オブジェクト指向と
ゲームプログラミング

▷　フレームワーク編 － 第２回　アプリケーションクラスの作成　◁

課　　題

ゲームアプリケーションを表現するクラスCGameAppを作成しましょう。

オブジェクト指向では、継承や集約を用いて既存のクラスを拡張していくことにより、作業効率を高めます。ゲーム向けアプリケーションクラスの場合は、さまざまなアプリケーションに共通する部分をクラスまたはインタフェース化し、それを継承または集約して拡張するような設計にします。
今回は、テーマをゲームに絞るので、そのような設計にはせず、CGameAppクラスにすべてを定義することにします。

(１)CGameAppクラスのヘッダファイル(GameApp.hpp)を以下のように作成しましょう。

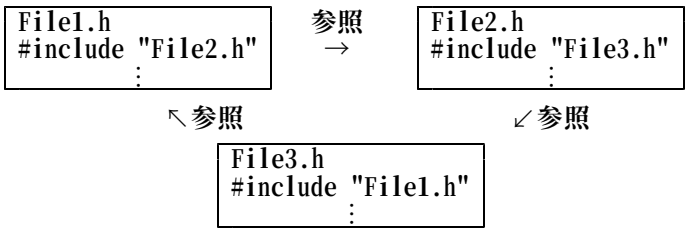
－GameApp.hpp－

```
/*
===========================================================================================
                        オブジェクト指向ゲームプログラミング
     Programmed by Hibikino software. Copyright (c) 2005 Hibikino software. All rights reserved.
===========================================================================================

  【対象ＯＳ】
     Microsoft Windows2000/XP

  【コンパイラ】
     Microsoft Visual C++ 2005

  【プログラム】
     GameApp.hpp
                ゲームアプリケーションクラスヘッダ

  【履歴】
     * Version     1.00       2005/03/dd  hh:mm:ss

===========================================================================================
*/

#pragma once

/*********************************************************************************************/
/*                                      インクルードファイル                                 */
/*********************************************************************************************/
#include <windows.h>

/*********************************************************************************************/
/*                                   ゲームアプリケーションクラス定義                        */
/*********************************************************************************************/
class CGameApp {
public:

private:
    CGameApp(const CGameApp&);              // コピーコンストラクタを生成しない
    CGameApp& operator=(const CGameApp&);   // 代入演算子を生成しない
};
```

ヘッダファイル(.hまたは.hpp)は、ソースファイルやほかのヘッダファイルの冒頭でインクルードされるファイルです。
ヘッダファイルの目的は、ほかのファイルで必要となる情報(対応するソースファイルで定義されている関数のプロトタイプ、ほかのファイルでも参照する定数、クラス、構造体、列挙体などの宣言および定義を行うことです。

ヘッダファイルの冒頭には、必ず「#pragma once」と記述しておきます。これは、同じヘッダファイルは１度しかインクルードしないという指定です。間違って(または暗黙のうちに)２回以上インクルードしたときの「すでに定義されています」といったエラーを防ぎ、次のようにヘッダファイルを互いに

参照しあって無限にインクルードしてしまうという現象を防ぐことができます。



(２)CGameAppクラスのソースファイル(GameApp.cpp)を以下のように作成しましょう。

－GameApp.cpp－

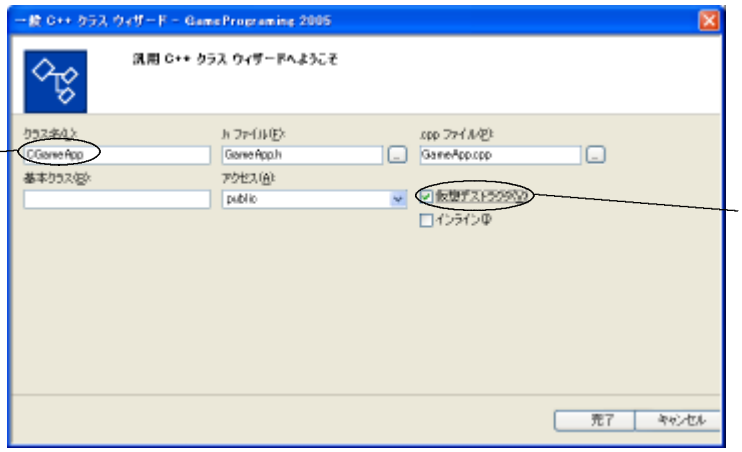
```
/*
========================================================================================================
                                オブジェクト指向ゲームプログラミング
     Programmed by Hibikino software. Copyright (c) 2005 Hibikino software. All rights reserved.
========================================================================================================

  【対象ＯＳ】
     Microsoft Windows2000/XP

  【コンパイラ】
     Microsoft Visual C++ 2005

  【プログラム】
     GameApp.cpp
                ゲームアプリケーションクラス

  【履歴】
     * Version     1.00       2005/03/dd  hh:mm:ss

========================================================================================================
*/

/*******************************************************************************************************/
/*                                          インクルードファイル                                       */
/*******************************************************************************************************/
#include "???????.???"
```

(３)CGameAppクラスのコンストラクタとデストラクタを作成します。ヘッダファイルに、コンストラクタとデストラクタのプロトタイプを記述しましょう。なお、デストラクタは、念のため仮想デストラクタとして宣言しましょう。

(４)CGameAppクラスのソースファイルに、コンストラクタとデストラクタを実装します。以下のプログラムの足りない部分を補って完成させましょう。

```
/*******************************************************************************************************/
/*                                          コンストラクタ                                             */
/*******************************************************************************************************/
????????::CGameApp()
{
}

/*******************************************************************************************************/
/*                                          デストラクタ                                               */
/*******************************************************************************************************/
????????::~CGameApp()
{
}
```

※(１)～(４)の作業は、クラスウィザードにある程度行わせることができます。クラスウィザードは、メニューから「ﾌﾟﾛｼﾞｪｸﾄ(P)→ｸﾗｽの追加(C)」を選択します。クラスの追加ダイアログが開いたら、「ｶﾃｺﾞﾘ(C)」の"一般"、「ﾃﾝﾌﾟﾚｰﾄ(T)」の"一般 C++ ｸﾗｽ"が選択された状態で追加ボタンをクリックします。



汎用C++クラスウィザードが表示されるので、「ｸﾗｽ名(L)」にクラス名"CGameApp"と入力すると、ヘッダファイルやソースファイルの名前が自動的に設定されます。「仮想ﾃﾞｽﾄﾗｸﾀ(V)」にチェックを入れて完了ボタンをクリックすると、CGameAppクラスが最低限作成されます。

(５)CGameAppクラスに必要な機能を追加します。最低限必要なのは、アプリケーションの初期化を行う機能、使用していた資源を解放する機能、そしてメイン処理を実行する機能です。それぞれ、Initialize関数、Release関数、Main関数とします。これらの関数のプロトタイプは、以下のようになります。ヘッダファイルの適切な場所に追加しましょう。

```
bool Initialize();
void Release();

int Main();
```

(６)ソースファイルにInitialize関数、Release関数、Main関数を実装します。以下のプログラムの足りない部分を補って完成させましょう。

```
/*****************************************************************************************************/
/*                                          初期化                                                   */
/*****************************************************************************************************/
bool ????????::Initialize()
{
    return ????;    // 成功
}

/*****************************************************************************************************/
/*                                          解　放                                                   */
/*****************************************************************************************************/
void ????????::Release()
{
}

/*****************************************************************************************************/
/*                                          メイン                                                   */
/*****************************************************************************************************/
int ????????::Main()
{
    return 0;       // 正常終了
}
```

(７)CGameAppクラスのメンバは、以下のとおりです。

CGameApp　コンストラクタ

CGameAppオブジェクトを構築します。

~CGameApp　デストラクタ

CGameAppオブジェクトを解放します。

Initialize 初期化

アプリケーションを初期化します。

試bool Init();

Return 初期化成功：true　それ以外：false

Release 解放

アプリケーションが使用していた資源を解放し、プログラム終了に備えます。

Main メイン処理

アプリケーションのメイン処理を行います。

試int Main();

Return 正常終了：0　それ以外：0以外

(8)WinMain.cppのコメント「インクルードファイル」以下の部分を、以下のように変更しましょう。
なお、足りない部分は補ってください。

```
/*****************************************************************************************************/
/*                                          インクルードファイル                                     */
/*****************************************************************************************************/
#include "GameApp.hpp"

/*****************************************************************************************************/
/*                                            WinMain関数                                            */
/*****************************************************************************************************/
int ?????? WinMain(????????? hInstance, ????????? hPrevInstance, LPTSTR lpCmdLine, ??? nShowCmd)
{
    // CGameAppオブジェクト生成
    ????????   App;

    // アプリケーション初期化
    if(App.??????????() == false)
        return -1;

    // メイン処理
    const int  ret = App.????();

    // 解放
    App.???????();

    return ret;
}
```

※WinMain関数は、CGameAppクラスのデストラクタを正しく記述することにより、以下のように省略することができます。

```
int ?????? WinMain(????????? hInstance, ????????? hPrevInstance, LPTSTR lpCmdLine, ??? nShowCmd)
{
    // CGameAppオブジェクト生成
    ????????   App;

    // アプリケーション初期化
    if(App.??????????() == false)
        return -1;

    // メイン処理
    return App.????();
}
```

(9)ソリューションエクスプローラに、フレームワーク用のフォルダを作成しましょう。

フォルダを作成するには、ソリューションエクスプローラの「ｿｰｽ ﾌｧｲﾙ」で右クリックし、「追加(D)▶新しいﾌｨﾙﾀ(F)」を選択します。作成されたフォルダを"ﾌﾚｰﾑﾜｰｸ"に変更し、GameApp.cppを移動します。

(10)ヘッダファイルGameApp.hppも同様にフォルダを作成し、移動しましょう。