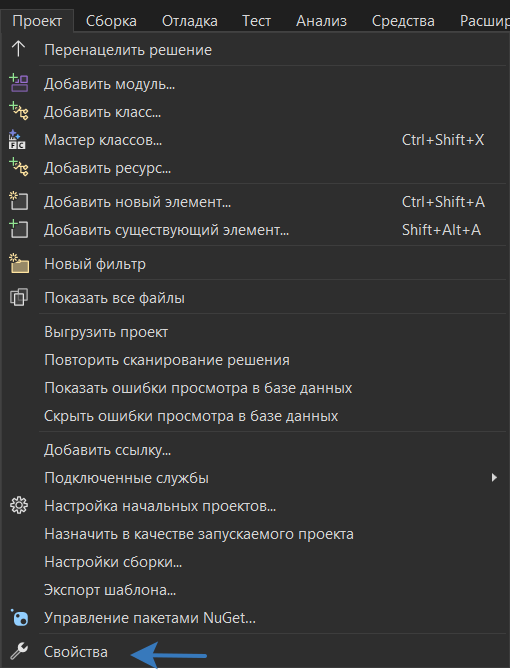
Дополнительная информация для прототипа SeqAdm:

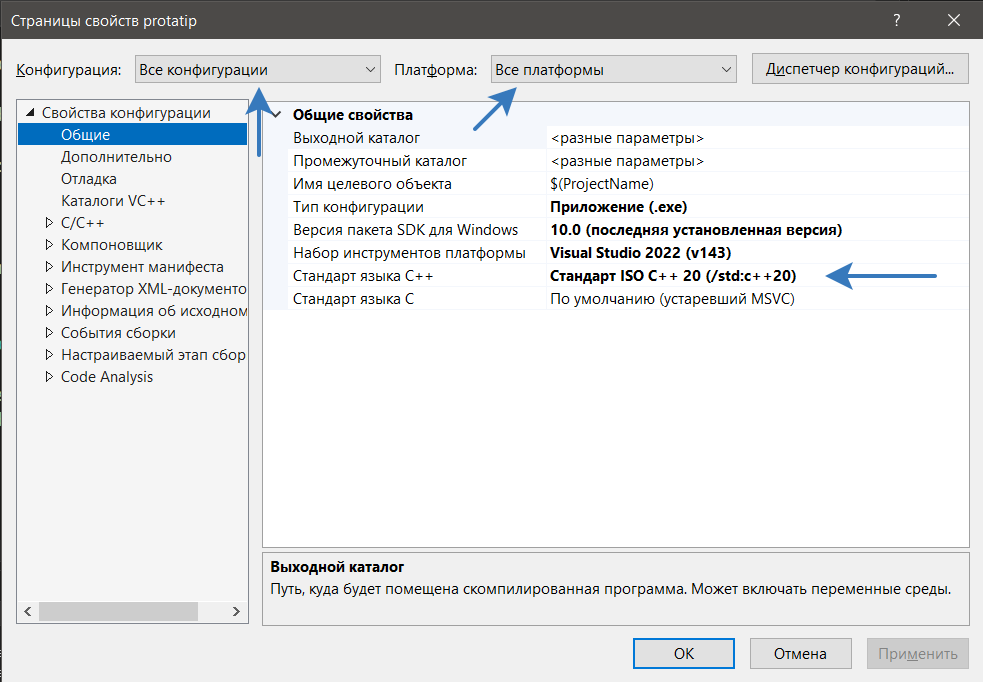
**Подключение библиотеки Json в Microsoft VS 2022:**

По умолчанию библиотека автоматически подключена к среде разработки, но если что-то пошло не так, то вот инструкция по ее подключению:

В верхней части программы нажинаем на вкладку Проект => Свойства

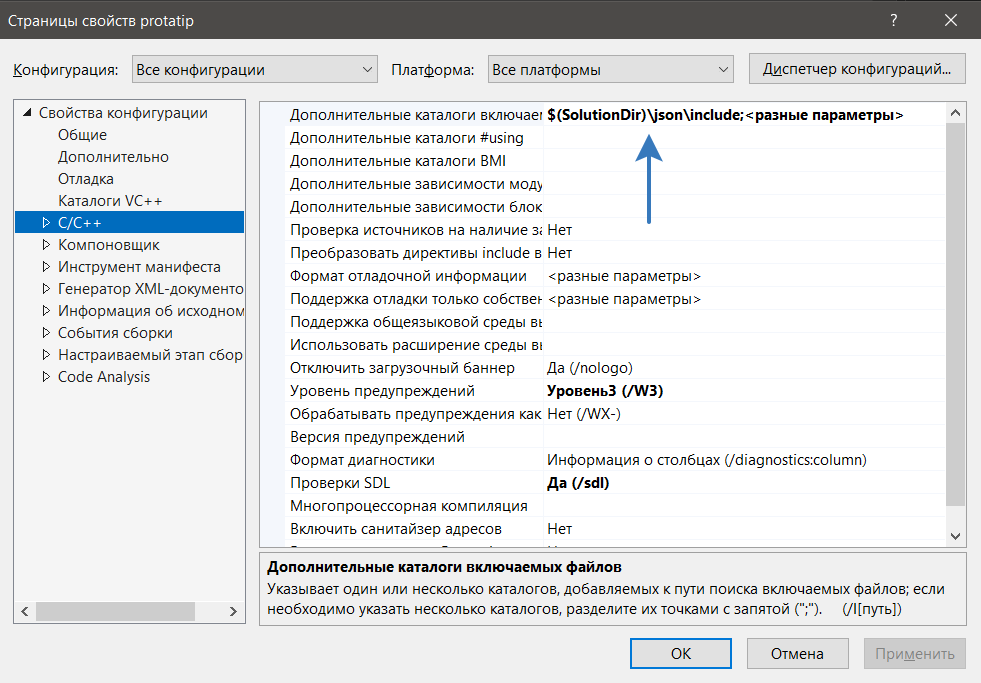
****

Во вкладке “Свойства” переходим в общие и сверяем информацию:



Далее переходим в каталог C/C++:

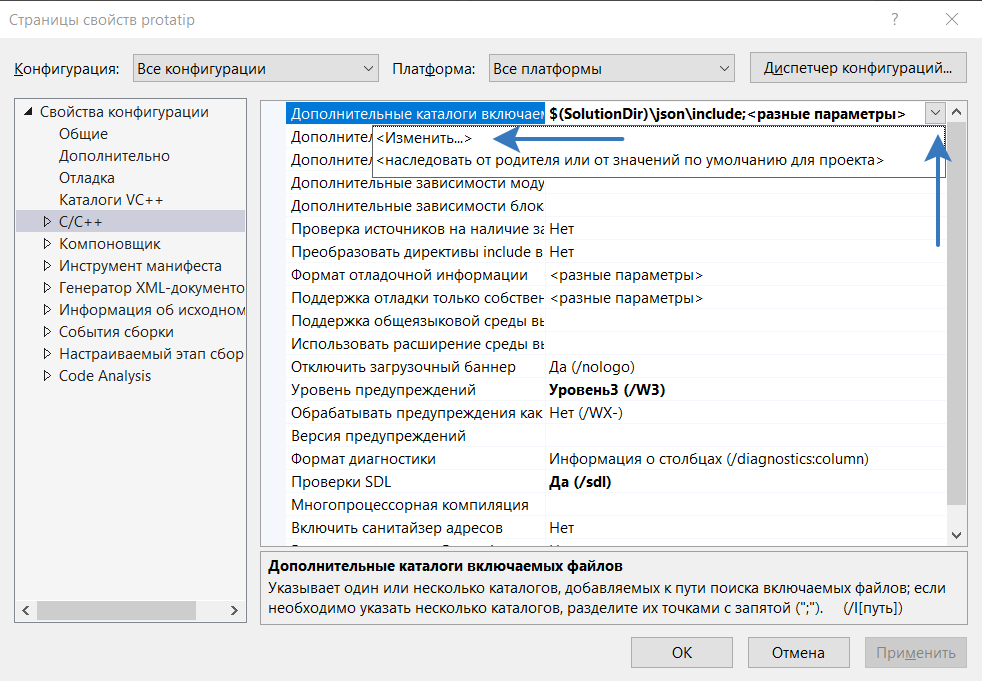
По умолчанию стоят такие каталоги включения:



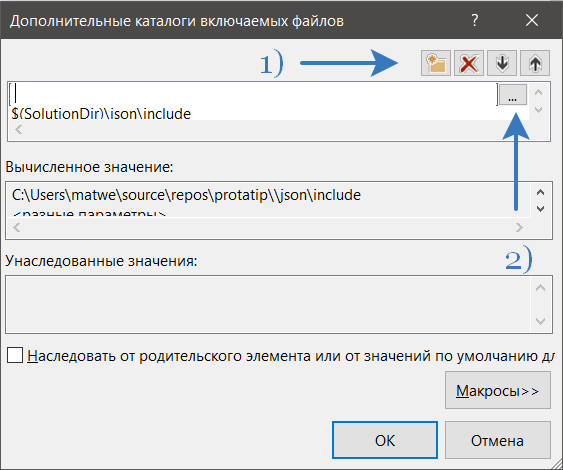
$(SolutionDir) – означает, что библиотека Json располагается по тому же пути, что и cpp файл.

Если же в каталоге ничего не стоит, то стоит явно указать путь расположения библиотеки, для этого надо сделать:

Нажать на “Дополнительные каталоги включаемых файлов” => “Изменить”



В появившемся окне выбрать:



Указать библиотек Json по пути Диск: \Ваш путь\Json\include, либо же есть библиотека находится рядом с cpp файлом, то можно вписать параметры по умолчанию: $(SolutionDir)\json\include

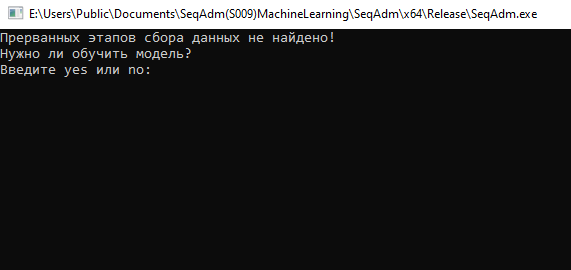
После данных действий программа должна скомпилироваться.

**Демонстрация работы программы:**

**Демонстрация обучения модели и сбора данных:**

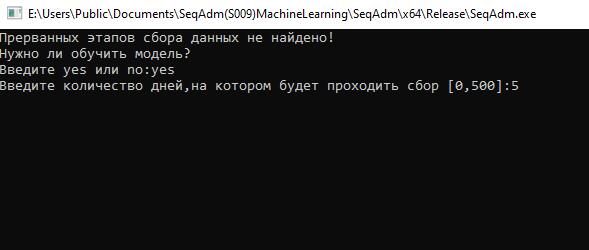
Шаг 1:

Вписать в диалоговое окно “yes” либо 1.

****

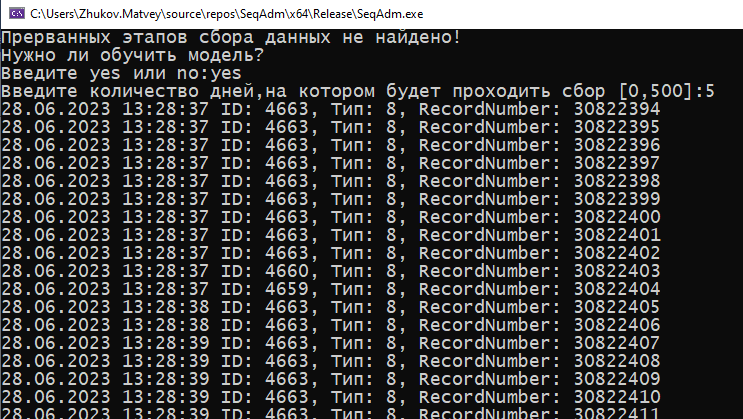
Шаг 2:

Вписать в диалоговое окно количество дней, в течение которых будет собираться выборка. Оптимальное время сбора в районе 5-7 дней, но в некоторых случаях может потребоваться и больше, если, например, не так часто эксплуатируют компьютер, на котором собираются данные.

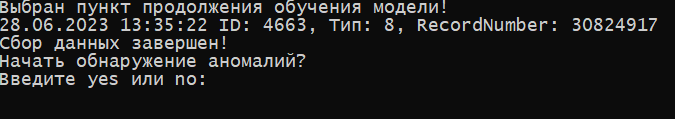


Шаг 3:

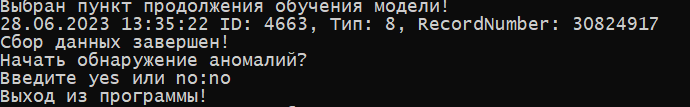
После ввода информации, начинается процесс сбора тренировочной выборки:



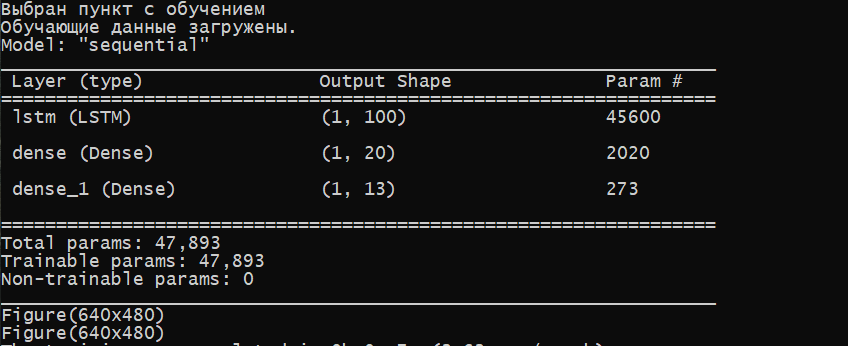
По ее окончанию, выпадет диалоговое окно:



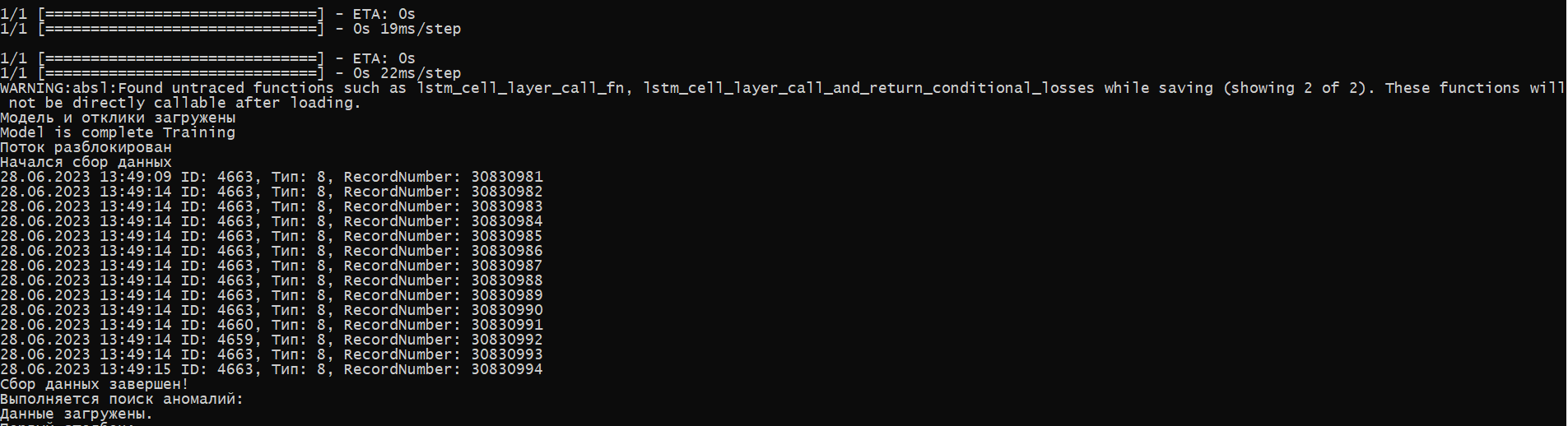
В нем если вписать “no”, то программа завершится:



Если вписать “yes”, то начнётся обучение модели нейронной сети:



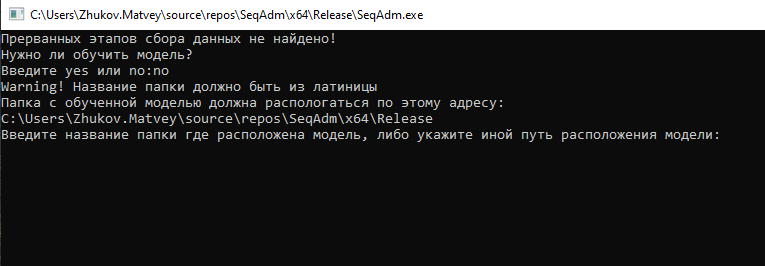
После ее обучения начнется поиск аномалий:



**Демонстрация загрузки уже обученной модели:**

Шаг 1:

Вписать в диалоговое окно “no” либо 0.



Директива, которая указана на скриншоте, подгружается автоматически и меняется в зависимости от того, где вы расположили exe файл.

Шаг 2:

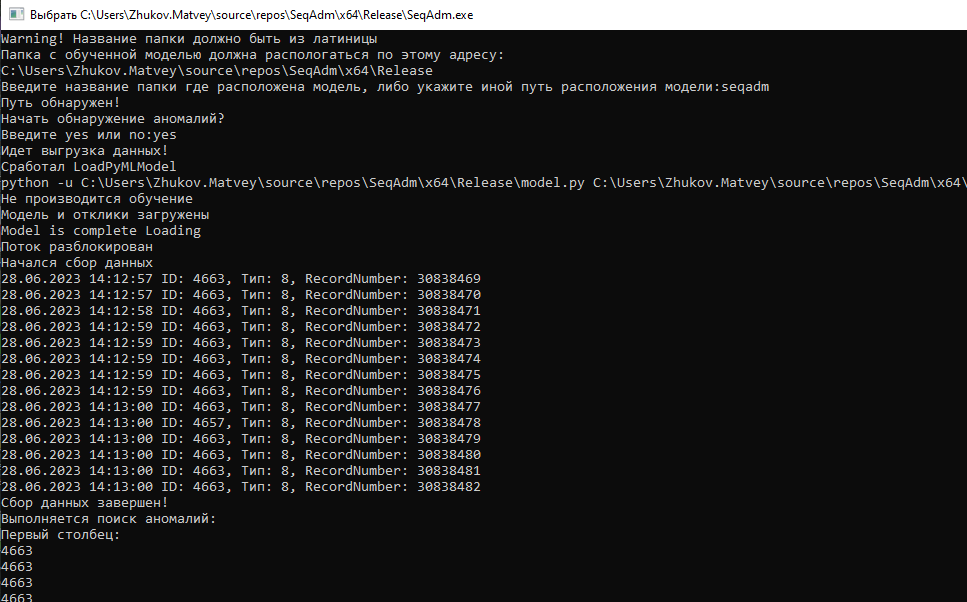
В примере вписана папка, в которой находится обученная модель, и которая находится там же, где и сам exe файл. Но также можно указать иной адрес расположения модели, это поддерживается программой. Если же данные обученной модели лежат непосредственно возле exe файла,

то нужно вписать “\”.

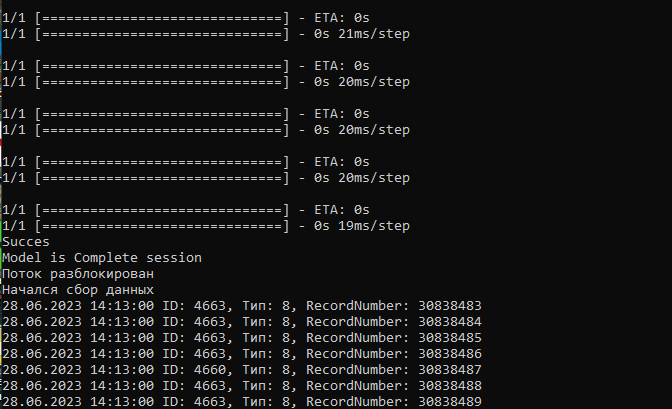
Шаг 3:

В появившемся диалоговом окне, надо вписать “yes”, что бы пошла загрузка модели и начался поиск аномалий:

Скриншот 1:



Скриншот 2:

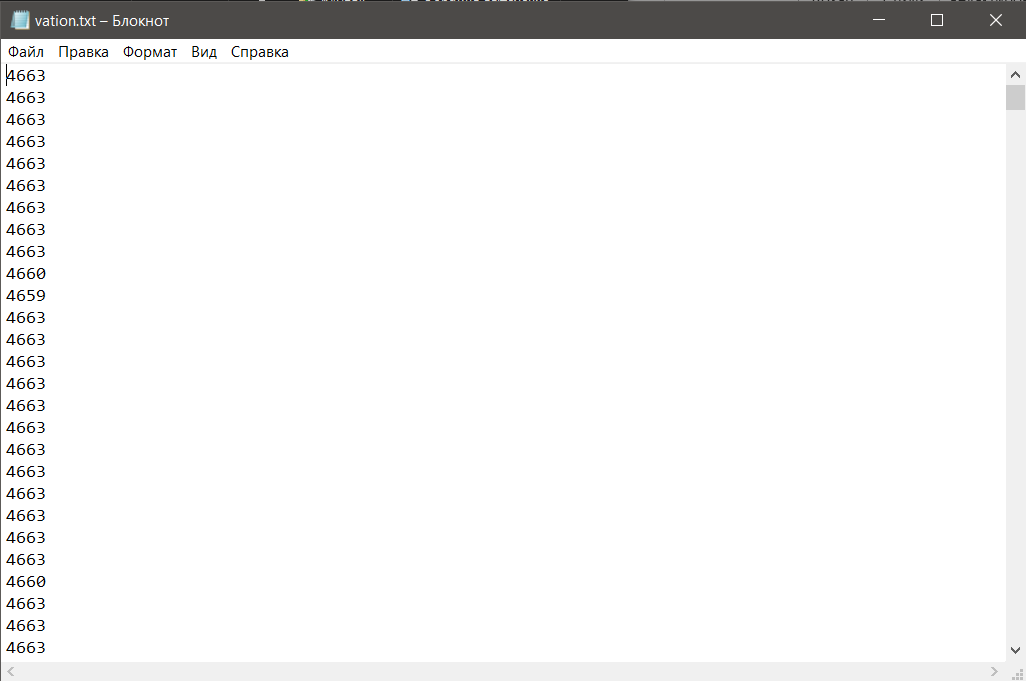


По мере возникновения новых событий, программа собирает их и обрабатывает, если возникает какая-то аномальная последовательность, то информация выводится в “detected\_anomalies.txt”.

Создаваемые программой Файлы:

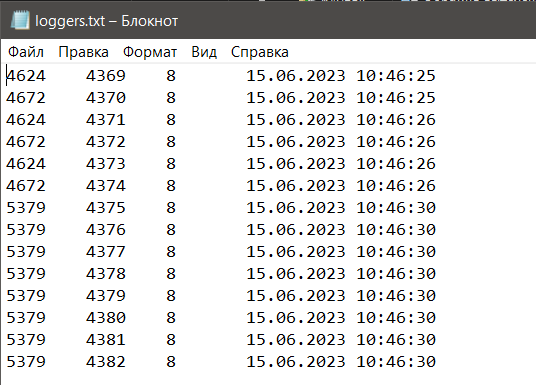
1. **“vation.txt”**

В этом файле собираются последовательность {Event ID}, для дальнейшего обучения модели на ней.



1. **“loggers.txt”**

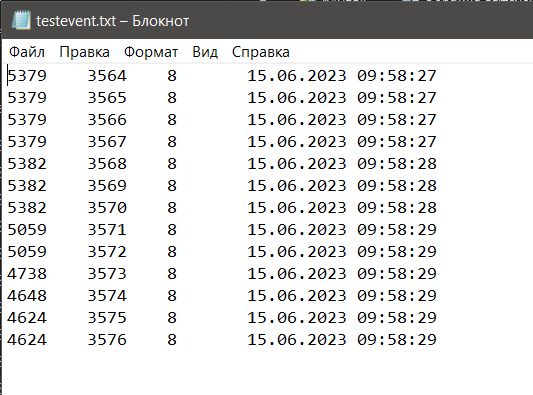
В этом файле записываются все данные о собранных Event id



Формат записи: Event ID, RecordNumber, Тип значения, Дата записи события в журнал.

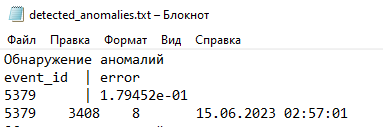
RecordNumber – уникальный номер записи каждого события в журнал.

1. **“testevent.txt”**



В этот файл записываются 13 событий, которые в дальнейшем идут на анализ в нашу нейронную сеть. С этим файлом программа начинает работать после выбора пункта в диалоговом окне “Начать обнаружение аномалий”.

1. **“detected\_anomalies.txt”**

****

В этот файл выводится информация о аномальной последовательности.