ca-1. プロセッサ，マシン語

（コンピュータ・アーキテクチャ演習）

URL: https://www.kkaneko.jp/cc/ca/index.html

金子邦彦

CC BY NC SA

アウトライン

1-1 プロセッサ

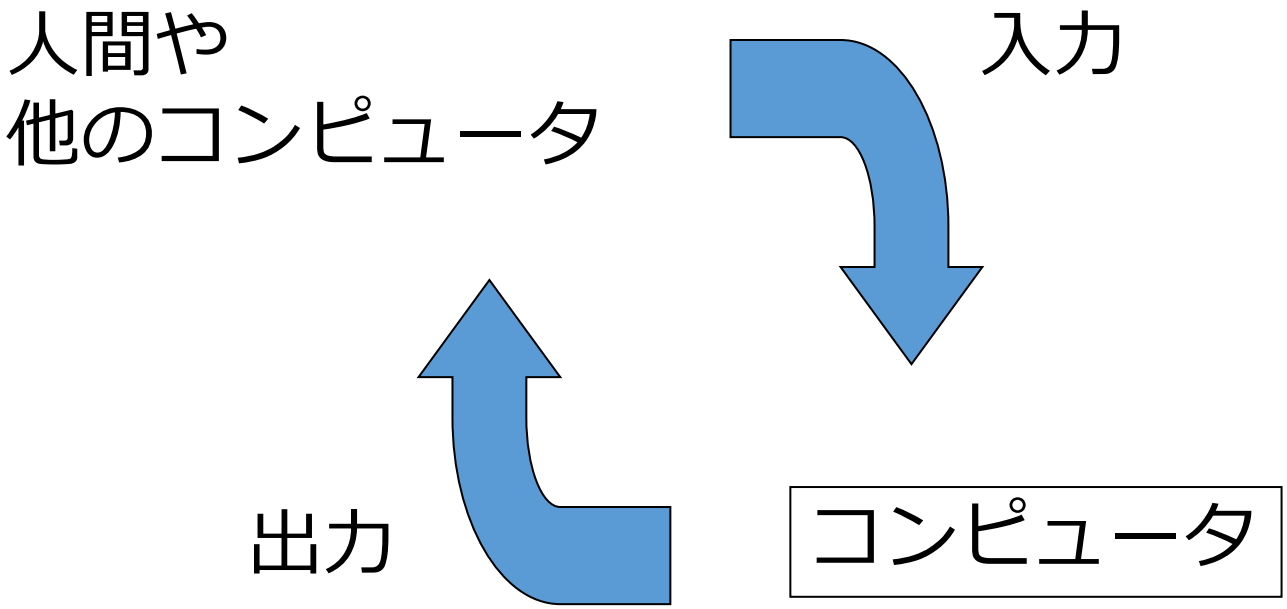
1-2 マシン語と実行型ファイル

1-3 ソースファイルとビルド

1-4 Visual Studio でビルドと実行

1-1 プロセッサ

コンピュータの入力と出力



ハードウエアとソフトウエア

ソフトウエア　アプリケーション
オペレーティングシステム

ハードウエア　プロセッサ，メモリ
その他，入力装置，出力装置など

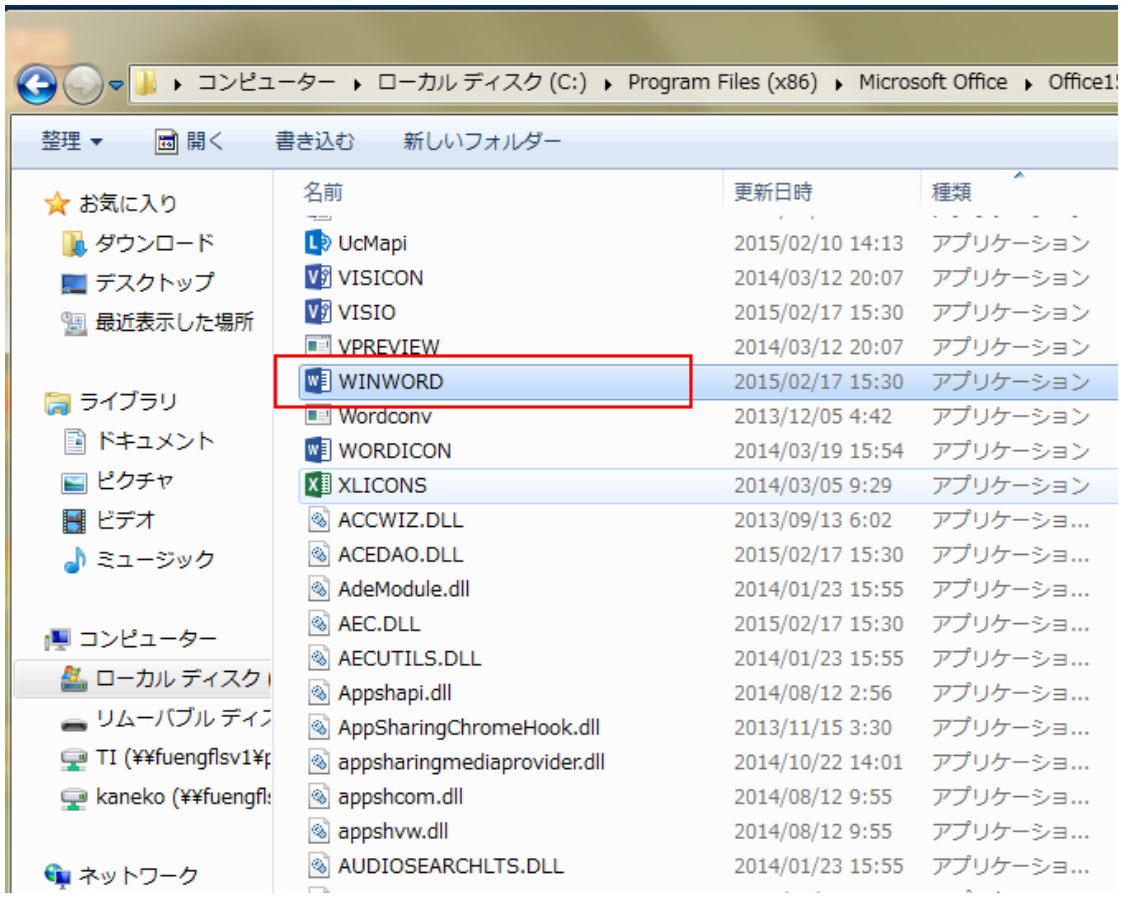
1-1 プロセッサ

- プロセッサは，コンピュータの中で，計算，制御を担う

1-2 マシン語と実行型ファイル

実行型ファイルの例

Word
のプログラム
のファイル



マシン語とは

プログラムのファイル

実行型ファイルの中身は
マシン語

```
4D 5A 90 00 03 00 00 00-04 00 00 00 FF FF 00 00
B8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00-00 00 00 00 10 01 00 00
0E 1F BA 0E 00 B4 09 CD-21 B8 01 4C CD 21 54 68
69 73 20 70 72 6F 67 72-61 6D 20 63 61 6E 6E 6F
74 20 62 65 20 72 75 6E-20 69 6E 20 44 4F 53 20
6D 6F 64 65 2E 0D 0D 0A-24 00 00 00 00 00 00 00
0F 93 F3 1B 4B F2 9D 48-4B F2 9D 48 4B F2 9D 48
24 84 36 48 48 F2 9D 48-50 6F 01 48 49 F2 9D 48
```

マシン語とは
プロセッサで実行される命令の言語

1-2 マシン語と実行型ファイル

- マシン語とは，プロセッサが実行できる言語のこと．プロセッサごとに異なる

- 実行型ファイルは，マシン語が格納されたファイル．実行ファイル，実行形式ファイルともいう

1-3 ソースファイルとビルド

ソースファイルとビルド

- ビルドとは，一般のプログラミング言語のソースファイルから，マシン語の入った実行型ファイルを生成すること

```
#include "stdafx.h"
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f ¥n", d, sqrt(d));
    return 0;
}
```

ソースファイル

ビルド

```
00000000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ..............
00000010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ........@.......
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00  00 00 00 00 F0 00 00 00  ................
00000040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ........!..L.!Th
00000050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
00000080  FF FA 97 C6 BB 9B F9 95  BB 9B F9 95 BB 9B F9 95  ................
00000090  FD CA 24 95 B8 9B F9 95  FD CA 26 95 BA 9B F9 95  ..$.......&.....
000000a0  FD CA 19 95 A8 9B F9 95  FD CA 18 95 BC 9B F9 95  ................
000000b0  66 64 32 95 B9 9B F9 95  BB 9B F8 95 83 9B F9 95  fd2.............
000000c0  B6 C9 18 95 B9 9B F9 95  B6 C9 22 95 BA 9B F9 95  ..........".....
000000d0  B6 C9 27 95 BA 9B F9 95  52 69 63 68 BB 9B F9 95  ..'.....Rich....
000000e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000000f0  50 45 00 00 4C 01 07 00  A8 13 37 55 00 00 00 00  PE..L.....7U....
00000100  00 00 00 00 E0 00 02 01  0B 01 0C 00 00 3C 00 00  .............<..
```

実行型ファイル

C言語などの
プログラミング言語

マシン語

1-3 ビルド

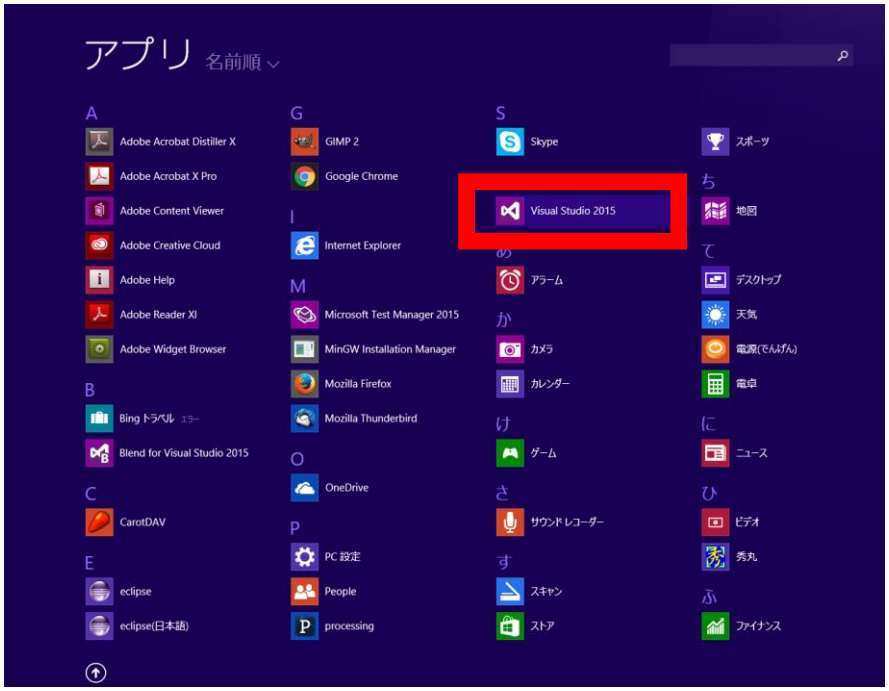
- ビルド（コンパイルともいう）とは，ソースファイルから，実行型ファイルを生成すること

ソースファイルが**エラー**を含むときは，**実行型ファイル**は生成されない

1-4 Visual Studio でビルドと実行

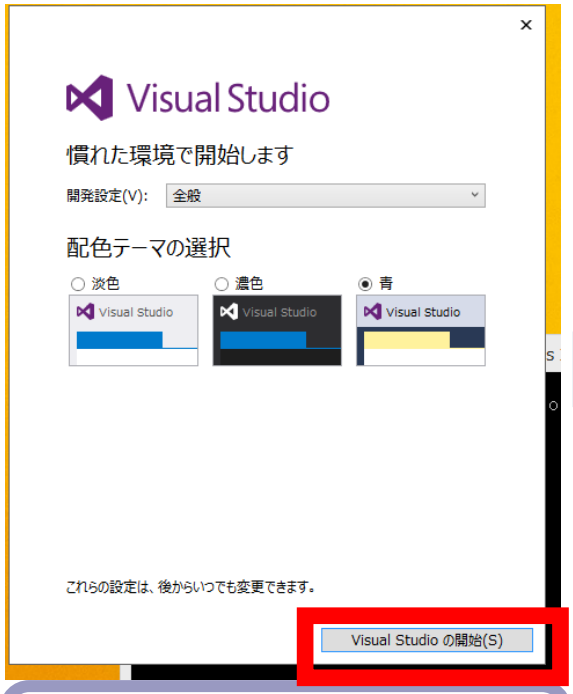
Visual Studioの起動手順

- **スタートメニュー**を使って，Visual Studioを起動
- 初回起動設定に数分かかる

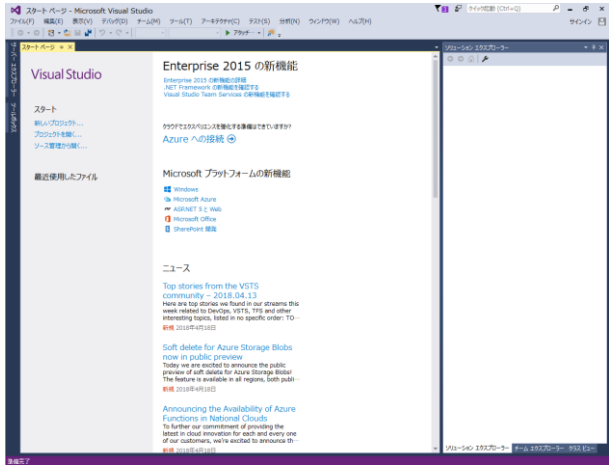


Visual Studioの初回起動設定（初回起動時のみ）

① 「後で行う。」を選んでおく

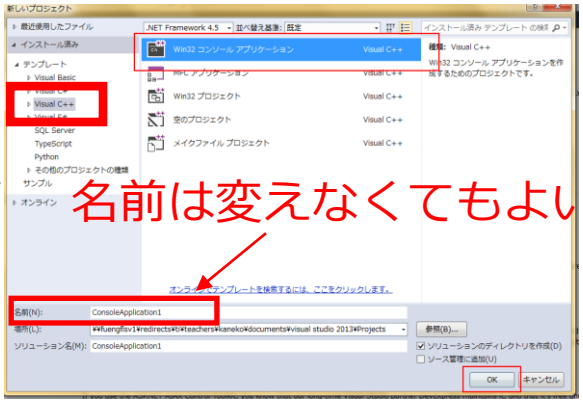


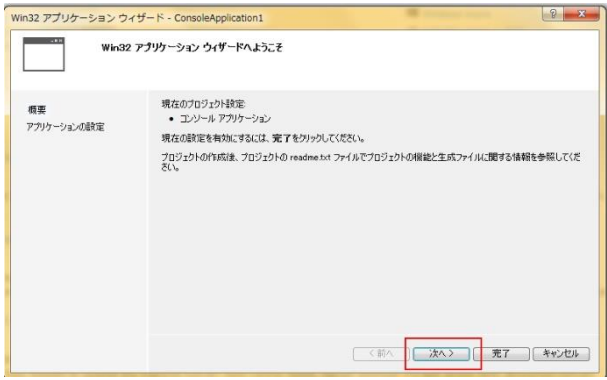
② 「Visual Studio の開始」をクリック



③ Visual Studio の画面が開く

Visual Studioで Win 32 コンソールアプリケーション用プロジェクトの新規作成 (1/2)





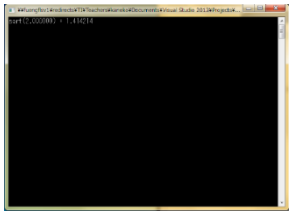
① 「ファイル」→「新規作成」→「**プロジェクト**」

② 「**Visual C++**」→「**Win32コンソールアプリケーション**」→「OK」

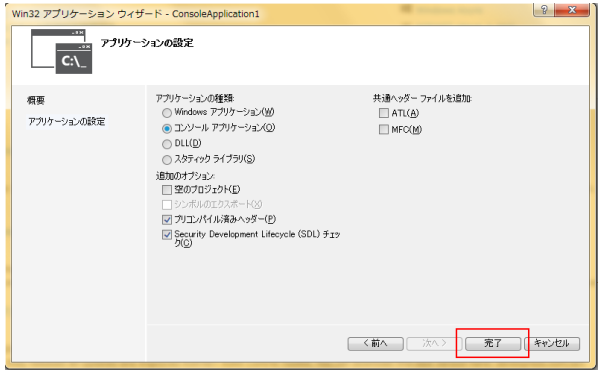
③ 「次へ」

※ 次ページに続く

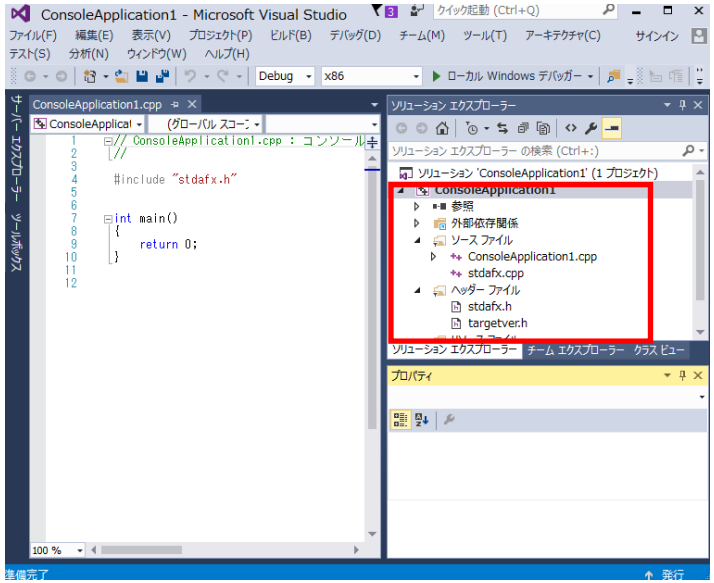
Win32コンソールアプリケーションとは，起動すると，**Win32コンソール**（例えば右図）が開くアプリのこと



Visual Studioで Win 32 コンソールアプリケーション用プロジェクトの新規作成 (2/2)

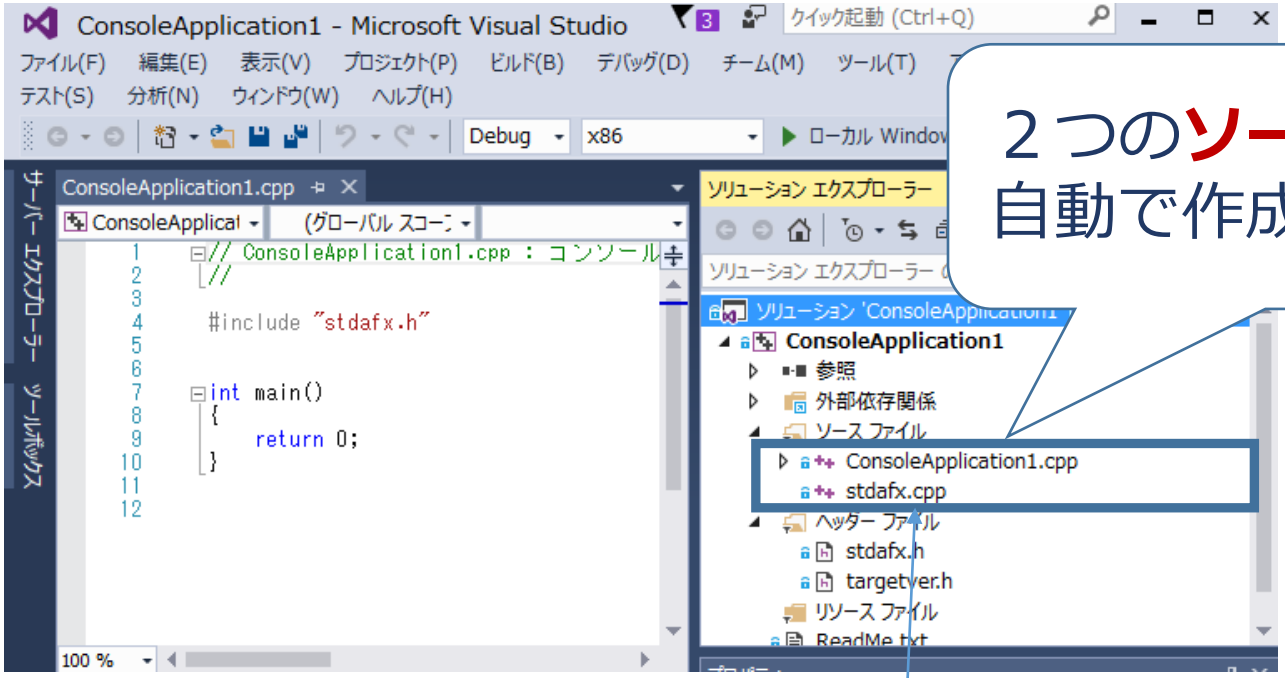


④ 「**完了**」をクリック



⑤ **プロジェクト**が新規作成されるので確認

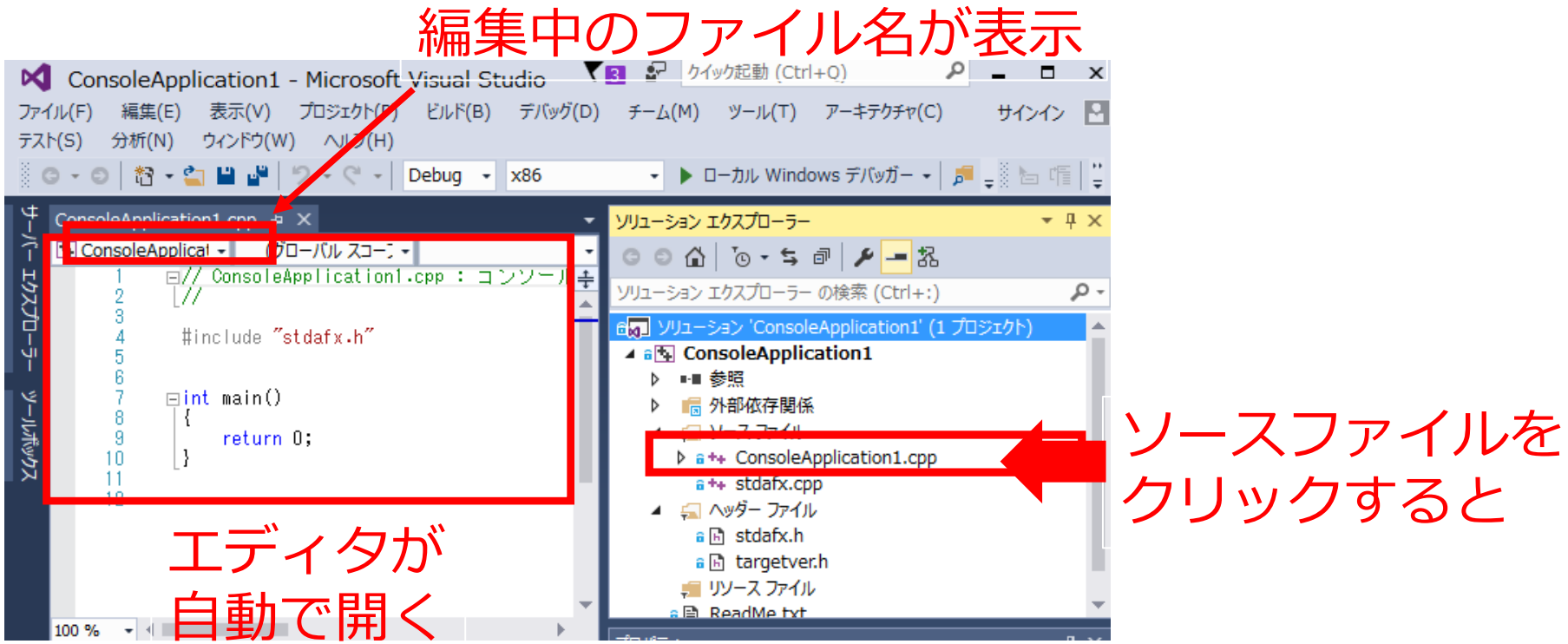
Visual Studioの Win 32 コンソールアプリケーション用プロジェクト



２つの**ソースファイル**が自動で作成されている

※ ここでは，stdafx.cpp は（システム用なので）編集することはない。**<u>もう１つの方を使う</u>**。

Visual Studioの Win 32 コンソールアプリケーション用プロジェクト



※ Win32 コンソールアプリケーション用プロジェクトを新規作成したとき，**<u>エディタが自動で開く</u>**ので，**<u>そのまま使ってよい</u>**

Visual C++ のソースファイル例

```
// ConsoleApplication1.cpp : コンソー
//

#include "stdafx.h"
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f ¥n", d, sqrt(d));
    return 0;
}
```

← 1行追加

← ２行追加

自動生成されたソースコードを活用（**追加するだけ**）

Visual C++ のソースファイル例

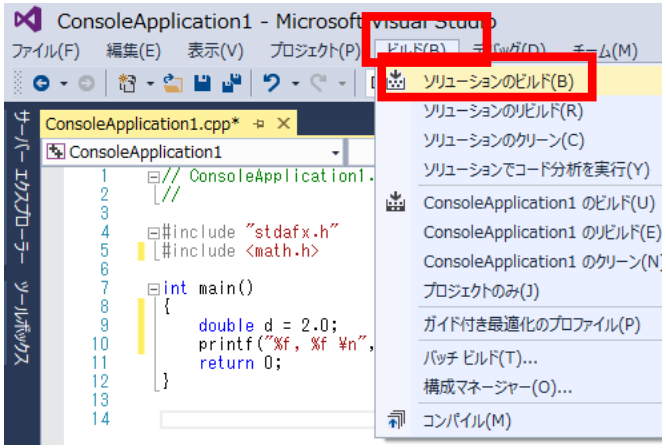
```
// ConsoleApplication1.cpp : コンソー
//

#include "stdafx.h"
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f ¥n", d, sqrt(d));
    return 0;
}
```

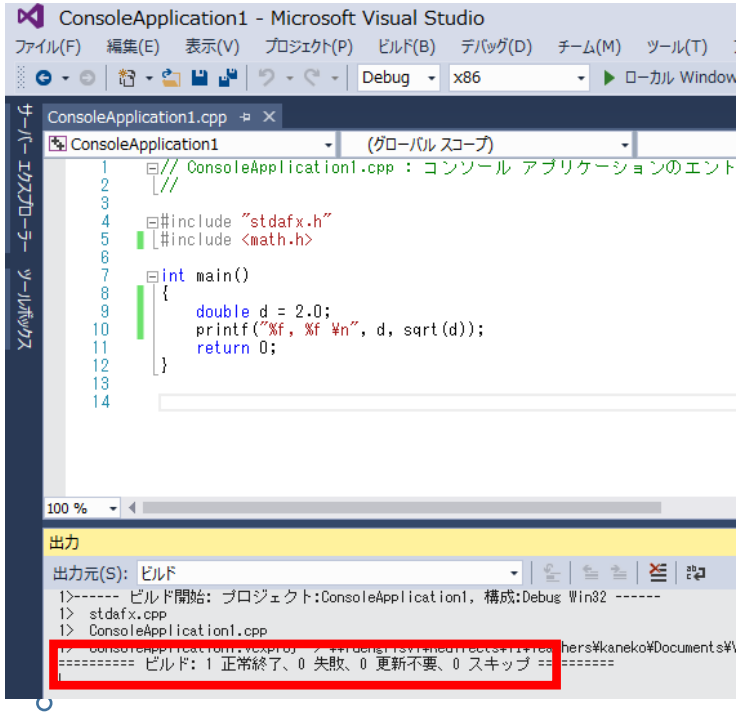
変数 d に，値 2.0 をセット

変数 d の平方根の計算。結果の表示

※ 「d = 2」と書かずに「d = 2.0」と書くのは，
小数付きの数として計算したい（整数ではない）ときの
C, C++ 言語での流儀

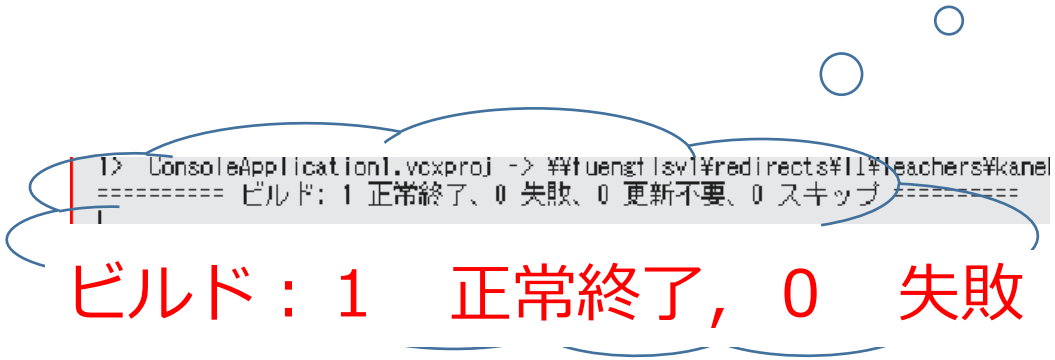
Visual Studioでのビルド手順





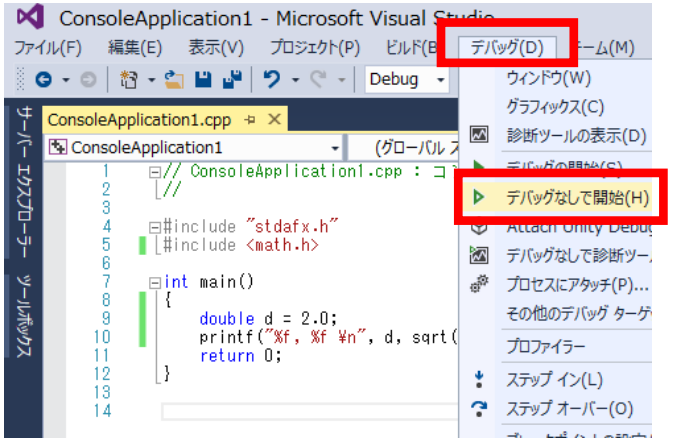
① 「ビルド」
→「**ソリューションのビルド**」

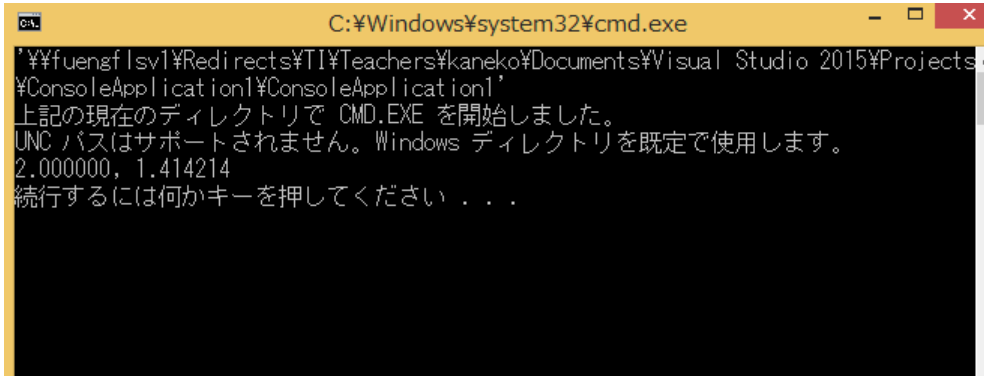
② 必ず「1　正常終了，0　失敗」の表示を確認



ビルド：1　正常終了，0　失敗

Visual Studioでの実行手順





① 「デバッグ」
→ 「**デバッグなしで開始**」

② Win32 コンソールが開く
（何かキーを押して終了）

ビルドが正常終了したら，
<u>実行</u>できる

演習

- Visual Studioのエディタを使って，ソースファイルを編集しなさい

```
#include "stdafx.h"
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f ¥n", d, sqrt(d));
    return 0;
}
```

← 1行追加

← ２行追加

- ビルドしなさい.
- ビルドのあと「１　正常終了，０　失敗」の表示を確認しなさい　→　表示されなければ，プログラムのミスを自分で確認し，修正して，ビルドをやり直す
- 実行しなさい.
- 実行結果を確認したら，Win32 コンソールで任意のキーを押して，Win32 コンソールを閉じなさい