



Глава II

Общие правила

Составление

-

Скомпилируйте свой код с помощью `c++` и флагов `-Wall -Wextra -Werror`

-

Ваш код все равно должен компилироваться, если вы добавите флаг `-std=c++98`

Соглашения о форматировании и именовании

-

Каталоги упражнений будут называться так: `ex00`, `ex01`, ...

, эксн

-

Назовите свои файлы, классы, функции, функции-члены и атрибуты в соответствии с требованиями руководства.

-

Запишите имена классов в формате `UpperCamelCase`. Файлы, содержащие код класса, всегда будут называться в соответствии с именем класса. Например: `className.hpp`/`className.h`, `ClassName.cpp`, или `className.tpp`. Затем, 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 заканчиваться символом новой строки и выводиться на стандартный вывод.

-

Прощай, Норма! В модулях `C++` не применяется никакой стиль кодирования. Вы можете следить за своим любимым. Но имейте в виду, что код, который ваши коллеги-оценщики не могут понять, - это код, который они не могут оценить. Сделайте все возможное, чтобы написать чистый и читаемый код.

Разрешено/Запрещено

Вы больше не кодируете на Си. Время для `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 версий функций `C++`, к которым вы привыкли.

-

Однако вы не можете использовать какую-либо другую внешнюю библиотеку. Это означает, что библиотеки `C++11` (и производные формы) и `Boost` запрещены. Запрещены также следующие функции: `*printf()`, `*alloc()` и `free()`. Если вы их используете, ваша оценка будет равна 0, и все.

С++ - Модуль 05

Повторения и Исключения

•

Обратите внимание, что если явно не указано иное, использование ключевых слов `<ns_name>` и `<ns_name>`

`friend` в пространстве имен запрещено. В противном случае ваша оценка будет -42.

•

Вы можете использовать STL только в модуле 08.

Это означает:

до тех пор никаких контейнеров (вектор/список/карта/и так далее) и никаких **Алгоритмов** (все, что

требует включения заголовка `<algorithm>`). В противном случае ваша оценка будет -42.

Несколько требований к дизайну

•

Утечка памяти происходит и в С++. При выделении памяти (с помощью ключевого слова `new`) необходимо избегать **утечек памяти**.

•

С модуля 02 по модуль 08 ваши классы должны быть оформлены 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 в себя все необходимые им зависимости. Однако вы должны избежать проблемы двойного включения, добавив **охранников включения**. В противном случае ваша оценка будет равна 0.

Прочти меня

•

При необходимости вы можете добавить несколько дополнительных файлов (например, разделить код). Поскольку эти назначения не проверяются программой, не стесняйтесь делать это до тех пор, пока вы включаете обязательные файлы.

•

Иногда руководящие принципы упражнения выглядят короткими, но примеры могут показать требования, которые явно не прописаны в инструкциях.

•

Прочитайте каждый модуль полностью перед началом работы! Действительно, сделай это.

•

Клянусь Одином, клянусь Тором! Используй свой мозг!!!

Вам придется реализовать множество классов.

Это может показаться утомительным,

если только вы не умеете писать сценарий в своем любимом текстовом редакторе.

Вам предоставляется определенная свобода для выполнения упражнений.

Однако соблюдайте обязательные правила и не ленитесь.

Ты бы

пропустили много полезной информации!

Глава III

Упражнение 00: Мамочка, когда я вырасту, я хочу быть бюрократом!

Упражнение : 00

Мамочка, когда я вырасту, я хочу быть бюрократом!

Каталог Turn-in : *ex*

00

/

Файлы, которые нужно сдать :

Makefile, main.cpp, Бюрократ.{h, hpp}, Bureaucrat.cpp

Запрещенные функции :

Нет

Обратите внимание, что классы исключений не обязательно должны быть разработаны в Православная каноническая форма.

Но любой другой класс должен это сделать.

Давайте создадим искусственный кошмар из офисов, коридоров, формуляров и очередей ожидания.

Звучит забавно? Нет? Очень жаль.

Во-первых, начните с самого маленького винтика в этой огромной бюрократической машине: **бюрократа**.

Бюрократ должен иметь:

-

Постоянное имя.

-

И оценка, которая колеблется от **1** (максимально возможная оценка) до **150** (минимально возможная оценка).

Любая попытка создать экземпляр Бюрократа с использованием недопустимой оценки должна вызвать

исключение:

либо Бюрократ::GradeTooHighException, либо Бюрократ::GradeTooLowException.

5

C++ - Модуль 05

Повторения и Исключения

Вы предоставите геттеры для обоих этих атрибутов: getName() и getGrade().

Добавьте также две функции-члена для увеличения или уменьшения класса бюрократа. Если класс находится вне диапазона, оба они будут выдавать те же исключения, что и конструктор. Помни.

Поскольку 1-й класс-самый высокий, а 150-самый низкий, увеличение 3-го класса должно дать бюрократу 2-й класс.

Выброшенные исключения должны быть уловимы с помощью блоков try и catch:

попробуй

```
{  
/* делай что-нибудь с бюрократами */  
}
```

улов

```
(std::exception & e)
{
    /* обработать исключение */
}
```

Вы реализуете перегрузку оператора insertion («), чтобы напечатать что-то вроде (без угловых скобок):

<имя>, класс бюрократа <имя><класс>.<класс>

Как обычно, сдайте несколько тестов, чтобы доказать, что все работает так, как ожидалось.

6

Глава IV

Упражнение 01: Стройтесь, личинки!

Упражнение : 01

Стройтесь, личинки!

Каталог Turn-in : *ex*

01

/

Файлы, которые нужно сдать :

Файлы из предыдущего упражнения + Форма.{h, hpp}, Form.cpp

Запрещенные функции :

Нет

Теперь, когда у вас есть бюрократы, давайте дадим им чем-нибудь заняться. Что может быть лучше занятия

может ли быть что-то большее, чем заполнение стопки бланков?

Затем давайте создадим класс формы. Он имеет:

-

Постоянное имя.

-

Логическое значение, указывающее, подписано ли оно (при построении-нет).

-

Для его подписания требовалась постоянная оценка.

-

И для его выполнения требовалась постоянная оценка.

Все эти атрибуты являются **частными**, а не защищенными.

Оценки **Формы** следуют тем же правилам, что и для Бюрократа. Таким образом, если класс формы выходит за рамки, будут созданы следующие исключения:

Form::GradeTooHighException и Form::GradeTooLowException.

Как и раньше, напишите геттеры для всех атрибутов и перегрузку вставки («).

оператор, который печатает всю информацию формы.

7

C++ - Модуль 05

Повторения и Исключения

Добавьте также функцию-член beSigned() в Форму, которая принимает Бюрократа в качестве параметра. Он меняет статус формы на подписанную, если ранг бюрократа достаточно высок (выше или равен требуемому). Помните, что оценка 1 выше, чем оценка 2.

Если оценка слишком низкая, бросьте исключение Form::GradeTooLowException.

Наконец, добавьте функцию-член signForm() в Бюрократ. Если бланк будет подписан, он напечатает что-то вроде:

<бюрократ> подпись <бюрократ><форма><форма>

В противном случае он напечатает что-то вроде:

<бюрократ> не смог подписать <бюрократ><форму>, потому что <форма><причина>.<причина>

Внедрите и включите некоторые тесты, чтобы убедиться, что все работает так, как ожидалось.

8

Глава V

Упражнение 02: Нет, вам нужна форма 28В,

а не 28С...

Упражнение : 02

Нет, вам нужна форма 28В, а не 28С...

Каталог Turn-in : *ex*

02

/

Файлы, которые нужно сдать :

Файлы из предыдущих упражнений +

ShrubberyCreationForm.[{h, hpp},cpp], RobotomyRequestForm.[{h, hpp},cpp],
PresidentialPardonForm.[{h, hpp},cpp]

Запрещенные функции :

Нет

Поскольку теперь у вас есть базовые формы, пришло время сделать еще несколько, которые действительно делают некоторые ...
вещь.

Во всех случаях форма базового класса должна быть абстрактным классом. Имейте в виду, что форма

атрибуты должны оставаться закрытыми и находиться в базовом классе.

Добавьте следующие конкретные классы:

•

ShrubberyCreationForm: Required grades: sign 145, exex 137

Создайте файл <target>_shrubbery в рабочем каталоге и запишите в него ASCII-деревья<target>

.

•

RobotomyRequestForm: Требуемые оценки: знак 72, exex 45

Издает некоторые сверлящие звуки. Затем сообщает, что <target> был успешно роботизирован<target>в 50% случаев. В противном случае сообщается, что роботомия не удалась.

•

PresidentialPardonForm: Требуемые оценки: знак 25, exex 5

Сообщает, что <цель> была помилована Зафодом Библброксом.<цель>

Все они принимают только один параметр в своем конструкторе: цель формы. Для например, "дом", если вы хотите посадить кустарник у себя дома.

9

C++ - Модуль 05

Повторения и Исключения

Теперь добавьте функцию-член const execute(Бюрократ const & исполнитель)

в базовую форму и реализуйте функцию для выполнения действия формы конкретных классов. Вы должны убедиться, что форма подписана и что оценка бюрократа, пытающегося выполнить форму, достаточно высока. В противном случае сделайте соответствующее исключение.

Хотите ли вы проверить требования в каждом конкретном классе или в базовом классе (затем вызвать другую функцию для выполнения формы), зависит только от вас. Однако один способ

красивее другого.

Наконец, добавьте функцию-член `executeForm(Form const & form)` в Бюро -

кратер. Он должен попытаться выполнить форму. Если это успешно, выведите что-то вроде:

<бюрократ> казнен <бюрократ><форма><форма>

Если нет, выведите явное сообщение об ошибке.

Внедрите и включите некоторые тесты, чтобы убедиться, что все работает так, как ожидалось.

10

Глава VI

Упражнение 03: По крайней мере, это лучше

, чем приготовление кофе

Упражнение : 03

По крайней мере, это лучше, чем приготовление кофе

Каталог Turn-in : *ex*

03

/

Файлы, которые нужно сдать :

Файлы из предыдущих упражнений + Стажер.{h, hpp}, Intern.cpp

Запрещенные функции :

Нет

Поскольку заполнение формуляров достаточно раздражает, было бы жестоко просить наших бюрократов

делать это весь день напролет. К счастью, стажеры существуют. В этом упражнении вы должны реализовать

класс Intern. У стажера нет ни имени, ни класса, ни уникальных характеристик. Единственное, о чем заботятся бюрократы, - это то, что они делают свою работу.

Однако у стажера есть одна важная способность: функция `makeForm()`. Для этого нужны две струны. Первый-это имя формы, а второй-ее цель

. Он возвращает указатель на **объект формы** (имя которого передается в качестве параметра), цель которого будет инициализирована вторым параметром.

Он напечатает что-то вроде:

Стажер создает <форма><форма>

Если имя формы, переданное в качестве параметра, не существует, выведите явное сообщение об ошибке.

11

С++ - Модуль 05

Повторения и Исключения

Вы должны избегать нечитаемых и уродливых решений, таких как использование леса if/elseif/else.

Такие вещи не будут приняты во время процесса оценки. Вы больше не в Piscine (бассейне). Как обычно, вы должны проверить, что все работает так, как ожидалось.

Например, приведенный ниже код создает **RobotomyRequestForm**, ориентированную на "Bender":

```
{
  Стажер
  someRandomIntern;
  Форма*
  ррф;
  rrf = someRandomIntern.makeForm(
    "запрос на роботехнику"
  ,
  "Bender"
);
}
```