

Просто Windows приложение

Създаването на приложение в **Borland C++ Builder 6** е комбинация от **визуално проектиране** (Drag & Drop) и **писане на C++ код**. Тъй като средата използва VCL (Visual Component Library), **процесът е обектно-ориентиран**.

Ето един опростен, но подробен алгоритъм за създаване на просто приложение .

Тук ще представим „Калкулатор за събиране на две числа“, който обяснява логиката на всеки етап.

1. Проектиране на интерфейса (Visual Design)

Когато стартирате нов проект (**File -> New -> Application**), се появява празна форма (Form1). Тук вие ще трябва да дефинирате как ще изглежда програмата.

- **Компоненти:** Използвайте палитрата с компоненти (**Standard**), за да поставите следните обекти върху формата:
 - 2 броя **TEdit** (за въвеждане на числа).
 - 1 брой **TLabel** (за извеждане на резултата).
 - 1 брой **TButton** (за извършване на действието).
- **Свойства (Object Inspector):** Маркирайте всеки обект и променете свойствата му:
 - За Edit1 и Edit2: изтрийте текста в свойството Text.
 - За Button1: променете Caption на "Събери".
 - За Label1: променете Caption на "Резултат:".

2. Деклариране на логиката (Свързване на интерфейса с кода)

За да накарате бутона да работи, трябва да създадете **Event Handler** (обработчик на събитие).

- **Действие:** Щракнете два пъти върху бутона в дизайнерския изглед.
- **Резултат:** IDE-то автоматично ще ви прехвърли в редактора на код (Unit1.cpp) и ще генерира празна функция:

```
void __fastcall TForm1::Button1Click(TObject *Sender) {  
  
    // Вашият код трябва да бъде тук  
  
}
```

3. Писане на алгоритъма (C++ Код)

Вътре във функцията `Button1Click` трябва да реализирате три стъпки:

Вход -> Изчисление -> Изход.

Важно: Тъй като потребителят въвежда текст в `TEdit`, трябва да преобразуваме този текст в число, за да можем после да ги съберем

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    // 1. Деклариране на променливи
    double num1, num2, sum;

    try {
        // 2. Вземане на входните данни и конвертиране (String към Double)
        num1 = Edit1->Text.ToDouble();
        num2 = Edit2->Text.ToDouble();

        // 3. Изчисление
        sum = num1 + num2;

        // 4. Извеждане на резултата (Double към String)
        Label1->Caption = "Резултат: " + FloatToStr(sum);
    }
    catch (...) {
        // Защита при грешно въведени данни (напр. текст вместо число)
        ShowMessage("Моля, въведете валидни числа!");
    }
}
```

4. Компилиране и Сглобяване (Build & Run)

Това е етапът, в който вашият C++ код се превръща в **ИЗПЪЛНИМ .exe** файл.

Дайте подходящи имена на проекта и файловете и запишете всички файлове, които сте създали чрез `File->Save ...` или `File->Save all!`

Натиснете **F9** (или зеленото триъгълниче) за да се стартира:
или **Project->Build Project** или **Project-> Make Project (Ctrl+F9)**.

1. **Compiler:** Проверява синтаксиса на вашия код.
2. **Linker:** Свързва вашия код с библиотеките на Borland (VCL).
3. **Execution:** **Ако всичко е наред** Програмата ви стартира като самостоятелен прозорец в Windows.

Как се работи с Visual Component Library (VCL):

Етап	Инструмент	Какво се случва
Визуален	Form Designer	Поставяте обекти и задавате <code>Parent</code> и <code>Owner</code> .
Свойства	Object Inspector	Променят цветове, шрифтове и начални стойности.
Логика	Code Editor	Пишете функциите, които реагират на <code>OnClick</code> , <code>OnKeyPress</code> и т.н.
Финал	Compiler/Linker	Създава се бинарният файл на приложението.