Романов Вячеслав гр. 931921

1) Постановка задачи

На основе паттерна Наблюдателя, написать программу для слежения за состоянием

выбранного файла.

Две характеристики:

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.

Паттерн Наблюдатель – это поведенческий паттерн проектирования, который создаёт механизм подписки, позволяющий одним объектам следить и реагировать на события, которые происходят в других объектах.  
  
Для реализации паттерна наблюдателя я реализовал следующие классы:  
  
Класс, обслуживающий наблюдателей ASubject, должен предоставить интерфейс добавления и удаления наблюдателя, иметь возможность оповещать всех подписчиков.

Класс наблюдателя, назовем его I\_Observer, будет абстрактным классом в целях поддержания полиформизма в интерфейсе класса ASubject. I\_Observer должен иметь виртуальный метод, вызываемый при получении оповещения со стороны наблюдаемого им объекта.

Для взаимодействия с файлом разработан класс Concrete\_File, который наследуется от класса ASubject.

Для наблюдения над состоянием файла будет отвечать класс File\_Observer, наследованный от абстрактного класса I\_Observer. Все изменения над файлом будут выводиться в консоль.  
  
3.Код программы:  
<https://github.com/Dontemon/Pattern_Observer/tree/develop>  
  
4. Инструкция пользователя

Для работы с программой необходимо создать в папке сборки программы создать файл с именем test.txt .

Также в программе написан цикл, во время работы которого наблюдатель отслеживает изменения файла и выводит их в консоль

Состояния файла:  
1) Файл не существует

2) Файл существует, но он пустой

3) Файл существует и он не пустой (также выведется размер файла)

4) Файл изменил свой размер(также выведется новый размер файла)  
После закрепления за файлом наблюдателя, во время работы цикла можно менять файл, удалять и восстанавливать его и отслеживать эти изменения в консоли.(длительность цикла также можно менять путём увеличения итераций, а также увеличения задержки, то есть можно поставить, чтобы i шёл до тысячи, и задержка была секунда std::this\_thread::sleep\_for( std::chrono::milliseconds(1000)) )  
//Открываем файл

Concrete\_File test\_File("test.txt");  
Concrete\_File test\_File("Название\_файла.txt");

//Создаём наблюдателя для файла  
File\_Observer fist\_Observer("test.txt", test\_File.file\_size());

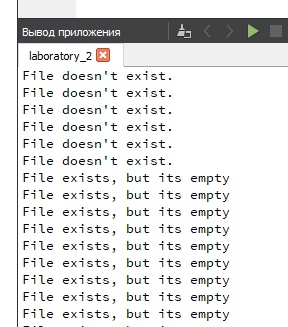
//Закрепляем наблюдателя за файлом

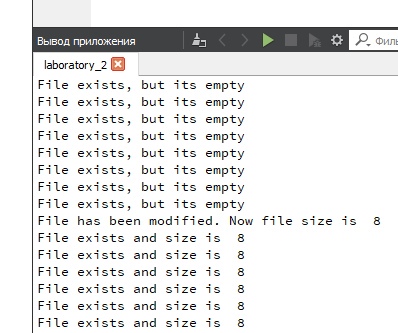
test\_File.Attach(&fist\_Observer);

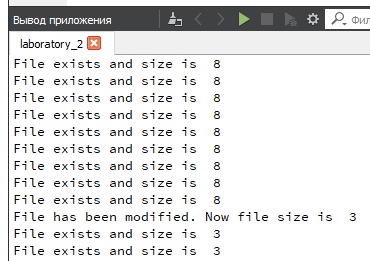
//Удаление наблюдателя

test\_File.Detach(&fist\_Observer);

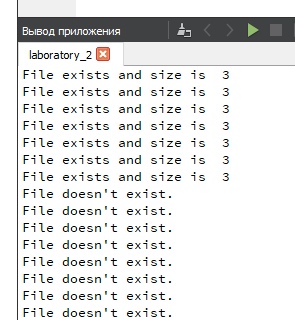
5. Тестирование:

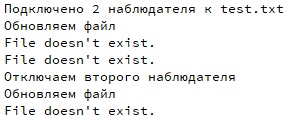
Если при запуске программы файла не существует, но во время работы программы файл создали, то в консоли будет выведено   


Далее если мы напишем в файле что-то и сохраним изменения, то в консоли выведется  


Если мы изменим количество символов в файле и сохраним изменения, то программа выведет  


Если удалить файл, в консоль будет выведено: File doesn't exist.



Также можно установить 2 наблюдателя за одним файлом(ниже представлен код, для проверки)  
Concrete\_File test\_File("test.txt");//создаём файл  
File\_Observer fist\_Observer("test.txt", test\_File.file\_size()), second\_Observer("test.txt", test\_File.file\_size());//создаём наблюдателя  
test\_File.Attach(&fist\_Observer);//привязываем 2-х наблюдателей к файлу  
test\_File.Attach(&second\_Observer);  
qDebug() « "Подключено 2 наблюдателя к test.txt";  
qDebug() « "Обновляем файл";  
test\_File.ChangeFile(); //обновляем файл  
test\_File.Detach(&second\_Observer); //отключаем наблюдателя  
qDebug() « "Отключаем второго наблюдателя";  
qDebug() « "Обновляем файл";  
test\_File.ChangeFile(); //обновляем файл   
  
В консоль будет выведено   
  
То есть информация о файле дублируется

