

# Платформа .NET и язык программирования C#

Ошибки и методы отладки  
программ

# Ошибки

1. Ошибки компиляции
2. Ошибки выполнения
3. Логические ошибки
4. Ошибки алгоритмов
5. Стилистические ошибки

# Обработка исключений

Ошибка при выполнении называется исключительной ситуацией (исключением).

С целью обезопасить программу от предполагаемых исключений используют структуру `try ... catch ... finally`:

```
try
{
    ... //Потенциально опасный код
}

catch (типИсключения имя)
{
    ... //Код, выполняемый при возникновении исключения,
        //если значение фильтра true
}
```

дополнительные блоки `catch` //Можно обрабатывать другие исключения

```
finally
{
    ... //Код, выполняемый всегда после блоков try и catch
}
```

# Метод TryParse

Статический метод типа данных для интерпретации текстовой строки как значения этого типа.

Используется вместо метода Parse, чтобы не обрабатывать возможные исключения, если интерпретация строки не удалась.

Метод возвращает логическое значение (true — интерпретация удалась, false — не удалась).

Первый параметр — интерпретируемая строка.

Последний параметр — полученное в результате значение заданного типа.

Возможны перегрузки с промежуточными параметрами, уточняющими способ интерпретации.

Пример:

```
string str;  
int n;  
bool result = int.TryParse(str, out n);
```

# Методы отладки

- Ищите красные и зеленые волнистые линии
- Проверяйте лампочку
- Исправляйте (или обрабатывайте) исключения
- Используйте отладчик в режиме пошагового выполнения
- Используйте просмотр значений переменных во время пошагового выполнения

# Отладчик

Процесс отладки подразумевает пошаговое выполнение кода в среде отладки в поисках точки, в которой вы допустили ошибку при написании программы.

Проанализируйте проблему, задавая себе правильные вопросы:

- Что именно должен был выполнить код?
- Что произошло вместо этого?

# Отладчик

Проверьте предположения, на основании которых вы ожидаете тот или иной результат:

- Используете ли вы нужный метод? Возможно, он работает не так, как вы ожидаете
- Правильно ли вы используете метод? Даже если вы выбрали нужный метод, он может использоваться неправильно.
- Нет ли в вашем коде опечаток?
- Вносили ли вы изменения в код и могут ли они быть связаны с возникшей проблемой?
- Должна ли переменная содержать определенное значение (или определенный тип значения) и соответствует ли это действительности?

# Режим пошагового выполнения

- Используйте точки остановки (breakpoints)
- Для пошагового выполнения метода используйте «шаг с обходом» (F10)
- Для пошагового выполнения вызываемого метода используйте «шаг с заходом» (F11)
- Для возвращения в вызывающий метод используйте «шаг с выходом» (Shift + F11)
- Для прекращения пошаговой отладки до следующей точки остановки используйте кнопку «Продолжить» (или F5)
- Для прекращения отладки используйте кнопку «Остановить отладку» (или Shift + F5)