



С++ - Модуль 00

Пространства имен, классы, функции-члены, потоки `stdio`, списки инициализации, `static`, `const` и некоторые другие базовые вещи.

Резюме:

Этот документ содержит упражнения модуля 00 из модулей С++.

Версия: 8

Содержание

I	Введение	2
II	Общие правила	3
III	Упражнение 00: Мегафон	5
IV	Упражнение 01: Моя удивительная телефонная книга	6
V	Упражнение 02: Работа вашей мечты	8

Глава I

Введение

С++ - это язык программирования общего назначения, созданный Бьярном Струstrupом как продолжение языка программирования С, или "С с классами" (источник: [Википедия](#)).

Цель этих модулей - познакомить вас с **объектно-ориентированным программированием**. Это будет отправной точкой вашего путешествия по С++. Многие языки рекомендуются для изучения ООП. Мы решили выбрать С++, поскольку он является производным от вашего старого друга С. Поскольку это сложный язык, и для того, чтобы все было просто, ваш код будет соответствовать стандарту С++98.

Мы понимаем, что современный С++ во многих аспектах сильно отличается. Поэтому, если вы хотите стать квалифицированным разработчиком С++, вам предстоит пройти дальше 42 Common Core!

Вы будете открывать новые понятия шаг за шагом. Сложность упражнений будет постепенно возрастать.

Глава II Общие правила

Компиляция

- Скомпилируйте ваш код с помощью `c++` и флагов `-Wall -Wextra -Werror`
- Ваш код будет компилироваться, если вы добавите флаг `-std=c++98`

Форматирование и соглашения об именовании

- Каталоги упражнений будут называться так: `ex00`, `ex01`, ... , `exp`
- Назовите свои файлы, классы, функции, функции-члены и атрибуты в соответствии с требованиями руководства.
- Записывайте имена классов в формате **UpperCamelCase**. Файлы, содержащие код класса, всегда будут именоваться в соответствии с именем класса. Например: `ClassName.hpp/ClassName.h`, `ClassName.cpp` или `ClassName.tpp`. Тогда, если у вас есть заголовочный файл, содержащий определение класса "BrickWall", обозначающего кирпичную стену, его имя будет `BrickWall.hpp`.
- Если не указано иное, каждое выходное сообщение должно завершаться символом новой строки и выводиться на стандартный вывод.
- *До свидания, Норминет!* В модулях C++ нет принудительного стиля кодирования. Вы можете следовать своему любимому стилю. Но имейте в виду, что код, который ваши коллеги-оценщики не могут понять, они не могут оценить. Делайте все возможное, чтобы писать чистый и читабельный код.

Разрешено/Запрещено

Вы больше не кодируете на C. Пора переходить на C++! Поэтому:

- Вам разрешено использовать почти все из стандартной библиотеки. Таким образом, вместо того чтобы придерживаться того, что вы уже знаете, было бы разумно использовать как можно больше C++-шных версий функций языка C, к которым вы привыкли.
- Однако вы не можете использовать никакие другие внешние библиотеки. Это означает, что библиотеки C++11 (и производные формы) и Boost запрещены. Также запрещены следующие функции: `*printf()`, `*alloc()` и `free()`. Если вы их используете, ваша оценка будет 0 и все.

- Обратите внимание, что если явно не указано иное, используемое пространство имен `<ns_name>` и ключевые слова-друзья запрещены. В противном случае ваша оценка будет равна -42.
- **Вам разрешено использовать STL только в модуле 08.** Это означает: никаких **контейнеров** (вектор/список/карта/и так далее) и никаких **алгоритмов** (все, что требует включения заголовка `<algorithm>`) до этого момента. В противном случае ваша оценка будет -42.

Несколько требований к дизайну

- Утечка памяти происходит и в C++. Когда вы выделяете память (с помощью функции `new` ключевое слово), вы должны избегать **утечек памяти**.
- С модуля 02 по модуль 08 ваши занятия должны быть построены в **православной канонической форме, за исключением случаев, когда прямо указано иное**.
- Любая реализация функции, помещенная в заголовочный файл (за исключением шаблонов функций), означает 0 для упражнения.
- Вы должны иметь возможность использовать каждый из ваших заголовков независимо от других. Таким образом, они должны включать все необходимые зависимости. Однако вы должны избегать проблемы двойного включения, добавляя **защитные элементы include**. В противном случае ваша оценка будет равна 0.

Читать

- Вы можете добавить несколько дополнительных файлов, если это необходимо (например, для разделения вашего кода). Поскольку эти задания не проверяются программой, не стесняйтесь делать это, если вы сдаете обязательные файлы.
- Иногда указания к упражнению выглядят кратко, но на примерах можно увидеть требования, которые не прописаны в инструкциях в явном виде.
- Перед началом работы полностью прочитайте каждый модуль! Действительно, сделайте это.
- Одином, Тором! Используйте свой мозг!!!



Вам придется реализовать множество классов. Это может показаться утомительным, если только вы не умеете писать сценарии в своем любимом текстовом редакторе.



Вам предоставляется определенная свобода в выполнении упражнений. Однако соблюдайте обязательные правила и не ленитесь. Иначе вы пропустите много полезной информации! Не стесняйтесь читать о теоретических концепциях.

Глава III

Упражнение 00: Мегафон

	Упражнение : 00
Мегафон	
Входящий каталог : <i>ex00/</i>	
Файлы для сдачи : Makefile, megaphone.cpp	
Запрещенные функции : Нет	

Чтобы убедиться, что все проснулись, напишите программу, которая ведет себя следующим образом:


```
$>./мегафон "шшшш... Кажется, студенты спят..." ШШШШШ... Я  
ДУМАЮ, ЧТО СТУДЕНТЫ СПЯТ...  
$>./megaphone Damnit " ! " "Извините, студенты, я думал, что эта штука  
отключена". ПРОКЛЯТЬЕ! ИЗВИНИТЕ, СТУДЕНТЫ, Я ДУМАЛ, ЧТО ЭТА ШТУКА  
ВЫКЛЮЧЕНА.  
$>./мегафон  
* ГРОМКИЙ И НЕВЫНОСИМЫЙ ШУМ ОБРАТНОЙ СВЯЗИ.
```



Решайте упражнения на C++.

Глава IV

Упражнение 01: Моя удивительная телефонная книга

	Упражнение : 01
	Моя удивительная телефонная книга
	Входящий каталог : <i>ex01/</i>
	Файлы для сдачи : Makefile, *.cpp, *.{h, hpp}
	Запрещенные функции : Нет

Добро пожаловать в 80-е годы и их невероятные технологии! Напишите программу, которая ведет себя как дерьмовая потрясающая программа телефонной книги.

Вы должны реализовать два класса:

- **Телефонная книга**

- Она имеет множество контактов.
- Он может хранить максимум **8 контактов**. Если пользователь попытается добавить 9-й контакт, замените самый старый контакт на новый.
- Обратите внимание, что динамическое распределение запрещено.

- **Связаться с**

- Обозначает контакт телефонной книги.

В вашем коде телефонная книга должна быть инстанцирована как экземпляр класса **PhoneBook**. То же самое касается контактов. Каждый из них должен быть создан как экземпляр класса **Contact**. Вы можете создавать классы по своему усмотрению, но помните, что все, что всегда будет использоваться внутри класса, является `private`, а все, что может быть использовано вне класса, является `public`.



Не забудьте посмотреть видеоролики в интранете.

При запуске программы телефонная книга пуста, и пользователю предлагается ввести одну из трех команд. Программа принимает только команды ADD, SEARCH и EXIT.

- **ДОБАВИТЬ:** сохранить новый контакт
 - Если пользователь вводит эту команду, ему предлагается ввести информацию о новом контакте по одному полю за раз. Как только все поля будут заполнены, добавьте контакт в телефонную книгу.
 - Поля контакта: имя, фамилия, прозвище, номер телефона и самый темный секрет. Сохраненный контакт не может иметь пустых полей.
- **ПОИСК:** отображение конкретного контакта
 - Отображение сохраненных контактов в виде списка из **4 колонок:** индекс, имя, фамилия и прозвище.
 - Ширина каждого столбца должна составлять **10 символов**. Их разделяет символ трубы ('|'). Текст должен быть выровнен по правому краю. Если текст длиннее колонки, он должен быть усечен, а последний отображаемый символ должен быть заменен точкой ('.').
 - Затем снова запросите у пользователя индекс записи, которую нужно отобразить. Если индекс находится вне диапазона или неверен, определите соответствующее поведение. В противном случае отобразите контактную информацию, по одному полю в строке.
- **ВЫХОД**
 - Программа завершает работу, и контакты теряются навсегда!
- **Любой другой входной сигнал отбрасывается.**

После правильного выполнения команды программа ожидает другой команды. Она останавливается, когда пользователь вводит команду EXIT.


Дайте соответствующее имя вашему исполняемому файлу.



<http://www.cplusplus.com/reference/string/string/> и, конечно же,
<http://www.cplusplus.com/reference/iomanip/>

Глава V

Упражнение 02: Работа вашей мечты

	Упражнение : 02
Работа вашей мечты	
Входящий каталог : <code>ex02/</code>	
Файлы для сдачи : <code>Makefile</code> , <code>Account.cpp</code> , <code>Account.hpp</code> , <code>tests.cpp</code>	
Запрещенные функции : Нет	



`Account.hpp`, `tests.cpp` и файл журнала доступны для загрузки на интранет-странице модуля.

Сегодня ваш первый день в *GlobalBanksters United*. После успешного прохождения повторных тестов при приеме на работу (благодаря нескольким приемам работы с *Microsoft Office*, которые показал вам друг) вы присоединились к команде разработчиков. Вы также знаете, что рекрутер был поражен тем, как быстро вы установили *Adobe Reader*. Это небольшое дополнение сыграло решающую роль и помогло вам победить всех ваших соперников (а именно других кандидатов): вы справились!

В любом случае, ваш менеджер только что поручил вам кое-какую работу. Ваша первая задача - воссоздать потерянный файл. Что-то пошло не так, и исходный файл был удален по ошибке. К сожалению, ваши коллеги не знают, что такое *Git*, и используют USB-носители для обмена кодом. В этот момент было бы разумно покинуть это место прямо сейчас. Однако вы решаете остаться. Вызов принят!

Ваши коллеги-разработчики дают вам кучу файлов. Компиляция `tests.cpp` показывает, что недостающим файлом является `Account.cpp`. К счастью, заголовочный файл `Account.hpp` был сохранен. Есть также файл журнала. Возможно, вы сможете использовать его для того, чтобы понять, как был реализован класс **Account**.

Вы начинаете воссоздавать файл Account.cpp. Всего за несколько минут вы написали несколько строк чистого потрясающего C++. После нескольких неудачных компиляций ваша программа проходит тесты. Ее вывод полностью совпадает с тем, что сохранен в лог-файле (**за исключением временных меток**, которые, очевидно, будут отличаться, поскольку тесты, сохраненные в лог-файле, были запущены до того, как вы были приняты на работу).

Черт, ты впечатляешь!



Порядок вызова деструкторов может отличаться в зависимости от вашего компилятора/операционной системы. Поэтому ваши деструкторы могут вызываться в обратном порядке.



Вы можете пройти этот модуль без выполнения упражнения 02.

