# STL Containers

Цель: Закрепить наваки работы STL Containers

Описание

Задания состоит из трех модулей:

• l4 (прилагается)

• Sale

• Product

Заключите весь исходный код в пространство имен sict и включите необходимые средства защиты в каждый файл заголовка. Вывод вашего исполняемого файла, работающего под управлением Visual Studio, со следующим аргументом командной строки должен выглядеть так:

**Command Line : C:\Users\...\Debug\lab4.exe Sls.dat**

**Product No Cost**

**10012 34.56**

**10023 45.67**

**10234 12.32**

**10056 67.54**

**10029 54.12**

**10034 96.30**

**Total 310.51**

Исходные данные для тестирования вашего решения хранятся в подготовленном пользователем файле. Имя файла указывается в командной строке, как показано красным выше. Файл прилагается.

Содержимое этого файла

**10012 34.56**

**10023 45.67**

**10234 12.32**

**10056 67.54**

**10029 54.12**

**10034 96.30**

Каждая запись в файле состоит из номера продукта и его цены.

Модуль Product

Разработайте иерархию наследования iProduct для хранения данных о товарах для продажи. Ваш интерфейс иерархии использует переменную (**int FW**) ширина полякоторая определена глобально и имеет следующие требования функции в общественных членах:

• double price () const - запрос, возвращающий цену товара.

• void display (std :: ostream & os) const - запрос, отображающий информацию о продукте. Используйте ширину поля, определенную в модуле l4.

Ваш интерфейс также включает в себя прототипы для удовлетворения следующих глобальных требований к функциям:

• operator << (std :: ostream & os, const iProduct & p) - оператор, который вставляет информацию о продукте в поток os.

• iProduct \* readRecord (std :: ifstream & file); - функция, которая считывает одну запись из объектного файла, выделяет память для продукта в иерархии, сохраняет считанную информацию в объекте продукта и возвращает его адрес.

Конкретный класс с именем Product реализует этот интерфейс и включает

• конструктор, который получает и сохраняет номер продукта и его цену.

Модуль Sale

Разработайте класс Sale для управления последовательностью объектов iProduct. Дизайн вашего класса включает следующие общедоступные функции-члены:

• Конструктор с одним аргументом, который получает адрес неизменяемой строки в стиле C с завершающим нулем, которая содержит имя файла, содержащего записи типа iProduct. Эта функция открывает файл, считывает одну запись из файла, перемещает эту запись в контейнер STL и продолжает чтение записей до тех пор, пока не останется ни одной записи для чтения. Если этой функции не удается открыть файл, она генерирует исключение с соответствующим сообщением.

• void display (std :: ostream & os) const - запрос, отображающий продукты на продажу, и общую цену в формате, показанном выше. При написании этой функции используйте диапазон на основе и ширину поля **FW**.

**Вторая часть этой лабораторной обновляет следующие модули, включая дополнительные расчети ввиде налогов:**

• Product

• Sale

Вывод вашего исполняемого файла, работающего под управлением Visual Studio, со следующим аргументом командной строки должен выглядеть так:

**Command Line : C:\Users\...\Debug\lab4.exe Sls.dat**

**Product No Cost Taxable**

**10012 34.56**

**10023 45.67 HST**

**10234 12.32 PST**

**10056 67.54**

**10029 54.12 HST**

**10034 96.30**

**Total 324.47**

Исходные данные для тестирования вашего решения хранятся в подготовленном пользователем файле. Имя файла указывается в командной строке, как показано красным выше. Файл прилагается.

Содержимое этого файла

**10012 34.56**

**10023 45.67 H**

**10234 12.32 P**

**10056 67.54**

**10029 54.12 H**

**10034 96.30**

Каждая запись в файле состоит из номера продукта, его цены и, возможно, налогооблагаемого статуса.

Модуль Product

Разработайте класс TaxableProduct из класса Product, чтобы добавить в иерархию функции налогообложения. Ваш класс включает следующее

• перечень видов налогов и переменная класса с соответствующими ставками

* + HT 0.13
  + PT 0.08

Ваш класс также включает следующие обновления:

• Конструктор с тремя аргументами, который получает и сохраняет номер продукта, его цену и налоговый статус.

• double price () const - функцию, которая возвращает цену с учетом налога на продукт.

• void display (std :: ostream & os) const - функцию, которая отображает информацию о продукте и добавляет налоговый статус, как показано в примеревыше.

• iProduct \* readRecord (std :: ifstream & file) - эта функция считывает одну запись из объектного файла, выделяет память для продукта в иерархии, сохраняет прочитанную информацию в объекте продукта и возвращает его адрес. Информация включает номер продукта и цену, а также, необязательно, налоговый статус, как показано в примере ввода выше. Если запись не содержит символа налогового статуса, эта функция выделяет память для объекта Product. Если запись имеет символ налогового статуса, эта функция выделяет память для объекта TaxableProduct.

Модуль Sale

Обновите модуль «Sale», чтобы включить «Taxable» в заголовок для отображения данных о продукте, как показано в примере выходных данных выше.