

■ #06 誤差の話 2019.2.26 修正版・・・修正遅れて申し訳ないっす。

「コンピュータは、正確に処理ができる」という言葉があります。これは本当なのでしょうか？今日はそんな話をします。



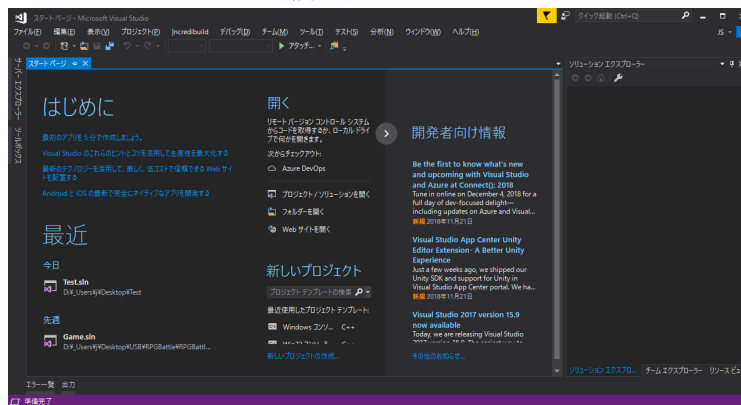
1. コンピュータは計算ができない？

いきなりな話ですが、「 $10 \div 3 \times 3$ 」できますか。馬鹿にしているわけではないのですが、実は一般的なコンピュータはこの計算ができません。

試しに、C++で計算してみましょう。黒い画面で面白くないと思いますが下記の手順でやってみましょう。

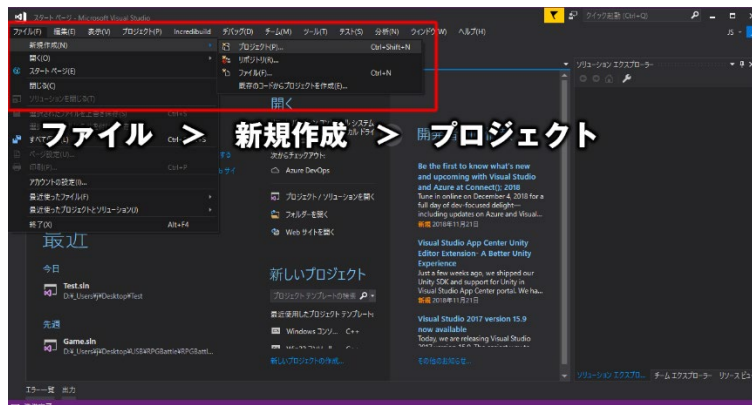
1.1. VisualStudio の起動

まずは、VisualStudio 起動します。

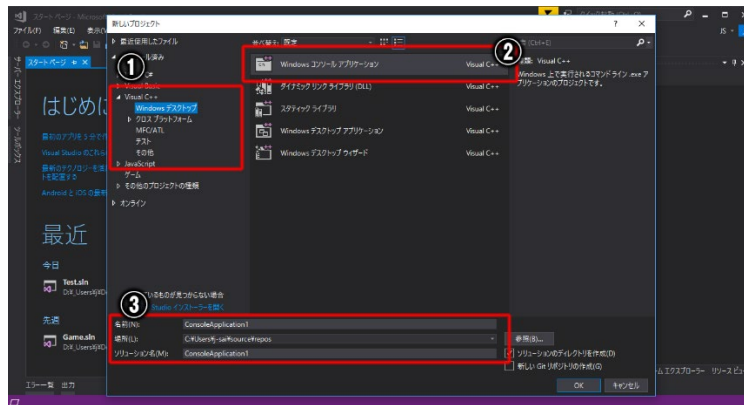


1.2. プロジェクトの作成

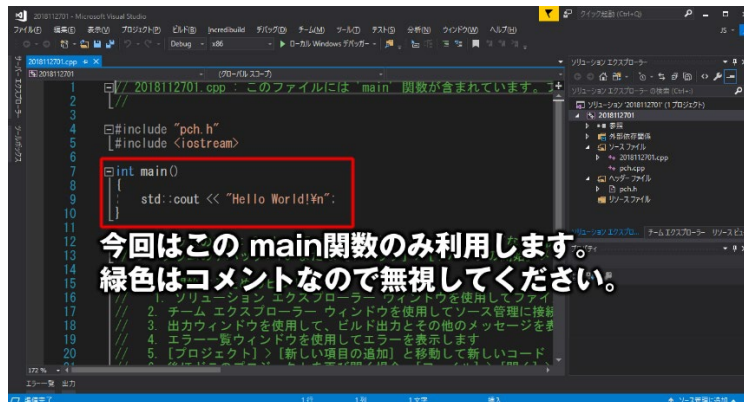
ファイル > 新規作成 > プロジェクト で新規プロジェクトを作成します。



ここでは、コンソール(黒い画面)のアプリケーション作成しますので。左側①のビューで “VisualStudioC++” を選択し、右側②に表示されたものの中から “Windows コンソールアプリケーション” を選択します。



ちなみに、このプロジェクトの場所は、下段③の “場所” に表示されています。“名前(N)” は作成するアプリケーションの名前になりますので、今回適当に “2018112701” と変更しておきます。ここで “OK” を押すとプロジェクトが作成され、ソースコードのテンプレートが開きます。



図にも書いていますが、赤枠で囲った部分だけ今回変更します。

1.3. コードの入力と実行

では、先ほどの計算実際やってみましょう。main 関数を下記の通り変更します。コメントなどは無視して記載しています。緑色の部分はコメントですので、ここでは入力しなくても大丈夫です。

2018112701.cpp 修正追加版 : 6

```

1  #include "pch.h"
2  #include <iostream>
3  #include <iomanip>           // この部分はおまじないです
4
5  void main()                 // int → void に変えておいてください
6  {
7      volatile int ans = 10/3*3;    // volatile (ボラタイル) は最適化させない指定です
8      // int ans = 10/3*3;          // こちらは最適化され表示が 10 になります。
9      std::cout << ans << std::endl;    // cout は文字をコンソールに表示する関数です。
10 }
```

入力が終わったら、“F5” を押して実行してください。どんな表示になりましたか？

多分 “9” じゃないでしょうか。“10”じゃないですね。何ででしょう。

1.4. 簡単な解説

最新のコンパイラは優秀なので、最適化でかなり不具合少なくなっていますが、浮動小数点など扱う場合に変換誤差が発生します。

先の例では、分かりやすいように int 型を利用していますが、内部処理を解説しておきます。

まずはおさらい

int 型変数：整数を扱う ← 小数点は切り捨て

ですね。そのため計算では下記の通りになっています。

```
volatile int ans = 10 / 3 * 3 ;
```

↓

$10/3 = 0.3333\cdots$ となるため、小数点以下切り捨てで “3” になります。

結局、3 に 3 をかけることになるため “9” になることがあります。

最近のコンパイラでは、式全体を評価して同じ数で剰余算行っている場合など無視するようになります。

賢い！

2. いろいろな計算

このようなことは、結構普通に発生します。実際にやってみましょう！

1. 0.01 を 10000 回足したら 100.003

Web でも結構話題になっていたりする。

nemui4 の日記： 0.01 を 10000 回足したら 100.003 になる：

<https://srad.jp/~nemui4/journal/625474/>

2. $1.0000001 - 1.0000000 = 0.0000000$

こちらは少し難しいか！

IEEE 演算機能 - Oracle Docs

https://docs.oracle.com/cd/E19957-01/806-4847/ncg_math.html

○チェックポイント・キーワード



- ・最適化 volatile
- ・変換誤差
- ・変数の型 int, float, double
- ・小数点の切り捨て