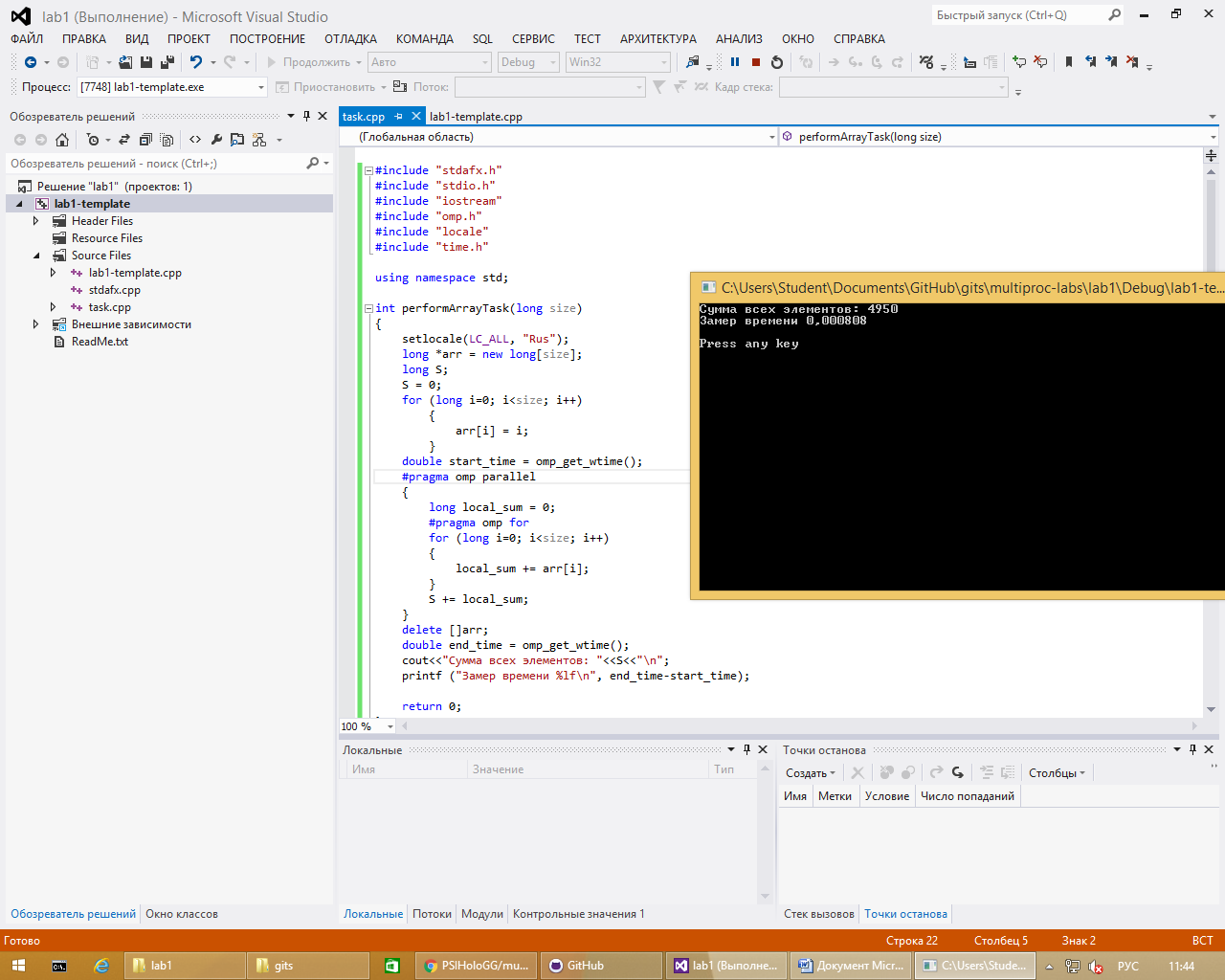
**Код многопоточности.**



#include "stdafx.h"

#include "stdio.h"

#include "iostream"

#include "omp.h"

#include "locale"

#include "time.h"

using namespace std;

int performArrayTask(long size)

{

setlocale(LC\_ALL, "Rus");

long \*arr = new long[size];

long S;

S = 0;

for (long i=0; i<size; i++)

{

arr[i] = i;

}

double start\_time = omp\_get\_wtime();

#pragma omp parallel

{

long local\_sum = 0;

#pragma omp for

for (long i=0; i<size; i++)

{

local\_sum += arr[i];

}

S += local\_sum;

}

delete []arr;

double end\_time = omp\_get\_wtime();

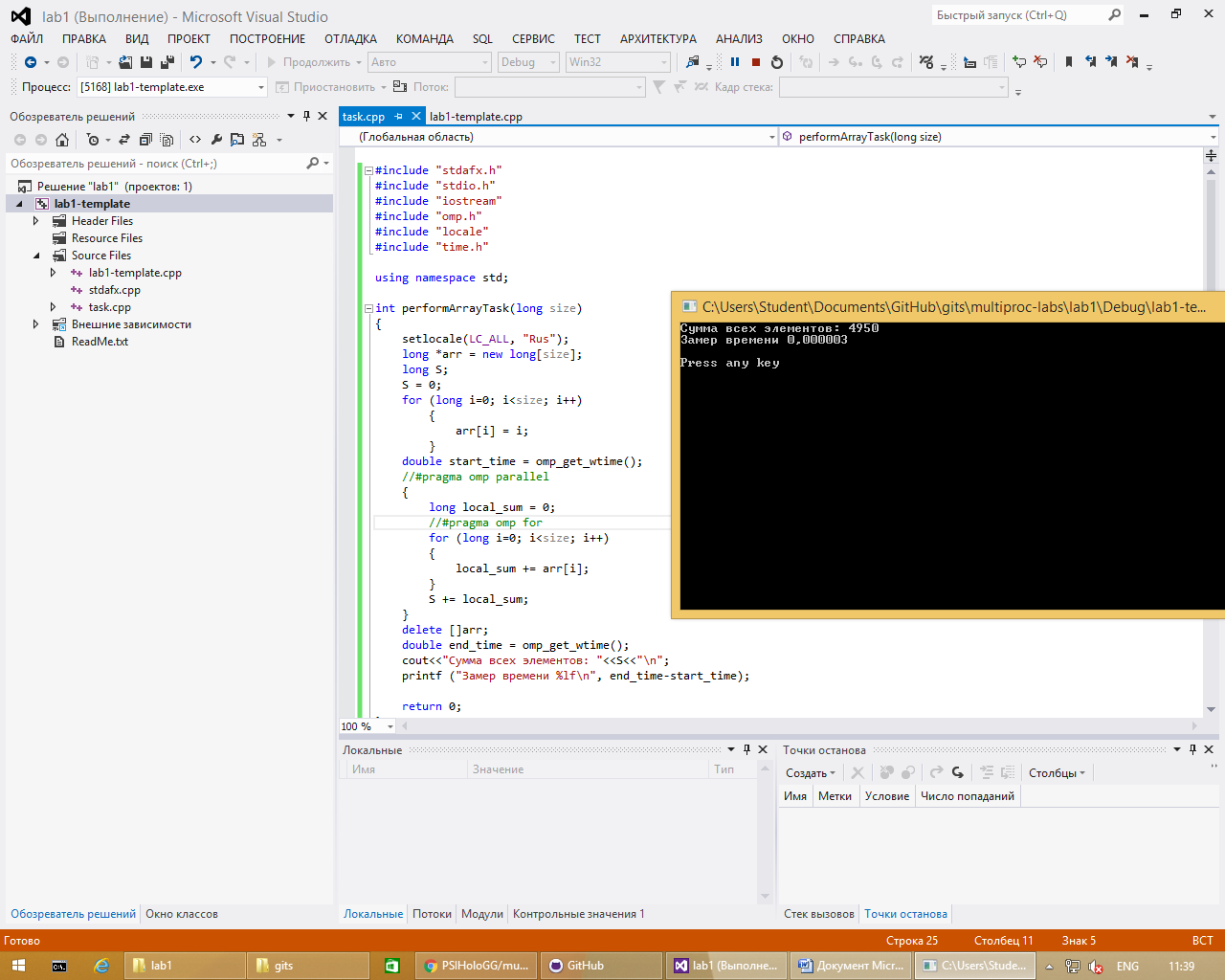
cout<<"Сумма всех элементов: "<<S<<"\n";

printf ("Замер времени %lf\n", end\_time-start\_time);

return 0;

}

**Код многопоточности с закомментированными потоками (1 поток).**



#include "stdafx.h"

#include "stdio.h"

#include "iostream"

#include "omp.h"

#include "locale"

#include "time.h"

using namespace std;

int performArrayTask(long size)

{

setlocale(LC\_ALL, "Rus");

long \*arr = new long[size];

long S;

S = 0;

for (long i=0; i<size; i++)

{

arr[i] = i;

}

double start\_time = omp\_get\_wtime();

//#pragma omp parallel

{

long local\_sum = 0;

//#pragma omp for

for (long i=0; i<size; i++)

{

local\_sum += arr[i];

}

S += local\_sum;

}

delete []arr;

double end\_time = omp\_get\_wtime();

cout<<"Сумма всех элементов: "<<S<<"\n";

printf ("Замер времени %lf\n", end\_time-start\_time);

return 0;

}

**Код многопоточности с заданным числом потоков**

Добавить строку #pragma omp parallel num\_threads (2)