

ce-2. Microsoft Visual Studio C++ を使ってみよう

(Cプログラミング応用) (全14回)

URL: <https://www.kkaneko.jp/pro/c/index.html>

金子邦彦



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
- 次ページのプログラムを使って, Microsoft Visual Studio C++でのC++ソースファイル編集, ビルド, テスト実行の一連の過程を体験する

例題 1 のプログラムの機能



- 計算の繰り返し
- キーボードからのデータ読み込み
- ファイルへの書き出し

```
#include "stdio.h"
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;

    printf( "start_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &start_x );
    printf( "step_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &step_x );

    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f¥n", x, y );
        fprintf( fp, "x=, %f, y=, %f¥n", x, y );
    }
    fprintf( stderr, "file d:¥¥data.csv created¥n" );
    fclose( fp );
    return 0;
}
```

データファイル名 d:¥¥data.csv は適切に設定すること

キーボードからの
データ読み込みを
行っている部分

計算を行っている部分

ファイルへの書き出し
を行っている部分

```
#include "stdio.h"
#include <math.h>
#pragma warning(disable:4996)
int main()
{
```

Cプログラムはメインの関数から 実行開始



```
double x;
double y;
char buf[256];
int i;
double start_x;
double step_x;
FILE* fp;
```

変数 **x, y, buf, i, start_x,**
step_x, fp をメモリエリア中に確保

```
printf( "start_x = " );
fgets( buf, 256, stdin );
sscanf_s( buf, "%lf%n", &start_x );
```

「**start_x=**」というメッセージを表示し、
キーボードから数値を読み込む
(読み込んだ値は変数 **start_x** に格納)

```
printf( "step_x = " );
fgets( buf, 256, stdin );
sscanf_s( buf, "%lf%n", &step_x );
```

「**step_x=**」というメッセージを表示し、
キーボードから数値を読み込む
(読み込んだ値は **step_x** に格納)

```
fp = fopen( "d:¥¥data.csv", "w" );
```

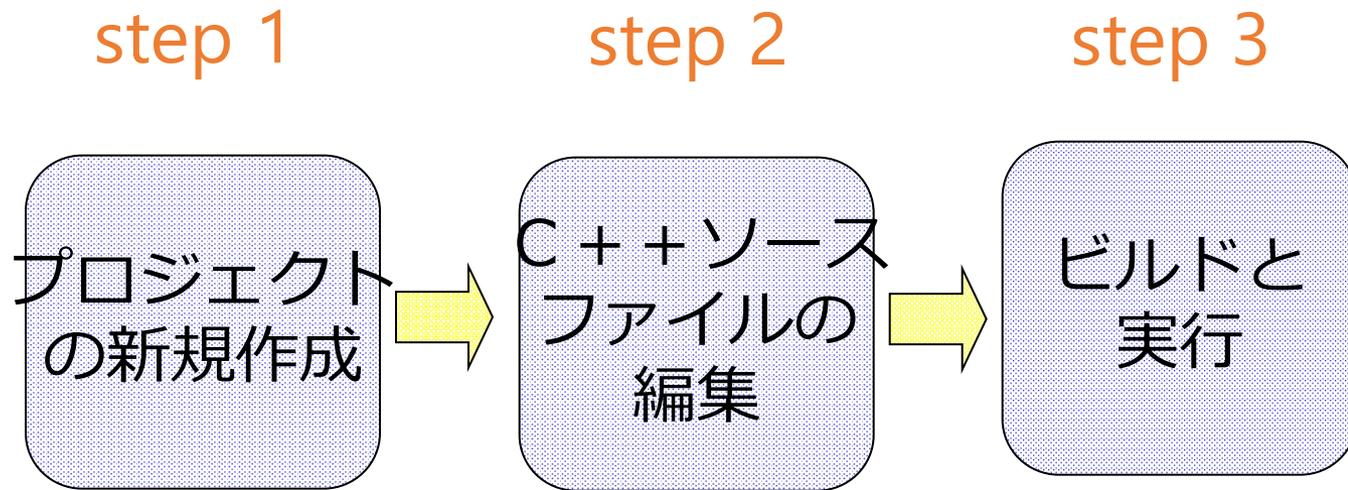
20回の繰り返し (i = 0, 1, ... 19)

```
for( i = 0; i < 20; i++ ) {
    x = start_x + ( i * step_x );
    y = sin( x );
    printf( "x= %f, y= %f¥n", x, y );
    fprintf( fp, "x=, %f, y=, %f¥n", x, y );
```

x の値から **sin(x)** を求め、
y に書き込む

```
}
fprintf( stderr, "file d:¥¥data.csv created¥n" );
fclose( fp );
return 0;
```

Microsoft Visual Studio C++での プログラム実行までの手順

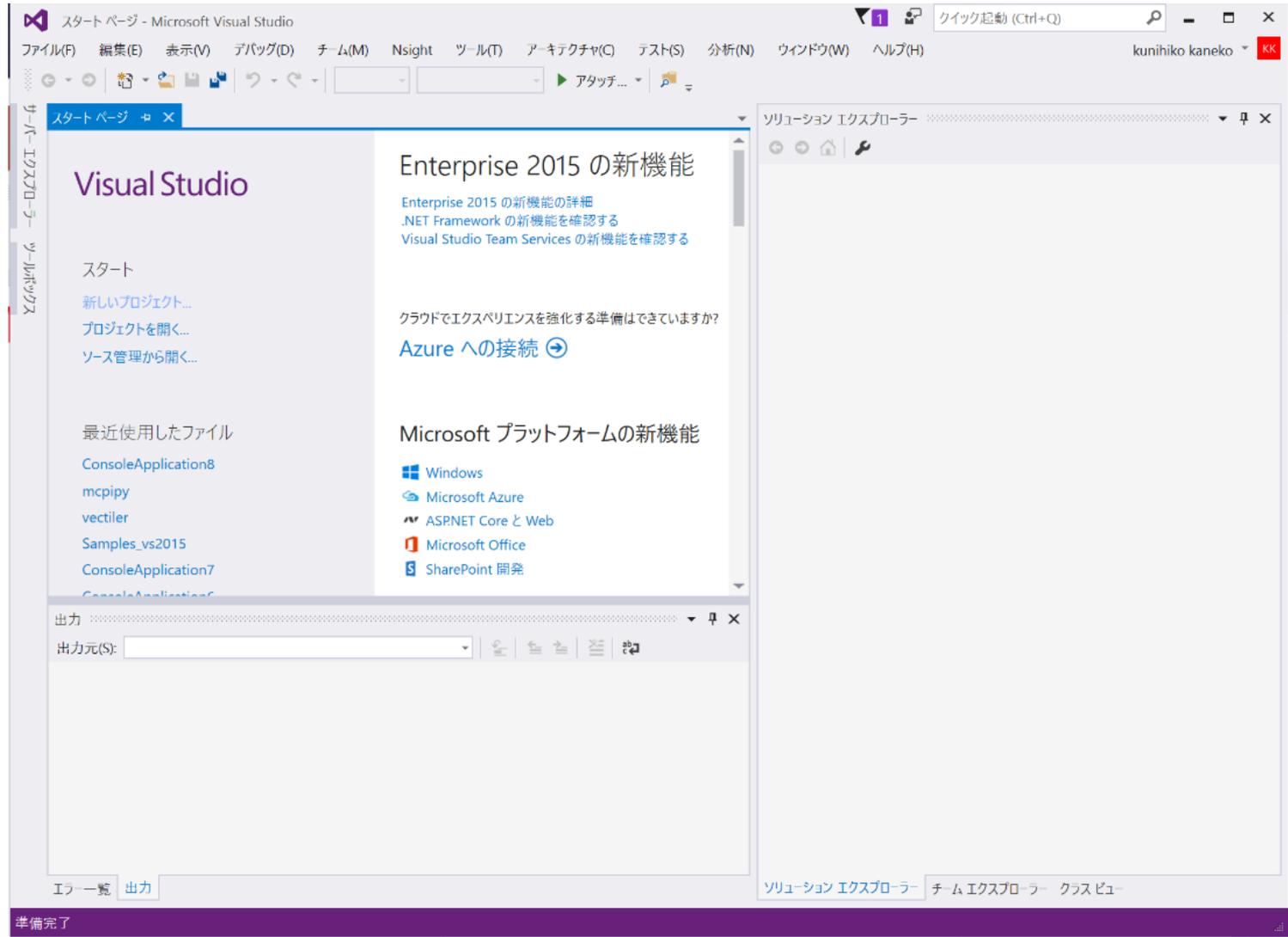


Microsoft Visual Studio C++ の起動(1/2)



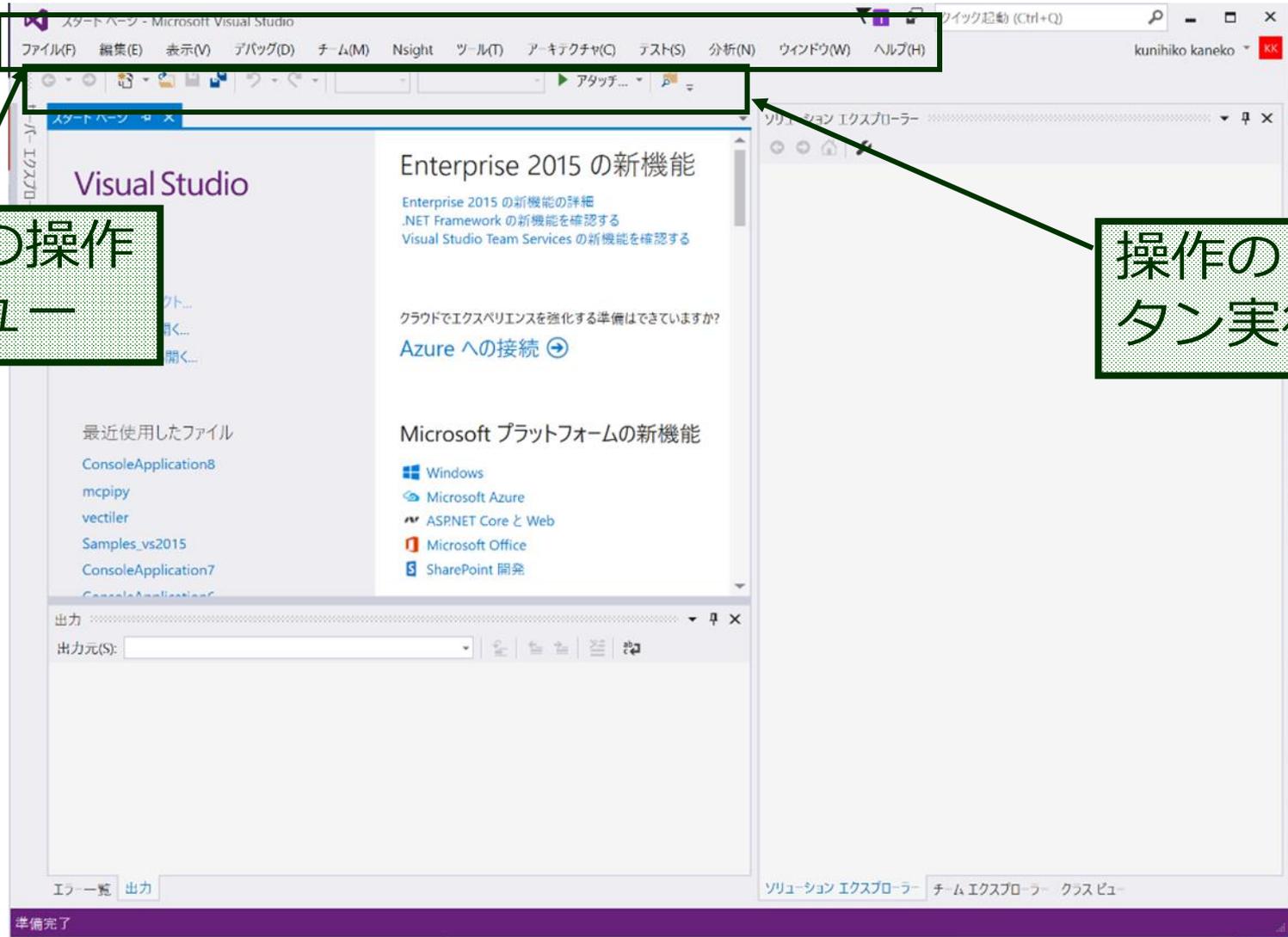
- 「スタート」
 - 「プログラム」
 - 「Visual Studio 2015」

Microsoft Visual Studio C++ の起動(2/2)



Microsoft Visual Studio 2015 の初回起動では、上のような画面が現

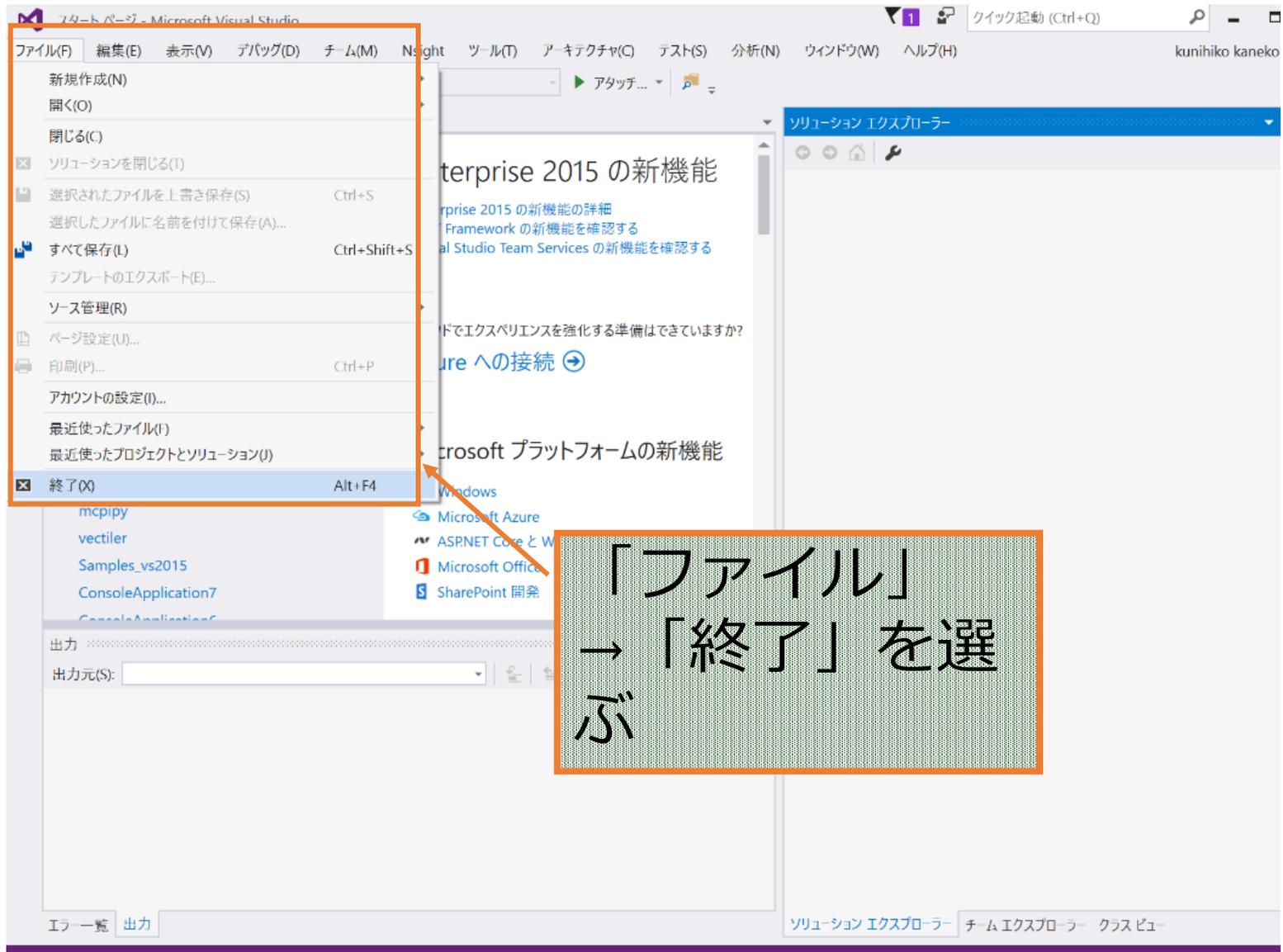
Microsoft Visual Studio C++ の起動(2/2)



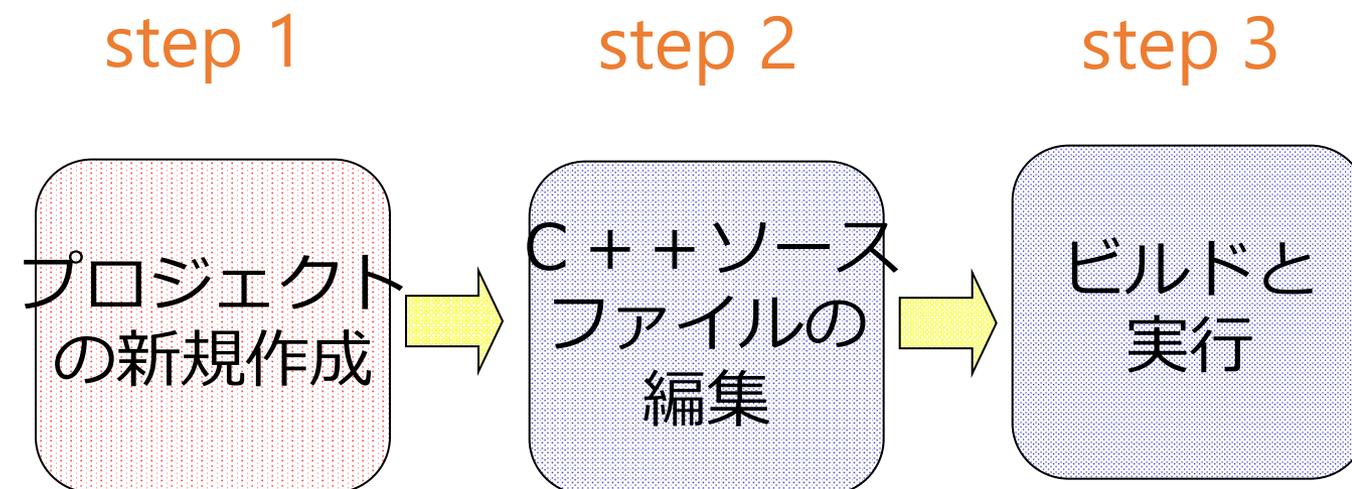
各種の操作
メニュー

操作の1ボ
タン実行

Microsoft Visual Studio C++ の終了



プロジェクトの新規作成



- 好きな「プロジェクト名」を入れる
- 好きな「場所」を入れる

プロジェクトの新規作成(1/8)



スタート ページ - Microsoft Visual Studio

ファイル(F) 編集(E) 表示(V) デバッグ(D) チーム(M) Nsight ツール(T) アーキテクチャ(C) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H) kunihiko kaneko

新規作成(N) ▶ プロジェクト(P)... Ctrl+Shift+N
開く(O) ▶ Web サイト(W)... Shift+Alt+N
閉じる(C) チーム プロジェクト(T)...
ソリューションを閉じる(I) ファイル(F)... Ctrl+N
選択されたファイルを上書き保存(S) Ctrl+S
選択したファイルに名前を付けて保存(A)...
すべて保存(L) Ctrl+Shift+S
既存のコードからプロジェクトを作成(E)...
Framework の新機能を確認する
Visual Studio Team Services の新機能を確認する

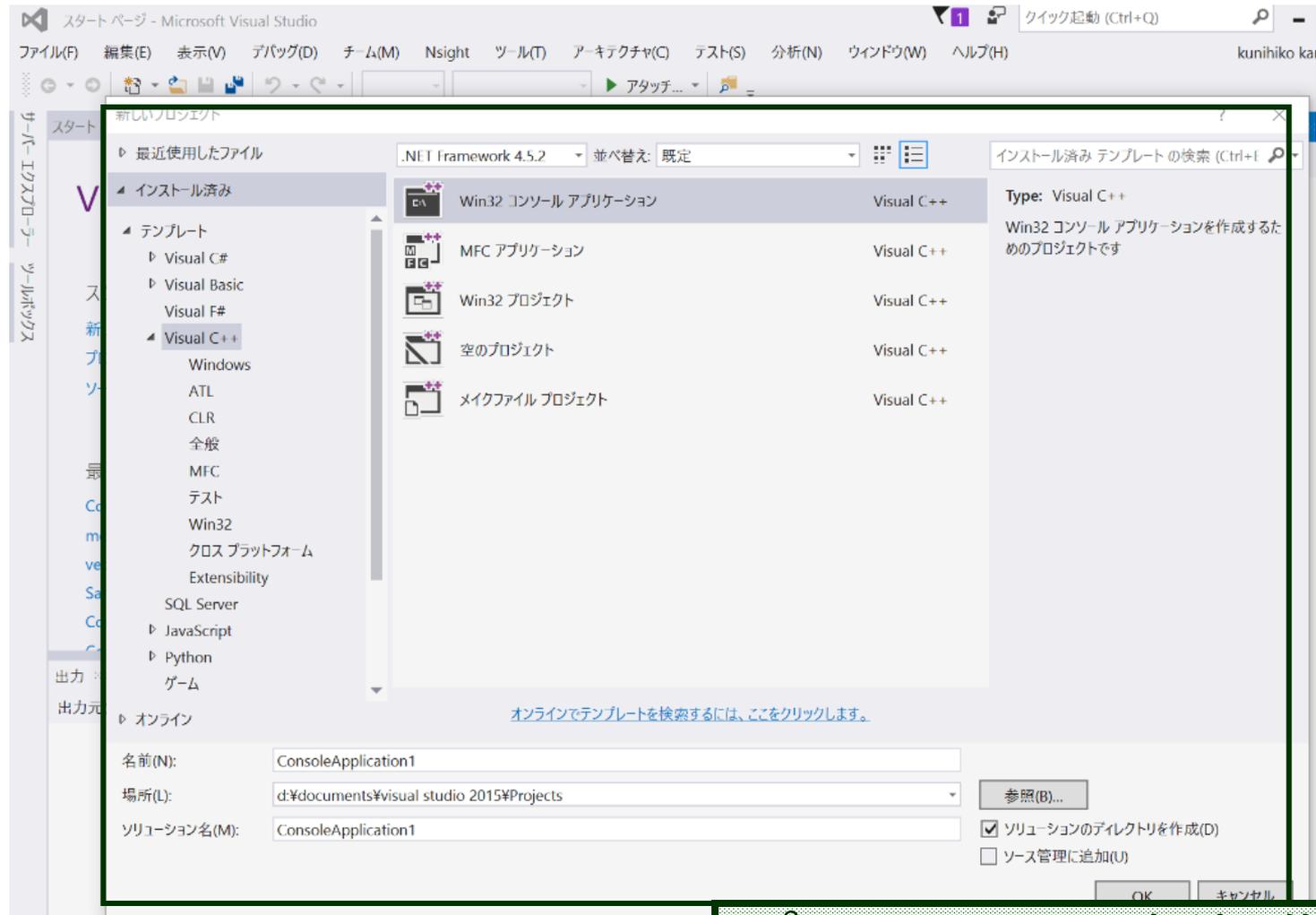
「ファイル」
→ 「新規作成」
→ 「プロジェクト」を選ぶ

出力
出力元(S):

エラー一覧 出力

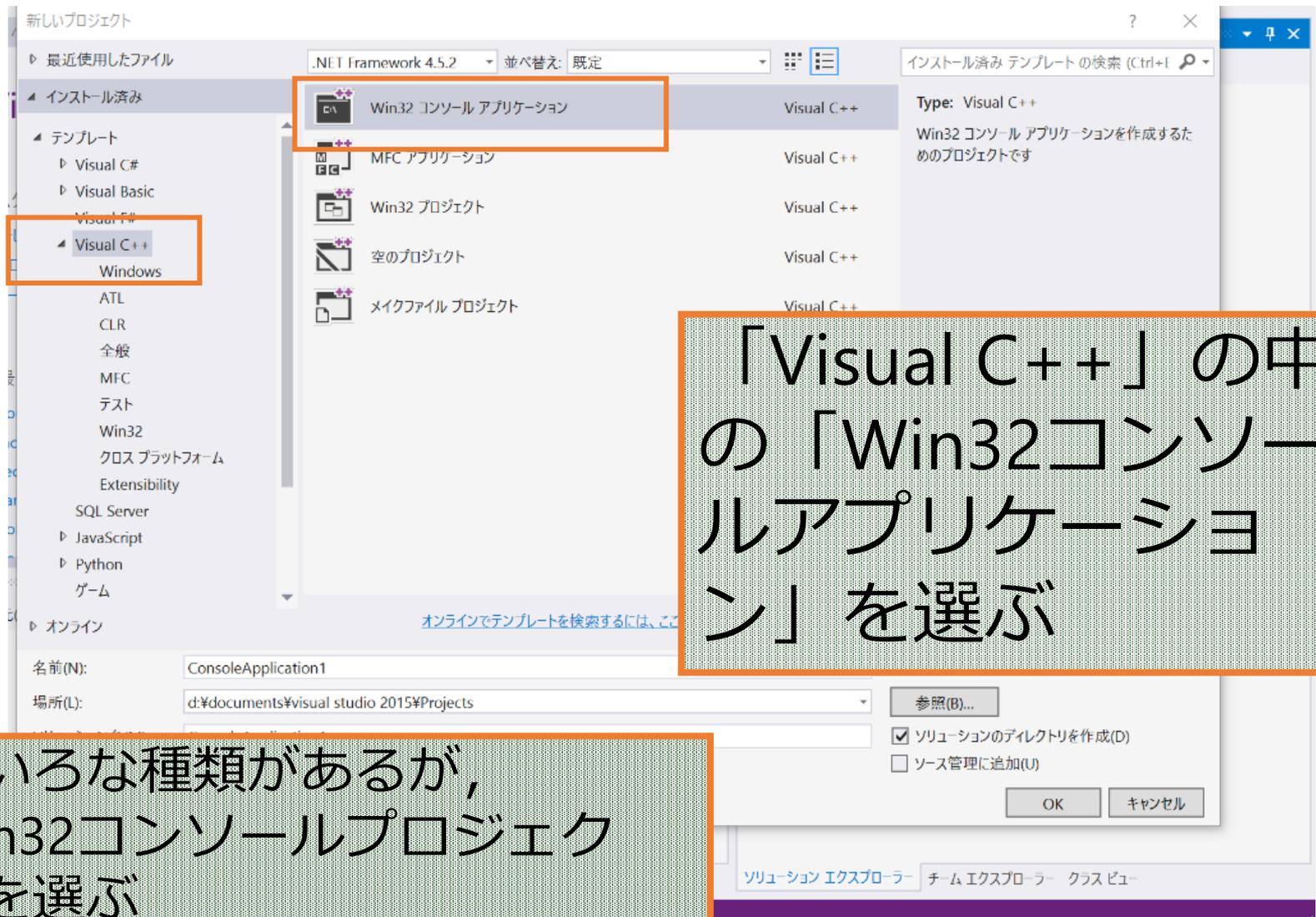
ソリューション エクスプローラー チーム エクスプローラー クラス ビュー

プロジェクトの新規作成(2/8)



プロジェクト新規作成のためのウィンドウが現れる

プロジェクトの新規作成(3/8)



「Visual C++」の中
の「Win32コンソール
アプリケーション」
を選ぶ

いろいろな種類があるが、
「Win32コンソールプロジェクト」
を選ぶ
(それ以外は使用しない)

プロジェクトの新規作成(4/8)



好きな「プロジェクト名」を入れる
ことができる (自分で命名できる)

名前(N): ConsoleApplication1
場所(L): d:\documents\visual studio 2015\Projects
ソリューション名(M): ConsoleApplication1

参照(B)...
 ソリューションのディレクトリを作成(D)
 ソース管理に追加(U)

OK キャンセル

プロジェクトの新規作成(5/8)



新しいプロジェクト

最近使用したファイル

.NET Framework 4.5.2 並べ替え: 既定

インストール済み テンプレートの検索 (Ctrl+I)

インストール済み

テンプレート

- Visual C#
- Visual Basic
- Visual F#
- Visual C++
 - Windows
 - ATL
 - CLR
 - 全般
 - MFC
 - テスト

テンプレート	言語	説明
Win32 コンソール アプリケーション	Visual C++	Win32 コンソール アプリケーションを作成するためのプロジェクトです
MFC アプリケーション	Visual C++	
Win32 プロジェクト	Visual C++	
空のプロジェクト	Visual C++	
メイクファイル プロジェクト	Visual C++	

Type: Visual C++

Win32 コンソール アプリケーションを作成するためのプロジェクトです

「場所」の設定もできる

USBメモリ, ネットワークドライブなどを使いたいときは, 場所の設定が必要

名前(N): ConsoleApplication1

場所(L): d:\documents\visual studio 2015\Projects

ソリューション名(M): ConsoleApplication1

参照(B)...

ソリューションのディレクトリを作成(D)

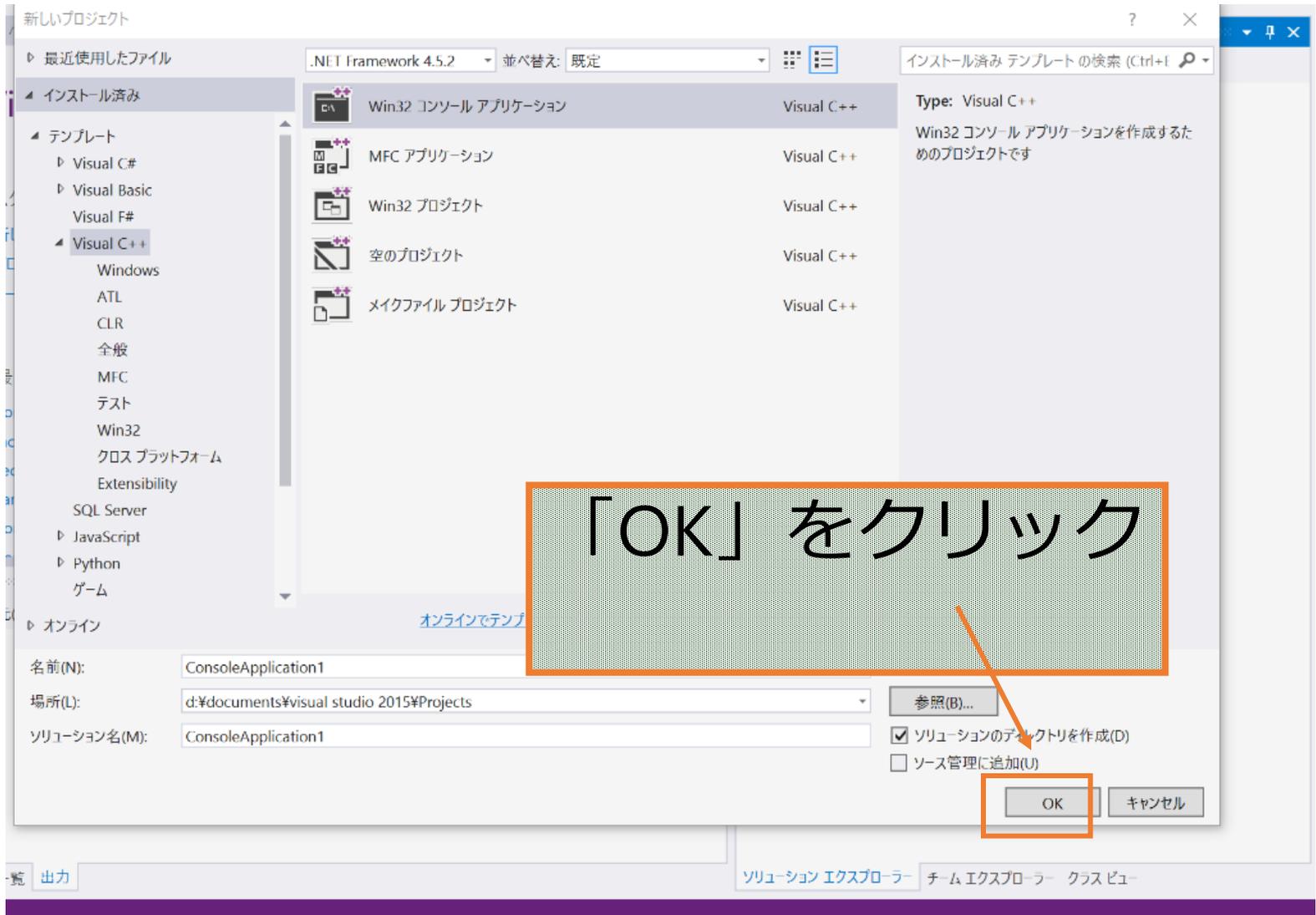
ソース管理に追加(U)

OK キャンセル

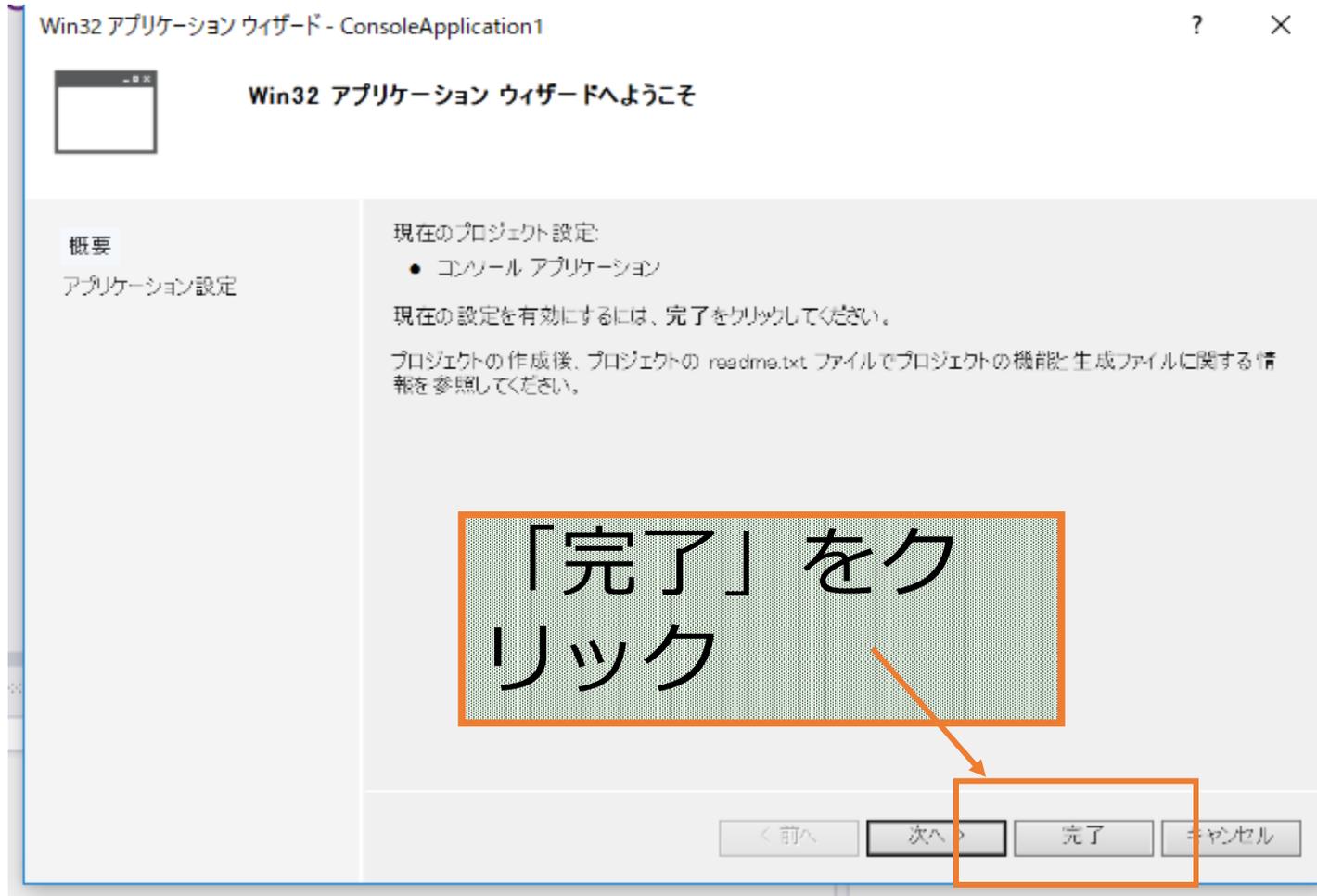
出力

ソリューション エクスプローラー チーム エクスプローラー クラス ビュー

プロジェクトの新規作成(6/8)



プロジェクトの新規作成(7/8)



プロジェクトの新規作成



ConsoleApplication1 - Microsoft Visual Studio

```
1 // ConsoleApplication1.cpp : コンソール アプリケーションのエントリ ポイントを定義します。
2
3
4 #include "stdafx.h"
5
6
7 int main()
8 {
9     return 0;
10 }
11
12
```

ソリューション エクスプローラー

- ソリューション 'ConsoleApplication1' (1 プロジェクト)
- ConsoleApplication1
 - 参照
 - 外部依存関係
 - ソース ファイル
 - ConsoleApplication1.cpp
 - stdafx.cpp
 - ヘッダー ファイル
 - stdafx.h
 - targetver.h
 - リソース ファイル
 - ReadMe.txt

出力

出力元(S):

100 %

エクスプローラー 出力

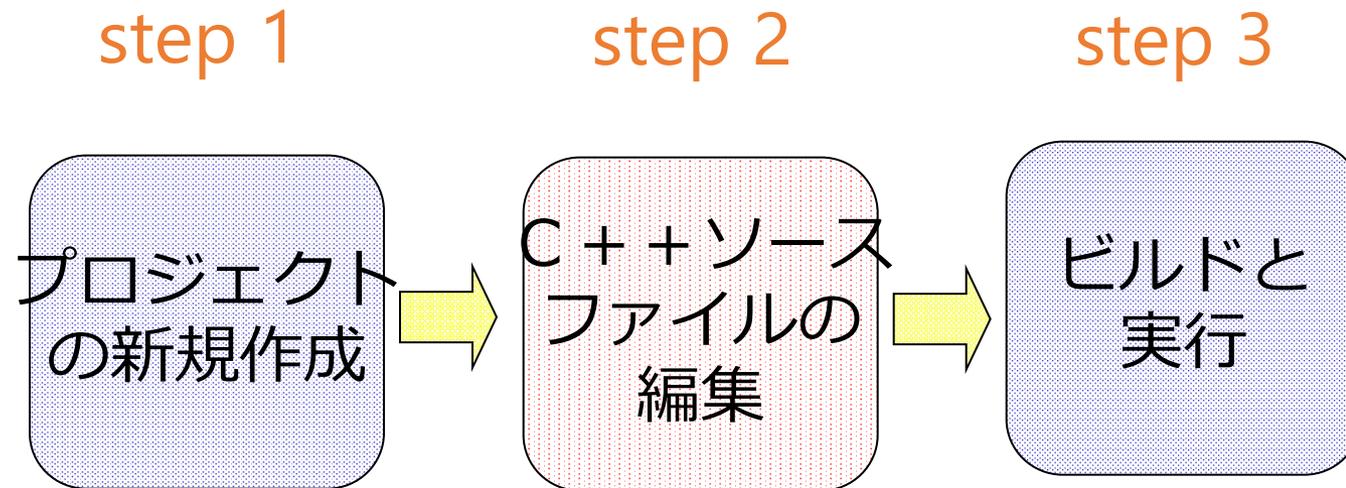
プロジェクトが新規作成されるので、確認する

プロジェクトの新規作成



- ① 「新規作成」 → 「プロジェクト」 を選ぶ
- ② 新規作成のためのウィンドウが現れる
- ③ プロジェクトの選択
「Win32 コンソールプロジェクト」 を選ぶ
- ④ プロジェクト名の入力
好きな「プロジェクト名」 を入れることができる
- ⑤ 場所の入力
好きな「場所」 を入れることができる
- ⑥ Win32アプリケーションウィザード
「完了」 を選ぶ
- ⑦ 先ほど入力した「プロジェクト名」 が現れるので、確認する

C++ソースファイルの編集



Microsoft Visual Studio C++ の画面構成



The screenshot displays the Microsoft Visual Studio C++ IDE interface. The main editor window shows the source code for `ConsoleApplication1.cpp`, with a green box highlighting the `int main()` function and the text "C++ソースファイルの編集はここで行う". The Solution Explorer on the right shows the project structure, with a green box highlighting the text "ファイルなどが表示される". The Output window at the bottom left shows the build results, with a green box highlighting the text "ビルド結果が現れる".

```
1 // ConsoleApplication1.cpp : コンソール アプリケーションのエントリ ポイントを定義します。
2 //
3
4 #include "stdafx.h"
5
6
7 int main()
8 {
9
10
11
12 }
```

出力

出力元(S):

ビルド結果が現れる

ファイルなどが表示される

C++ ソースファイルの編集

5
6
7 int main()
8 {
9 return 0;
10 }
11
12. A green box highlights the code editor with the text '編集画面'. Another green box highlights 'ConsoleApplication1.cpp' in the Solution Explorer with the text '編集集中のファイル名'. The bottom right shows the Properties window for the project."/>

ConsoleApplication1 - Microsoft Visual Studio

ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T) アーキテクチャ(C) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H) kunihiko kaneko

```
1 // ConsoleApplication1.cpp : コンソール アプリケーションのエントリ ポイント  
2 //  
3  
4 #include "stdafx.h"  
5  
6  
7 int main()  
8 {  
9     return 0;  
10 }  
11  
12
```

ソリューション エクスプローラー

ソリューション エクスプローラー の検索 (Ctrl+;)

ソリューション 'ConsoleApplication1' (1 プロジェクト)

- ConsoleApplication1
 - 参照
 - 外部依存関係
 - ソース ファイル
 - ConsoleApplication1.cpp
 - stdafx.cpp
 - ヘッダー ファイル
 - stdafx.h
 - target
 - リソース ファイル
 - ReadMe.t

ソリューション エクスプローラー チーム エクスプローラー クラス ビュー

プロパティ ConsoleApplication1 プロジェクトのプロパティ

その他

(名前)	ConsoleApplication1
プロジェクト ファイル	d:\documents\visual studio 2015\Projects\%
プロジェクトの依存関係	
ルート名前空間	ConsoleApplication1

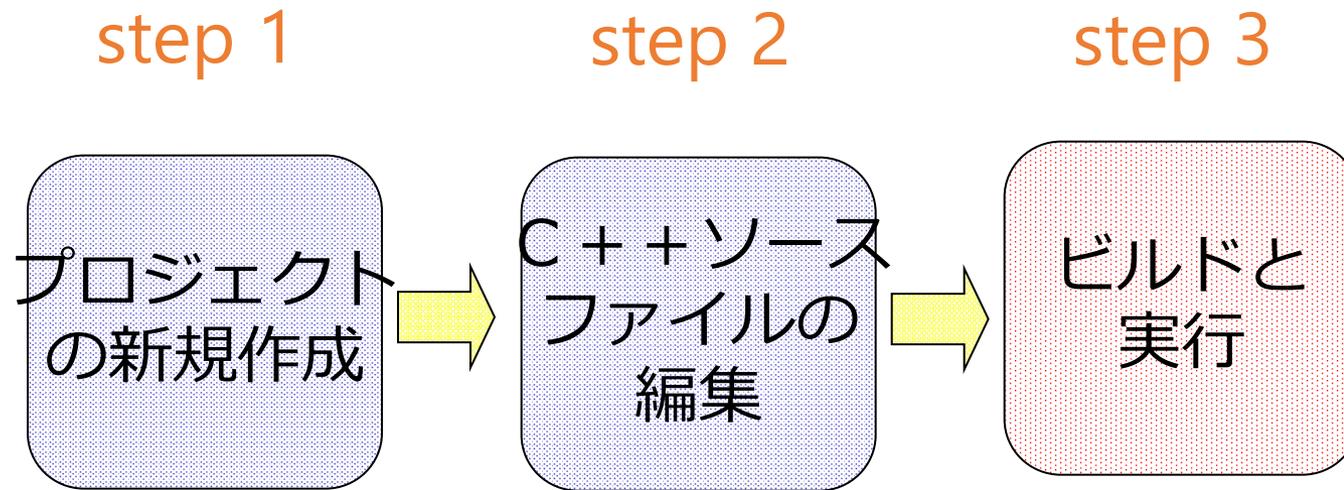
(名前)
プロジェクト名を指定します。

出力

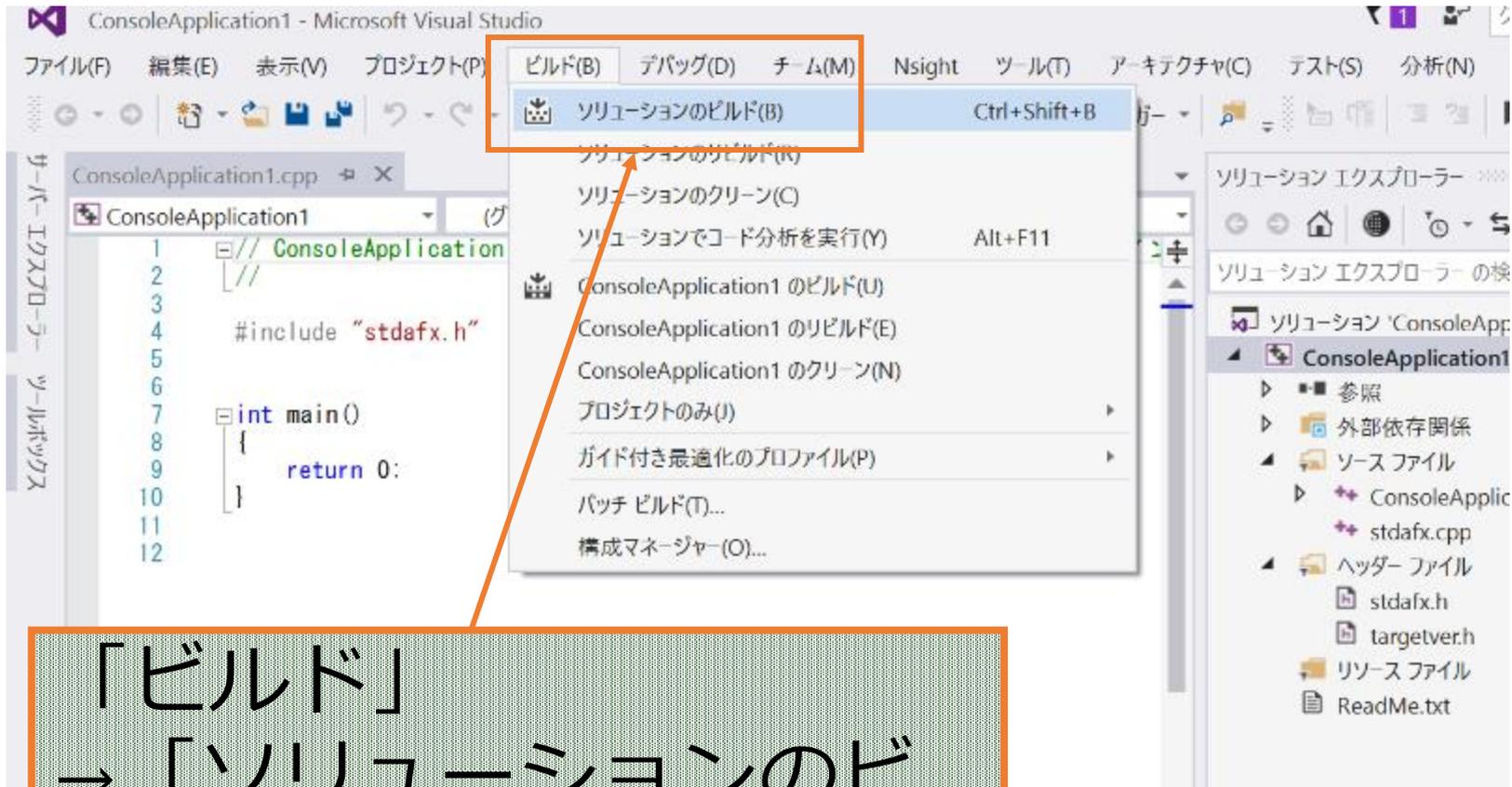
出力元(S):

エラー一覧 出力

Microsoft Visual Studio C++での プログラム実行までの手順



ビルドと実行(1/6)



The screenshot shows the Visual Studio interface for a project named 'ConsoleApplication1'. The 'ビルド(B)' menu is open, and 'ソリューションのビルド(B)' is selected. The menu items include: ソリューションのビルド(B), ソリューションのクリーン(C), ソリューションでコード分析を実行(Y) (Alt+F11), ConsoleApplication1 のビルド(U), ConsoleApplication1 のリビルド(E), ConsoleApplication1 のクリーン(N), プロジェクトのみ(I), ガイド付き最適化のプロファイル(P), バッチ ビルド(T)..., and 構成マネージャー(O)...

「ビルド」
→ 「ソリューションのビルド」を選ぶ

ビルドと実行(2/6)



ConsoleApplication1 - Microsoft Visual Studio

ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T) アーキテクチャ(C)

Debug x86 ローカル Windows デバッガー

```
1 // ConsoleApplication1.cpp : コンソール アプリケーションのエントリ ポイントを定義します。
2 //
3
4 #include "stdafx.h"
5
6
7 int main()
8 {
9     return 0;
10 }
11
12
```

ビルド結果が現れる。「1 正常終了, 0 失敗・・・」ならばビルドに成功

出力

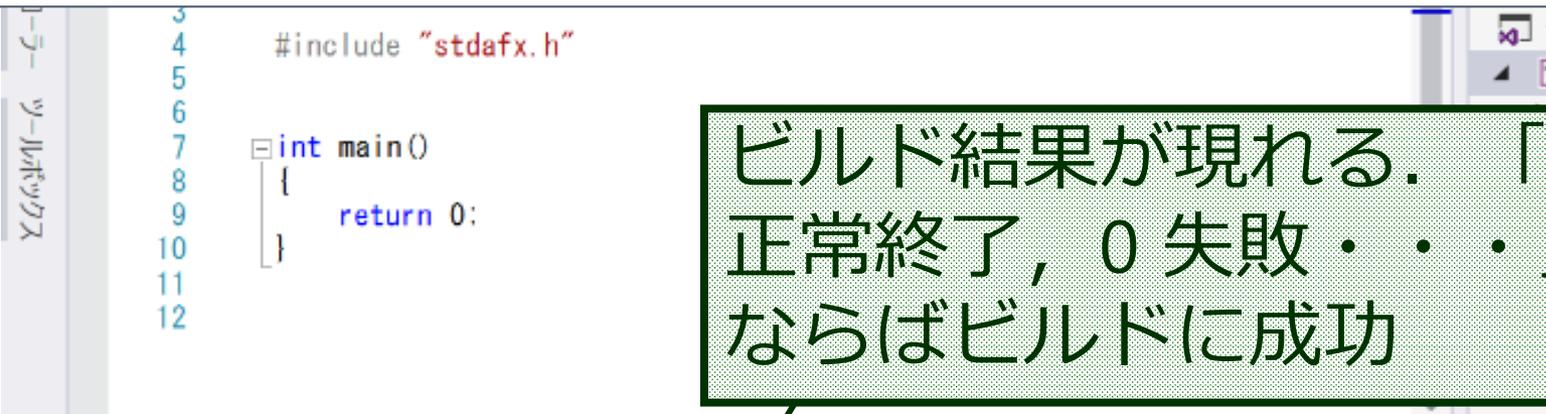
出力元(S): ビルド

```
1>----- ビルド開始: プロジェクト:ConsoleApplication1, 構成:Debug Win32 -----
1> stdafx.cpp
1> ConsoleApplication1.cpp
1> ConsoleApplication1.vcxproj -> d:\documents\visual studio 2015\projects\ConsoleApplication1\ConsoleApplication1.vcxproj
1> ConsoleApplication1.vcxproj -> d:\documents\visual studio 2015\projects\ConsoleApplication1\ConsoleApplication1.vcxproj
----- ビルド: 1 正常終了, 0 失敗, 0 更新不要, 0 スキップ -----
```

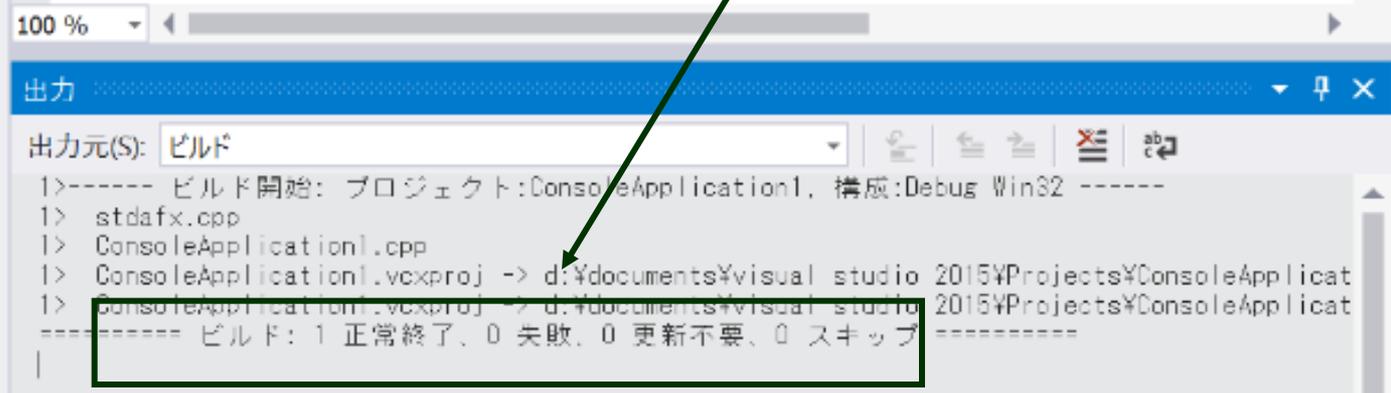
ビルドと実行(2/6)



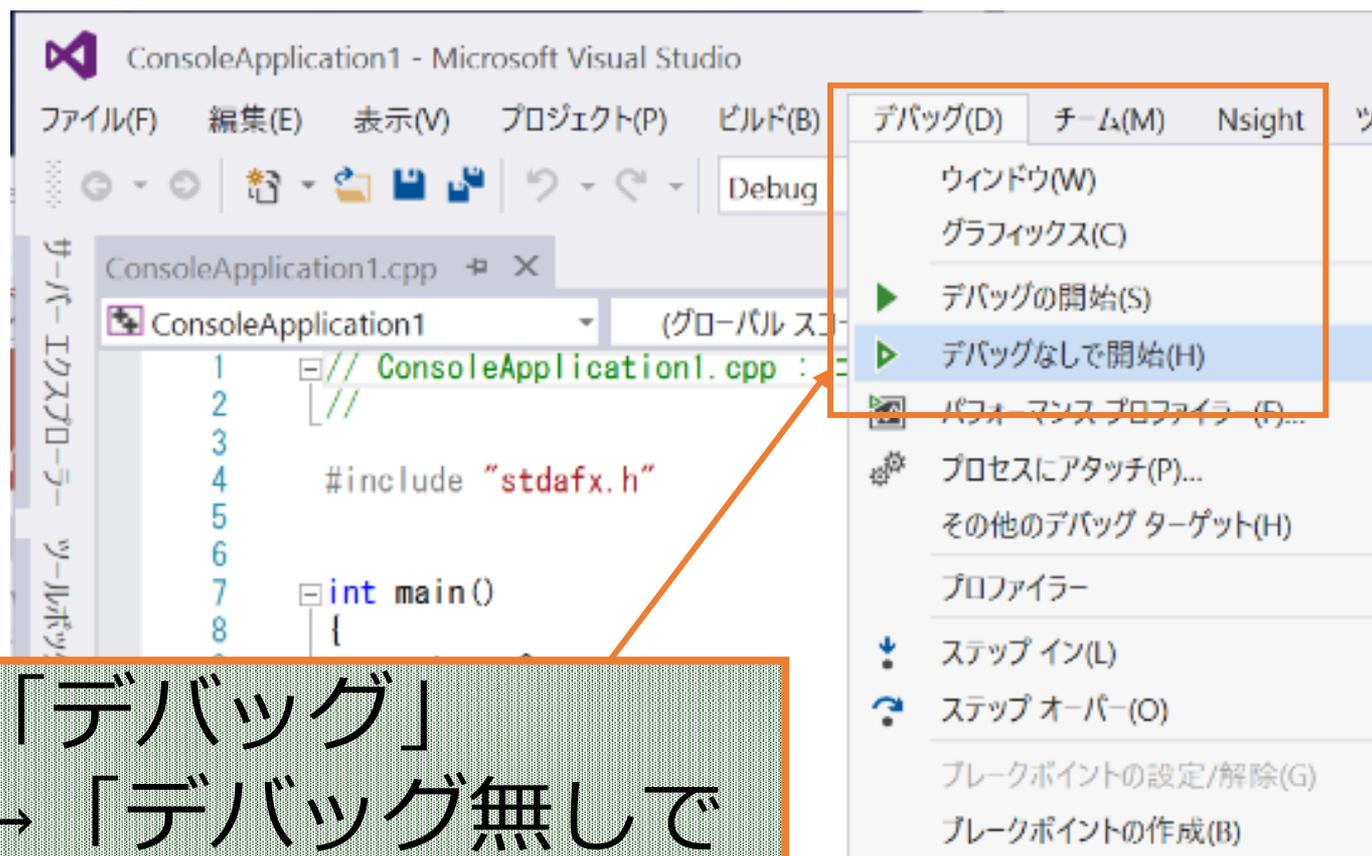
ビルドが正常終了するまで、プログラム編集を続ける
・文法ミスのある場所、種類は、コンピュータが探し指摘してくれる



ビルド結果が現れる。「1
正常終了, 0 失敗・・・」
ならばビルドに成功



ビルドと実行(3/6)



「デバッグ」
→ 「デバッグ無しで
開始」を選ぶ

当然だが、ビルドに成功し
ていないと、この操作は意
味が無い

ビルドと実行(4/6)



実行画面が現れる

ビルドと実行について



C:\WINDOWS\system32\cmd.exe

start_x =

この資料のページ5のプログラムを使っているときは、実行画面のメッセージが少し違う。
(次ページに続く)

ビルドと実行について



C:\> C:\WINDOWS\system32\cmd.exe

```
start_x = 0  
step_x = 0.1
```

数値を入れる

(プログラムに数値データを与える)

```
start_x = 0
```

```
step_x = 0.1
```

ビルドと実行について



```
C:\C:\WINDOWS\system32\cmd.exe
start_x =0
step_x =0.1
x= 0.000000, y= 0.000000
x= 0.100000, y= 0.099833
x= 0.200000, y= 0.198669
x= 0.300000, y= 0.295520
x= 0.400000, y= 0.389418
x= 0.500000, y= 0.479426
x= 0.600000, y= 0.564642
x= 0.700000, y= 0.644218
x= 0.800000, y= 0.717356
x= 0.900000, y= 0.783327
x= 1.000000, y= 0.841471
x= 1.100000, y= 0.891207
x= 1.200000, y= 0.932039
x= 1.300000, y= 0.963558
x= 1.400000, y= 0.985450
x= 1.500000, y= 0.997495
x= 1.600000, y= 0.999574
x= 1.700000, y= 0.991665
x= 1.800000, y= 0.973848
x= 1.900000, y= 0.946300
file z:\ydata.csv created
続行するには何かキーを押してください...
```

2つの数値を入れると、
計算結果の結果が得られる。

プログラム実行結果



```
C:\WINDOWS\system32\cmd.exe
start_x =0
step_x =0.1
x= 0.000000, y= 0.000000
x= 0.100000, y= 0.099833
x= 0.200000, y= 0.198669
x= 0.300000, y= 0.295520
x= 0.400000, y= 0.389418
x= 0.500000, y= 0.479426
x= 0.600000, y= 0.564642
x= 0.700000, y= 0.644218
x= 0.800000, y= 0.717356
x= 0.900000, y= 0.783327
x= 1.000000, y= 0.841471
x= 1.100000, y= 0.891207
x= 1.200000, y= 0.932039
x= 1.300000, y= 0.963558
x= 1.400000, y= 0.985450
x= 1.500000, y= 0.997495
x= 1.600000, y= 0.999574
x= 1.700000, y= 0.991665
x= 1.800000, y= 0.973848
x= 1.900000, y= 0.946300
file z:\data.csv created
続行するには何かキーを押してください...
```

キーボードから、データ「0」、「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

ビルドと実行



- ① 「ビルド」 → 「ソリューションのビルド」 を選ぶ
ビルド結果が現れる
- ② 「デバッグ」 → 「デバッグ無しで開始」 を選ぶ
- ③ 実行画面が出る

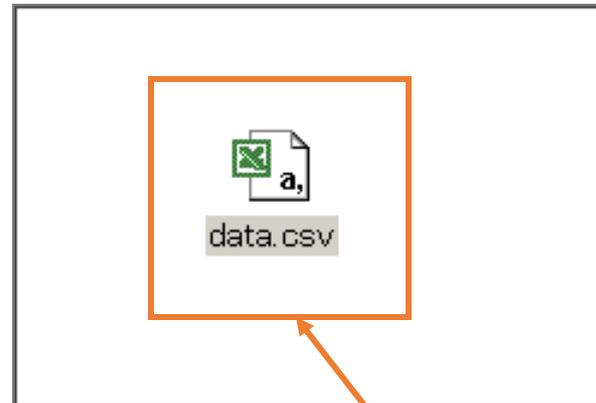
例題 2. 他のソフトとの連携



- 例題 1 のプログラムによって書き出されたファイルを使い, Microsoft Excel を使って折れ線グラフを作成する

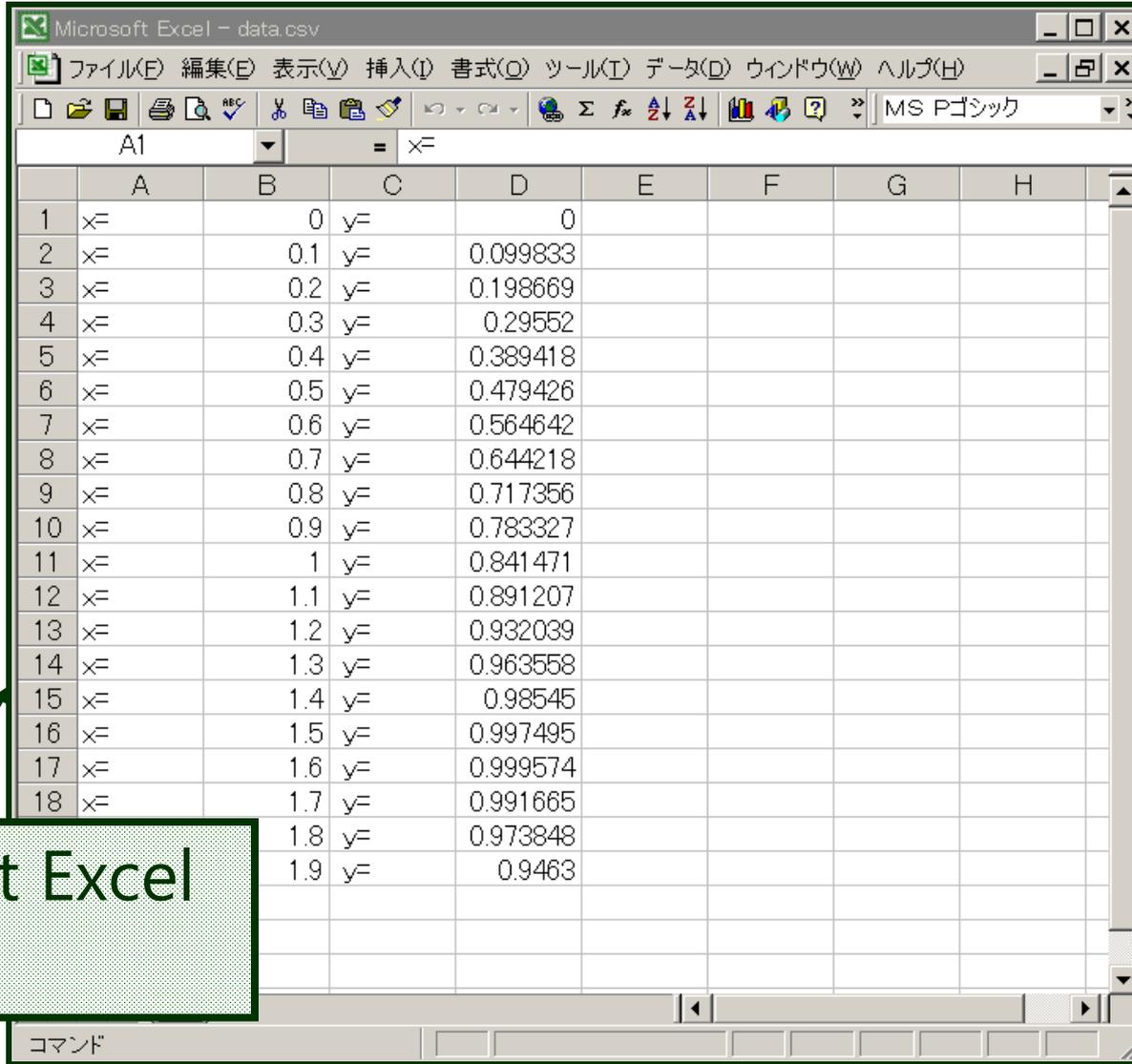
ファイル名 : d:¥data.csv

Microsoft Excel でグラフ作成 (1/8)



zドライブの
data.csv をダブルク
リックする

Microsoft Excel でグラフ作成 (2/8)



Microsoft Excel - data.csv

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)

MS Pゴシック

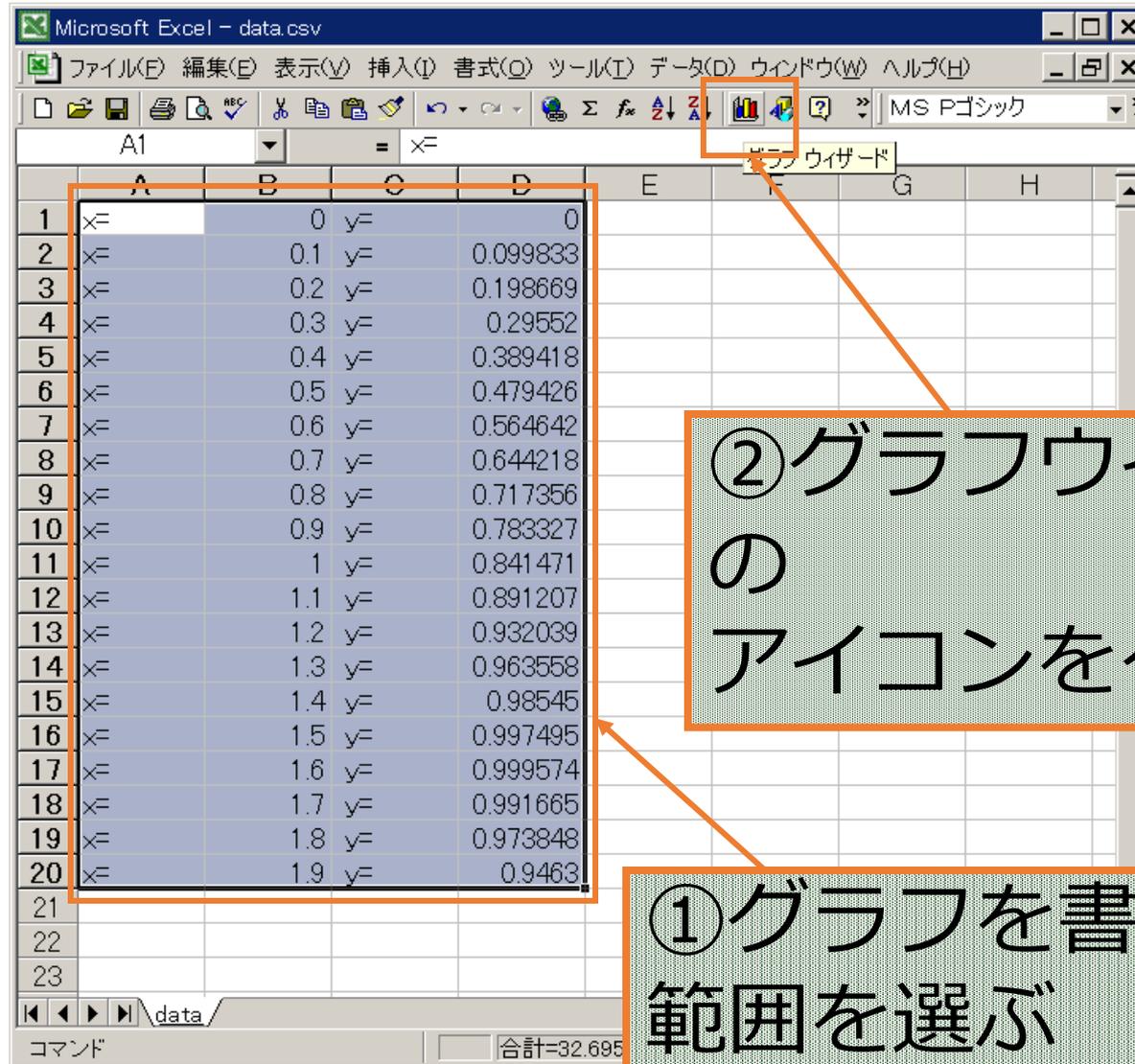
A1 = x^f

	A	B	C	D	E	F	G	H
1	x ^f	0	y ^f	0				
2	x ^f	0.1	y ^f	0.099833				
3	x ^f	0.2	y ^f	0.198669				
4	x ^f	0.3	y ^f	0.29552				
5	x ^f	0.4	y ^f	0.389418				
6	x ^f	0.5	y ^f	0.479426				
7	x ^f	0.6	y ^f	0.564642				
8	x ^f	0.7	y ^f	0.644218				
9	x ^f	0.8	y ^f	0.717356				
10	x ^f	0.9	y ^f	0.783327				
11	x ^f	1	y ^f	0.841471				
12	x ^f	1.1	y ^f	0.891207				
13	x ^f	1.2	y ^f	0.932039				
14	x ^f	1.3	y ^f	0.963558				
15	x ^f	1.4	y ^f	0.98545				
16	x ^f	1.5	y ^f	0.997495				
17	x ^f	1.6	y ^f	0.999574				
18	x ^f	1.7	y ^f	0.991665				
		1.8	y ^f	0.973848				
		1.9	y ^f	0.9463				

コマンド

Microsoft Excel
が現れる

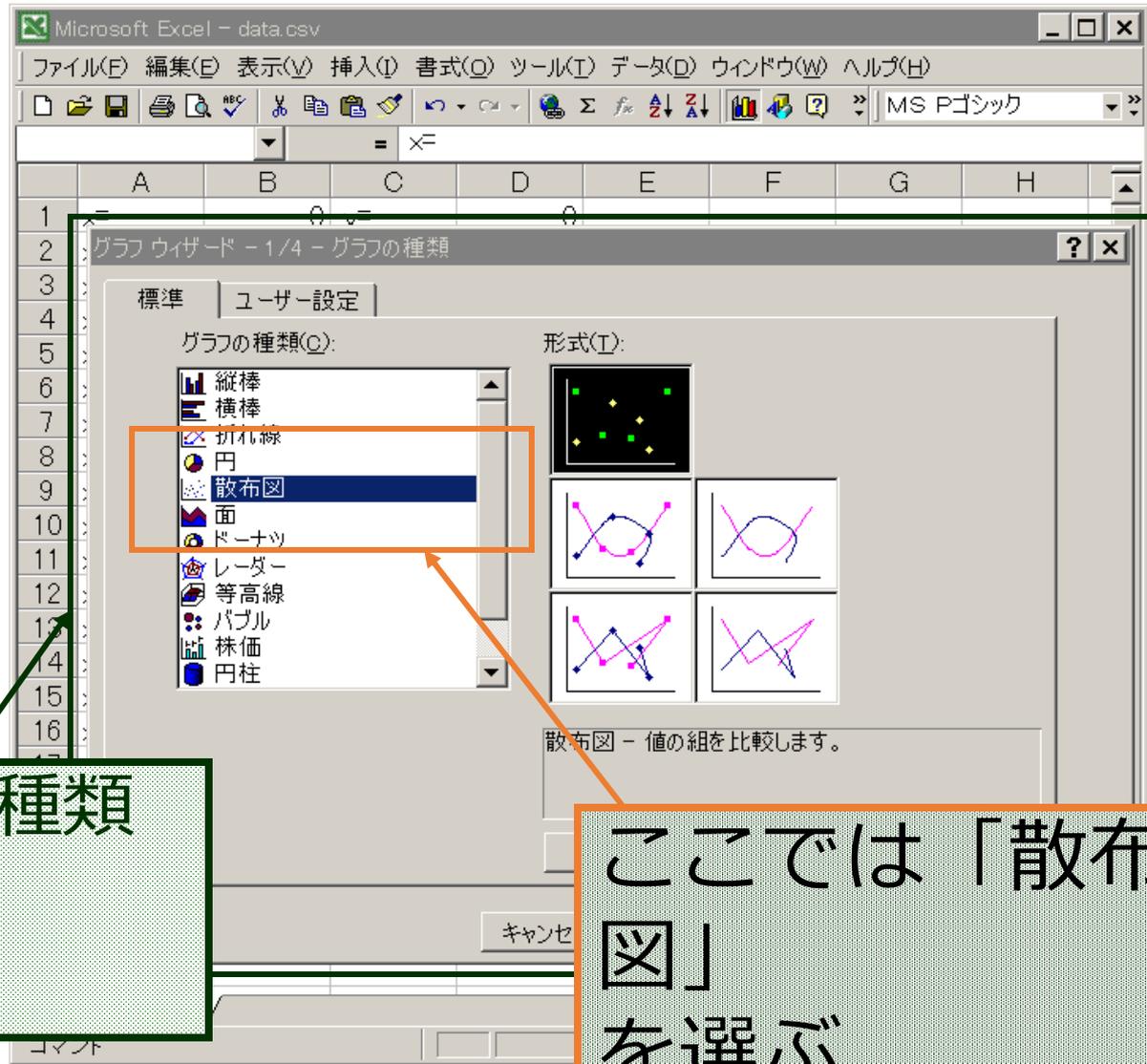
Microsoft Excel でグラフ作成 (3/8)



② グラフウィザードのアイコンをクリック

① グラフを書きたい範囲を選ぶ

Microsoft Excel でグラフ作成 (4/8)



Microsoft Excel - data.csv

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)

標準 ユーザー設定

グラフの種類(Q):

- 縦棒
- 横棒
- 折れ線
- 円
- 散布図**
- 面
- ドーナツ
- レーダー
- 等高線
- バブル
- 株価
- 円柱

形式(I):

散布図 - 値の組を比較します。

キャンセル

グラフの種類
を
選べる

ここでは「散布
図」
を選ぶ

Microsoft Excel でグラフ作成 (5/8)



Microsoft Excel - data.csv

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D)

グラフ ウィザード - 2/4 - グラフの元データ

データ範囲 系列

データ範囲(D): =data!\$A\$1:\$D\$20

系列: 行(R) 列(L)

次へ >

	A
1	x=
2	x=
3	x=
4	x=
5	x=
6	x=
7	x=
8	x=
9	x=
10	x=
11	x=
12	x=
13	x=
14	x=
15	x=
16	x=
17	x=

① グラフの形を確認

グラフ化すべき範囲の確認

② 「次へ」を押す

Microsoft Excel でグラフ作成 (6/8)



The screenshot shows Microsoft Excel with a data table and a chart wizard dialog box. The data table contains the following values:

	A	B	C	D	E	F	G	H
1	x=	0	y=	0				
2	x=	0.1	y=	0.099833				
3	x=	0.2	y=	0.198669				
4	x=	0.3	y=	0.29552				

The chart wizard dialog box is titled "グラフ ウィザード - 3/4 - グラフ オプション". It has tabs for "タイトルとラベル", "軸", "目盛線", "凡例", and "データラベル". The "凡例" tab is selected. The "凡例を表示する(S)" checkbox is checked. The "表示位置" section has radio buttons for "下(M)", "右上(Q)", "上(T)", "右(R)", and "左(L)". The "右(R)" option is selected. A preview of the chart is shown, with a legend box labeled "系列1". The "次へ >" button is highlighted with an orange box and an arrow pointing to it from a text box.

凡例などを選べる

「次へ」を押す

Microsoft Excel でグラフ作成 (7/8)



Microsoft Excel - data.csv

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)

MS Pゴシック

	A	B	C	D	E	F	G	H
1	x=		0 y=	0				
2	x=		0.1 y=	0.099833				
3	x=		0.2 y=	0.198669				
				0.29552				
				0.389418				
				0.479426				
				0.564642				
				0.644218				
				0.717356				
10	x=		0.9 y=	0.783327				
11	x=		1 y=	0.841471				
12	x=		1.1 y=	0.891207				
13	x=							
14	x=							
15	x=							
16	x=							
17	x=							
18	x=							
19	x=							
20	x=							
21								
22								
23								

グラフの作成場所
を選べる

グラフウィザード - 4/4 - グラフの作成場所

グラフの場所:

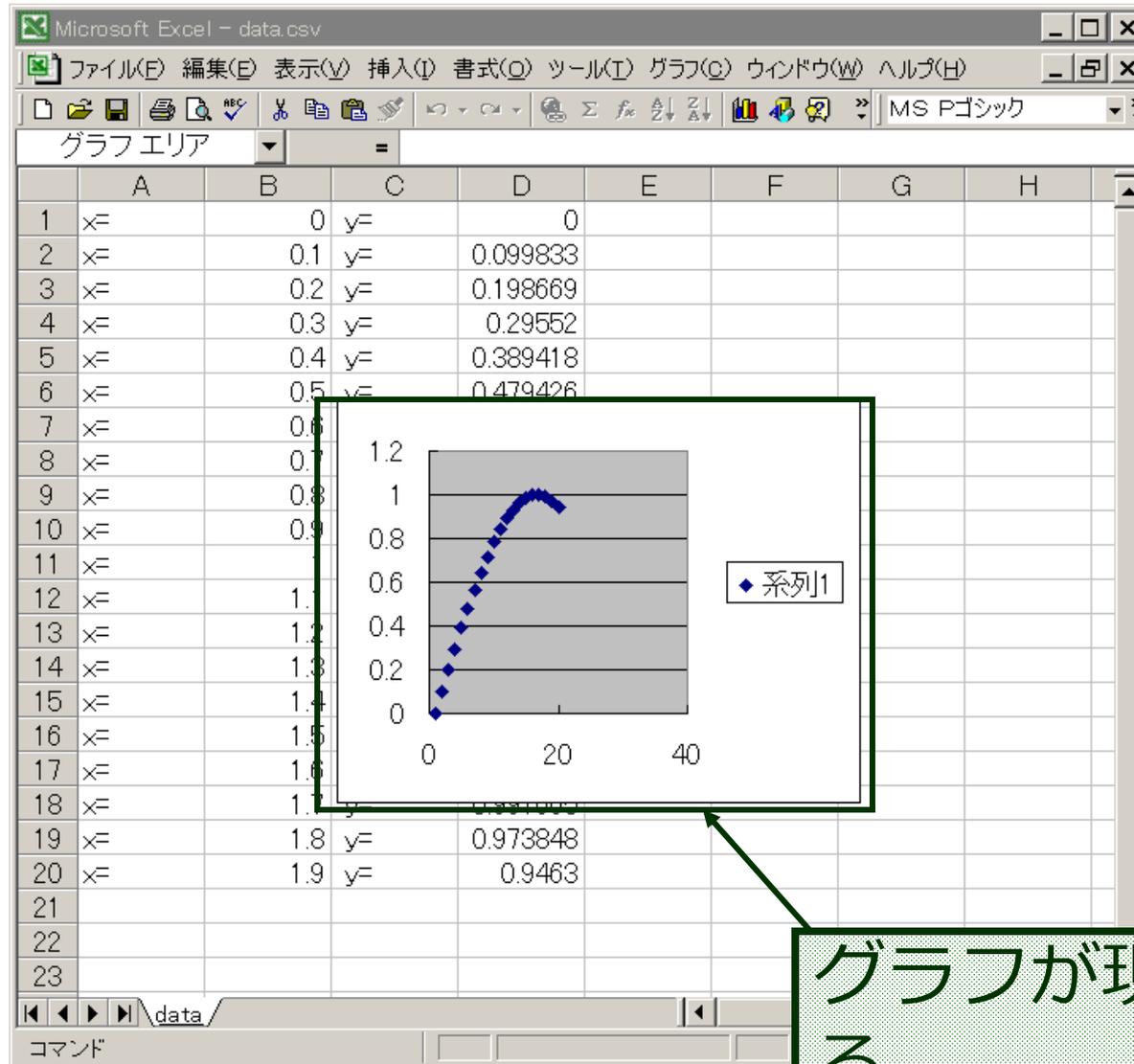
新しいシート(S): Graph2

オブジェクト(O): data

キャンセル < 戻る(B) 次へ > 完了(F)

「完了」を押す

Microsoft Excel でグラフ作成 (8/8)

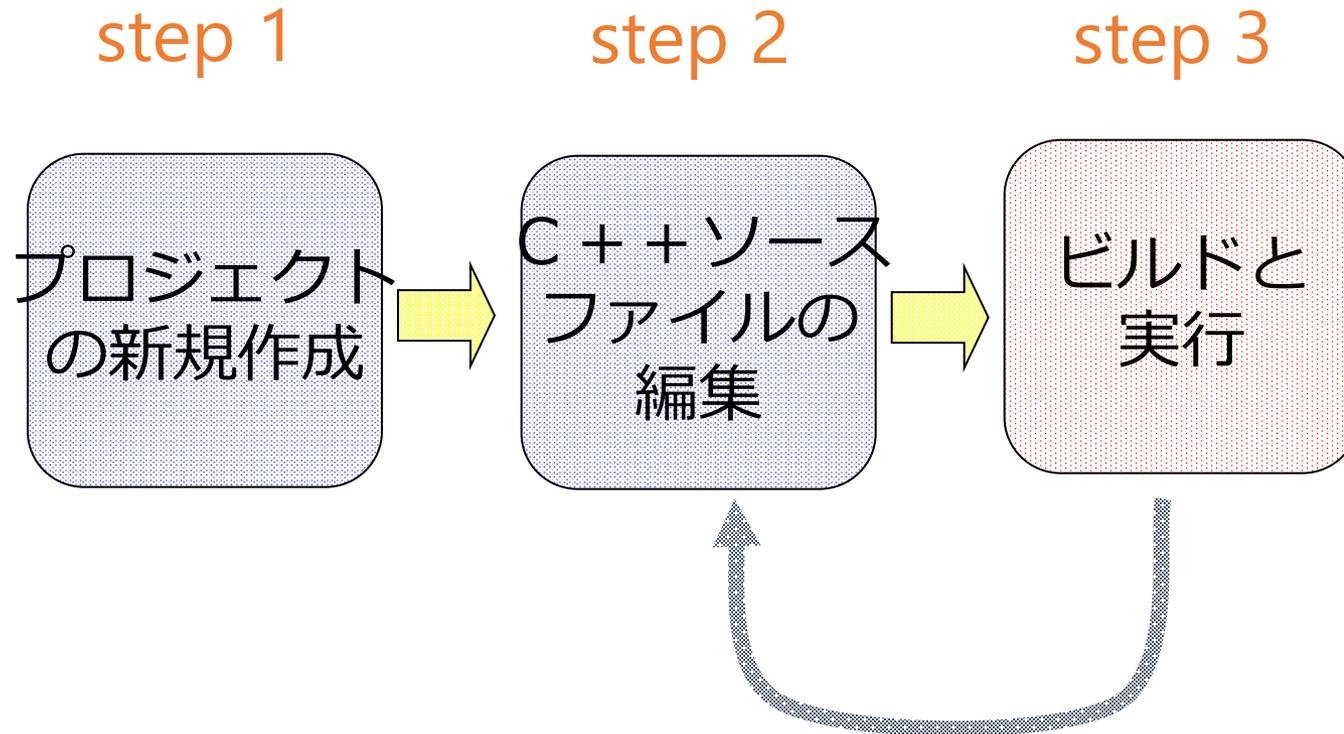


プログラムの機能



- 計算等の実行手順を記述したもの
- 多量の計算の繰り返し
 - 計算は自動で繰り返し
- キーボードからの読み込み
 - 自動で読み込み
- ファイルへの書き出し
 - ファイルを介して, 他のソフトと連携
 - など

プログラム作成の流れ

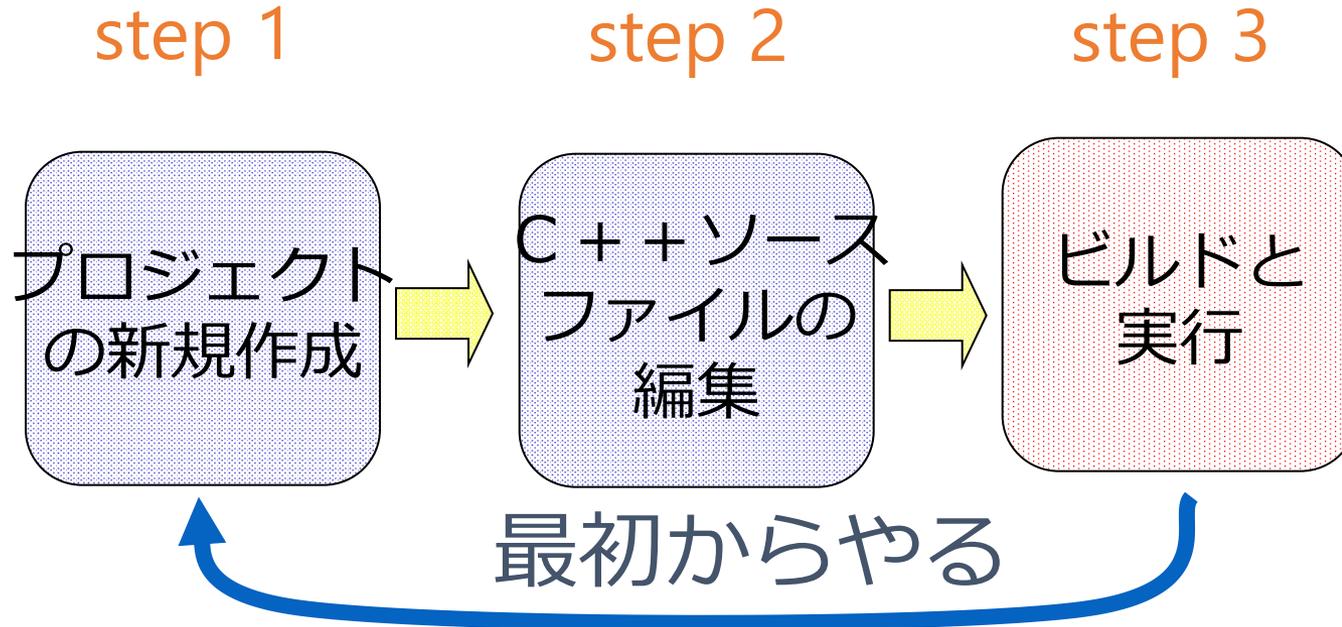


ビルド, 実行の結果, 間違いが見つかるたびに, 繰り返す

別のプログラムを作成する場合



ある課題のプログラム作成が終了し、次の課題をやる場合。
C++のソースファイルを変えるだけでは、エラーとなる。



1つのプロジェクトには、メインの関数が1つしか許されないなので、新しくプロジェクトを作成する。



ConsoleApplication1 - Microsoft Visual Studio

ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T) アー...

Debug x86 ローカル Windows デバツ...

ConsoleApplication1.cpp* + X

ConsoleApplication1 (グローバル スコープ)

```
19 fp = fopen( "d:\\data.csv", "w" );
20 for (i = 0; i < 20; i++) [
21     x = start_x + (i * step_x);
22     y = sin(x);
23     printf("x= %f, y= %f\n", x, y);
24     fprintf(fp, "x=, %f, y=, %f\n", x, y);
25 ]
26 fprintf(stderr, "file d:\\data.csv created\n");
27 fclose(fp);
28 return 0;
29
30
```

ここに全角文字「;」が入っている
(人間の目では良く分からない)

100 %

出力

```
ConsoleApplication1 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) Nsight ツール(T) アーキテクチャ(C)
Debug x86 ローカル Windows デバッガー
ConsoleApplication1.cpp
ConsoleApplication1 (グローバル スコープ)
19 fp = fopen( u.**data.csv , w );
20 for (i = 0; i < 20; i++) [
21     x = start_x + (i * step_x);
22     y = sin(x);
23     printf("x= %f, y= %f\n", x, y);
24     fprintf(fp, "x=, %f, y=, %f\n", x, y);
25 ]
26 fprintf(stderr, "file d:**data.csv created\n");
27 fclose(fp);
28 return 0;
29
30
```

ビルドを実行すると「文字'0x81'は認識
できません」というエラーメッセージが出
→ エラーの種類と場所が分かる

エラー一覧

ソリューション全体 4 エラー 0 警告 0 メッセージ

エラー一覧を検索

コード	説明	プロジェクト	ファイル
	ユーザー定義のリテラル演算子が見つかりません	ConsoleApplicatio	Console/
	';' が必要です	ConsoleApplicatio	Console/
C3688	リテラル サフィックス ';' が無効です。リテラル演算子またはリテラル演算子テンプレート 'operator "";' が見つかりません	ConsoleApplicatio	consolelea
C2143	構文エラー: ';' が ']' の前にありません。	ConsoleApplicatio	consolelea

課題

課題 1 . Microsoft Visual Studio C++ 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，例題 1 と例題 2 の手順を自分で行いなさい

```
#include "stdio.h"
#include <math.h>
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &start_x );
    printf( "step_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f\n", x, y );
        fprintf( fp, "x=, %f, y=, %f\n", x, y );
    }
    fprintf( stderr, "file d:¥¥data.csv created\n" );
    fclose( fp );
    return 0;
}
```

データファイル名
d:¥¥data.csv
は適切に設定すること

補足説明資料

例題のプログラムの機能

- キーボードからのデータ読み込み
- 型のキャスト
- 整数の範囲で四則演算
- 結果の画面表示

例題プログラム



```
#include "stdio.h"
void main()
{
    int i,j,k; /* 変数宣言 */
    float f0,f1;
    printf("整数を2つ入力してください: ");
    scanf_s("%d %d", &i, &j); /* キーボードから読み込み */
    k = i + j; /* 算術演算 */
    f0 = (float)(i/j); /* キャスト */
    f1 = (float)i / (float) j;
    printf("%5d + %5d = %5d¥n", i,j,k);
    printf("%5d - %5d = %5d¥n", i,j,i-j);
    printf("%5d x %5d = %5d¥n", i,j,i*j);
    printf("%5d / %5d = %5d¥n", i,j,i/j);
    printf("(float)(%d / %d) = %5.3f¥n", i,j,f0);
    printf("(float)%d / (float)%d = %5.3f¥n", i,j,f1);
}
```

ビルドと実行(1/2)



```
C:\WINDOWS\system32\cmd.exe
整数を2つ入力してください:
```

実行画面が現れる

ビルドと実行(2/2)

A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The prompt is "整数を2つ入力してください: ". The user has entered "3 7". A blue box highlights the input "3 7", and a blue arrow points from this box down to a larger text box below.

```
C:\WINDOWS\system32\cmd.exe
整数を2つ入力してください: 3 7
```

数値を入れる

(プログラムに数値データを与える)

ここでは, 「3 7 Enter」

3 と 7 は半角. 間に半角のスペース.

プログラム実行結果



```
C:\> C:\WINDOWS\system32\cmd.exe
```

```
整数を2つ入力してください: 3 7
```

```
3 + 7 = 10
3 - 7 = -4
3 x 7 = 21
3 / 7 = 0
```

```
(float)(3 / 7) = 0.000
```

```
(float)3 / (float)7 = 0.429
```

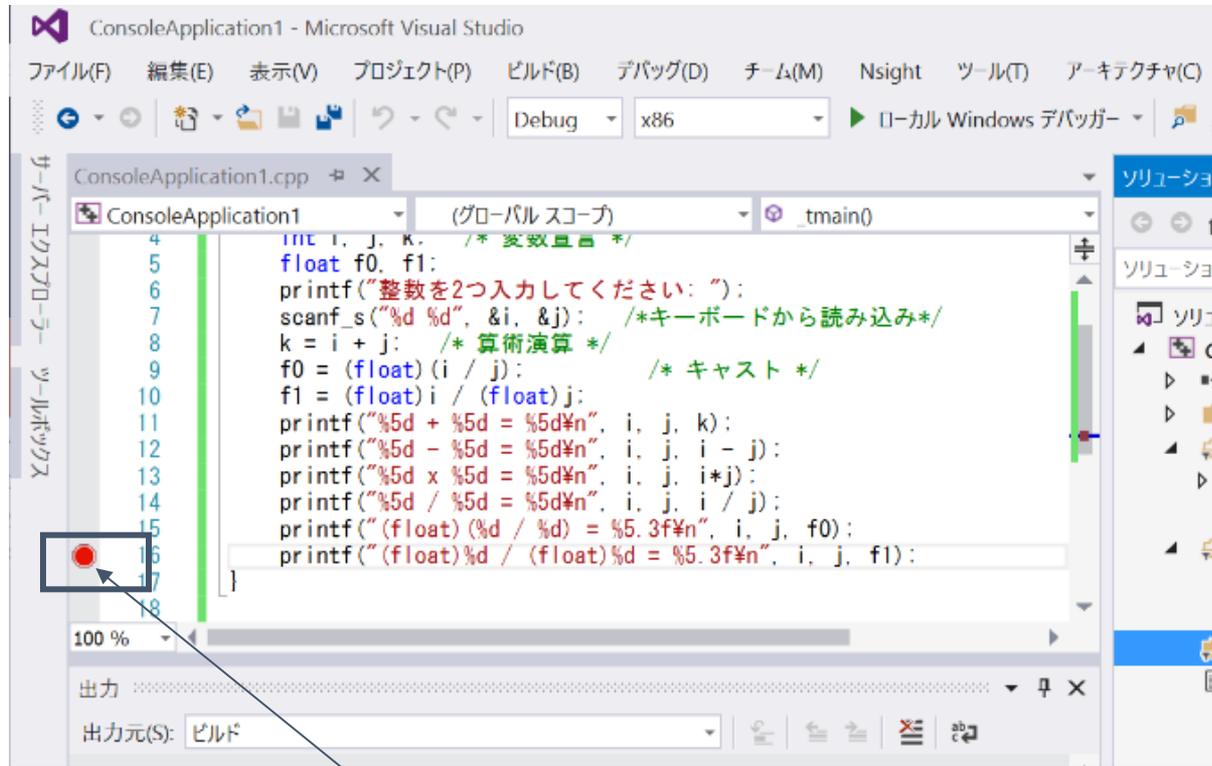
```
続行するには何かキーを押してください . . .
```

整数の範囲での演算

整数として計算した後で実数に変換するので、0.000になる。

浮動小数として計算するので、3/7の値が小数で表示される。

Microsoft Visual Studio C++ でのブレークポイント設定



ここをクリックして、ブレークポイントの設定
「●」はブレークポイント設定済みの印

ブレークポイントでプログラム実行が中断するので、
実行結果の画面が消えずに残る