

Как проще выполнять замеры в методе AddQueriesStream

Нам надо посчитать, сколько времени суммарно работает каждая стадия обработки поискового запроса. Это удобно делать с помощью вот таких классов:

```
using namespace std;
using namespace chrono;

struct TotalDuration {
    string message;
    steady_clock::duration value;

    explicit TotalDuration(const string& msg = "")
        : message(msg + ": ")
        , value(0)
    {
    }

    ~TotalDuration() {
        ostringstream os;
        os << message
            << duration_cast<milliseconds>(value).count()
            << " ms" << endl;
        cerr << os.str();
    }
};

class AddDuration {
public:
    explicit AddDuration(steady_clock::duration& dest)
        : add_to(dest)
        , start(steady_clock::now())
    {
    }

    explicit AddDuration(TotalDuration& dest)
        : AddDuration(dest.value)
    {
    }

    ~AddDuration() {
        add_to += steady_clock::now() - start;
    }

private:
    steady_clock::duration& add_to;
    steady_clock::time_point start;
};

#define ADD_DURATION(value) \
    AddDuration UNIQ_ID(__LINE__){value};
```

Пример применения:

```
istream& ReadLine(istream& input, string& s, TotalDuration& dest) {  
    ADD_DURATION(dest);  
    return getline(input, s);  
}  
  
int main() {  
    TotalDuration read("Total read");  
    TotalDuration parse("Total parse");  
  
    for (string line; ReadLine(cin, line, read); ) {  
        ADD_DURATION(parse);  
        const auto words = SplitIntoWordsView(line);  
    }  
}
```