

## Borland Delphi 6 を使ってみよう

## Borland Delphi 6 とは

- PASCAL プログラムコードの編集, 実行などの機能を持ったソフト

## 本日の内容

### 例題1. プログラム実行の体験

- Borland Delphi 6 を使って, sin 関数の繰り返し計算を行ってみる
- 作成されたプログラムコードを, 「D:ドライブ」に保存してみる

### 例題1. プログラム実行の体験

- コンピュータを役に立つ道具として実感するために, 次ページのプログラムを使って, Borland Delphi 6 でのプログラム実行を体験する

```

program sum;
{$APPTYPE CONSOLE}
uses SysUtils;
var start_x, step_x, x, y:real;
var i:integer;
begin

```

```

write('Please Enter start_x:');
readln(start_x);
write('Please Enter step_x:');
readln(step_x);

```

```

for i:=1 to 20 do
begin
  x := start_x + ( i * step_x );
  y := sin(x);
  writeln('sin(', x:8:3, ') =', y:8:3);
end;

```

```

readln
end.

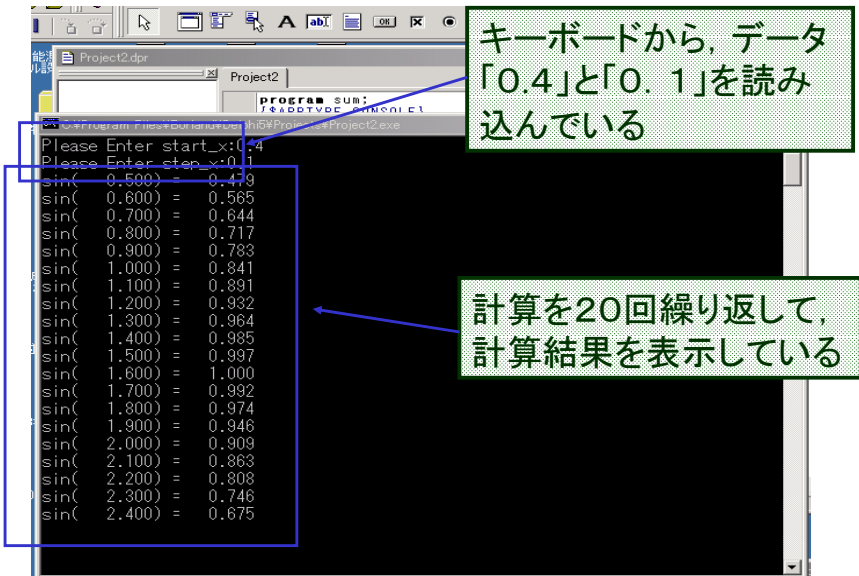
```

キーボードからの  
データの読み込みを  
行っている部分

計算の繰り返しを  
行っている部分

画面へのデータの  
書き出しを行って  
いる部分

## 例題1のプログラム実行結果(例)



キーボードから、データ  
「0.4」と「0.1」を読み  
込んでいる

計算を20回繰り返して、  
計算結果を表示している

## 課題1. 例題1のプログラム実行まで

- ① 例題1のプログラム
- ② Borland Delphi 6 の起動  
「スタート」→「プログラム」→「Borland Delphi 6」→「Delphi 6」
- ③ コンソールアプリケーションの新規作成  
「ファイル」→「新規作成」→「その他」⇒ ウィンドウが現れる  
「コンソールアプリケーション」→「OK」⇒ ウィンドウが現れる
- ④ ①のプログラムコードを「コピー」して「貼り付け」
- ⑤ コンパイル  
「プロジェクト」→「OOをコンパイル」
- ⑥ 実行  
「実行」→「実行」
- ⑦ 数値データを2つ、キーボードから与える

## プログラムの機能

### 計算等の実行手順を記述したもの

- 計算の繰り返し
  - 計算は自動で繰り返し
- キーボードからの読み込み
- 画面への書き出し
- など

## 例題1のプログラムの機能

1. キーボードからのデータの読み込み  
次の2つの値を読み込む

start\_x, step\_x

2. 計算の繰り返し

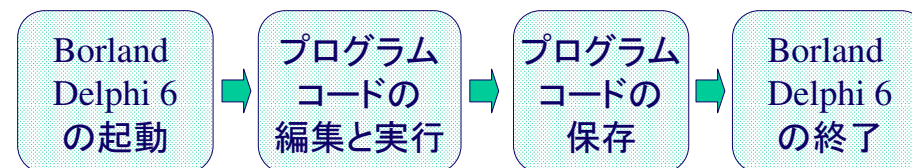
sin( x ) の計算を20回繰り返す

x = start\_x + step\_x,  
start\_x + 2 × step\_x,  
...  
start\_x + 20 × step\_x

} 20回分

3. 画面へのデータの書き出し  
計算した sin( x ) の値を書き出す

## Borland Delphi 6 の 起動から終了まで



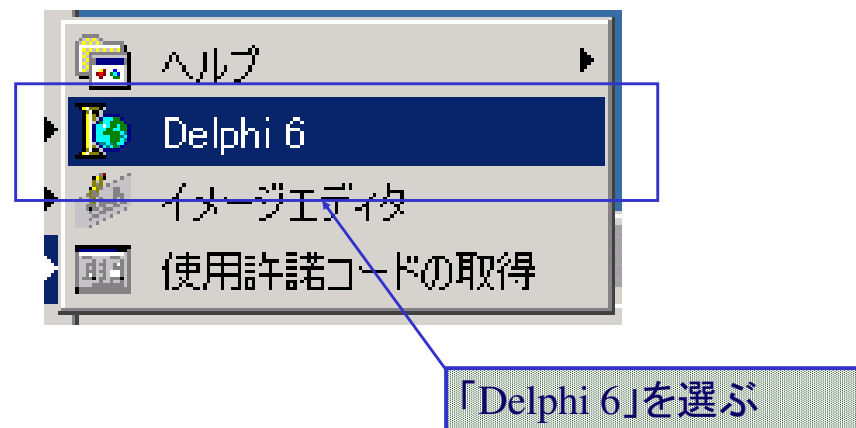
以前に作っておいた  
プログラムコードの  
ファイルを読み込むこと  
も簡単にできる

プログラムコードなど  
が入ったファイルがで  
きる

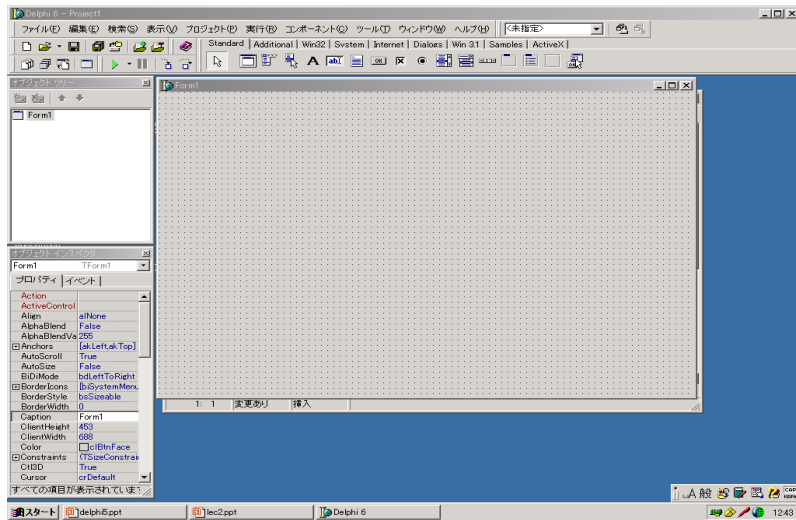
## Borland Delphi 6 の起動 (1/3)

- 「スタート」  
→「プログラム」  
→「Borland Delphi 6」  
→「Delphi 6」

## Borland Delphi 6 の起動 (2/3)

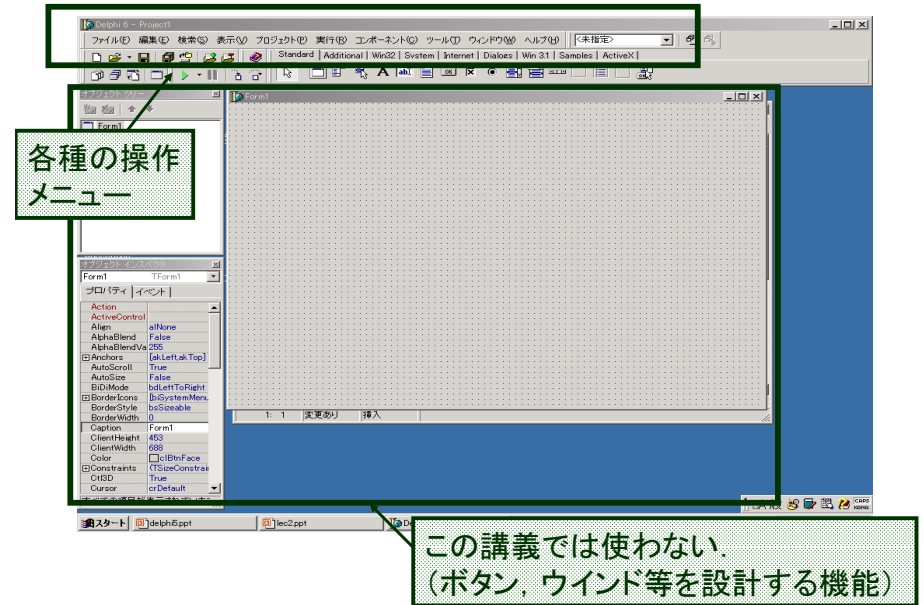


## Borland Delphi 6 の起動(3/3)

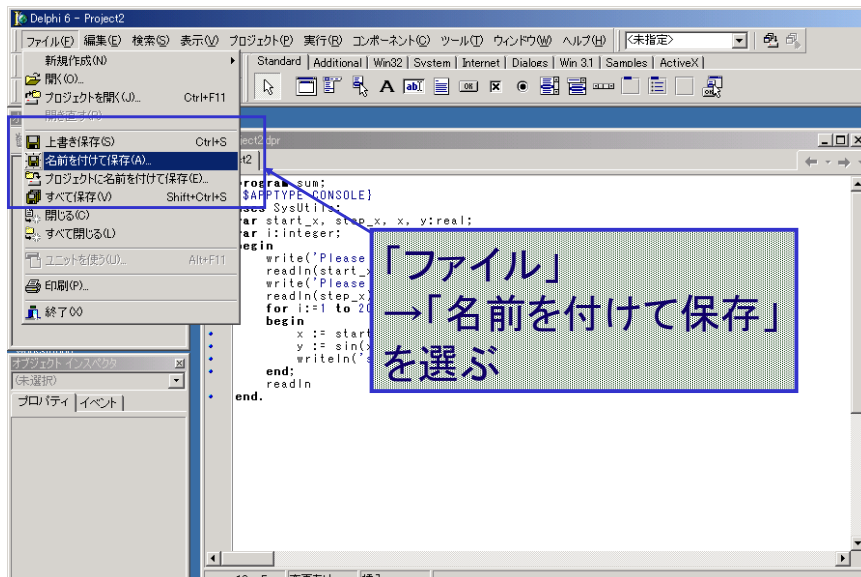


Borland Delphi 6 を起動すると、上のような画面が現れる

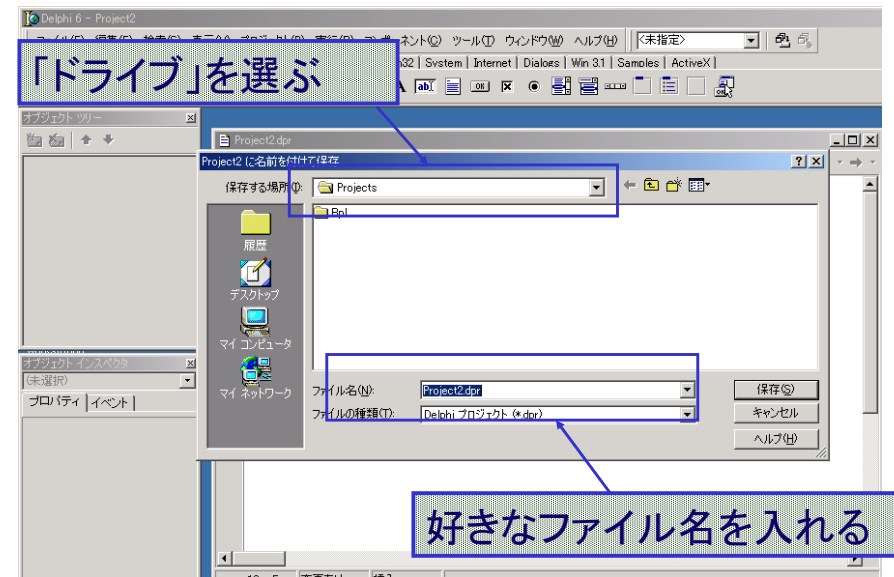
## Borland Delphi 6 の画面構成



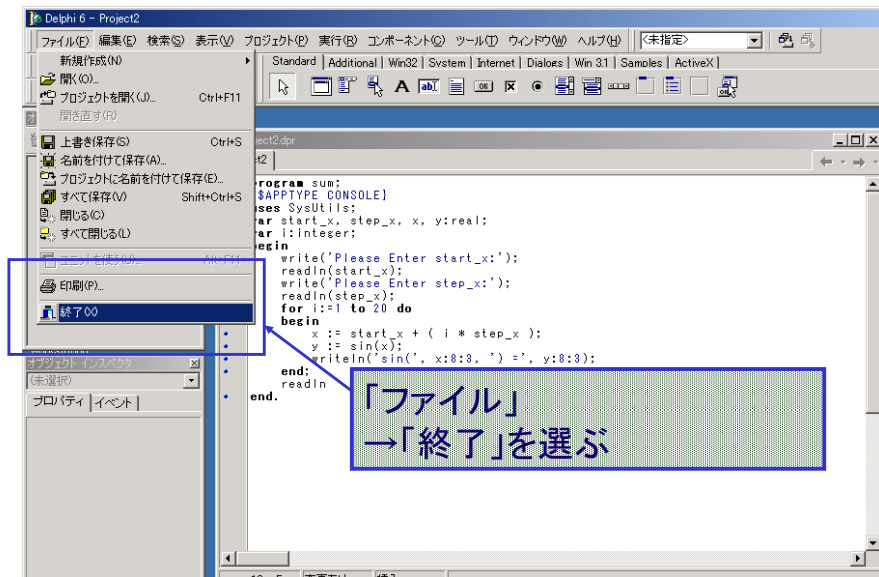
## プログラムコードの保存(1/2)



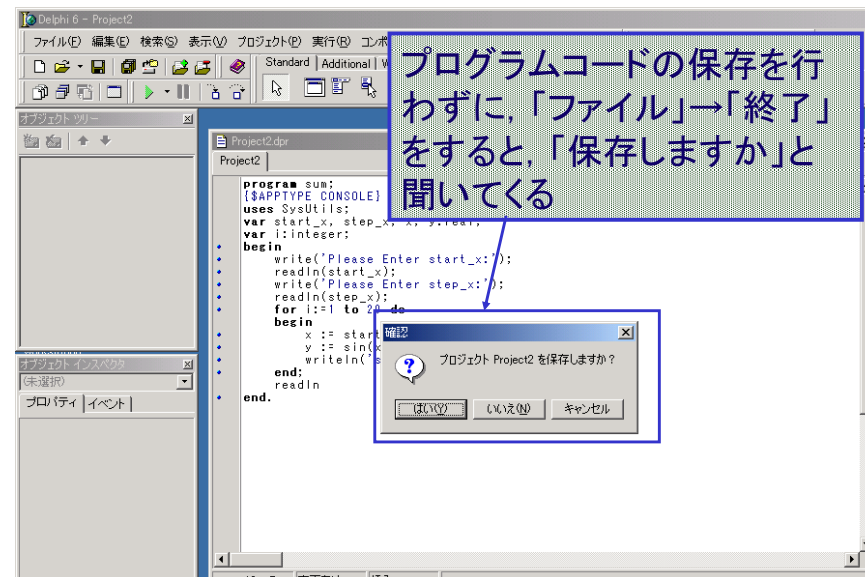
## プログラムコードの保存(2/2)



## Borland Delphi 6 の終了



## Borland Delphi 6 の終了



## 課題2. プログラムコードの保存と終了

課題1に続いて次の手順を行う

### ⑧ プログラムコードの保存

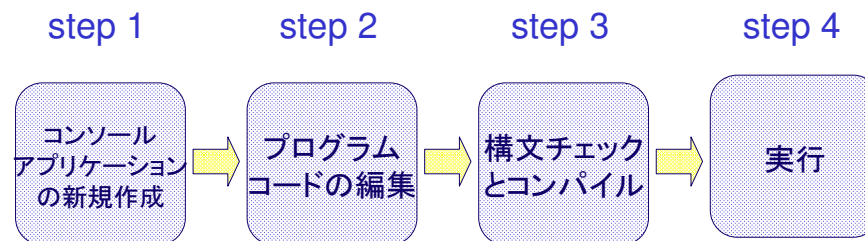
ドライブ: 好きなドライブ名

ファイル名: 好きなファイル名

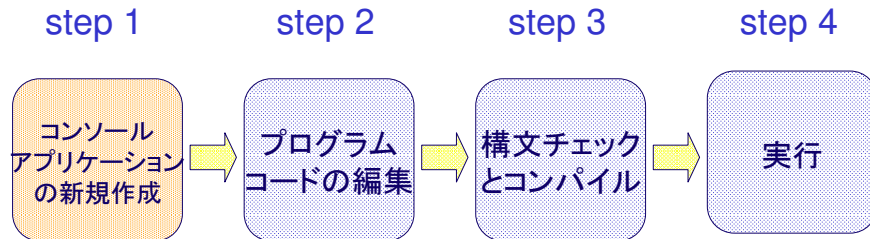
### ⑨ Borland Delphi 6 の終了

「ファイル」→「終了」

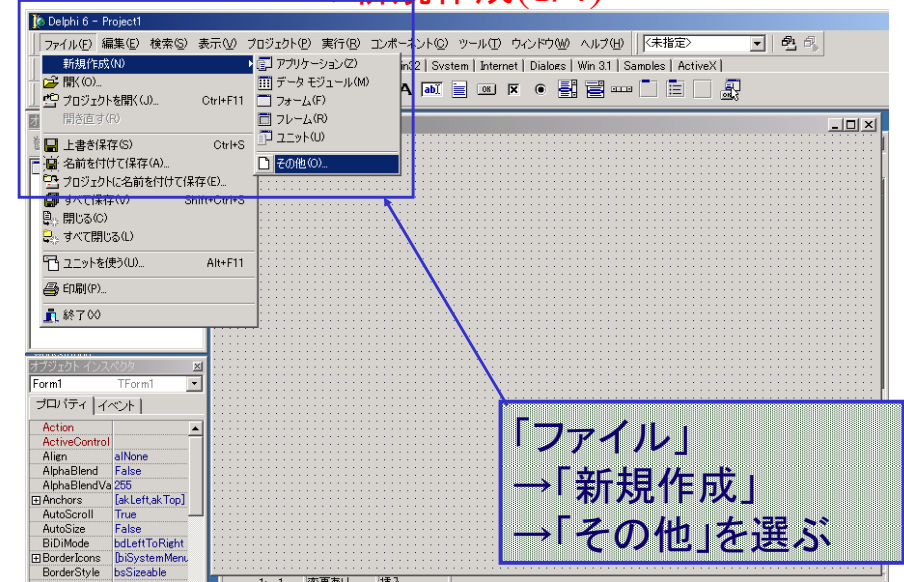
## Borland Delphi 6 での プログラムコード編集と実行の手順



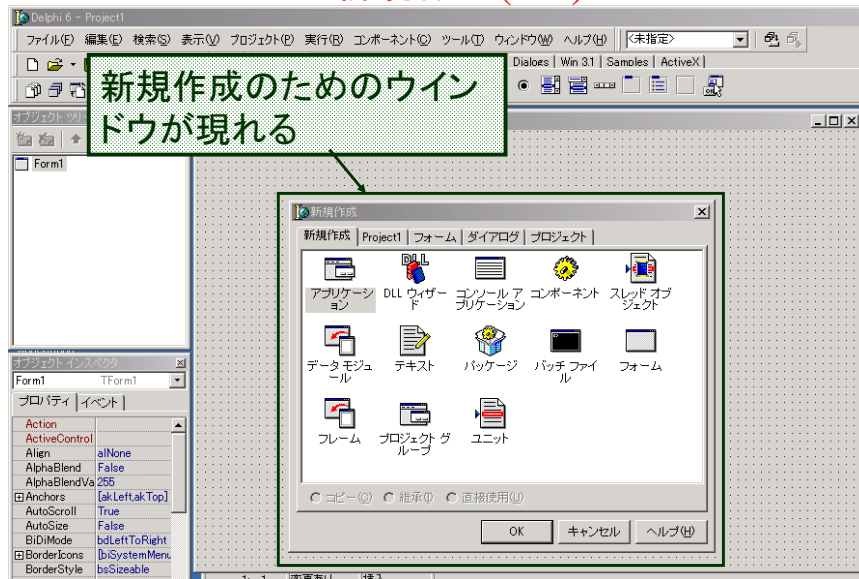
# コンソールアプリケーション の新規作成



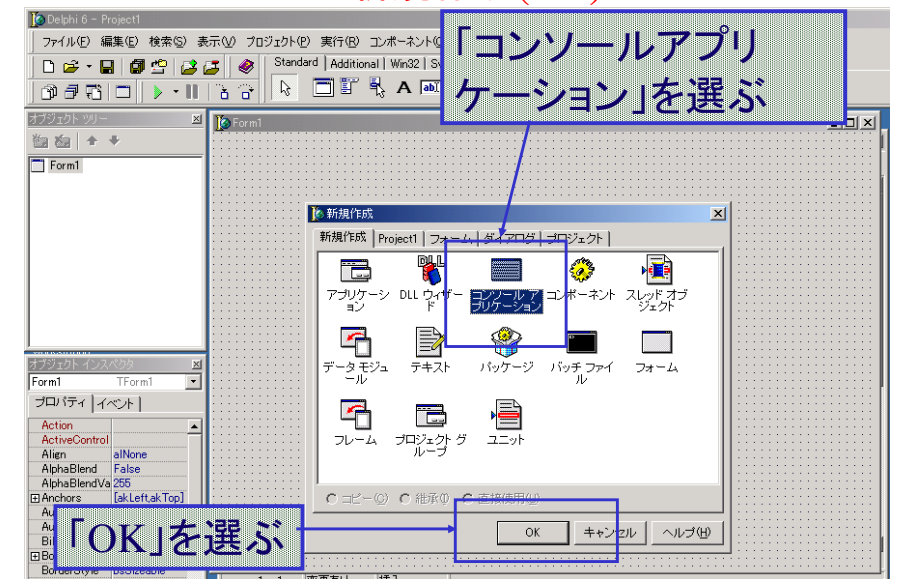
# コンソールアプリケーション の新規作成(1/4)



# コンソールアプリケーション の新規作成(2/4)

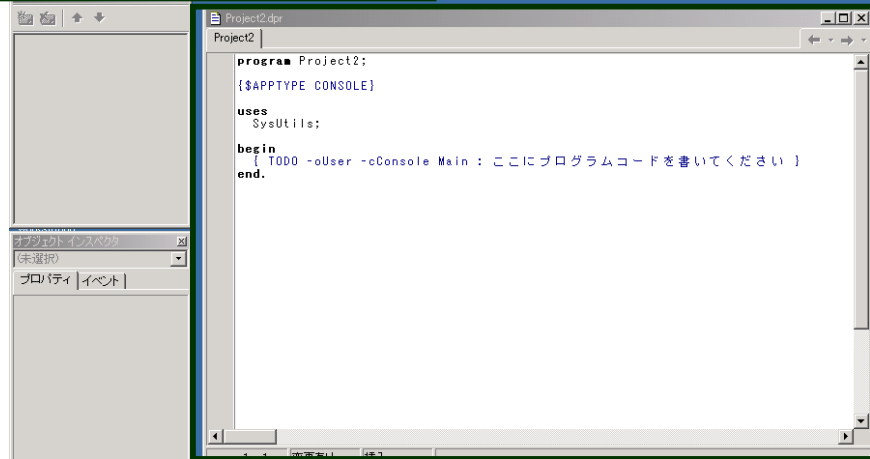


# コンソールアプリケーション の新規作成(3/4)



## コンソールアプリケーション の新規作成(4/4)

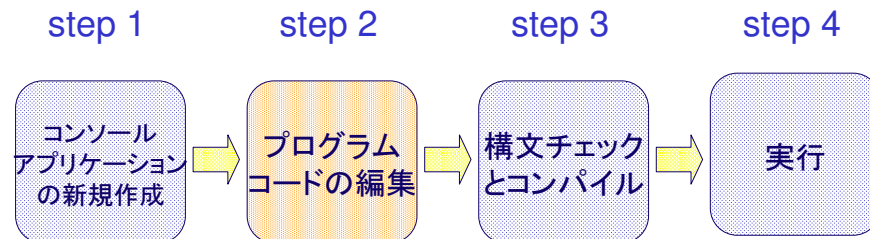
プログラムコード編集用の  
ウィンドウが現れる



## コンソールアプリケーション の新規作成

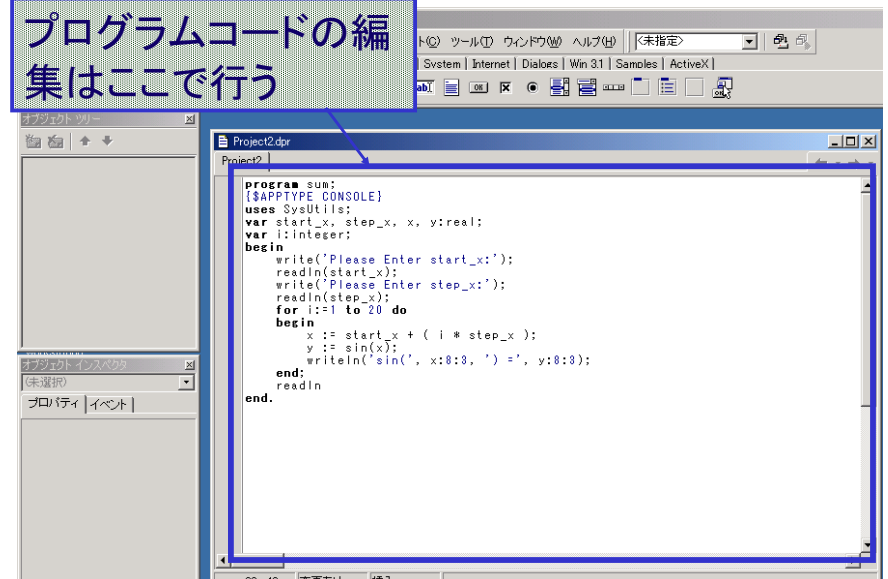
- ① 「ファイル」→「新規作成」→「その他」を選ぶ
- ② 新規作成のためのウィンドウが現れる
- ③ 新規作成では  
「コンソールアプリケーション」を選ぶ  
「OK」を選ぶ
- ④ プログラムコード編集のためのウィンドウが  
現れる

## プログラムコードの編集

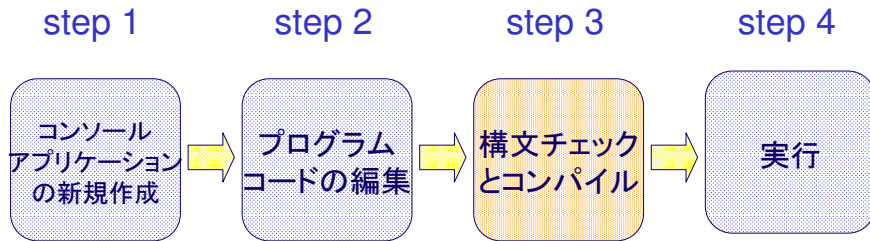


## プログラムコードの編集

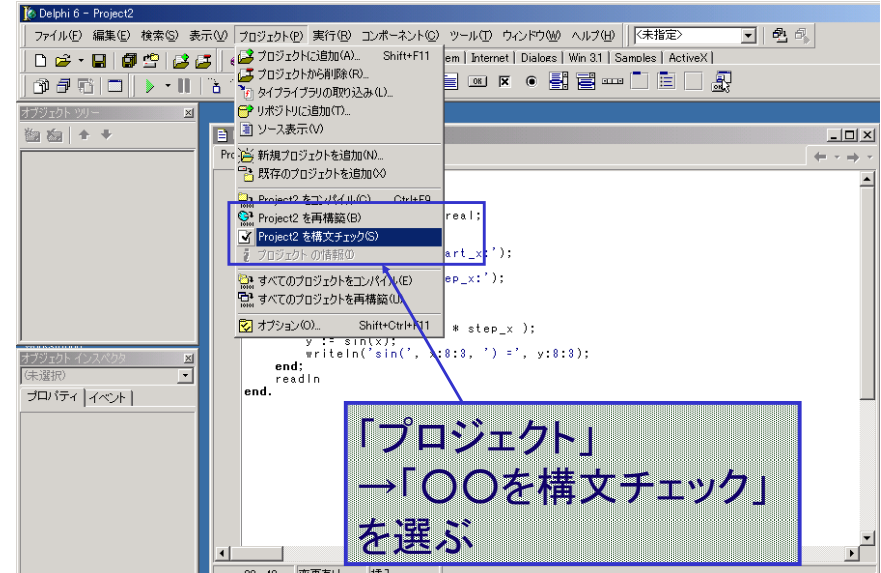
プログラムコードの編  
集はここで行う



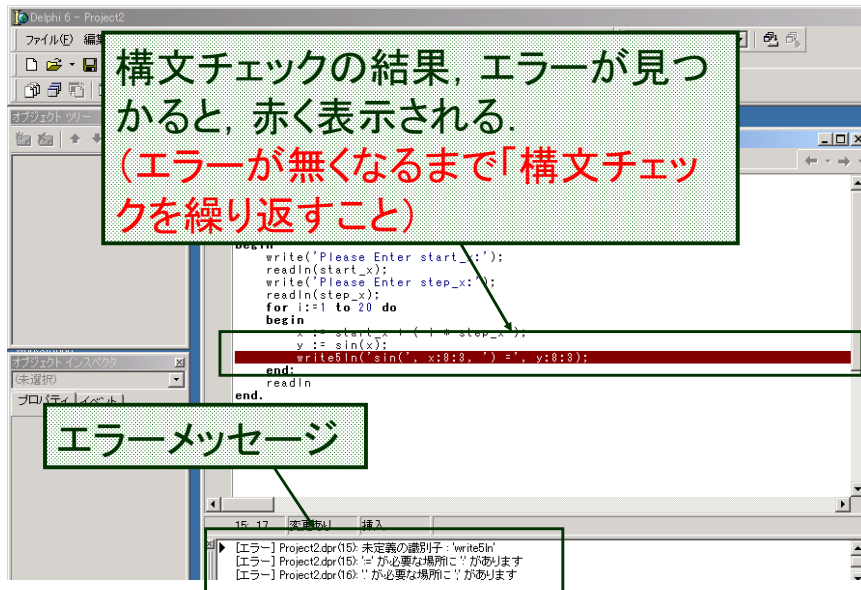
# 構文チェックとコンパイル



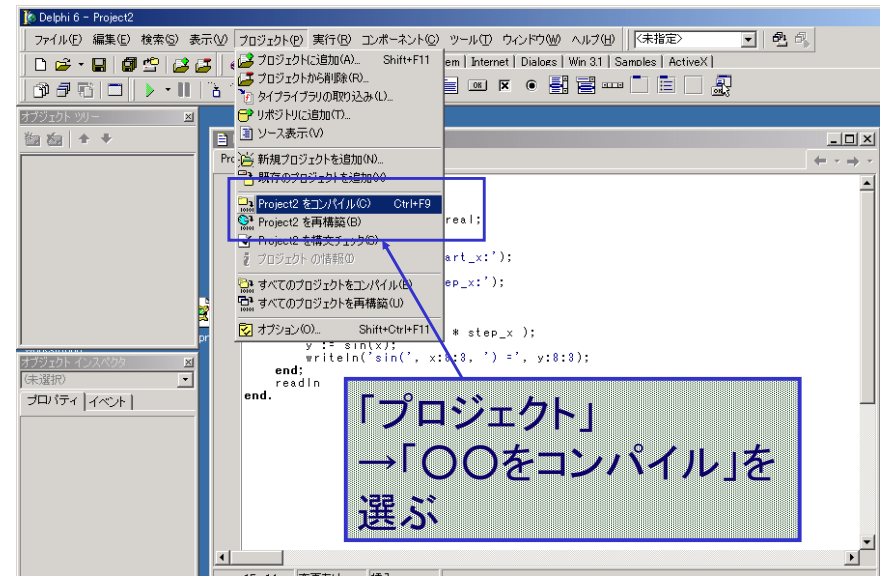
# 構文チェックとコンパイル(1/4)



# 構文チェックとコンパイル(2/4)

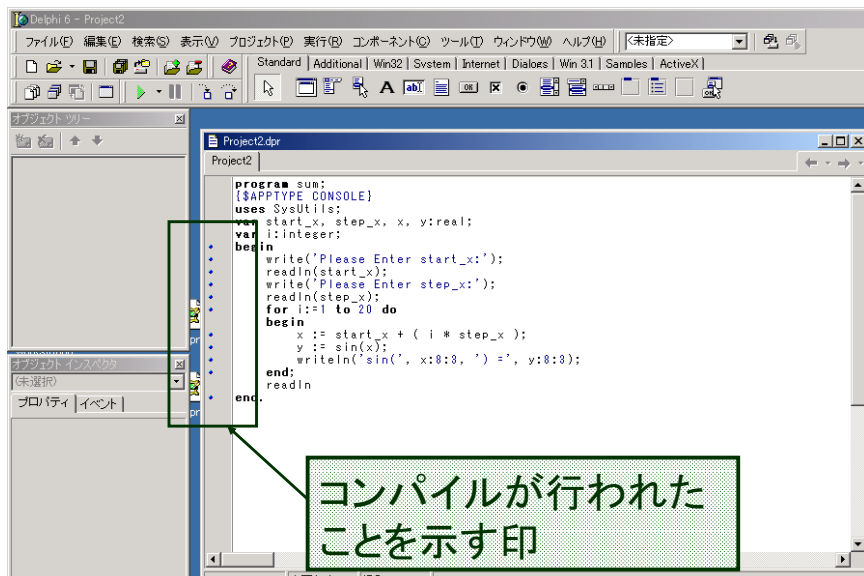


# 構文チェックとコンパイル(3/4)





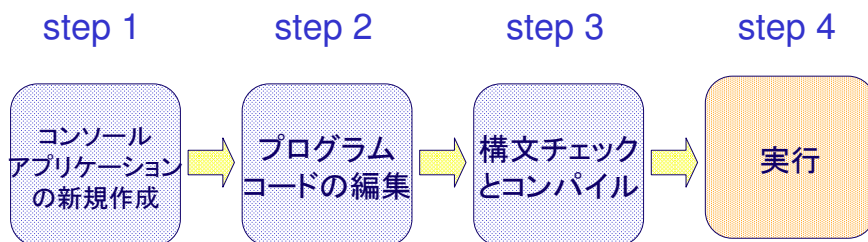
# 構文チェックとコンパイル(4/4)



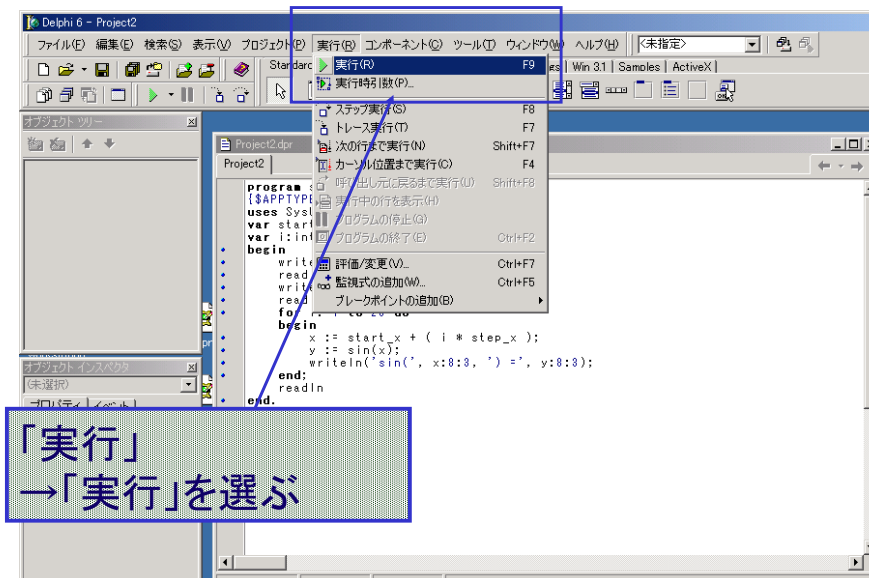
# 構文チェックとコンパイル

- ① 「プロジェクト」→「OOを構文チェック」を選ぶ
- ② 構文チェック結果が現れる  
エラーが出たら、プログラムコードを修正する
- ③ 「プロジェクト」→「OOをコンパイル」を選ぶ
- ④ コンパイル結果が現れる  
エラーが出たら、プログラムコードを修正する

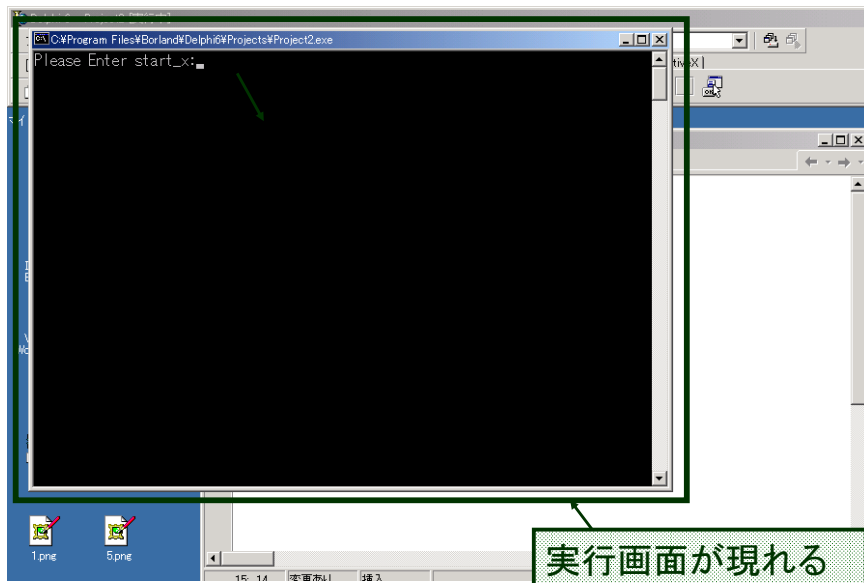
# 実行



# 実行(1/4)



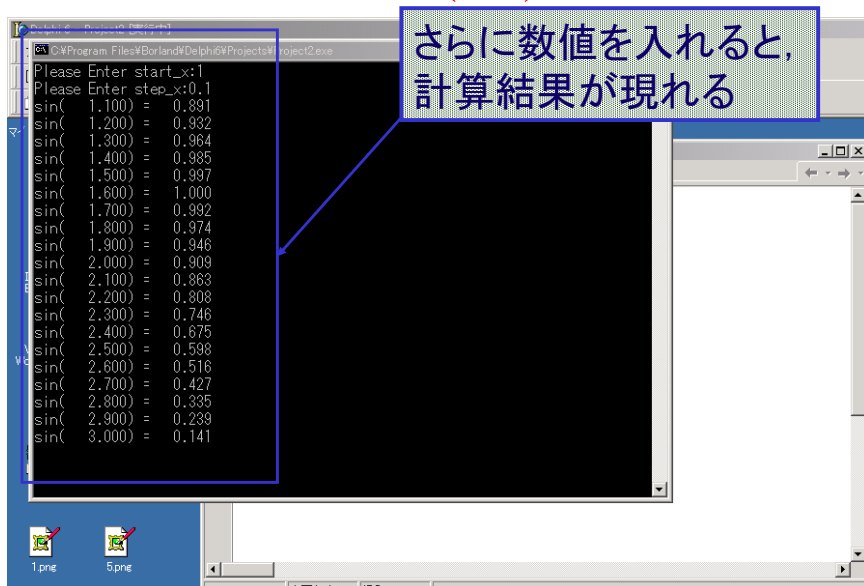
## 実行(2/4)



## 実行(3/4)



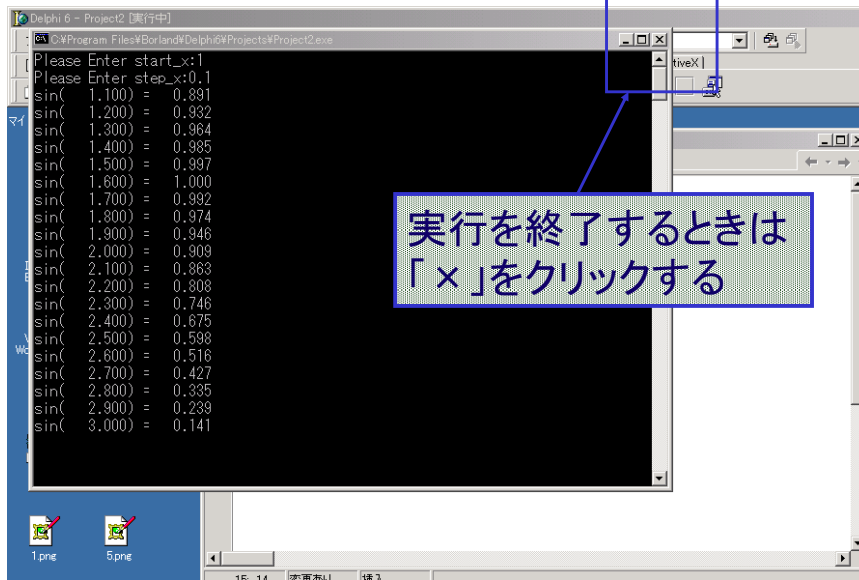
## 実行(4/4)



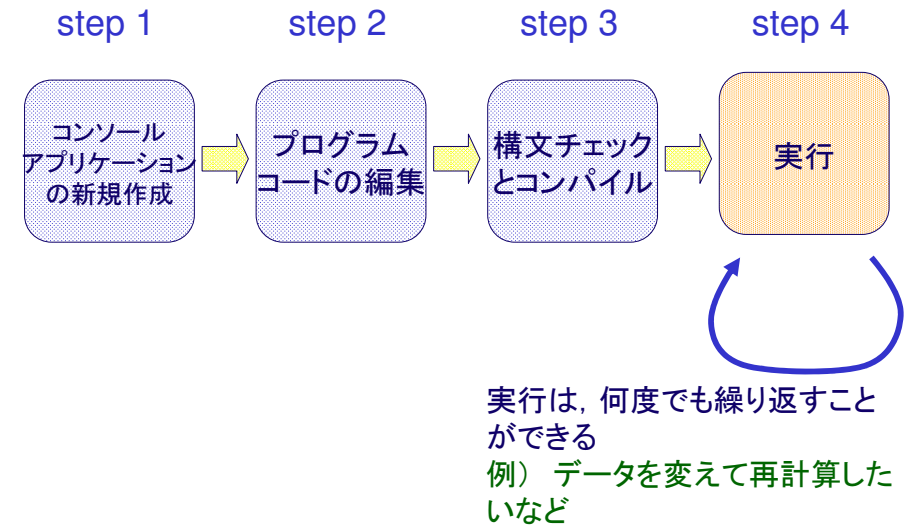
## 実行

- ① 「実行」→「実行」を選ぶ  
実行画面が現れる
- ② 実行画面で、数値データなどを入れる  
順次、計算結果が現れる

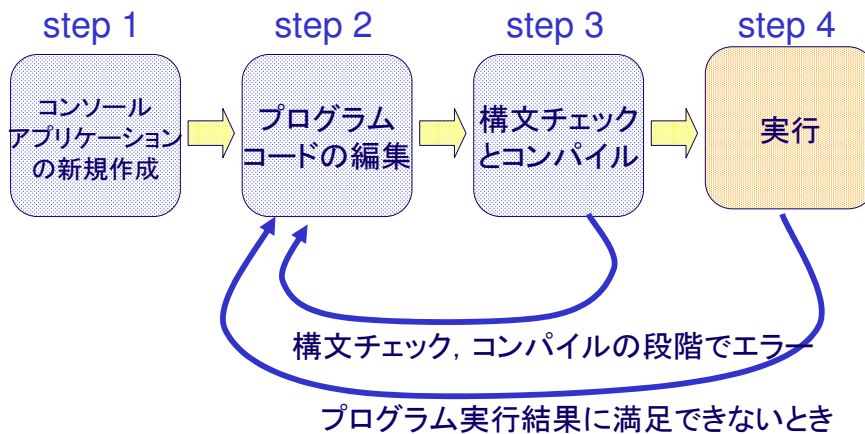
## 実行の終了



## 実行の繰り返し



## プログラムコードの編集の繰り返し



プログラムコードを「変更」したら、実行の前に再度の構文チェックと、コンパイルを行うこと

## 課題1. Delphi 6 でのプログラム作成と実行

- 次のプログラム(例題1のプログラム)について、この資料の手順を繰り返し、「実行」まで至りなさい

```
program sum;
{$APPTYPE CONSOLE}
uses SysUtils;
var start_x, step_x, x, y:real;
var i:integer;
begin
  write('Please Enter start_x:');
  readln(start_x);
  write('Please Enter step_x:');
  readln(step_x);
  for i:=1 to 20 do
  begin
    x := start_x + ( i * step_x );
    y := sin(x);
    writeln('sin(', x:8:3, ') =', y:8:3);
  end;
  readln
end.
```

## 課題1について

- この資料に記載の手順に従うこと
- プログラムそのものは、この資料ファイルからコピーできる(「コピー」と「貼り付け」)
- 操作を間違っても、基本的には、「作成したプログラム」が消えることはない(ある手順を踏めば、たいてい、復活できる)

## 課題2

- 例題1 のプログラムを実行して、 $\sin(0.4)$  の値を報告しなさい

## 課題3

- 例題1のプログラムの「sin」の部分を、わざと間違えて(「son」のように)、Borland Delphi 6 の構文チェック機能が有効に働いていることを確認しなさい

## 課題4

- 例題1のプログラムを  $\cos$  の値を計算できるように適当に書き換えた後に、構文チェック、コンパイル、実行を繰り返しなさい。  $\cos(0.4)$  の値を報告しなさい。