

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

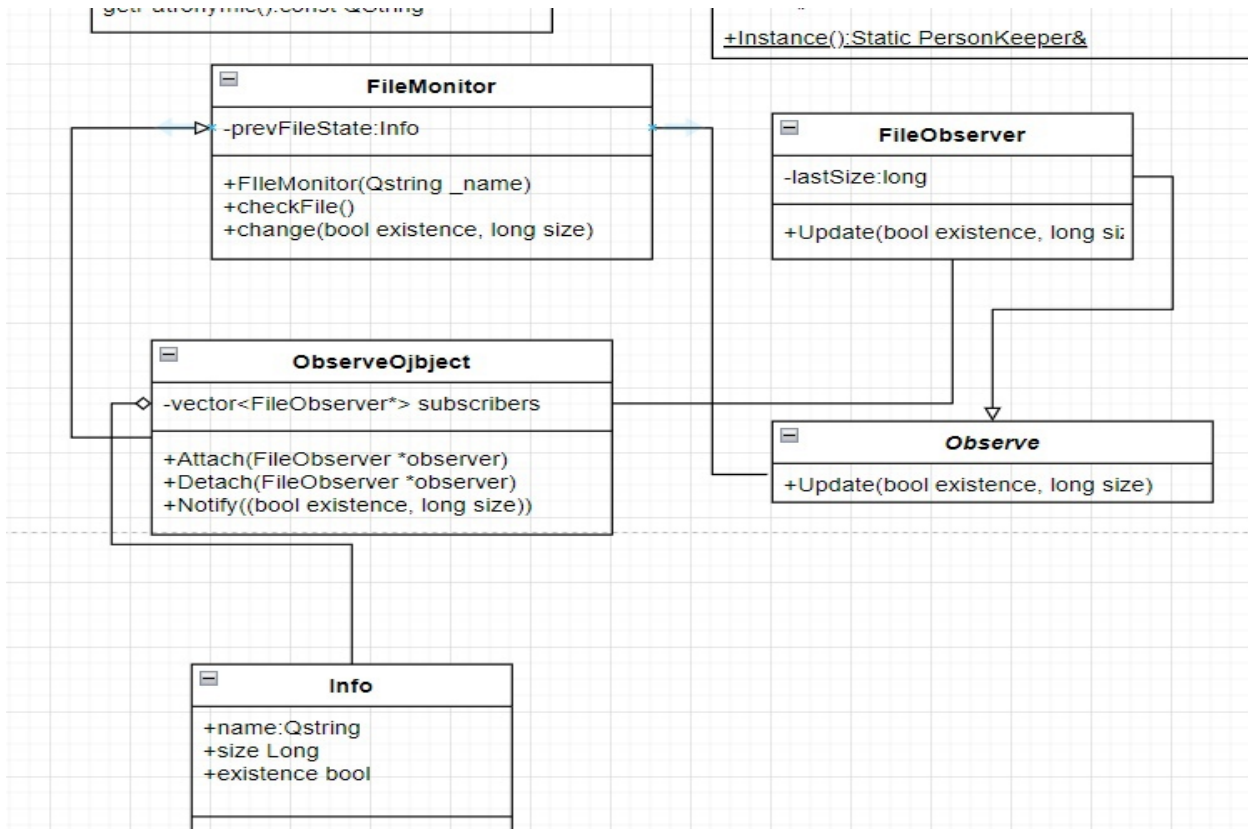
1. Файл существует, файл не пустой - на экран выводится факт существования файла и его размер.
2. Файл существует, файл был изменен - на экран выводится факт существования файла, сообщение о том что файл был изменен и его размер.
3. Файл не существует - на экран выводится информация о том что файл не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

В главной программе создаем объект для отслеживания состояния конкретного файла, затем определяется объект наблюдатель, после чего связываем Наблюдателя с Источником.

Можно использовать бесконечный цикл, в котором будем обновлять состояние объекта Источника каждые 100 миллисекунд (например так `std::this_thread::sleep_for(std::chrono::milliseconds(100));`).

2. Предлагаемое решение.

1. Паттерн "Наблюдатель" (Observer) представляет поведенческий шаблон проектирования, который использует отношение "один ко многим". В этом отношении есть один наблюдаемый объект и множество наблюдателей. И при изменении наблюдаемого объекта автоматически происходит оповещение всех наблюдателей. (Определение с metanit)



Observe - интерфейс, с помощью которого наблюдатель получает оповещение;

FileObserver- конкретный класс, который реализует интерфейс Observer.

FileMonitor - конкретный класс, который реализует интерфейс Observable;

ObserveObject -интерфейс, определяющий методы для добавления, удаления и оповещения наблюдателей;

Info - класс содержащий данные файла

Разработка началась самого легкого создание класса Info.В нем обрабатывает данные файла.Который содержит название, размер и пустоту файла.Данный класс был реализован через <sys\stat.h> путем получение данных файла

Далее был реализован Observe с его реализацией.Интерфейс observe состоит из одной виртуальной функцией Update(bool extinct,int size) *FileObserver- конкретный класс наблюдателя состоит из*

lastSize-последний размер файла и функции Update(bool extinct,int size) туда входит размер и существование файла.Далее идет проверки на существование и изменение файла ,если все выполнено идет вывод размера файла

Далее идет реализация интерфейса ObserveObject и его реализации вначале реализуется 3 функций Attach,Detach,Notify и хранилища подписчиков subscribers реализованных через вектор

Attach - добавление наблюдаемого путем добавление subscribers

Detach- удаление элемента из subscribers

Notify проходка по subscribers и обновление их данных

Далее рассмотрим сам наблюдаемый объект FileMonitor

Change- запускает Notify

CheckFile - обрабатывает состояние файла и сохраняет обновление,если оно случилось

3 Файлы

info.cpp/h

observe.cpp/h

observerobject.cpp/h

4 Руководства пользователя

В начале создается переменная с полным путем к файлу типа FileMonitor(объект наблюдения)

После с помощью метода Attach и Detach можно добавить или убрать объект

Метод CheckFile() смотреть изменения файла от первоначального запуска программы


Чтоб смотреть изменения в реальном был использован цикл задержкой

Далее после запуска программы пользователю высвечивается консоль в которой показано текущие состояние файла.Изменение размера файла фиксируется на консоли также фиксируется существование.Пользователь может менять данных ,что будет высвечиваться на консоли

5 Тестирование

Info

1. Файл пустой но существует



The image shows a Qt Creator window with a file named 'filename.txt' open. The code in the editor is as follows:

```
1  
  
int main()  
{  
    Info tes("C:/Users/andru/Documents/Lab2/filename.txt");  
    cout<<tes.existence<<endl;  
    cout<<tes.name.toStdString()<<endl;  
    cout<<tes.size<<endl;  
}
```

Below the code editor, a terminal window shows the output of the program:

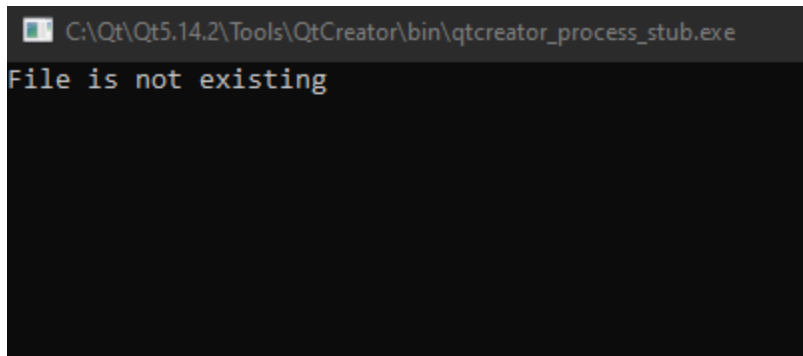
```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe  
1  
C:/Users/andru/Documents/Lab2/filename.txt  
0
```

1 в первой строчке значит что файл существует

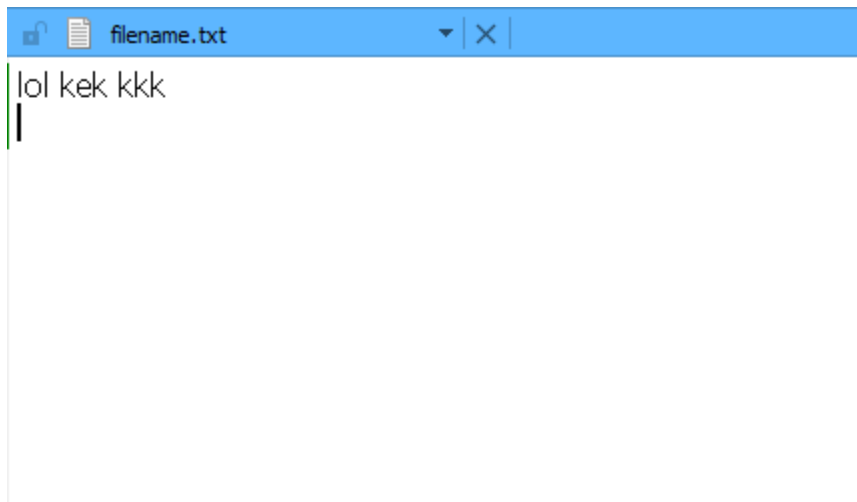
Вторая строка имя

Третья размер

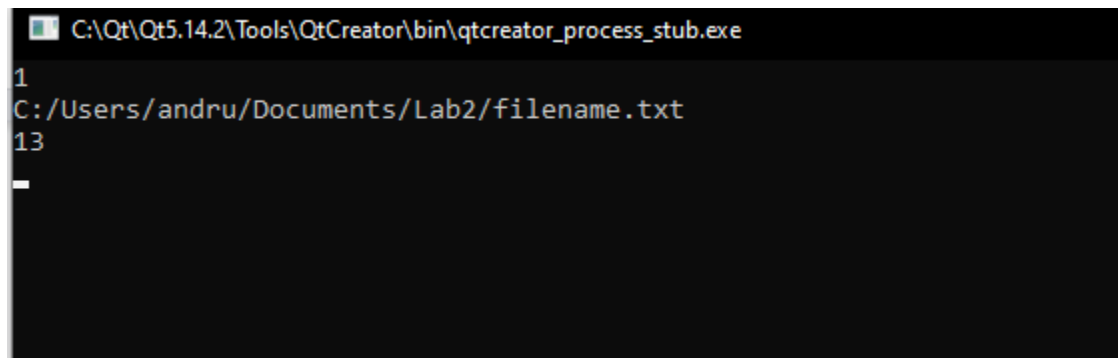
2. Файл существует и заполнен



```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe
File is not existing
```



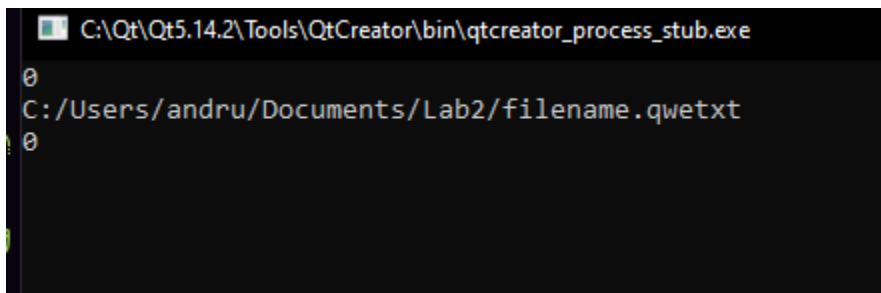
```
filename.txt
lol kek kkk
```



```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe
1
C:/Users/andru/Documents/Lab2/filename.txt
13
-
```

3. Файла не существует

```
int main()
{
    Info tes("C:/Users/andru/Documents/Lab2/filename.qwebxt");
    cout<<tes.existence<<endl;
    cout<<tes.name.toStdString()<<endl;
    cout<<tes.size<<endl;
}
```

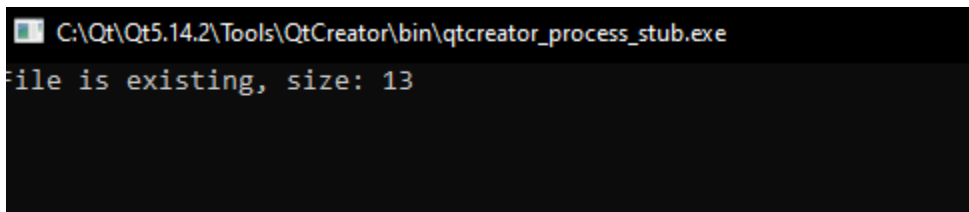


```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe
0
C:/Users/andru/Documents/Lab2/filename.qwebxt
0
```

Observe

1)Файл существует

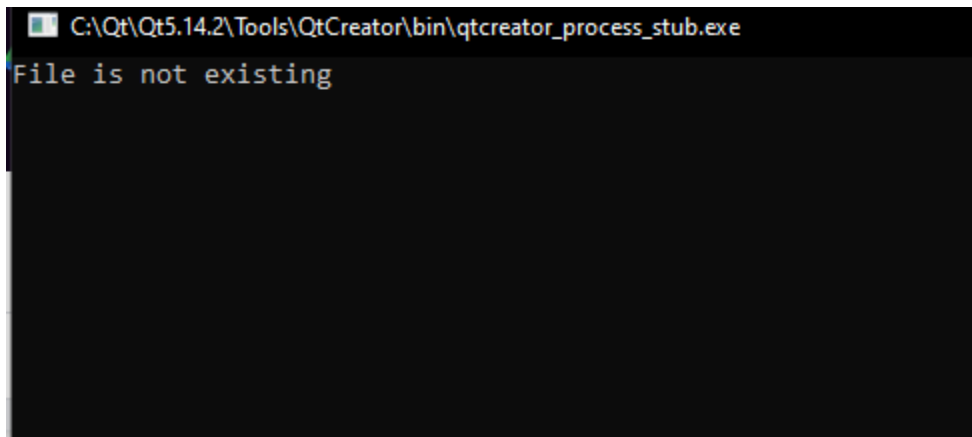
```
int main()
{
    Info tes("C:/Users/andru/Documents/Lab2/filename.txt");
    FileObserver observer;
    observer.Update(tes.existence,tes.size);
}
```



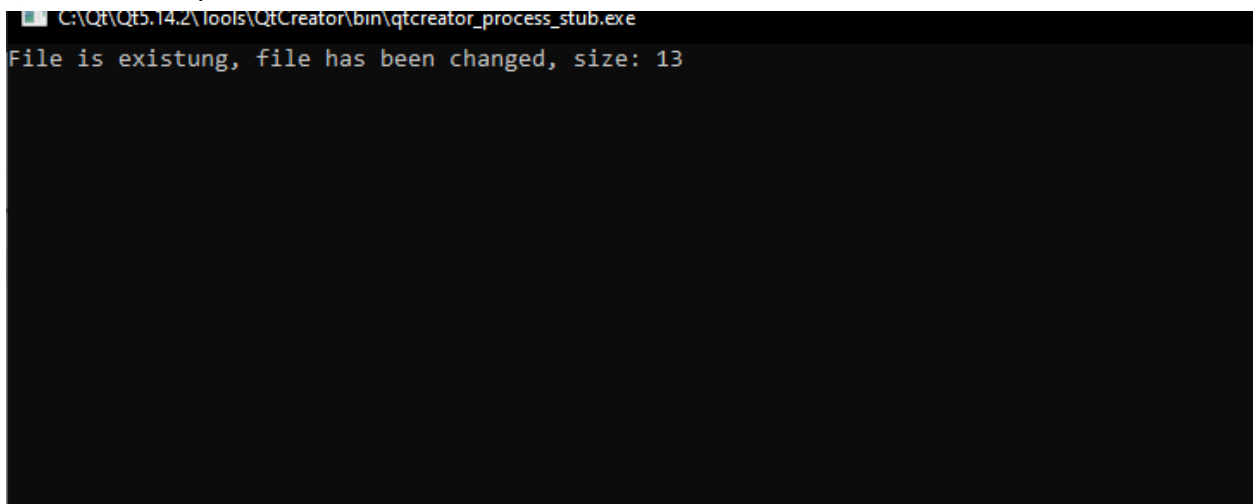
```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe
file is existing, size: 13
```

2)Файла нет

```
int main()
{
    Info tes("C:/Users/andru/Documents/Lab2/filename.txt");
    FileObserver observer;
    observer.Update(tes.existence, tes.size);
}
```

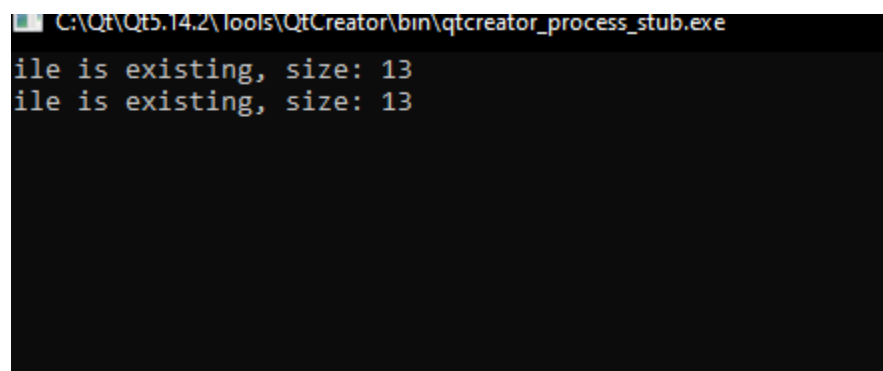
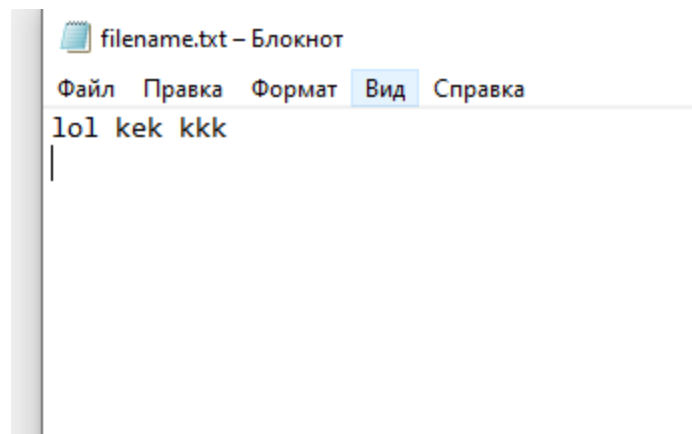


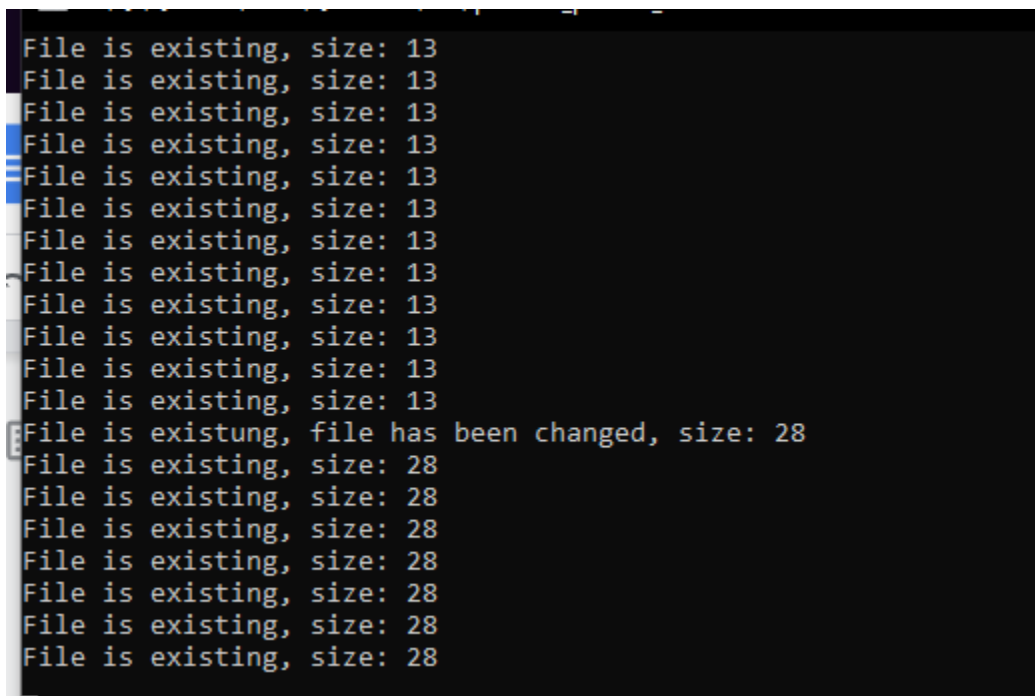
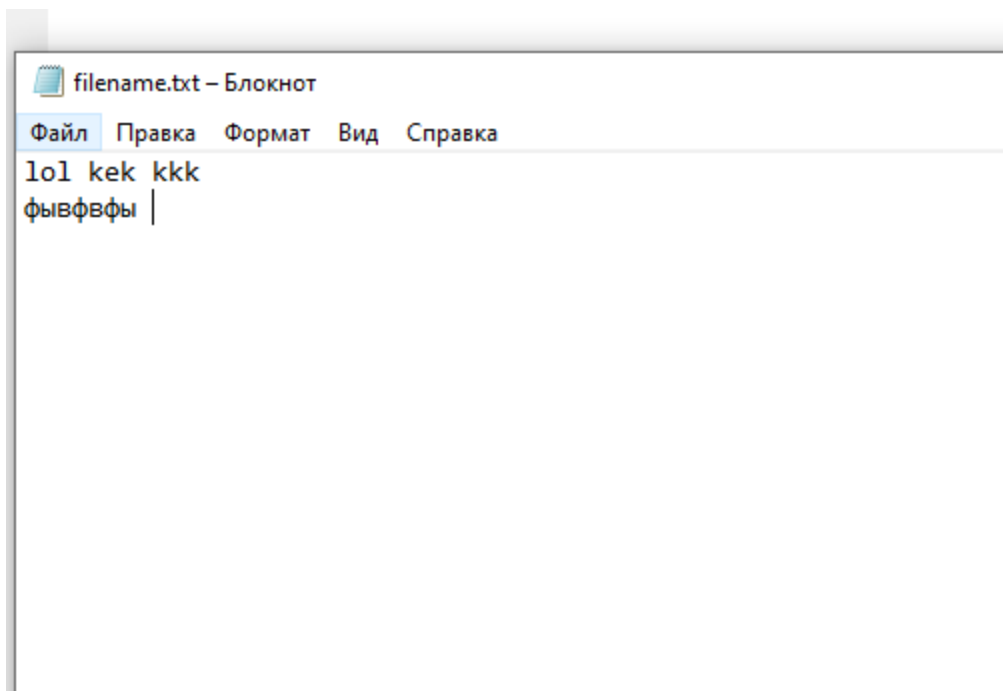
3) Поменял дефолтное значение для теста -1 на 3



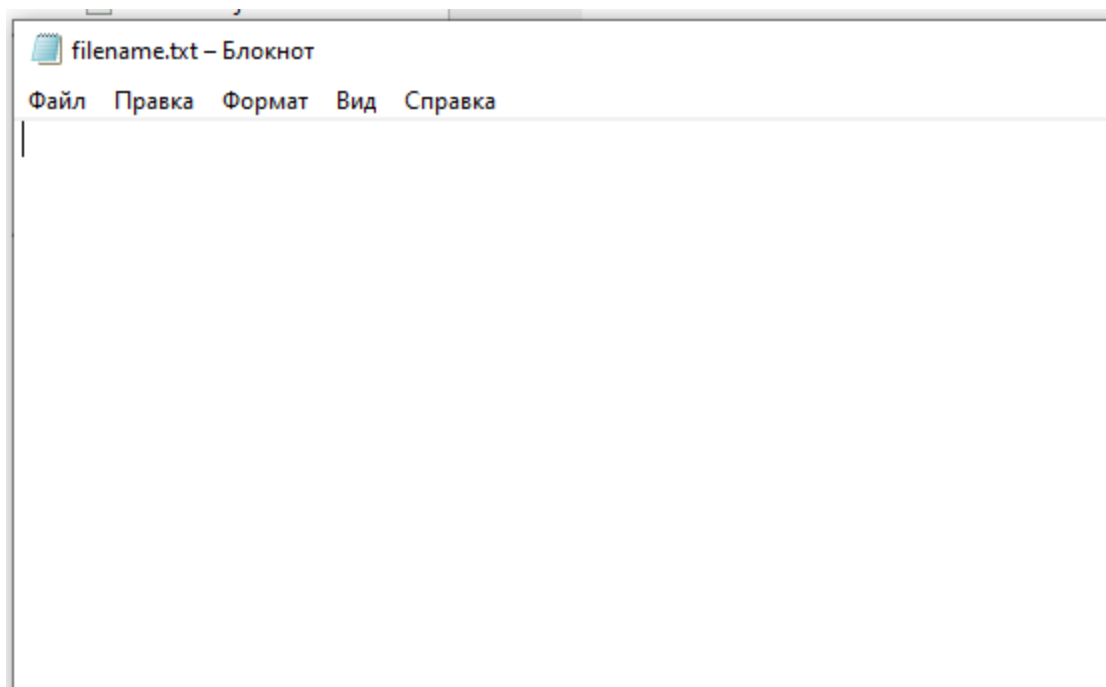
Тест готовой программы

1) Файл существует и заполнен

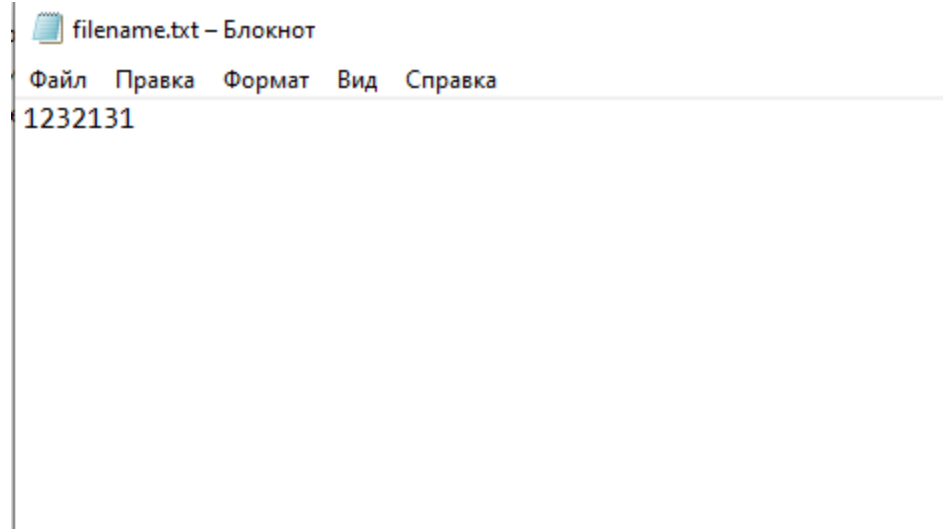




2)Файл пустой но существует



```
..._process_start.exe
file is existing, size: 0
file is existing, size: 0
```



3)Файла не существует

```
using namespace std;

int main()
{
    FileMonitor tets("C:/Users/andru/Documents/Lab2/filename.txt");
    FileObserver observer;
    tets.Attach(&observer);
    while (true)
    {
        tets.CheckFile();
        this_thread::sleep_for(chrono::milliseconds(1000));
    }
}
```

```
C:\Qt\Qt5.14.2\Tools\QtCreator\bin\qtcreator_process_stub.exe
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, size: 0
File is existing, file has been changed, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
File is existing, size: 7
```

Тест Attach

Функция добавляет последователя это подтверждается по примером выше

Тест Detach

Не был проведен

Тест Notify и update аналогичен программе

Тест переименованием файла не был проведен