

Урок №10

Ошибки это круто!

Отладка

Инструменты для избавления программ от ошибок.
Подпрограммы, методы.

Отладка необходима, когда программа работает неправильно

- ▶ Чтобы программа была без ошибок
- ▶ Для лучшего понимания работы программы
- ▶ Для удобства пользователя
- ▶ Для предотвращения потери данных

Инструменты отладки

- Вывод в узлах
- Точки останова
- Пошаговое выполнение программы
- Просмотр переменных

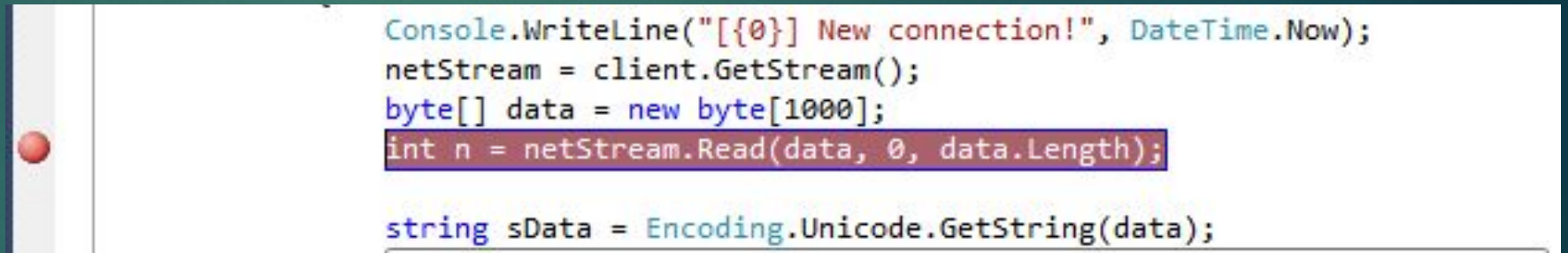
Вывод в узлах

- ▶ Самый понятный метод
- ▶ В нужных местах программы выводим нужные нам переменные в многострочный текстовый элемент вывода

Точки останова

- ▶ Останавливает программу на нужной строке и дает возможность посмотреть переменные

- ▶ Выглядит так:

A screenshot of a code editor window. On the left side, there is a vertical gray bar representing the margin. A red circular breakpoint icon is placed on this bar, aligned with the fourth line of code. The code is written in C# and is as follows:

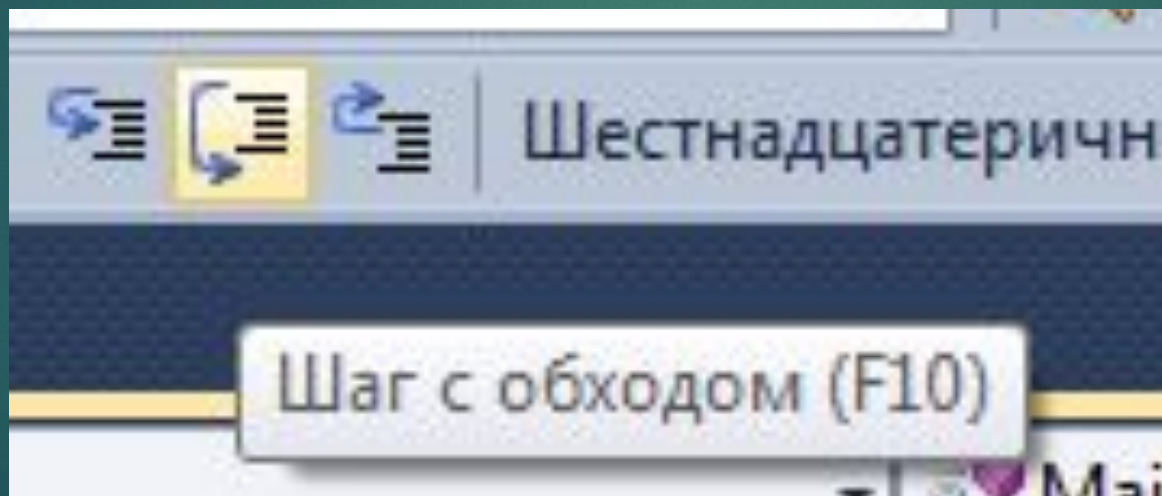
```
Console.WriteLine("[{0}] New connection!", DateTime.Now);  
netStream = client.GetStream();  
byte[] data = new byte[1000];  
int n = netStream.Read(data, 0, data.Length);  
  
string sData = Encoding.Unicode.GetString(data);
```

The line containing `int n = netStream.Read(data, 0, data.Length);` is highlighted with a purple background.

- ▶ Ставится и снимается щелчком по серой панели слева от кода или в контекстном меню кода

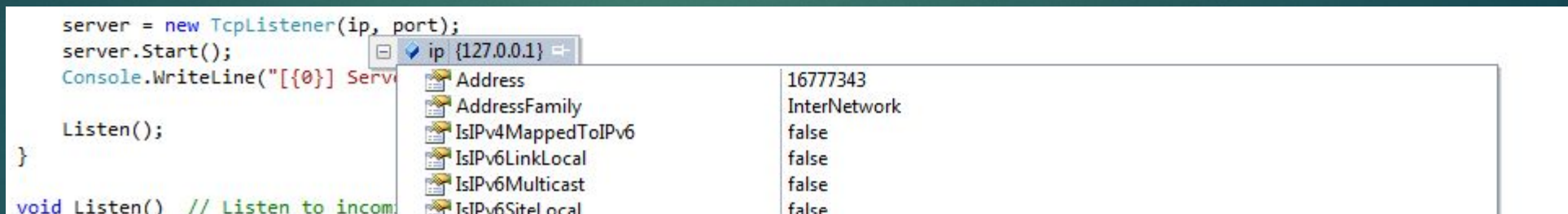
Пошаговое выполнение программы

- ▶ Используется после точки останова.
- ▶ Нажатием на кнопку или на клавишу F10 заставляем программу выполнить следующую строку кода

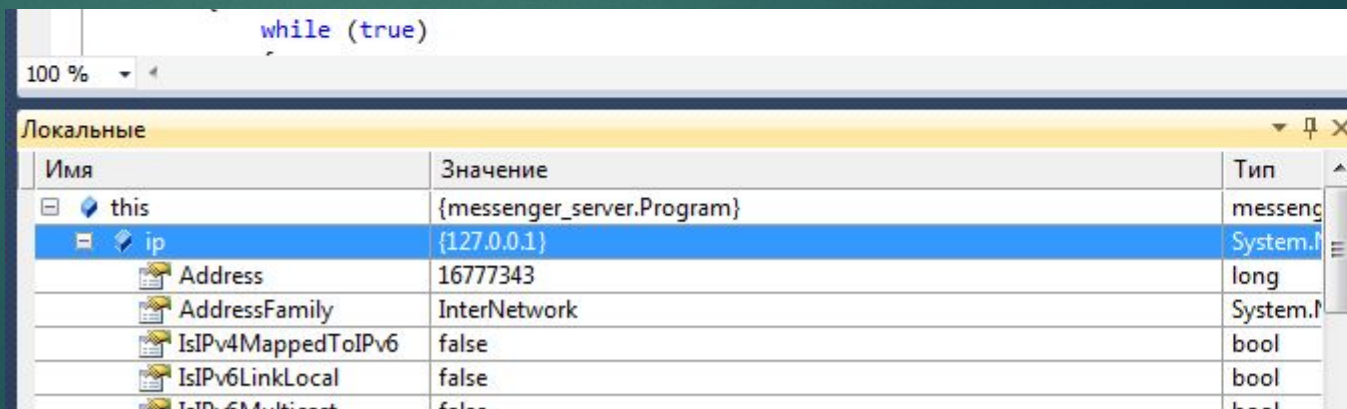


Просмотр переменных

- ▶ Используется после точки останова.
- ▶ Наведение на переменную открывает её содержимое



- ▶ Или нужно использовать панель внизу экрана разработки



Подпрограммы

Процедуры, функции, методы

- ▶ Язык C# позволяет создавать подпрограммы для повторяющихся действий.
- ▶ Все, что мы делали до этого — это писали код программы в созданных для нас методах
- ▶ Выглядит метод вот так:

```
void Method(int start, double finish)
{
    label1.Text = start.ToString();
    label2.Text = finish.ToString();
}
```


Как писать метод

Вот так

```
тип_возвращаемого_значения Название_метода(Входные_параметры_метода)  
{  
    действия  
}
```

Пример

```
void Method(int start, double finish)  
{  
    label1.Text = start.ToString();  
    label2.Text = finish.ToString();  
}
```

Как разработать игру

- ▶ Придумать правила и цель игры для **пользователя** и для **программы**
- ▶ Написать пользовательский сценарий – расписать по пунктам как должен действовать пользователь
- ▶ Написать алгоритм – расписать по пунктам как программа должна отвечать на действия пользователя
- ▶ Создать пользовательский интерфейс – накидать на форму элементов управления
- ▶ Реализовать алгоритм – запрограммировать все элементы управления
- ▶ Отладить программу – избавить программу от ошибок.
- ▶ Добавить красивостей и заветушек

Домашнее задание

- ▶ Описать будущую игру ту, что вы хотите создать
 - ▶ Описать все её правила и цели.
 - ▶ Описать пользовательские сценарии
 - ▶ Описать алгоритм
 - ▶ Создать пользовательский интерфейс для игры
- ▶ Отладить уже написанную азартную игру.