

Visual Studio .NET 2003 によるプログラム開発の手順

目次

Microsoft Visual Studio .NET 2003 の起動手順と初期設定	2
既存のソースファイルを用いたプロジェクトの新規作成手順	4
プログラムの翻訳・エラーへの対処・実行の手順	8
プロジェクトの閉じ方 (終了する時・他のプロジェクトへの移行の時)	10
既存プロジェクトの開き方	11
新規のソースファイルの作成手順	12
コマンドラインでの開発作業の手順	14

Microsoft Visual Studio .NET 2003 の起動手順と初期設定

手順 1

ログオンを終えたら、通常のアプリケーションと同様に、**スタート**ボタン → **すべてのプログラム** → **Microsoft Visual Studio .NET 2003** → **Microsoft Visual Studio .NET 2003** と辿って起動する (図 A)。



(講師の環境での画面コピーなので、実習室のメニュー構成とは異なります)

図 A 起動時の辿り方

(最初の起動時に、2回警告メッセージが表示されるが、2回とも **はい** で応答する)

手順 2

スタートページのプロフィールのうち、キーボードスキームおよびウィンドウレイアウトをそれぞれ、Visual C++ 6 に切り替えておく (図 B)。

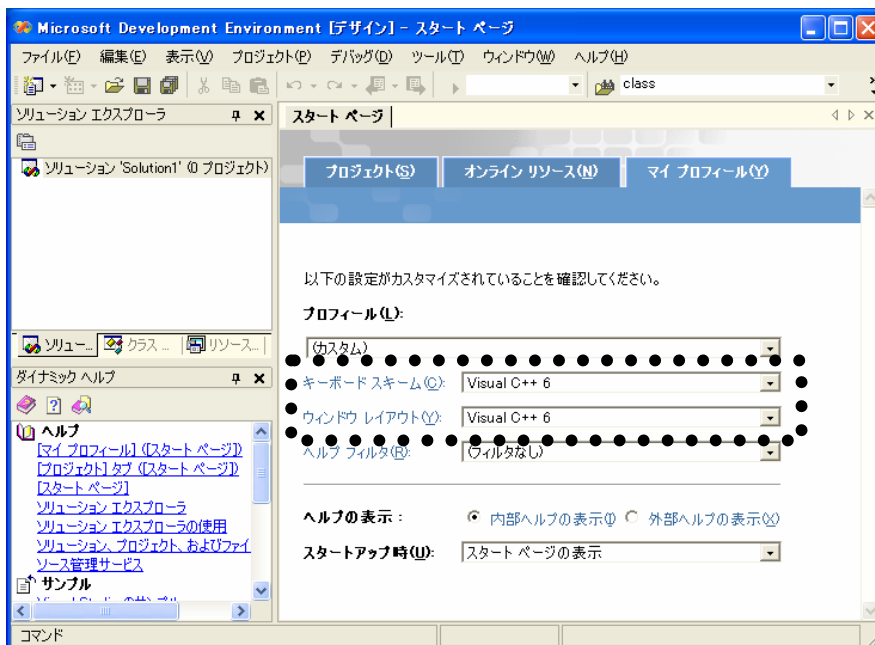


図 B スタートページの設定箇所

手順 3

ソースプログラムに行番号を表示させておくと、印刷テキストとの対応が取り易くなるので、**ツール**メニュー→**オプション**と辿ったら、画面左側のパネルを**テキストエディタ**→**すべての言語**→**全般**と辿る。右側に現れた設定パネルの**表示**：**行番号**にチェックを入れておく (図 C)。

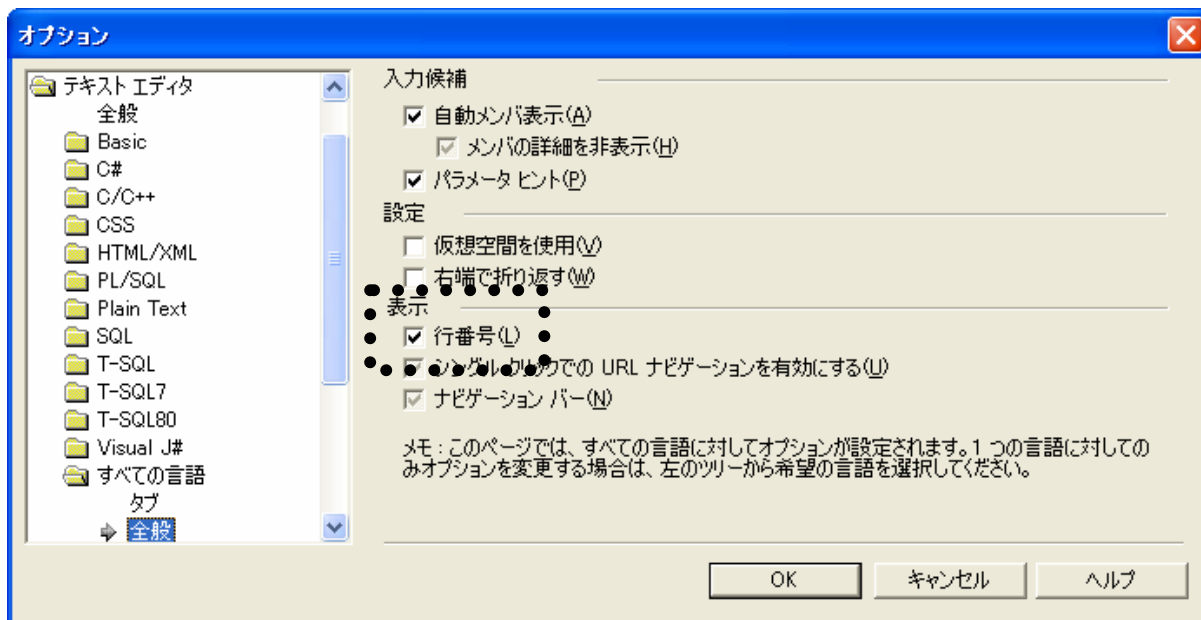


図 C 行番号表示の設定

以上の設定は、ログオンし直さない限り、1度だけ行っておけばよい。

既存のソースファイルを用いたプロジェクトの新規作成手順

手順 4

初期設定を終えたら、既存のソースプログラムを使って、翻訳・実行させるための環境 (Visual Studio の用語ではプロジェクト) を作成する。

まず、ファイルメニュー→新規作成→プロジェクトと辿る (図 D)。

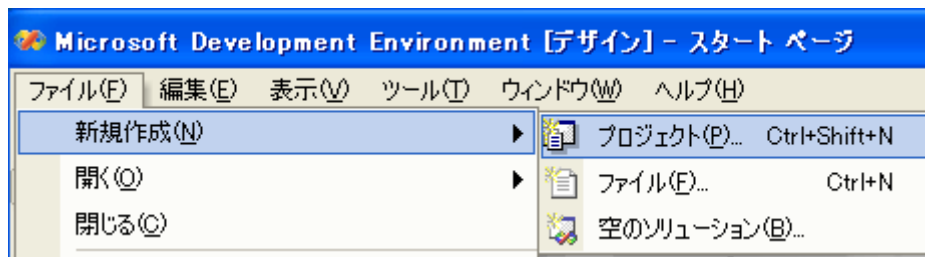


図 D 新規プロジェクトの作成

手順 5

プロジェクトの種類から、Visual C++ プロジェクト→Win32 と辿り、テンプレートを Win32 コンソール プロジェクトにする。

プロジェクト名はこれから使おうとする教材のファイル名に合わせて入力する (図 E の例では sample1)。場所は、教材を保管したフォルダ名に直接書き換えるか、参照ボタンを用いて当該のフォルダ名となるように切り替えておく。

ソリューションのオプションが見えている場合は、ソリューションを閉じるに切り替えておく (図 E)。設定事項を確認したら、OK ボタンをクリックする。

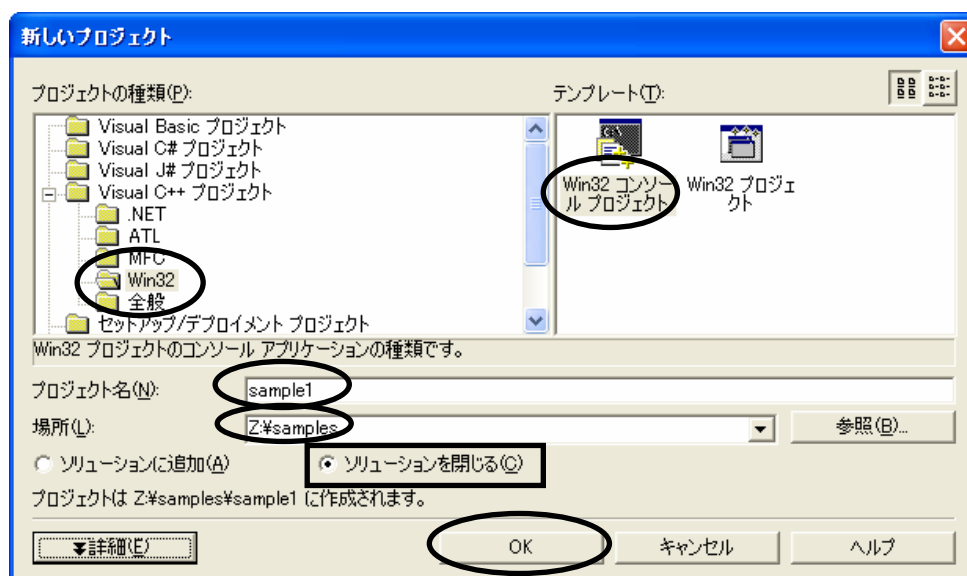


図 E (C 言語用) 新規プロジェクトの設定項目

手順 6

アプリケーションウィザードが表示されたら、**アプリケーションの設定**に切り替えて、**追加のオプション**：**空のプロジェクト**にチェックを入れてから**完了**ボタンをクリックする (図 F)。

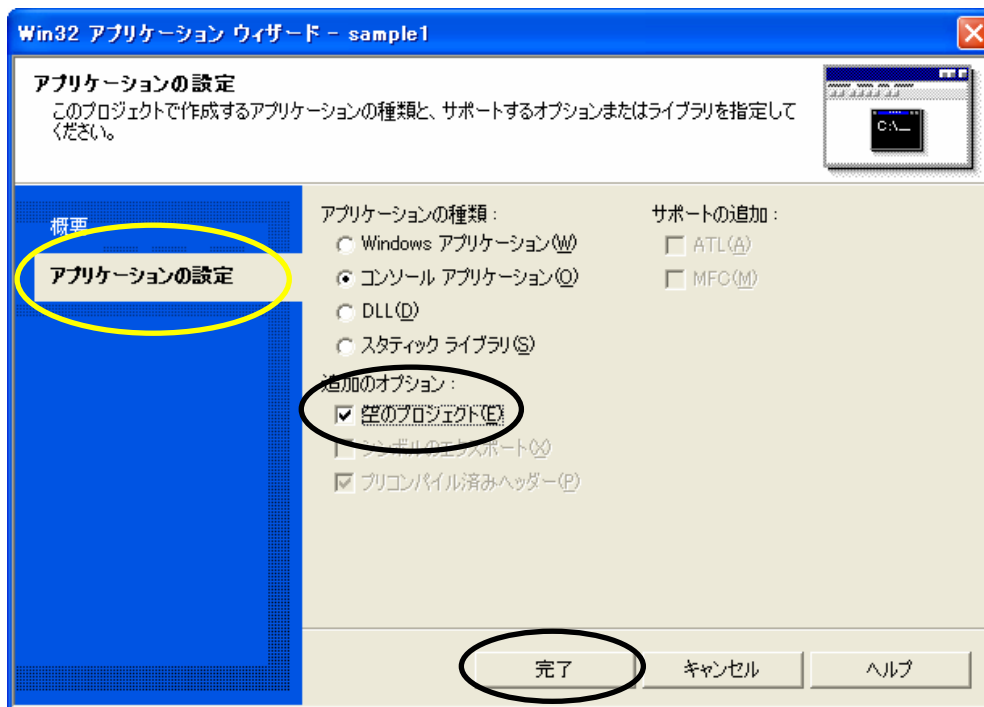


図 F アプリケーションウィザードのオプションの変更

手順 7

プロジェクトが開かれた状態になったら、**ファイル**メニューから**既存項目の追加**を選び、ファイルの選択画面にて教材のフォルダから当該のファイルを選んで開く (図 G)。

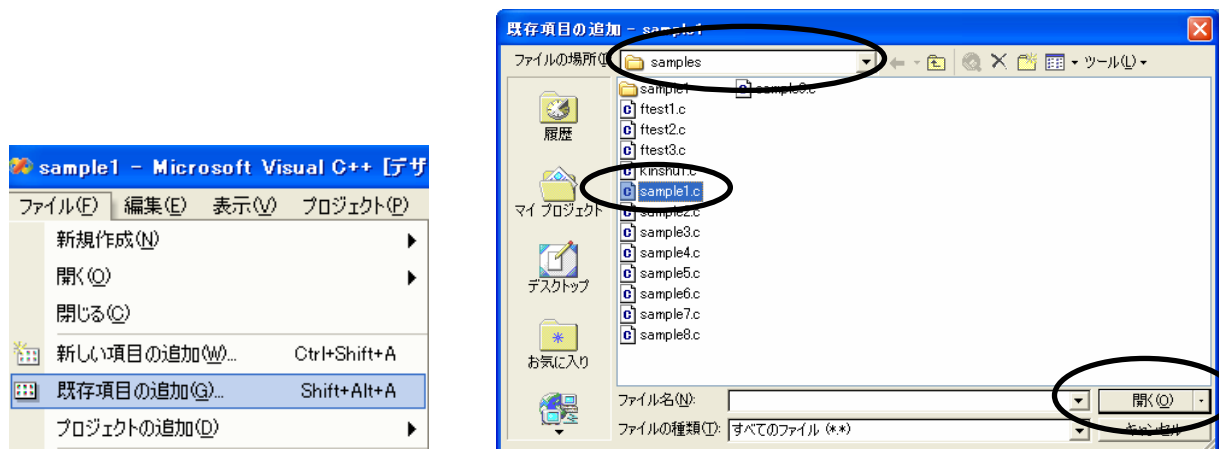


図 G 既存のソースファイルのプロジェクトへの追加

手順 8

ソリューションエクスプローラの **ソースファイル** に、指定したファイルが追加されたのを確認したら、その内容の確認のため、ファイル名をダブルクリックしてみる。右側にソースファイルの内容が表示されたらプロジェクトの作成は完了となる (図 H)。

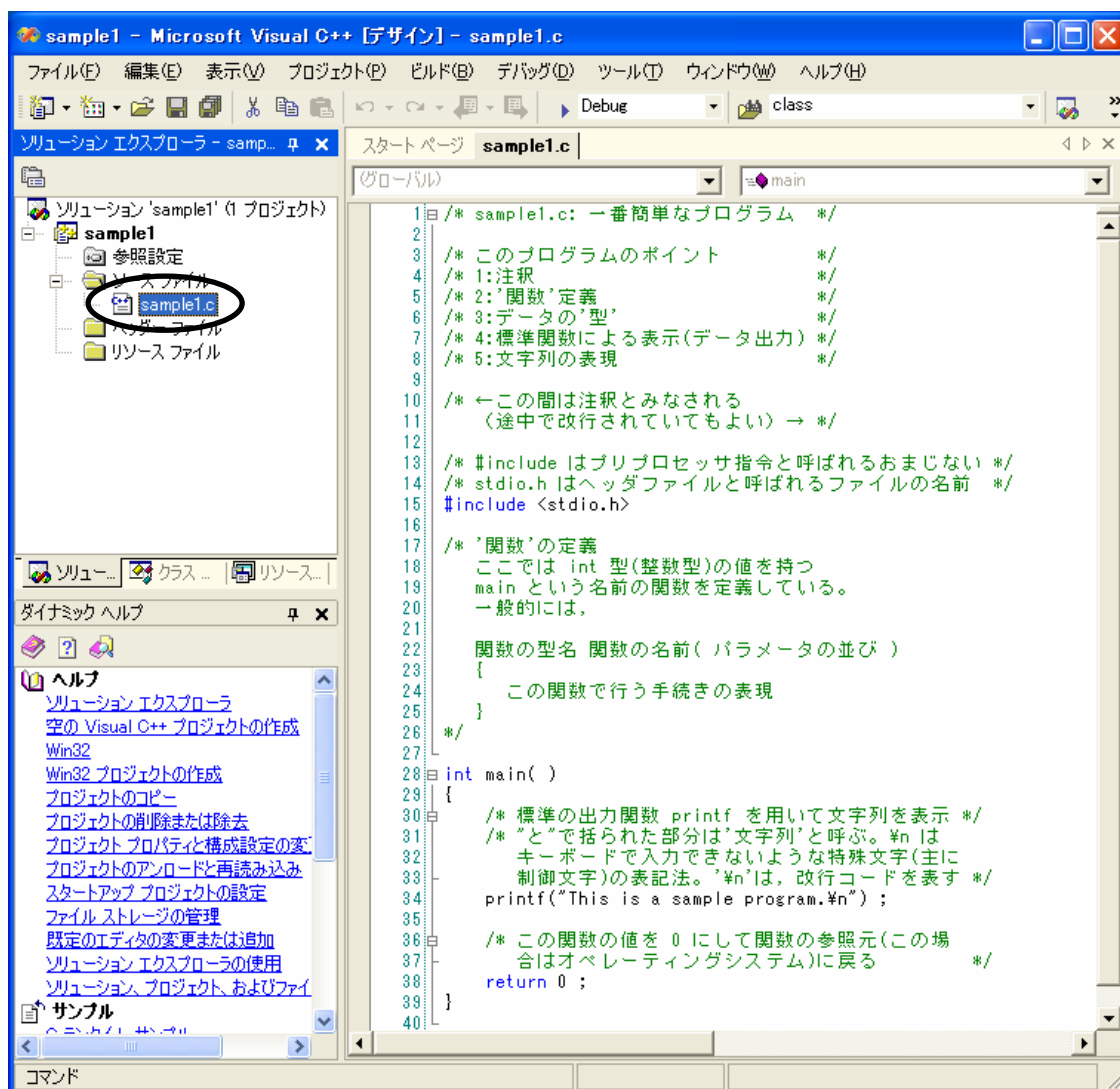


図 H ソースファイルの登録とその内容の表示

以上の手順は、ソースファイルを変えるたびに行なう。一旦プロジェクトを作成しておけば、閉じた後でも、既存プロジェクトの開き方 (11 ページ) の手順を使って編集状態を呼び戻すことができる。

(空白ページ)
メモなどに利用しましょう

プログラムの翻訳・エラーへの対処・実行の手順

手順 9

プロジェクトに追加されたソースファイルを翻訳 (コンパイル) して, 実行可能なプログラムを生成 (ビルド) するには, **ビルド**メニューから **xxxx (プロジェクト名) のビルド** を選ぶ (図 I 左)。ソースファイルに誤り (エラー) がなければ, ソースプログラムの下方にビルド作業の結果が報告されるので, 必ず確認しておく (図 I 右)。

失敗の数が 0 でない時は, ソースプログラムにエラーが含まれているので, ビルド画面を前方にスクロールして最初に報告されているエラーメッセージの行をダブルクリックすることでソースファイル中の当該箇所に自動的に移動してくれる。スペルミスや記号の抜け (特に ;) に注意して修正する。

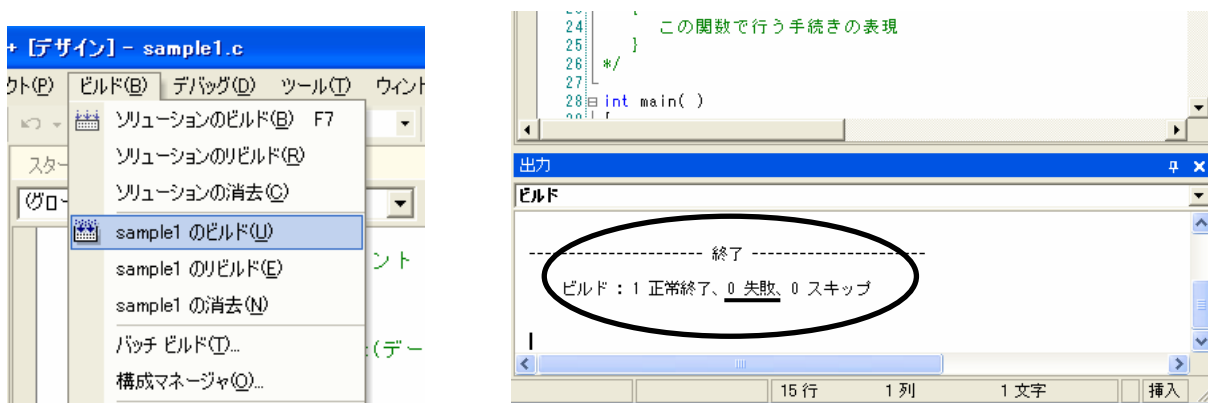


図 I ビルドの手順と結果の確認

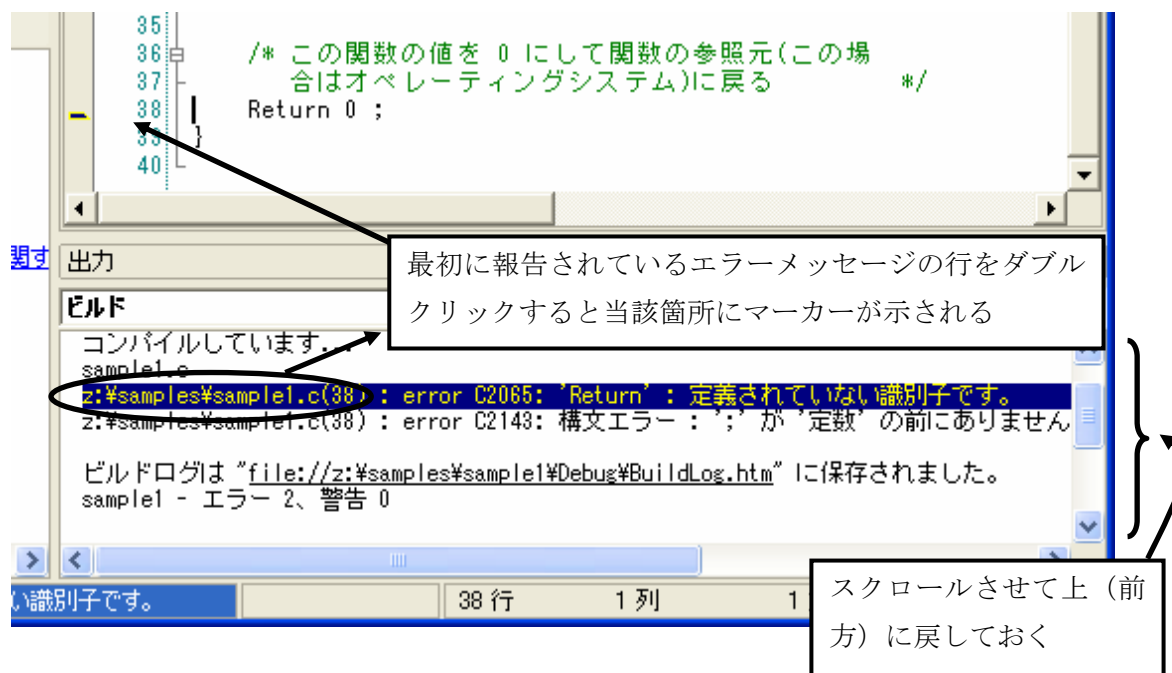


図 J エラー箇所の見つけ方

手順 10

ビルドされた (実行可能) プログラムを実行させるには、**デバッグ**メニューから**デバッグなし****で開始**を選ぶ (図 K 上)。実行中のプログラムは別ウィンドウになって表示される (図 K 下)。

内容を確認し (必要な時は、そのウィンドウでデータを入力するなどの応答をして) 最後に、Enter キーを使ってウィンドウを閉じておく。

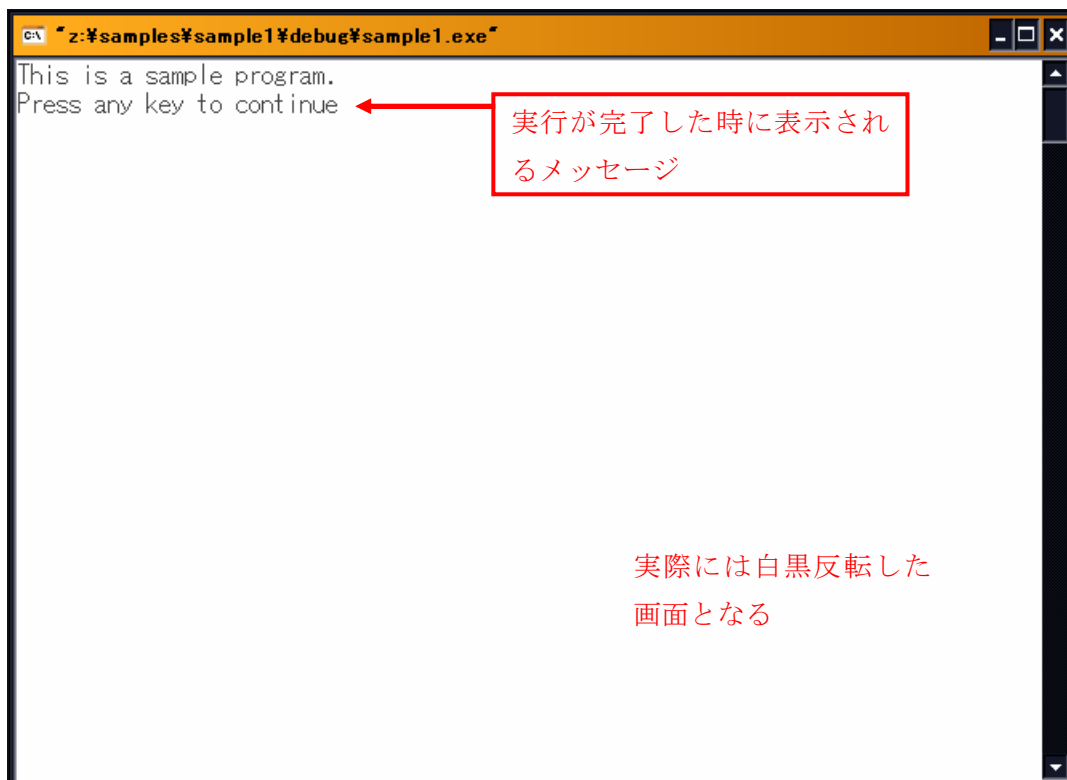


図 K 実行手順と実行画面 (コンソール型アプリケーション)

プロジェクトの閉じ方（終了する時・他のプロジェクトへの移行の時）

手順 1 1

ログオフする前や、別のプロジェクトを作成する場合には、現在のプロジェクトを保管して閉じておく。ファイルメニューからソリューションを閉じるで一括して閉じておく（図 L）。

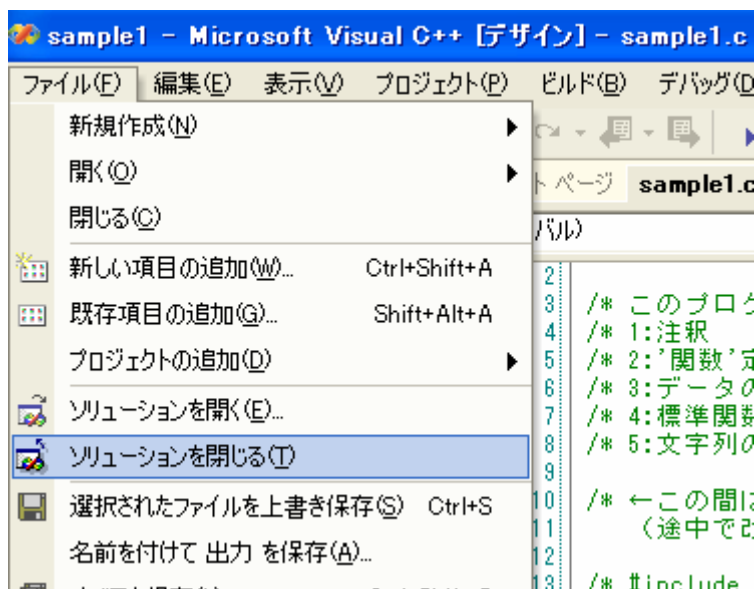


図 L プロジェクトの閉じ方

既存プロジェクトの開き方

手順 1 2

既存のプロジェクトを開くには、**ファイル**メニューから**開く**→**プロジェクト**と辿る(図 M 上)。保存した場所のフォルダに移動したら、拡張子が `sln` または `vcproj` となっているファイルが見つかるはずなのでどちらかを開けばよい(図 M 下)。

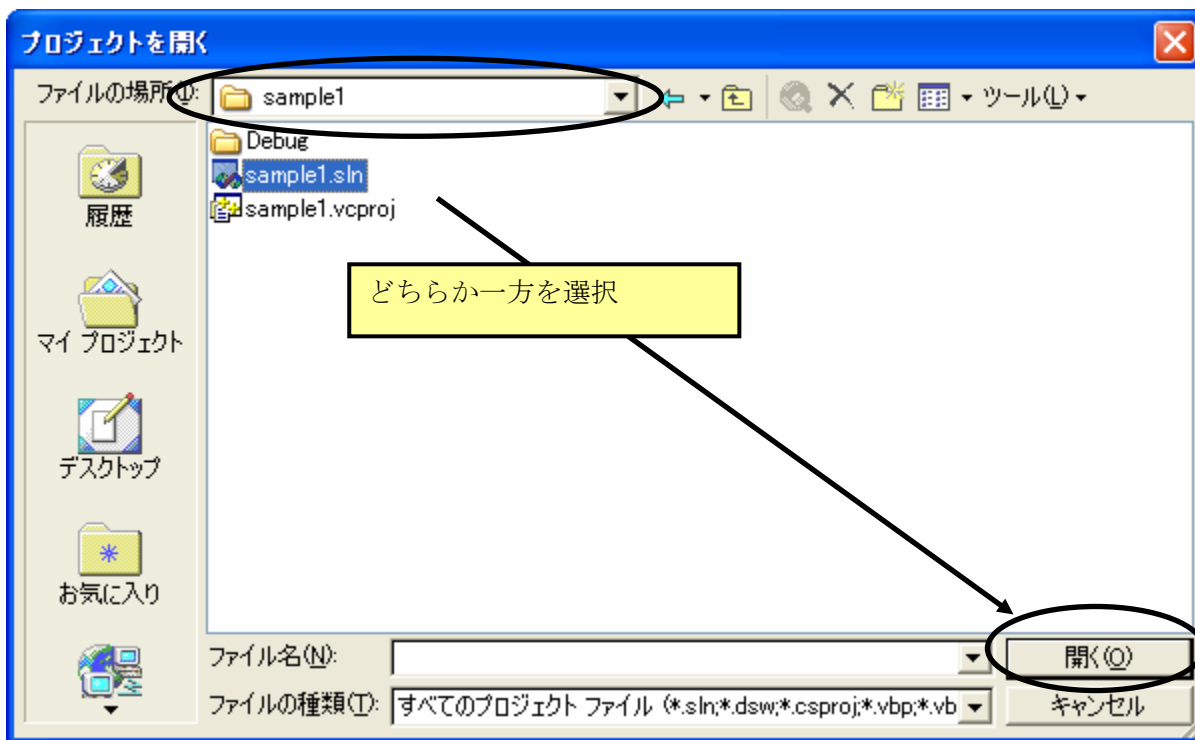


図 M 既存プロジェクトの開き方

新規のソースファイルの作成手順

手順 1 3

新たにソースファイルを作成する時は、そのファイル名をあらかじめ決めておき、手順 4~6 (4~5 ページ) と同様にしてプロジェクトを作成する (以下の例では hanpuku としておく)。手順 7 の代わりに、**ファイル**メニュー→**新しい項目の追加**を選ぶ (図 N 上)。

テンプレートから C++ファイル(.cpp) を選び、**ファイル名**にソースファイルとなるファイルの名前 (.c は省く)、**場所**を (特に理由がなければ) 教材のフォルダに書き換えておく。

内容を確認したら、**開く**ボタンをクリックする。

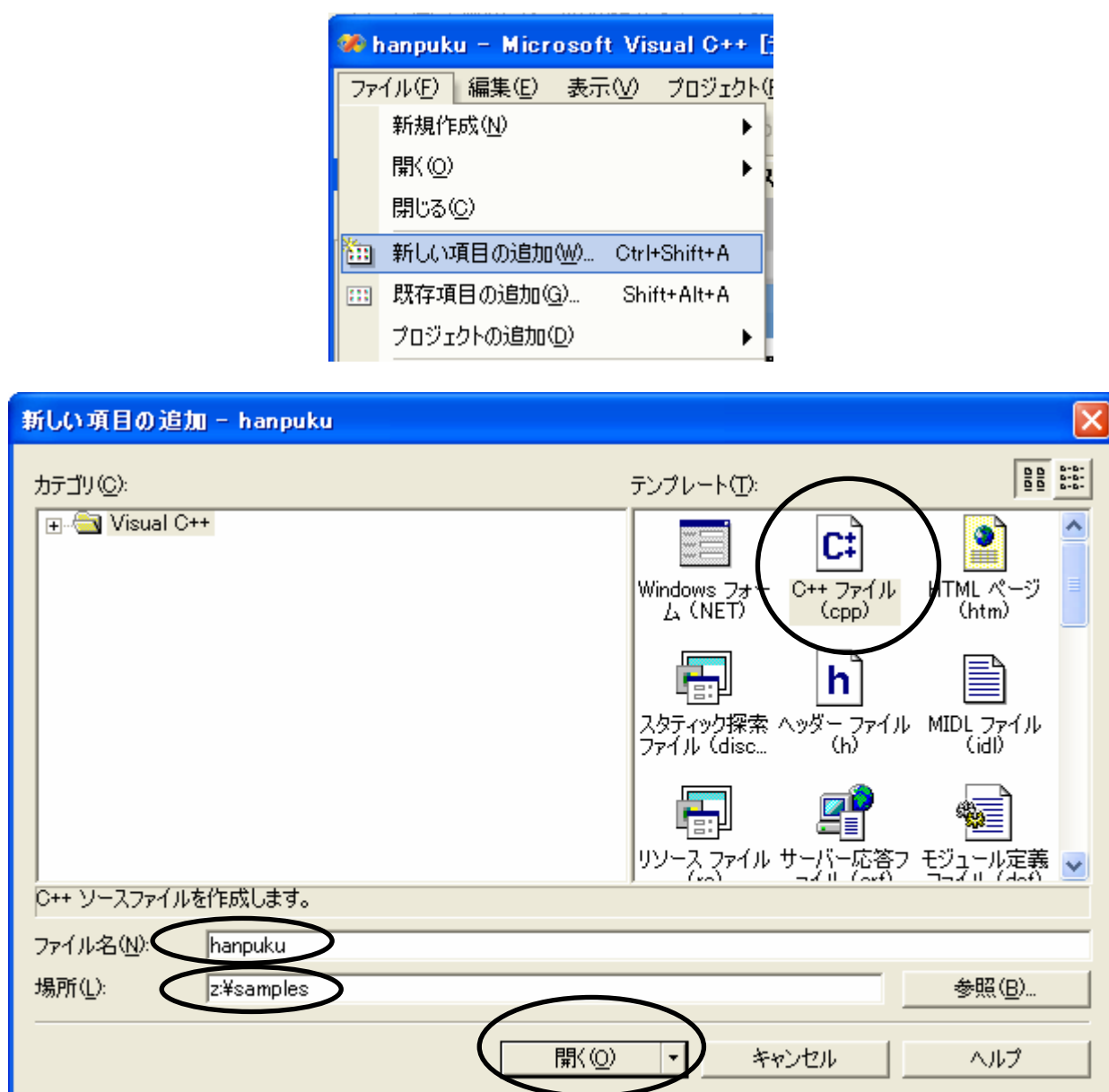


図 N プロジェクトへの新規ソースファイルの追加

手順 1 4

図 0 のように空っぽの状態のファイルが追加されるので、必要な文を追加入力していく。翻訳・実行などの手順は、既存のものを使う場合と同じである。

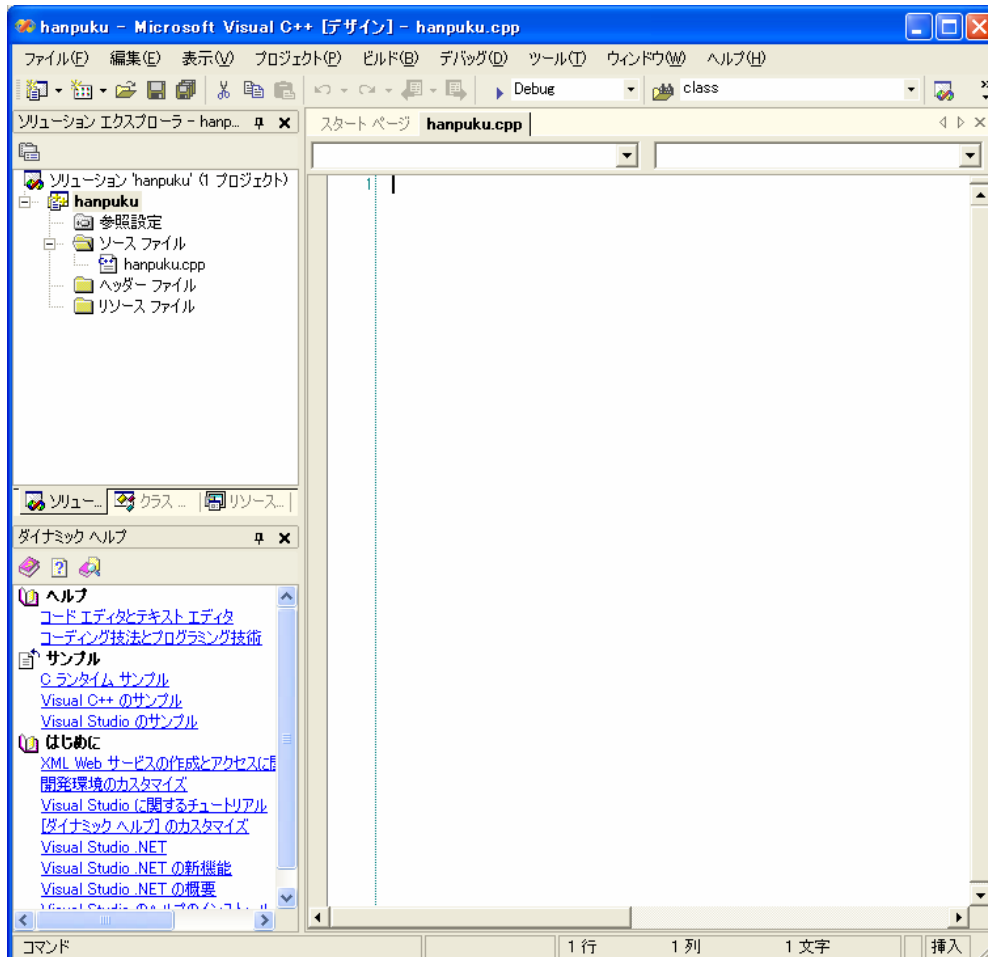


図 0 新規作成直後のソースファイルの様子

コマンドラインでの開発作業の手順

統合環境を使う時は、プロジェクトの作成など、本筋ではない部分での操作が煩雑なので、キーボード操作に自信がある人は、コマンドモードでの使い方も体験してみるとよい (Unix や Linux を使ったマシンでは、むしろこちらの使い方が標準でもある)。

手軽な反面、エラー箇所の自動的な特定は行なわれないので、デバッグ作業はやや面倒になる。小規模なプログラム開発ではこの手順でも十分実用的であるので、TPO に応じて使い分けの得策であろう。

手順 1 5

スタートボタン → すべてのプログラム → Microsoft Visual Studio .NET 2003 → Visual Studio .NET ツール → Visual Studio .NET 2003 コマンド と辿る (図 P)。

いわゆるコマンドを入力するための (統合環境ではプログラムの実行ウィンドウと同じもの) コマンドプロンプトのウィンドウが開かれる。



図 P 開発環境用コマンドプロンプトの呼び出し

手順 1 6

操作はすべてキーボードから入力する文字列 (コマンド) によって行なう (図 Q)。

① フォルダの移動

コマンドプロンプトで行なう場合は、教材のフォルダにすべてのファイルを置くことになるので、CD コマンドを使ってフォルダに移動しておく。一度だけ実行しておけばよい。

`cd フォルダ名`

② ソースファイルの編集 (既存のソースファイルを実行させる場合は、この手順は省いてよい)

プロジェクトを作成していなくても個々のファイルの編集はできるので、編集作業のみ統合環境を使うか、Windows 標準のメモ帳 (notepad.exe) をコマンドレベルで呼び出して、テキストエディタの代用とすることもできる。

`notepad "ファイル名.c"` 前後のダブルクォート記号も必要

③ 翻訳

翻訳は CL (入力時は小文字のままでよい) コマンドに続けてソースファイルの名前を指定する。

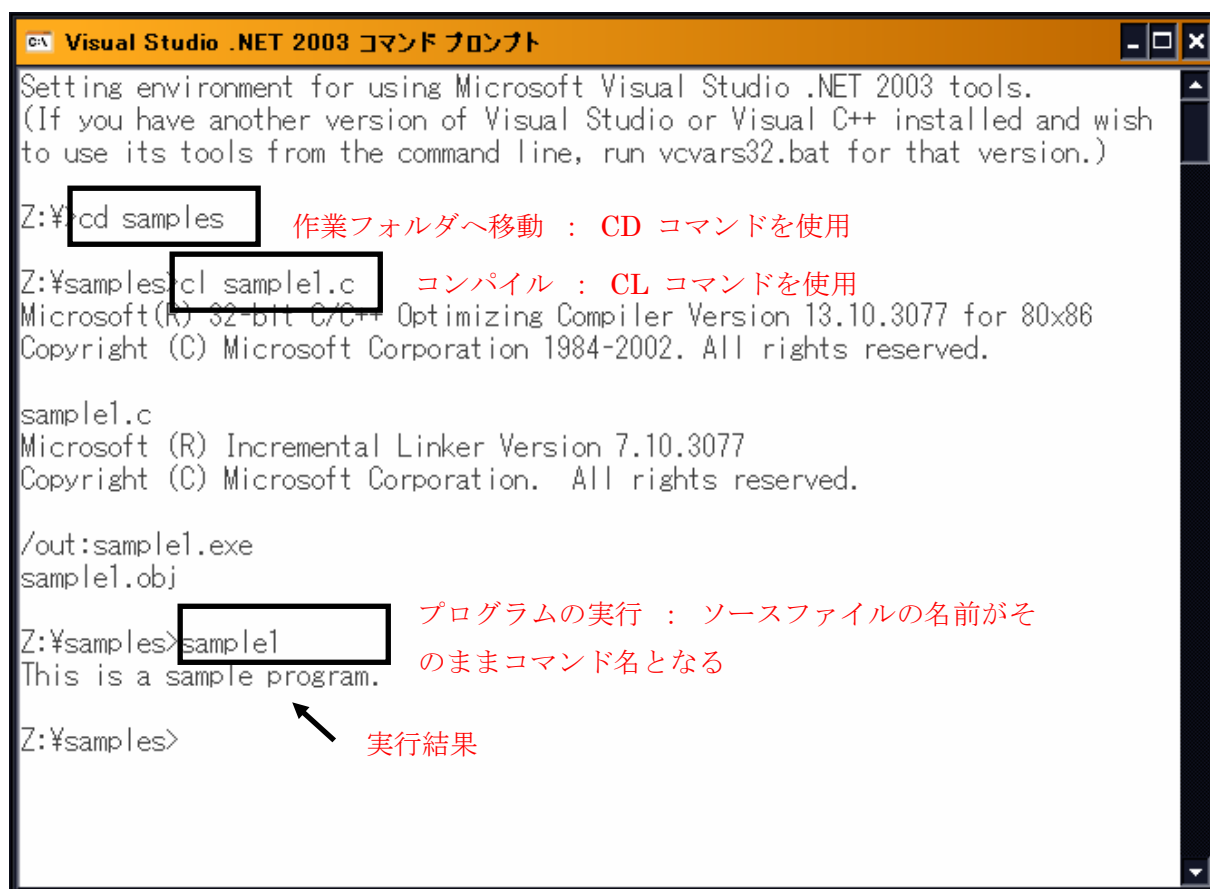
`cl ファイル名.c` (.c が .cpp の場合もある)

エラーが発生した時は、最初に報告されているエラーの行番号を確認して修正の手がかりとしておく。②に戻って必要な修正と保存を行なう。

④ 実行

翻訳が正常終了したらファイル名と同じ名前の実行可能ファイル (拡張子.exe のファイル) が同じフォルダ内に作成され、これが新たなコマンドとして利用できる。つまり、自分のプログラムを起動するには、ファイル名 (拡張子.c や.cpp を省いたもの) を入力する。

ファイル名



(実際には白黒反転した画面となる)

図 Q コマンドプロンプト上での作業の様子