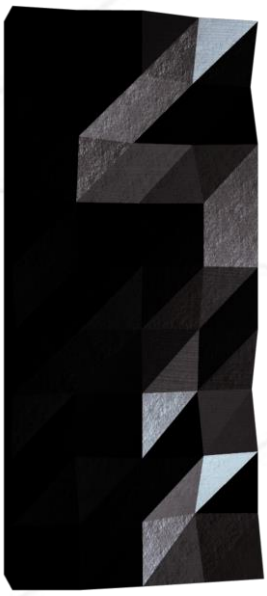

C++ - モジュール 07

C++テンプレート

概要
本書は、C++モジュールのうち、Module 07の演習問題を収録しています。

バージョン：6

内容

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++が多くの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!

第II章 総則

コンパイル

- c++と-Wall -Wextra -Werrorフラグでコードをコンパイルします。
- 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、そのファイル名は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。さもなければ、あなたの成績は -42 となります。

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。メモリを確保する際に（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、トールによって!頭を使え!!!

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、退屈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、必須のルールを守り、怠慢にならないようにしましょう。 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

練習問題00：いくつかの関数から始める

	演習：00
いくつかの機能から始める	
ターンインディレクトリ : *ex00/*	
提出するファイル:Makefile, main.cpp, whatever.{h, hpp}.	
禁止されている機能 :なし	

以下の機能テンプレートを実装する。

- をスワップする。与えられた2つの引数の値を交換する。何も返さない。
- min: 引数で渡された2つの値を比較し、最も小さいものを返す。もし2つの値が等しければ、2番目の値を返す。
- max:引数で渡された2つの値を比較し、最も大きいものを返します。もし2つの値が等しければ、2番目の値を返す。

これらの関数は、どのような型の引数でも呼び出すことができます。唯一の要件は、2つの引数が同じ型であり、すべての比較演算子をサポートしていなければならないことです。

テンプレートはヘッダーファイルで定義する必要があります。

以下のコードを実行する。

```
int		main( void ) {

	int a = 2;
	int b = 3;
```

```
a = 3, b = 2
min(a, b) = 2
max(a, b) = 3
c = chaine2, d = chaine1
min(c, d) = chaine1 max(c,
d) = chaine2
```

```
	chaine2 );

	::swap(c, d)です。
	std::cout << "c = " << c << ", d = " << d << std::endl; std::cout <<
	"min( c, d ) = " << ::min( c, d ) << std::endl;
	std::cout << "max( c, d ) = " << ::max( c, d ) << std::endl;

	0を返します。
}
```

第IV章 練習問題

01：イター

	演習：01
イター	
ターンインディレクトリ : *ex01/*	
提出するファイル:Makefile, main.cpp, iter.{h, hpp}。	
禁止されている機能 :なし	

3つのパラメータを受け取り、何も返さない関数テンプレート iter を実装してください。

- 第1引数は配列のアドレスである。
- 2つ目は、配列の長さです。
- 3つ目は、配列の各要素に対して呼び出される関数です。

テストを含む main.cpp ファイルを提出する。テストの実行ファイルを生成するのに十分なコードを提供すること。

iter関数テンプレートは、任意の型の配列で動作する必要があります。第3パラメータには、インスタンス化された関数テンプレートを指定することができます。

第V章 演習02：配列

	演習：02
アレイ	
ターンインディレクトリ：*ex02/*	
提出するファイル:Makefile, main.cpp, Array.{h, hpp}とオプションファイル。 Array.tpp	
禁止されている機能 :なし	

型Tの要素を含み、以下の動作と機能を実装したクラステンプレート**Array**を開発しなさい。

- パラメータを指定しない構築。空の配列を作成します。
- unsigned int n をパラメータに持つ構築。デフォルトで初期化されたn個の要素を持つ配列を作成する。
  ヒント：int * a = new int(); をコンパイルして、*a を表示してみてください。
- コピーと代入演算子による構築。どちらの場合も、コピー後に元の配列またはそのコピーを変更しても、もう一方の配列に影響を与えることはありません。
- メモリを割り当てるには、必ず new[] 演算子を使用しなければなりません。予防的割り当て（事前にメモリを確保すること）は禁止されています。プログラムは決して割り当てられたメモリ以外のものにアクセスしてはいけません。
- 要素は、添え字演算子でアクセスすることができます。[ ].
- 演算子で要素にアクセスする際、そのインデックスが範囲外である場合は std::exception がスローされます。
- 配列の要素数を返すメンバ関数 size()。このメンバ関数はパラメータをとらず、現在のインスタンスを変更してはいけません。

いつものように、すべてが期待通りに動作することを確認し、テストを含む main.cpp ファイルを提出します。