

Задание: реализовать параллельные вычисления с помощью асинхронного делегата

Код программы:

```
namespace Async1
{
    class Program
    {
        static void Main(string[] args)
        {
            //создание экземпляра класса от родителя
            Parser parse = new Client();
            parse.EventParse += Args;
            while (true)
            {
                Console.WriteLine("Параллельное вычисление");
                System.Threading.Thread.Sleep(1000);
            }
        }
        static void Args(object sander, EventHandlerParseStr e)
        {
            //вывод значения messege
            Console.WriteLine(e.Messege);
        }
    }
    class Parser
    {
        //объявление делегата
        protected delegate string DelegateGetMessege();
        //объявление события
        public event EventHandler<EventHandlerParseStr> EventParse;
        public Parser() { }
        protected void ParseStr(string message)
        {
            if (EventParse != null)
                //передача сообщения с сервера в конструктор класса
                EventParse(this, new
                EventHandlerParseStr(string.Format("Количество шагов: {0}", message)));
        }
    }
    class Client : Parser
    {
        public Client()
        {
            //передача метода в делегат
            DelegateGetMessege handl = new DelegateGetMessege(GetMessege);
            //запуск асинхронного делегата
            handl.BeginInvoke(null, null);
        }
        protected string GetMessege()
        {

```

```

        //функция выполнения последовательных шагов
        int cont = 0;
        while (true)
        {
            ParseStr(cont.ToString());
            cont++;
            System.Threading.Thread.Sleep(1000);
        }
    }
}
class EventHendlerParseStr : EventArgs
{
    //передача переменной messege значение в конструкторе
    public EventHendlerParseStr(string messege)
    {
        Messege = messege;
    }
    //переменную можно только считывать
    public string Messege { get; private set; }
}

```

Результат:

```

Параллельное вычисление
Количество шагов: 0
Параллельное вычисление
Количество шагов: 1
Параллельное вычисление
Количество шагов: 2
Параллельное вычисление
Количество шагов: 3
Параллельное вычисление
Количество шагов: 4
Параллельное вычисление
Количество шагов: 5
Параллельное вычисление
Количество шагов: 6
Параллельное вычисление
Количество шагов: 7
Параллельное вычисление
Количество шагов: 8
Параллельное вычисление
Количество шагов: 9
Параллельное вычисление
Количество шагов: 10
Параллельное вычисление
Количество шагов: 11
Параллельное вычисление
Количество шагов: 12
Параллельное вычисление

```

Вывод:

Разработано консольное приложение, реализовывающее получение строки на сервер и её автоматическую обработку. Для этого были созданы класс Parse и наследуемый от него класс Client. В первом классе реализовывается объявление делегата DelegateGetMessege, а во втором ему

присваивается метод GetMessage и запускается асинхронный делегат. Также был создан класс EventHandlerParseStr, в конструкторе которого происходит присвоение значения переменной message. Создан метод Args для вывода этой переменной, т.е. самого сообщения.