



C++ - モジュール 00

名前空間、クラス、メンバ関数、stdioストリーム、初期化リスト、static、const、その他いくつかの基本的なもの

概要

本書は、C++モジュールのうち、モジュール00の演習問題を収録したものです。

バージョン : 8

内容

I	はじめに	2
II	一般規定	3
III	練習曲00：メガホン	5
IV	練習問題01：私のすごいPhoneBook	6
V	Exercise 02: 夢をかなえる仕事	8

第一章 はじめに

C++は、*Bjarne Stroustrup*が「C with Classes」(出典：[Wikipedia](#))というプログラミング言語の元祖として作った汎用プログラミング言語である。

これらのモジュールの目標は、オブジェクト指向プログラミングを紹介することです。これは、あなたのC++の旅の出発点となるでしょう。オブジェクト指向を学ぶには、多くの言語が推奨されます。この言語は複雑な言語であり、物事を単純化するために、あなたのコードはC++98標準に準拠することになります。

私たちは、現代のC++が多く側面で大きく異なっていることを認識しています。ですから、もしあなたが熟練したC++開発者になりたいのであれば、42のコモンコアの後にさらに進むのはあなた次第なのです。

新しいコンセプトを段階的に発見することができます。演習は徐々に複雑さを増していきます。

第II章 総則

コンパイル

- `c++`と`-Wall -Wextra -Werror`フラグでコンパイルしてください。
- フラグ `-std=c++98` を追加しても、あなたのコードはコンパイルできるはずです。

フォーマットと命名規則

- 演習用のディレクトリは、`ex00`, `ex01`, ...のように命名されます。 `exn`
- ファイル、クラス、関数、メンバ関数、属性にガイドラインに必要な名前を付ける。
- クラス名は、**UpperCamelCase** 形式で記述します。クラスコードを含むファイルには、常にクラス名に従った名前が付けられます。例えば例えば、`ClassName.hpp/ClassName.h`, `ClassName.cpp`, または `ClassName.tpp`.例えば、レンガの壁を表すクラス "`BrickWall` "の定義を含むヘッダーファイルがあれば、その名前は`BrickWall.hpp`となります。
- 特に指定がない限り、すべての出力メッセージは改行文字で終了し、標準出力に表示されなければならない。
- さよならノルミネット!C++モジュールでは、コーディングスタイルは強制されません。あなたの好きなものに従えばいいのです。しかし、同僚評価者が理解できないコードは、評価できないコードであることを心に留めておいてください。きれいで読みやすいコードを書くために、最善を尽くしてください。

許可/禁止

あなたはもうCでコーディングしていない。C++の時代だ!というわけで。

- 標準ライブラリのほとんどすべてを使用することが許されています。したがって、すでに知っているものに固執するのではなく、使い慣れたC関数のC++的なバージョンをできるだけ使うのが賢い方法でしょう。
- ただし、それ以外の外部ライブラリは使用できません。つまり、C++11（および派生形式）とBoostライブラリは禁止されています。以下の関数も禁止されています。`*printf()`、`*alloc()`、`free()`。これらを使用した場合、成績は0点、それで終わりです。

- 明示的に指定しない限り、using名前空間<ns_name>と友人キーワードは禁止です。そうでない場合、あなたの成績は-42となります。
- **STL の使用はモジュール 08 のみ許可されています。**つまり、コンテナ (vector/list/map/ など) とアルゴリズム (<algorithm> ヘッダを含む必要があるもの) はそれまで使わないでください、ということです。さもないと、あなたの成績は -42 になります。

いくつかのデザイン要件

- C++でもメモリリークは発生します。メモリを確保するときに (new キーワード)を使用する場合は、メモリリークを回避する必要があります。
- モジュール02からモジュール08までは、特に明記されている場合を除き、正教会の正書法で授業を設計する必要があります。
- ヘッダーファイルに書かれた関数の実装は (関数テンプレートを除いて)、演習では0を意味します。
- それぞれのヘッダーは、他のヘッダーと独立して使用できるようにする必要があります。従って、必要な依存関係はすべてインクルードしなければなりません。しかし、インクルードガードを追加することによって、二重インクルードの問題を回避しなければなりません。そうでなければ、あなたの成績は0点となります。

リードミー

- 必要であれば、いくつかのファイルを追加することができます (コードを分割するためなど)。これらの課題は、プログラムによる検証を行わないので、必須ファイルを提出する限り、自由に行ってください。
- 演習のガイドラインは短く見えても、例題には明示的に書かれていない要件が示されていることがあります。
- 始める前に、各モジュールを完全に読んでください本当に、そうしてください。
- オーディンによって、ツールによって!頭を使い!!!



多くのクラスを実装する必要があります。これは、お気に入りのテキストエディタでスクリプトを書くことができる人でなければ、面倒に思えるかもしれません。



演習をこなすには、ある程度の自由度が与えられています。ただし、必須のルールは守り、怠慢は禁物です。多くの有益な情報を見逃すことになりますよ。理論的な概念については、ためらわずに読んでください。

第三章

練習曲00：メガホン

	エクササイズ: 00 メガ
ホン	
ターンインディレクトリ: <i>ex00/</i>	
提出するファイル: Makefile、megaphone.cpp	
禁止されている機能: なし	

みんなが起きていることを確認するために、次のような動作をするプログラムを書いてみてください。


```
$>./megaphone "しーっ...生徒が寝ているようです..."しーっ...しーっ...生徒
が寝ているようです...
$>./megaphone くそったれ!"!" "学生諸君、申し訳ないが、これはオフだと思った。"く
そったれ!生徒の皆さん、ごめんなさい、これはオフだと思ってたんです。
$>./megaphone
*大きく、耐え難いフィードバック音*
$>
```



C++で演習問題を解く。

第四章

練習問題01：私のすごいPhoneBook

	演習：01
マイ・オーサム・フォン・ブック	
ターンインディレクトリ: <i>ex01/</i>	
提出するファイル: <i>Makefile</i> 、 <i>*.cpp</i> 、 <i>*.{h, hpp}</i> 。	
禁止されている機能: なし	

80年代とその信じられないような技術へようこそ!くだらない凄い電話帳ソフトのような振る舞いをするプログラムを書いてください。

2つのクラスを実装する必要があります。

- 電話帳

- 接点が充実している。
- 最大**8件**の**連絡先**を保存することができます。ユーザーが**9番目**の連絡先を追加しようとした場合、最も古い連絡先を新しい連絡先で置き換えます。
- なお、ダイナミックアロケーションは禁止されています。

- お問い合わせ先

- 電話帳の連絡先を表します。

あなたのコードでは、電話帳は**PhoneBook**クラスのインスタンスとして作成する必要があります。連絡先も同じです。それぞれ**Contact**クラスのインスタンスとしてインスタンス化されなければなりません。クラスは自由に設計できますが、クラス内部で常に使用されるものは**private**、クラス外部で使用されるものは**public**であることに留意してください。



イントラネットのビデオも忘れずに見てください。

プログラム起動時、電話帳は空で、ユーザーは3つのコマンドのうち1つを入力するよう促されます。プログラムは、ADD、SEARCH、EXITのみを受け付けます。

- **ADD** : 新しいコンタクトを保存する

- ユーザーがこのコマンドを入力すると、新しい連絡先の情報を1フィールドずつ入力するように促されます。すべてのフィールドの入力が完了したら、その連絡先を電話帳に追加します。
- 連絡先のフィールドは、名前、苗字、ニックネーム、電話番号、そして最も暗い秘密です。保存された連絡先は、空のフィールドを持つことはできません。

- **SEARCH** : 特定のコンタクトを表示する

- 保存された連絡先をインデックス、名、姓、ニックネームの**4列**のリストとして表示します。
- 各列の幅は**10文字**です。各列はパイプ文字('|')で区切られます。テキストは右寄せにしなければならない。テキストが列より長い場合、テキストは切り捨てられ、表示可能な最後の文字はドット('.')
- その後、表示するエントリーのインデックスを再度ユーザーに入力させる。インデックスが範囲外であるか、間違っている場合は、関連する動作を定義する。そうでない場合は、連絡先情報を1行に1フィールドずつ表示する。

- **エグジット**

- プログラムが終了し、連絡先が永遠に失われてしまいます。

- **それ以外の入力**は破棄される。

あるコマンドが正しく実行されると、プログラムは次のコマンドを待ちます。ユーザーがEXITを入力すると停止する。


実行ファイルに関連する名前を付けます。



<http://www.cplusplus.com/reference/string/string/>、もちろん
<http://www.cplusplus.com/reference/iomanip/>。

第五章

Exercise 02: 夢をかなえる仕事

	演習 : 02
夢のある仕事	
ターンインディレクトリ : <i>ex02/</i>	
提出するファイル: <i>Makefile, Account.cpp, Account.hpp, tests.cpp</i> 。	
禁止されている機能 : なし	



Account.hpp、*tests.cpp*、ログファイルは、このモジュールのイントラネットページからダウンロードできます。

今日は*GlobalBanksters United*の入社初日です。再採用試験に無事合格し（友人が教えてくれた*Microsoft Office*の小技のおかげで）、あなたは開発チームに参加しました。*Adobe Reader*のインストールがとても早かったので、採用担当者が驚いていたことも知っている。その小さな積み重ねが、他の応募者たちとの差を縮めたのです。

とにかく、あなたは上司から仕事を任されただけです。最初の仕事は、失われたファイルを再生することです。何かの間違いで、ソース・ファイルを削除してしまったのです。残念ながら、あなたの同僚は*Git*が何であるかを知らず、コードを共有するために*USB*キーを使っています。この時点で、今すぐこの場所を離れるのが筋でしょう。しかし、あなたはここに留まることにしました。チャレンジを受け入れました

開発者仲間から大量のファイルを渡される。*tests.cpp*をコンパイルしてみると、足りないファイルは*Account.cpp*であることがわかる。ラッキーなことに、ヘッダーファイルの*Account.hpp*が保存されていた。また、ログファイルもあります。**Account**クラスがどのように実装されたかを理解するために、もしかしたらあなたはそれを使うことができるかもしれません。

Account.cppファイルの再作成を開始します。わずか数分で、あなたは純粋に素晴らしいC++を数行コーディングしました。何回かコンパイルに失敗した後、あなたのプログラムはテストに合格しました。その出力は、ログファイルに保存されたものと完全に一致します（ログファイルに保存されたテストは、あなたが雇われる前に実行されたので、明らかに異なるタイムスタンプを除きます）。

くっそー、感動したー。



デストラクタが呼ばれる順序は、コンパイラやオペレーティングシステムによって異なる場合があります。そのため、デストラクタが呼ばれる順番が逆になっている可能性があります。



演習02を行わずとも、このモジュールに合格することができます。

