



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、その他基本的な
もの

概要

このドキュメントには、C++ モジュールのモジュール00の練習問題が含まれて
います。

バージョン: 9

内容

| | | |
|-----|------------------|----|
| I | はじめに | 2 |
| II | 一般規定 | 3 |
| III | 練習00: メガホン | 5 |
| IV | 練習01: 私の素晴らしい電話帳 | 6 |
| V | エクササイズ02: 夢の仕事 | 8 |
| VI | 提出と相互評価 | 10 |

第一章 はじめに

C++は、*Bjarne Stroustrup*（ビャルネ・ストルストラップ）により、C言語の発展形として開発された汎用プログラミング言語である（出典：[Wikipedia](#)）。

これらのモジュールの目的は、**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。OOPを学ぶには多くの言語が推奨されています。これは複雑な言語であり、物事をシンプルに保つために、あなたのコードはC++98標準に準拠することになります。

現代のC++は多くの面で大きく異なっていることは承知しています。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です！

段階を追って新しいコンセプトを発見していきます。練習問題は徐々に複雑になっていきます。

第II章 一般規定

コンパイル

- `c++`と`-Wall -Wextra -Werror`フラグでコードをコンパイルする。
- フラグ`-std=c++98`を追加しても、コードはコンパイルできるはずだ。

フォーマットと命名規則

- 演習用のディレクトリは次のように命名されます: `ex00`, `ex01`, ..., `exn`.
`exn`
- ファイル名、クラス名、関数名、メンバ関数名、属性名は、ガイドラインの要求に従ってください。
- クラス名は**UpperCamelCase**形式で記述する。クラス・コードを含むファイルは常にクラス名に従って命名されます。例えば例えば、
`ClassName.hpp/ClassName.h`、`ClassName.cpp`、`ClassName.hpp`。例えば、
`ClassName.hpp/ClassName.hpp/ClassName.cpp/ClassName.hpp`のようになります。
- 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。
- さようならノルミネット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。あなたの好きなスタイルに従ってください。しかし、相互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よう、ベストを尽くしてください。

許可/禁止

あなたはもうC言語でコーディングしていない。C++の出番だ！ だから

- 標準ライブラリのほとんどすべてを使うことが許されている。したがって、すでに知っていることに固執するのではなく、使い慣れたC関数のC++っぽいバージョンをできるだけ使うのが賢いやり方だろう。
- ただし、それ以外の外部ライブラリーは使用できない。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る。以下の関数も禁止されている：*printf()、*alloc()、free()です。これらを使用した場合、あなたの成績は0点となり、それで終わりです。

- 明示的に指定されていない限り、using 名前空間 `<ns_name>` と友達キーワードは禁止。さもないと、あなたの成績は-42点となる。
- **モジュール 08 と 09 でのみ STL を使用できます。**つまり、それまでは**コンテナ**（vector/list/map/など）や**アルゴリズム**（`<algorithm>`ヘッダーを含む必要があるもの）を使用しないこと。そうでなければ、成績は-42点となる。

いくつかの設計要件

- メモリ・リークはC++でも発生する。メモリを確保するときに（new キーワード）、**メモリー・リーク**を避けなければならない。
- モジュール02からモジュール09まで、**特に明記されている場合を除き**、あなたのクラスは**正教会正書式**でデザインされなければなりません。
- 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）、練習にとっては0を意味する。
- それぞれのヘッダーは、他のヘッダーから独立して使えるようにしなければならない。したがって、必要な依存関係をすべてインクルードしなければならない。しかし、**インクルード・ガード**を追加することで、二重インクルードの問題を避け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。

続きを読む

- 必要であれば（コードを分割するためなど）ファイルを追加しても構いません。これらの課題はプログラムによって検証されるわけではないので、必須ファイルを提出する限り、自由に行ってください。
- 時には、練習のガイドラインが短く見えても、例題を見れば、指示には明示的に書かれていない要件がわかることもある。
- 始める前に各モジュールを完全に読むこと！ 本当にそうしてください。

- オーディンによって、ソーによって！頭を使え！



たくさんのクラスを実装しなければならない。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ない限り、退屈に思えるかもしれない。



練習をこなすにはある程度の自由が与えられている。しかし、義務的なルールは守り、怠けてはいけません。多くの有益な情報を見逃すことになります！理論的な概念を読むことをためらわないでください。

第三章


練習00：メガホン

| | |
|---|--------|
|  | エクササイズ |
| | : 00 |
| ターニン・ディレク | |
| 提出するファイル: Makefile, メガホン | |
| megaphone.cpp | |

禁止されている関数： 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よう：


```
$>./メガホン "シーっ...生徒が寝ているようです..."シーッ...生徒たちは眠
っていると思います...
$>./megaphone Damnit "！"生徒諸君、申し訳ないが、これは休みだと思っていた。"ちく
しょう！生徒の皆さん、ごめんなさい。
メガホン
* ハウリングノイズが大きく、耐え難い。
```



C++の方法で練習問題を解く。

第四章

練習01：私の素晴らしい電話帳

| | |
|---|-----------|
|  | エクササイズ：01 |
| 私の素晴らしい電話帳 | |
| ターンイン・ディレクトリ：ex01/ | |
| 提出するファイル：Makefile、*.cpp、*.{h、hpp}。 | |
| 禁止されている関数：なし | |

80年代とその信じがたいテクノロジーへようこそ！電話帳ソフトのようなプログラムを書いてみよう。

つのクラスを実装しなければならない：

- **フォンブック**

- さまざまなコンタクトがある。
- 最大**8件の連絡先**を保存できる。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に置き換えます。
- なお、動的割り当ては禁止されている。

- **連絡先**

- 電話帳の連絡先を表す。

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インスタンス化されなければなりません。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する必要があります。クラスをどのように設計するかは自由ですが、クラス内部で常に使用されるものはprivateであり

、クラス外部で利用できるものはpublicであることを覚えておいてください。



イントラネットのビデオを見ることもお忘れなく。

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る。プログラムはADD、SEARCH、EXITのみを受け付ける。

- **追加:** 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姓、名、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:** 特定の連絡先を表示する

- 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**でなければならない。パイプ文字 (|) で区切ります。テキストは右揃えにしなければならない。テキストが列より長い場合は切り捨てられ、表示可能な最後の文字はドット (.) に置き換えられなければなりません。
- 次に、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。そうでなければ、連絡先情報を1行に1フィールドずつ表示する。

- **退場**

- プログラムは終了し、連絡先は永遠に失われる!

- **それ以外の入力**は破棄される。

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つ。ユーザ

一がEXITを入力すると停止する。


実行ファイルに適切な名前を付ける。



<http://www.cplusplus.com/reference/string/string/> そしてもちろん
<http://www.cplusplus.com/reference/iomanip/>

第五章

練習02：夢の仕事

| | |
|---|-----------|
|  | エクササイズ：02 |
| 夢の仕事 | |
| ターンイン・ディレクトリ：ex02/ | |
| 提出ファイル：Makefile、Account.cpp、Account.hpp、tests.cpp | |
| 禁止されている関数：なし | |



Account.hpp、tests.cpp、ログ・ファイルは、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*への入社初日だ。再採用テストに無事合格し（友人が教えてくれた*Microsoft Office*の小技のおかげ）、あなたは開発チームに加盟した。あなたが*Adobe Reader*を素早くインストールしたことに、採用担当者が驚いていたこともご存知だろう。その小さな積み重ねがすべての違いを生み、すべての敵（他の応募者）を打ち負かすのに役立った！

とにかく、上司から仕事を任されたんだ。最初の仕事は紛失したファイルの再作成だ。何かの間違いでソース・ファイルを削除してしまったのだ。残念なことに、あなたの同僚はGitが何なのか知らず、USBキーでコードを共有している。この時点で、今すぐここを離れるのが筋だろう。しかし、あなたは残ることにした。挑戦は受け入れられた！

開発者仲間からたくさんのファイルを渡された。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は保存されていた。ログファイルもある。**Account**クラスがどのように実装されたかを理解するために、それを使うことができるかもしれない。

Account.cppファイルを作り直す。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た。何度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する（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ものなので、**タイムスタンプ**が明らかに異なることを除いて）。

すごいね！



デストラクタが呼ばれる順番は、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る。そのため、デストラクタは逆の順序で呼び出されるかもしれません。



練習問題02を解かなくても、このモジュールに合格することができます。

第六章

提出と相互評価

通常通り、Gitリポジトリに課題を提出してください。あなたのリポジトリ内の作品だけが、ディフェンス中に評価されます。ファイル名が正しいかどうか、ためらわずにダブルチェックしてください。



?????????????XXXXXXXXXX = \$3\$f15bc138aca1e76ec6f4cfd0797ec037

