



## C++ - 모듈 09

### STL

*요약:*

*이 문서에는 C++ 모듈 중 모듈 09의 연습 문제가 포함되어 있습니다.*

*버전: 2*

# 콘텐츠

I	소개	2
II	일반 규칙	3
III	모듈별 규칙	5
IV	연습 00: 비트코인 거래소	6
V	연습 01: 역 폴란드어 표기법	8
VI	연습 02: P머지미	10
VII	제출 및 동료 평가	13



# 1장 소개

*C++는 비야른 스트로스트룹이 C 프로그래밍 언어 또는 '클래스가 있는 C'의 변형으로 만든 범용 프로그래밍 언어입니다(출처: [위키백과](#)).*

이 모듈의 목표는 **객체 지향 프로그래밍**을 소개하는 것입니다. 이것이 C++ 여정의 시작점이 될 것입니다. OOP를 배우기 위해 많은 언어가 권장됩니다. C++는 여러분의 오랜 친구인 C에서 파생된 언어이기 때문에 C++를 선택하기로 결정했습니다. C++는 복잡한 언어이며, 코드를 단순하게 유지하기 위해 C++98 표준을 준수합니다.

최신 C++는 여러 측면에서 많이 다르다는 것을 알고 있습니다. 따라서 능숙한 C++ 개발자가 되고 싶다면 42개의 공통 코어를 넘어서는 것은 여러분에게 달려 있습니다!



## 2장 일반 규칙

### 컴파일

- c++와 -Wall -Wextra -Werror 플래그를 사용하여 코드를 컴파일합니다.
- 플래그 -std=c++98을 추가하면 코드가 계속 컴파일됩니다.

### 서식 및 이름 지정 규칙

- 연습 디렉터리의 이름은 ex00, ex01, ..., exn
- 가이드라인에서 요구하는 대로 파일, 클래스, 함수, 멤버 함수 및 속성의 이름을 지정하세요.
- 클래스 이름을 **대문자 대소문자** 형식으로 작성합니다. 클래스 코드가 포함된 파일은 항상 클래스 이름에 따라 이름이 지정됩니다. 예를 들어 ClassName.hpp/ClassName.h, ClassName.cpp 또는 ClassName.hpp. 예를 들어, 벽돌 벽을 의미하는 "BrickWall" 클래스의 정의가 포함된 헤더 파일이 있다면 그 이름은 BrickWall.hpp가 됩니다.
- 달리 지정하지 않는 한, 모든 출력 메시지는 새 줄 문자로 끝나야 하며 표준 출력에 표시되어야 합니다.
- *안녕, 노미네트!* C++ 모듈에는 코딩 스타일이 강제되지 않습니다. 좋아하는 스타일을 따를 수 있습니다. 하지만 동료 평가자가 이해할 수 없는 코드는 채점할 수 없는 코드라는 점을 명심하세요. 깔끔하고 읽기 쉬운 코드를 작성하기 위해 최선을 다하세요.

### 허용/금지

더 이상 C로 코딩하지 마세요. 이제 C++로 전환하세요! 그러므로:

- 표준 라이브러리의 거의 모든 것을 사용할 수 있습니다. 따라서 이미 알고 있는 것을 고수하는 대신 익숙한 C 함수의 C++ 버전을 최대한 많이 사용하는 것이 현명할 것입니다.
- 하지만 다른 외부 라이브러리는 사용할 수 없습니다. 즉, C++11(및 파생 형식) 및 Boost 라이브러리는 금지됩니다. 다음 함수도 금지됩니다: `*printf()`, `*alloc()` 및 `free()`. 이 함수를 사용하면 성적은 0점이 됩니다.

- 명시적으로 달리 명시되지 않는 한 네임스페이스 <ns\_name> 및 친구 키워드는 금지되어 있습니다. 그렇지 않으면 성적은 -42점이 됩니다.
- **모듈 08과 09에서만 STL을 사용할 수 있습니다.** 즉, 그때까지는 **컨테이너**(벡터 /리스트/맵 등)와 **알고리즘**(<알고리즘> 헤더를 포함해야 하는 모든 것)을 사용할 수 없습니다. 그렇지 않으면 성적이 -42점이 됩니다.

### 몇 가지 설계 요구 사항

- 메모리 누수는 C++에서도 발생합니다. 메모리를 할당할 때(새로운 키워드)의 경우 **메모리 누수**를 방지해야 합니다.
- 모듈 02부터 모듈 09까지, **명시적으로 달리 명시된 경우를 제외하고** 클래스는 반드시 **정통 정석 양식**으로 디자인해야 합니다.
- 헤더 파일에 있는 모든 함수 구현(함수 템플릿 제외)은 연습에 0을 의미합니다.
- 각 헤더를 다른 헤더와 독립적으로 사용할 수 있어야 합니다. 따라서 필요한 모든 종속성을 포함해야 합니다. 그러나 **include 가드**를 추가하여 이중 포함 문제를 피해야 합니다. 그렇지 않으면 성적이 0점이 됩니다.

### 읽기

- 필요한 경우 추가 파일을 추가할 수 있습니다(예: 코드를 분할하는 경우). 이러한 과제는 프로그램에서 확인하지 않으므로 필수 파일을 제출하는 한 자유롭게 추가할 수 있습니다.
- 간혹 연습 지침이 짧아 보이지만 지침에 명시되지 않은 요구 사항이 예제에 나와 있는 경우가 있습니다.
- 시작하기 전에 각 모듈을 완전히 읽으세요! 정말 그렇게 하세요.
- 오딘의, 토르의! 머리를 써라!!!



많은 클래스를 구현해야 합니다. 자주 사용하는 텍스트 편집기를 스크립팅할 수 없다면 이 작업이 지루해 보일 수 있습니다.





운동을 완료하는 데는 어느 정도의 자유가 주어집니다. 그러나 필수 규칙을 따르고 게으르지 마세요. 유용한 정보를 많이 놓칠 수 있습니다! 이론적 개념에 대해 주저하지 말고 읽어보세요.

## 제3장

### 모듈별 규칙

이 모듈의 각 연습을 수행하려면 반드시 표준 컨테이너를 사용해야 합니다.

컨테이너를 한 번 사용하면 나머지 모듈에는 사용할 수 없습니다.



연습을 하기 전에 주제 전체를 읽는 것이 좋습니다.



두 개의 용기를 사용해야 하는 운동 02를 제외하고 각 운동마다 최소 하나의 용기를 사용해야 합니다.

각 프로그램에 대해 소스 파일을 필요한 출력으로 컴파일하는 메이크파일을 -Wall, -Wextra 및 -Werror 플래그와 함께 제출해야 합니다.


c++를 사용해야 하며 메이크파일이 링크되지 않아야 합니다.

메이크파일에는 최소한 \$(NAME), all, clean, fclean, re 규칙이 포함되어야 합니다



## 제4장

### 연습 00: 비트코인 거래소

	운동 : 00
	비트코인 거래소
제출 디렉토리 : <i>ex00/</i>	
제출할 파일 : 메이크파일, 메인.cpp, 비트코인익스체인지.{cpp, hpp}	
금지된 기능 : 없음	

특정 날짜에 특정 금액의 비트코인 가치를 출력하는 프로그램을 만들어야 합니다.

이 프로그램은 시간 경과에 따른 비트코인 가격을 나타내는 csv 형식의 데이터베이스를 사용해야 합니다. 이 데이터베이스는 이 주제와 함께 제공됩니다.

이 프로그램은 두 번째 데이터베이스를 입력으로 받아 평가할 다양한 가격/날짜를 저장합니다.

프로그램은 이러한 규칙을 준수해야 합니다:

- 프로그램 이름은 btc입니다.
- 프로그램은 파일을 인수로 받아야 합니다.
- 이 파일의 각 줄은 다음 형식을 사용해야 합니다: "날짜 | 값".
- 유효한 날짜는 항상 다음 형식입니다: 연-월-일.
- 유효한 값은 0에서 1000 사이의 실수 또는 양의 정수여야 합니다.



이 연습의 유효성을 검사하려면 코드에서 컨테이너를 하나 이상 사용해야 합니다.  
발생 가능한 오류는 적절한 오류 메시지로 처리해야 합니다.

다음은 input.txt 파일의 예입니다:

```
$>
머리 입력.txt 날
짜 | 값
2011-01-03 | 3
2011-01-03 | 2
2011-01-03 | 1
2011-01-03 | 1.2
2011-01-09 | 1
2012-01-11 | -1
2001-42-42
2012-01-11 | 1
2012-01-11 | 2147483648
```

프로그램에서 입력 파일의 값을 사용합니다.

프로그램은 데이터베이스에 표시된 날짜에 따라 환율을 곱한 값의 결과를 표준 출력에 표시해야 합니다.



입력에 사용된 날짜가 DB에 존재하지 않는 경우 DB에 포함된 가장 가까운 날짜를 사용해야 합니다. 상위 날짜가 아닌 하위 날짜를 사용하도록 주의하세요.

다음은 프로그램 사용 예시입니다.

```
$> ./btc
오류: 파일을 열 수 없습니다.
./btc input.txt 2011-
01-03 => 3 = 0.9
2011-01-03 => 2 = 0.6
2011-01-03 => 1 = 0.3
2011-01-03 => 1.2 = 0.36
2011-01-09 => 1 = 0.32
오류: 양수가 아닙니다. 오류: 잘못
된 입력 => 2001-42-42 2012-01-
11 => 1 = 7.1
오류: 숫자가 너무 큼니다
```




경고: 이 연습의 유효성을 검사하는 데 사용하는 컨테이너는 이 모듈의 나머지 부분에서는 더 이상 사용할 수 없습니다.



## 5장

### 연습 01: 역 폴란드어 표기법

	운동 : 01
	RPN
	체크인 디렉토리 : <i>ex01/</i>
	제출할 파일 : 메이크파일, <i>main.cpp</i> , <i>RPN.{cpp, hpp}</i>
	금지된 기능 : 없음

이러한 제약 조건이 있는 프로그램을 만들어야 합니다:

- 프로그램 이름은 RPN입니다.
- 프로그램은 반전된 폴란드 수학 식을 인수로 사용해야 합니다.
- 이 연산에 사용되며 인수로 전달되는 숫자는 항상 10보다 작습니다. 계산 자체뿐만 아니라 결과도 이 규칙을 고려하지 않습니다.
- 프로그램은 이 표현식을 처리하여 표준 출력에 올바른 결과를 출력해야 합니다.
- 프로그램 실행 중에 오류가 발생하면 표준 출력에 오류 메시지가 표시되어야 합니다.
- 프로그램은 이러한 토큰으로 작업을 처리할 수 있어야 합니다: "+ - / \*".



이 연습의 유효성을 검사하려면 코드에서 컨테이너를 하나 이상 사용해야 합니다.







괄호나 소수를 관리할 필요가 없습니다.

C++ - 모듈 09

STL

다음은 표준 사용의 예입니다:

```
$> ./rpn "8 9 * 9 - 9 - 4 - 1 +"  
42  
$> ./rpn "7 7 * 7 -"  
42  
$> ./rpn "1 2 * 2 / 2 * 2 4 - +"  
0  
$> ./rpn "(1 + 1)"  
오류  
$
```




경고: 이전 연습에서 사용한 컨테이너는 여기서는 금지되어 있습니다. 이 연습의 유효성을 검사하는 데 사용한 컨테이너는 이 모듈의 나머지 부분에서는 사용할 수 없습니다.



## 제6장

### 연습 02: P머지미

	운동 : 02
	PmergeMe
	제출 디렉토리 : <i>ex02/</i>
	제출할 파일 : Makefile, main.cpp, PmergeMe.{cpp, hpp}
	금지된 기능 : 없음

이러한 제약 조건이 있는 프로그램을 만들어야 합니다:

- 프로그램 이름은 PmergeMe입니다.
- 프로그램은 양수 정수 시퀀스를 인자로 사용할 수 있어야 합니다.
- 프로그램은 병합-삽입 정렬 알고리즘을 사용하여 양의 정수 시퀀스를 정렬해야 합니다.



명확히 말씀드리자면, 예, 포드-존슨 알고리즘을 사용해야 합니다.

- 프로그램 실행 중에 오류가 발생하면 표준 출력에 오류 메시지가 표시되어야 합니다.



이 연습의 유효성을 검사하려면 코드에서 최소 두 개의 서로 다른 컨테이너를 사용해야 합니다. 프로그램은 최소 3000개의 서로 다른 정수를 처리할 수 있어야 합니다.



각 컨테이너마다 알고리즘을 구현하는 것이 좋으므로 일반 함수를 사용하지 않는 것이 좋습니다.

C++ - 모듈 09

STL

다음은 표준 출력에 한 줄씩 표시해야 하는 정보에 대한 몇 가지 추가 지침입니다:

- 첫 번째 줄에는 명시적인 텍스트와 정렬되지 않은 양의 정수 시퀀스를 표시해야 합니다.
- 두 번째 줄에는 명시적인 텍스트와 정렬된 양의 정수 시퀀스를 표시해야 합니다.
- 세 번째 줄에는 양의 정수 시퀀스를 정렬하는 데 사용된 첫 번째 컨테이너를 지정하여 알고리즘에서 사용한 시간을 나타내는 명시적인 텍스트를 표시해야 합니다.
- 마지막 줄에는 양의 정수 시퀀스를 정렬하는 데 사용되는 두 번째 컨테이너를 지정하여 알고리즘에서 사용한 시간을 나타내는 명시적인 텍스트를 표시해야 합니다.



분류를 수행하는 데 사용된 시간을 표시하는 형식은 자유이지만, 선택한 정밀도는 사용된 두 컨테이너의 차이를 명확하게 볼 수 있어야 합니다.

다음은 표준 사용의 예입니다:

```
$> ./PmergeMe 3 5 9 7 4
이전: 3 5 9 7 4
After:   3 4 5 7 9
다양한 범위를 처리하는 데 걸리는 시간 표준::[...]이 있는 요소 5개
::[...] : 0.00031 초 다음 범위를 처리하는 데 걸리는 시간 5개의 요
소::[...] : 0.00014 us
$> ./PmergeMe `shuf -i 1-100000 -n 3000 | tr "\n" " "` 이전:
141 79 526 321 [...]
After:   79 141 321 526 [...]
std::[...]를 사용하여 3000개의 요소 범위를 처리하는 데 걸린 시간:
62.14389 초 std::[...]를 사용하여 3000개의 요소 범위를 처리하는 데 걸린
시간: 69.27212 초
```

./PmergeMe "-1" "2" 오류



이 예제에서는 시간 표시가 일부러 이상하게 되어 있습니다. 물론 정렬 부분과 데이터 관리 부분 모두 모든 작업을 수행하는 데 사용된 시간을 표시해야 합니다





경고: 이전 연습에서 사용한 컨테이너는 여기에서 사용할 수 없습니다.

C++ - 모듈 09

STL



중복과 관련된 오류 관리는 회원님의 재량에 맡겨집니다.





## 제 7장

# 제출 및 동료 평가

과제는 평소처럼 Git 리포지토리에 제출하세요. 방어 기간 동안에는 리포지토리 내의 작업만 평가됩니다. 폴더와 파일 이름을 다시 한 번 확인하여 정확한지 확인하세요.



16D85ACC441674FBA2DF65190663F33F793984B142405F56715D5225FBAB6E3D6A4F  
167020A16827E1B16612137E59ECD492E47AB764CB10B45D979615AC9FC74D521D9  
20A778A5E

