

HttpClient

В этой лабораторной требуется написать простого http клиента.

Клиент должен уметь кидать запросы на сервер с адресом `http://127.0.0.1:{port}/{method}` , где `port` – целочисленное, приходит первой строкой со стандартного потока ввода, а `method` – вызываемый на сервере метод.

Далее расписаны поддерживаемые сервером методы.

/Ping

Метод пинг служит признаком того, что сервер находится в рабочем состоянии, в ответе запроса приходит `HttpStatusCode.Ok (200)`. В любом другом случае сервер считается недоступным.

/GetInputData

С помощью этого метода участник может получить входные данные для задачи. Входные данные приходят в теле ответа в виде сериализованного в Json объекта типа `Input` в кодировке `Utf-8`.

/WriteAnswer

С помощью этого метода можно отдать ответ задаче серверу. Ответ нужно отдавать в теле запроса в виде сериализованного объекта `Output` в Json в кодировке `Utf-8`.

Задача считается решенной, если решение участника отработало без исключений, и на сервер прислали верный ответ. Начинать работу с сервера следует после того, как он будет готов принимать запросы.

```
public class Input
{
    public int K { get; set; }
    public decimal[] Sums { get; set; }
    public int[] Muls { get; set; }
}

public class Output
{
    public decimal SumResult { get; set; }
    public int MulResult { get; set; }
    public decimal[] SortedInputs { get; set; }
}
```

где `SumResult` сумма всех чисел из массива `Sums` входного объекта, умноженная на коэффициент `K`
`MulResult` произведение всех чисел из массива `Muls` входного объекта `SortedInputs` отсортированные числа из полей `Sums`, `Muls` входного объекта/

Пример объектов в Json

```
{"K":10,"Sums":[1.01,2.02],"Muls":[1,4]}
```

```
{"SumResult":30.30,"MulResult":4,"SortedInputs":[1.0,1.01,2.02,4.0]}
```