Група методів призначена для роботи з буфером обміну Windows. Для копіювання виділеного тексту в буфер обміну застосовується метод CopyToClipboard (), для вирізання тексту - метод CutToClipboard (), для вставки тексту з буфера - метод PasteFromClipboard.
- Коли в текстовій області відбувається зміна тексту, генерується подія OnChange

СТАНДАРТНИЙ КОМПОНЕНТ TListBox

- Список **TListBox**.

Список рядків зберігається у властивості `Items`. Список рядків допускає виділення рядка. Одночасно може бути виділена або один рядок, або декілька. Визначає можливість виділення кількох рядків логічне властивість `MultiSelect`. Номер виділеного рядка можна отримати за допомогою цілочисельного властивості `ItemIndex`. Якщо ця властивість одно `-1`, то жоден рядок не виділена. Відзначимо, що рядки нумеруються з нуля. Якщо дозволено виділення кількох елементів списку, то можна перевірити виділена рядок з номером `I` за допомогою логічного векторного властивості `Selected [i]`. Кількість виділених рядків містить властивість `SelCount`. Логічне властивість `Sorted` вказує, чи повинні рядки автоматично сортуватися в алфавітному порядку. Для очищення списку використовується метод `Clear`. Видалити `i`-ю рядок можна за допомогою методу `DeleteString (I)`. Событіє за замовчуванням для списку - `OnClick`.