

Условия лабораторной

На основании данных с сайта ЦБ РФ написать программу, которая раз в 10 секунд опрашивает данные о курсе валют и выводит текущие значения. По окончании работы выводит среднее и медианное значение.

main.cpp

```
#include<iostream>
#include<ctime>
#include<conio.h>
#include "curl-7.82.0/include/curl/curl.h"
#include "json-develop/single_include/nlohmann/json.hpp"
#include "Valute.h"
#include "Request.h"

size_t write_data(char *ptr, size_t size, size_t nmemb, std::string *data)
{
    if (data)
    {
        data->append(ptr, size * nmemb);
        return size * nmemb;
    }
    else
        return 0;
}

int main()
{
    std::string data, names[35];
    size_t(*f)(char * ptr, size_t
size, size_t
nmemb, std::string * data) = write_data;
    Request request(f, &data);
    Valute valutes;
    nlohmann::json json_data;
    while (!kbhit())
    {
        request.get_data();
        json_data = nlohmann::json::parse(data);
        valutes.add(json_data);
        data.clear();
        for (auto it = json_data["Valute"].begin(); it !=
json_data["Valute"].end(); it++)
        {
            std::cout << it.key() << " : " << valutes.get_value(it.key())
<< '\n';
        }
        Sleep(10000);
    }
}
```

```

        for (auto it = json_data["Valute"].begin(); it !=
json_data["Valute"].end(); it++)
        {
            std::cout << it.key() << " " << "Average : " <<
valutes.get_average(it.key()) << " Mid: " << valutes.get_mid(it.key()) <<
'\n';
        }
        return 0;
    }
}

```

Request.h

```

#ifndef LAB2_REQUEST_H
#define LAB2_REQUEST_H

#include "curl-7.82.0/include/curl/curl.h"

class Request
{
public:
    Request(){}

    Request(size_t (*f)(char *ptr, size_t size, size_t nmemb, std::string
*data), std::string *data)
    {
        curl = curl_easy_init();
        curl_easy_setopt(curl, CURLOPT_URL, "https://www.cbr-xml-
daily.ru/daily_json.js");
        curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, 0);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, f);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, data);
    }

    Request(Request const &) = delete;

    Request &operator=(Request const &) = delete;

    bool get_data() const
    {
        CURLcode res = curl_easy_perform(curl);
        if (res != CURLE_OK)
            return false;
        else
            return true;
    }

    ~Request()
    {
        curl_easy_cleanup(curl);
    }
}

```

```
private:
    CURL *curl;
};

#endif
```

Valute.h

```
#ifndef LAB2_VALUTE_H
#define LAB2_VALUTE_H

#include<map>
#include<set>
#include "json-develop/single_include/nlohmann/json.hpp"

class Valute
{
public:
    Valute(){}

    void add(nlohmann::json data)
    {
        for (auto it = data["Valute"].begin(); it != data["Valute"].end();
it++)
        {
            values[it.key()] = data["Valute"][it.key()]["Value"];
            save_values[it.key()].insert(values[it.key()]);
        }
    }

    float get_value(std::string valute)
    {
        return values[valute];
    }

    float get_average(std::string valute)
    {
        float sum = 0;
        for (auto it = save_values[valute].begin(); it !=
save_values[valute].end(); it++)
        {
            sum = sum + *it;
        }
        return sum / save_values[valute].size();
    }

    float get_mid(std::string valute)
    {
        int mid = save_values[valute].size() / 2, pos = 1;
```

```
        auto it = save_values[valute].begin();
        if (mid == 0)
            return *it;
        while (true)
        {
            if (pos == mid)
                return *it;
            pos++;
            it++;
        }
    }

private:
    std::map<std::string, float> values;
    std::map<std::string, std::multiset<float>> save_values;
};

#endif
```

Ввод и вывод

Программа выводит курсы валют.

Вывод

Используя принципы ООП, я выполнил предложенное мне задание. Также я научился подключать сторонние библиотеки.