**COMP7406 Software development for quantitative finance**

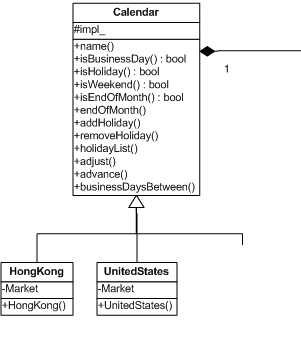
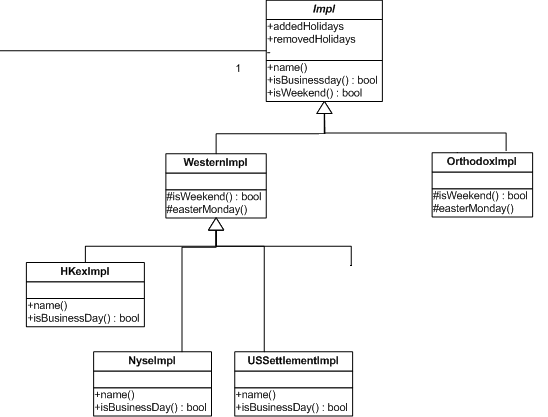
*Assignment 3*

**Introduction**

To implement the BusinessCalendar as described in the assignment guidelines, I designed a ***BusinessCalendar*** class that inherited from the ***Quantlib::Calendar*** generic class. My BusinessCalendar class allows the upload of holidays from a .csv file, based on a particular *Market*. An inner class *BusImpl*, which inherits from Calendar’s inner-class *Impl*, facilitates the implementation of the calendar. *BusImpl* includes a constructor to differentiate between different markets, a naming convention, and an *isBusinessDay()* function. I borrowed much of the implementation layout from the ***QuantLib::BespokeCalendar*** class, though I didn’t use that class as a parent. More details are provided in the class diagram and the file descriptions below.

**Class Diagram**

The below diagram shows how my new files relate to the Quantlib classes. The main class layout of Quantlib was copied from the lecture notes, and my new classes are shaded in green.



|  |
| --- |
| **BusinessCalendar** |
| +BusinessCalendar(string filename, string market) |
| +busImpl\_ |

|  |
| --- |
| **BusImpl** |
| +BusImpl(string name) |
| +name() |
| +isBusinessDay(): bool |

***BusinessCalendar* class**

My BusinessCalendar class is split into BusinessCalendar.hpp (header file) and BusinessCalendar.cpp (class implementation file). These files are found in QuantLib-1.3\ql\time and I have included the text of both files in businessCalendar\_hpp.txt and businessCalendar\_cpp.txt in my zipped submission folder.

Class Declaration

class BusinessCalendar : public Calendar

***BusinessCalendar*** inherits from Quantlib’s ***Calendar*** class, and includes all its member functions.

Constructor

BusinessCalendar(std::string holidayfile, std::string market);

The constructor takes two std::string arguments, *holidayfile* and *market*:

*holidayfile* - Name of the .csv file that has a list of holiday/market dates in the format: “Market”,“Date”. Example: AMS,19950414

*market*  - Name of the market of which to take the holiday dates from the .csv file

A default constructor is not provided, so the filename and market have to be used upon construction.

Constructor Implementation

BusinessCalendar::BusinessCalendar(std::string holidayfile, std::string market)

Calling the constructor will first create a new boost::shared\_ptr of type *BusImpl* (with a *name* equal to *market*)and sets it equal to Calendar’s *impl\_* pointer, which will be described in the **BusImpl** section.

The constructor also takes care of the file input using the std::fstream library:

1. Each line of the file is read with *getline* and stored in the variable <std::string> *line.*
2. <std::string> variables *mkt* and *date* are parsed from *line*. The *date* is then parsed into day, month, year, and re-combined into a <Quantlib::Date> type variable named *holidayDate*.
3. *holidayDate* is then inserted into **addedHolidays**, a std::set of type *Date* that is a member function of the inherited BusImpl class:

busImpl\_->addedHolidays.insert(holidayDate);

***BusImpl* class**

Class Declaration

class BusImpl : public Calendar::WesternImpl

***BusImpl*** is an inner class (and a private member) within BuisinessCalendar, and inherits from Calendar’s ***WesternImpl*** class (which is itself inherited from the ***Impl*** class).

Constructor

BusImpl(const std::string& name = "");

The constructor takes a std::string argument called *name*, which will be equal to the *market* variable in the BusinessCalendar constructor.

Public Members

std::string name() const;

bool isBusinessDay(const Date&) const;

name(): Returns the *name* of the ***BusImpl***, which is determined by the *market* variable in the BusinessCalendar constructor.

isBusinessDay(Date): Takes a QuantLib::Date variable and tests whether the date is a business day. Returns **false** if the Date is a weekend day, which is checked using the inherited *isWeekend(Date)* function from the inherited ***Impl*** class, or if the date is found in the *addedHolidays* set. Returns **true** otherwise.

**Test Cases**

To test my classes, I created three instances of BusinessCalendar, calling the Tested "HKEx", "NY", and "AMS" Calendars:

QuantLib::Calendar hkCal = QuantLib::BusinessCalendar("Holiday.csv", "HKEx");

QuantLib::Calendar nyCal = QuantLib::BusinessCalendar("Holiday.csv", "NY");

QuantLib::Calendar amsCal = QuantLib::BusinessCalendar("Holiday.csv", "AMS");

I created several Date variables and tested the various Calendar functions, including *isHoliday()*, *isBusinessDay(),businessDaysBetween(),endOfMonth(), advance(),* and *adjust().* I made sure to test dates that were holidays in one calendar but not the others.

The full code for my test cases is found in ***testCases.txt***, included in the zip file for submission.