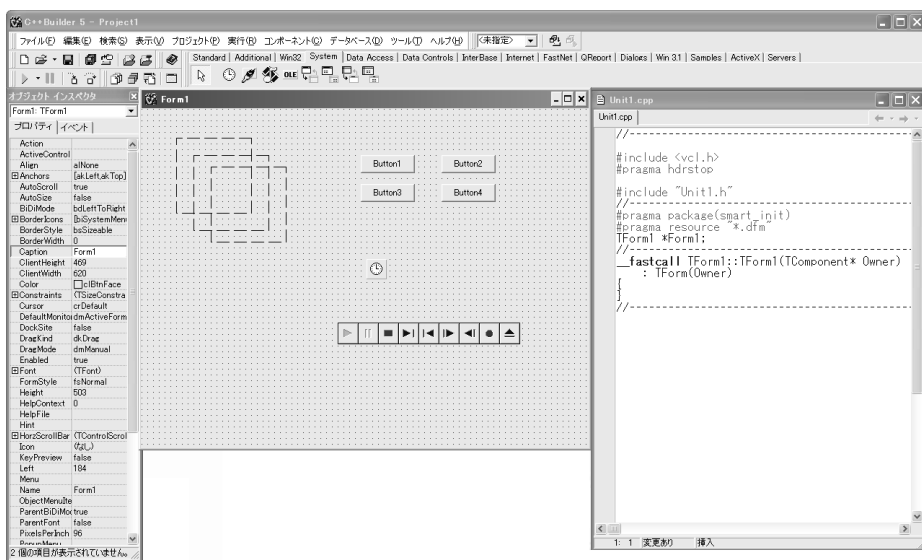


3. C++Builder

3-1 : プログラムについて

・ C++Builder

ここで使っている C++Builder は Borland 社が開発したプログラミングソフトです。このソフトは C++言語を使って Windows アプリケーションを作る事ができます。



① コンポーネントパレット



部品棚です。ここから必要な物を選んで、フォームに配置しましょう。

② オブジェクトインスペクタ

配置したコンポーネントのプロパティを設定できます。ここで設定した値は『初期値』として使用できますが、プログラム上には書かれませんが、混乱をさけるためにもできるだけプログラム上に初期値を書いて設定するようにしましょう。



③この部分にプログラムを書きます。

The screenshot shows a code editor window titled 'Unit1.cpp'. The code is as follows:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{
```

オブジェクトインスペクタでイベントタブに切り替えて、空欄をダブルクリックイベントハンドラがここに作成されます。各イベントハンドラ内に動作させたいプログラムを書きましょう。

③ 実行



書き終わったプログラムを動かすには、上の三角ボタンを使います。キーボードの「F9」キーでも可能です。

④ 保存

実行ボタンの上のフロッピーマークが保存ボタンです。トラブルが起きて保存していないデータが消えてしまう事も考えられるのでこまめに保存しましょう。

新しいプログラムの保存は右の「すべて保存」を使い、上書きする際には左の「保存」を使います。

※保存されるファイル名（デフォルトの場合）

自分で書いたプログラム : Unit1.cpp

ヘッダーファイル : Unit1.h

実行ファイル : Project.exe

・変数や関数の宣言

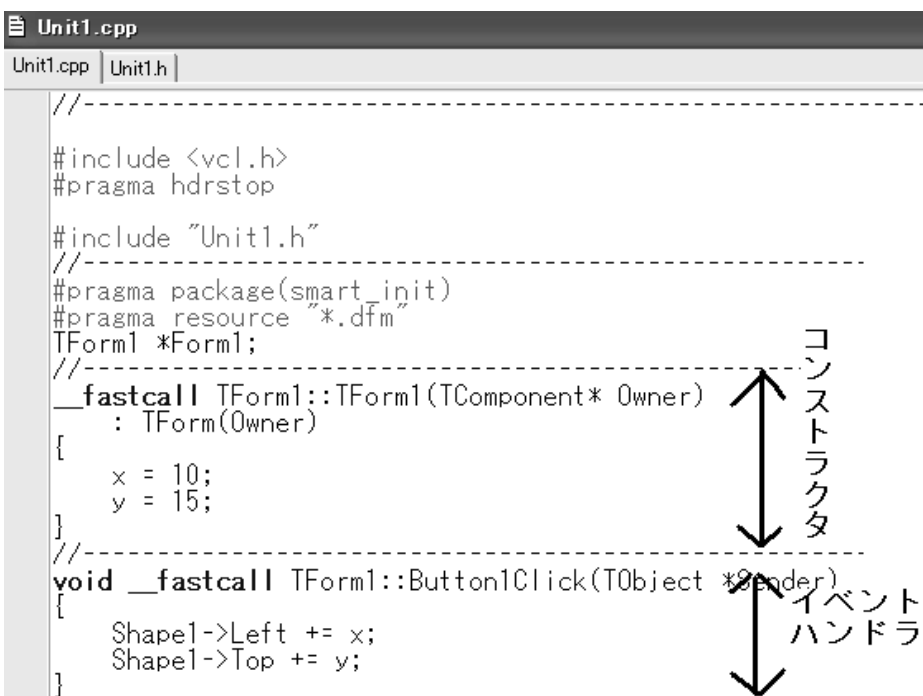
C++Builder では、イベントハンドラ内で変数を宣言した場合、その中でしか変数は使用できません。そこで、ヘッダー (Unit1.h) という別のファイルに宣言するのが普通です。ここで宣言した変数はプログラム上のどこでも使う事が出来ます。

ヘッダーファイルは初期状態では表示されていません。コードウィンドウの Unit1.cpp タブを右クリックして「ソース/ヘッダーファイルを開く」を選択すると開きます。

★変数の初期化と関数の定義

宣言はヘッダーに書きましたが、変数の初期値は Unit1.cpp のコンストラクタと言う部分に書きます。コンストラクタとはプログラムの実行を開始した時、最初に一度だけ実行されるプログラムのことです。

コンストラクタやイベントハンドラは自動的に作られますが、関数の定義は自分で作る必要があります。



```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{  
    x = 10;  
    y = 15;  
}  
//-----  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    Shape1->Left += x;  
    Shape1->Top += y;  
}
```

コンストラクタ

イベントハンドラ

また、C++のプログラム上にはふつう見えませんが、デストラクタというプログラム終了時に実行される部分があります。

関数の数が膨大になってくると、本来のプログラムが見つらなくなってしまいます。そこで、別の cpp ファイルを作成してそこに続きを書いた上で Unit.cpp に include して使うことができます。

例 : #include "tsuduki.cpp"

●文字列データ型 AnsiString 型

C++専用のデータ型で、文字列を扱えます。変数扱いなので int 型や char 型と同じ扱いです。AnsiString 型は文字列、つまり文章を代入することができます。

また「+=」で加えれば文字列に文字を追加することができます。”\n”を追加すれば改行も可能です。

<例>

```
AnsiString X;  
int A = 2468;  
X = “半角字も可能” ;  
X += A;
```

<結果>

半角字も可能 2468

このまま続けても改行もしてみましよう。

<例>

```
X += “\n” ;  
X += “改行できるよ。” ;
```

<結果>

半角字も可能 2468

改行できるよ

となります。