Шарко Полина, 931921

1) Постановка задачи

На основе паттерна Наблюдателя, написать программу для слежения за состоянием выбранного файла.

Ограничимся двумя характеристиками:

1. Существует файл или нет;

2. Каков размер файла.

Программа будет выводить на консоль уведомление о произошедших изменениях в файле.

Существует несколько ситуаций для наблюдаемого файла

1. Файл существует, файл не пустой - на экран выводится факт существования файла и его размер.

2. Файл существует, файл был изменен - на экран выводится факт существования файла, сообщение о том, что файл был изменен и его размер.

3. Файл не существует - на экран выводится информация о том, что файл не существует

2) Предлагаемое решение

Наблюдатель — это поведенческий паттерн проектирования, который создает механизм подписки, позволяющий одним объектам следить и реагировать на события, происходящие в других объектах. Применяется в случае, если изменение состояния одних объектов влияет на поведение других, но мы не знаем, каких.

Для реализации паттерна нам нужны два класса - класс Наблюдатель(у нас **IObserver**) и класс, ответственный за взаимодействие с наблюдателями(у нас **Subject**).

Разберем назначение предложенных в реализации классов.

**IObserver** – базовый класс наблюдателя.

Методы класса:

* virtual void *Update*(bool fileExist, int size) = 0 - метод,определяющий состояние файла (существует/ не существует, изменен/не изменен, информация о размере файла) и оповещающий о нем.

**FileObserver –** наследник **IObserver** для работы с объектами типа QFile.

Методы класса:

* конструктор по умолчанию
* void ***Update***(bool fileExist, int size); - метод,определяющий состояние файла (существует/ не существует, изменен/не изменен, информация о размере файла) и оповещающий о нем. Метод **Update** применяет при обработке информации, полученной от объектов класса **Subject**.

**Subject -** класс, ответственный за взаимодействие с наблюдателями. Регулирует обновление наблюдателей, предоставляя им методы для контроля своего нахождения в списке(позволяет добавлять/удалять себя из списка).

Методы класса:

* void **Attach**(IObserver \*product) - метод для присоединения наблюдателя
* void **Detach**(IObserver \*product) - метод для отключения наблюдателя
* void **Notify**(bool fileExist, int size) - метод, уведомляющий об обновлении данных

**FileSubject -** наследник **Subject** для работы с объектами типа QFile**,** отвечает за обновление данных в наблюдателях.

* конструктор по умолчанию
* void **updateObservers**() - метод, уведомляющий об обновлении данных при работе с файлами

3) Коды программ

Файлы можно найти по ссылке

**Observer.h** - содержит классы-наблюдатели: **IObserver** и **FileObserver.**

**Observer.cpp** - файл с определением метода класса **FileObserver.**

**subj.h** - содержит классы, ответственные за взаимодействие с наблюдателями: **Subject** и **FileSubject.**

**subj.cpp** - файл с определением метода класса **Subject.**

**main.cpp** - файл, используемый для тестирования

Файлы для тестирования:

**file.txt** - наблюдаемый файл

4) Инструкция пользователя

Диалоговое окно не предлагается. Изменение файла непосредственно в коде не производится, предполагается изменение извне. Результаты тестирования можно видеть в консоли.

5) Тестирование

Примеры для тестирования реализованы в файле main.cpp.

Протестируем метод void **updateObservers**().

Из условия, ситуации может быть 3:

*1. Файл существует, файл не пустой - на экран выводится факт существования файла и его размер.*

Для тестирования используем непустой наблюдаемый файл. Тогда в консоли можно будет увидеть сообщение:

File exists and is not empty

Size: 7

*2. Файл существует, файл был изменен - на экран выводится факт существования файла, сообщение о том, что файл был изменен и его размер.*

Для тестирования можно изменить наблюдаемый файл. Тогда в консоли можно будет увидеть сообщение:

File test.txt exists, has been modified

Size: 6

*3. Файл не существует - на экран выводится информация о том, что файл не существует.*

Этот случай можно проверить, удалив файл из папки сборки. Тогда в консоли можно будет увидеть сообщение:

File file.txt does not exist