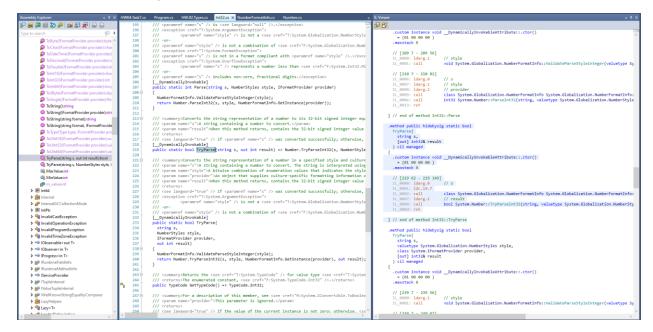
## Задание 7.

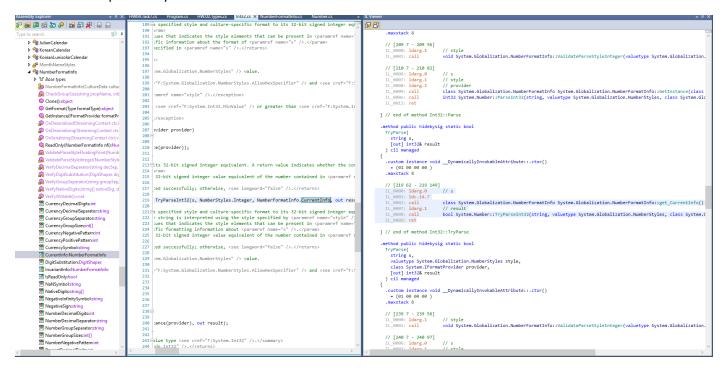
Цель: найти отличия между методами Int.Parse и Int.TryParse при помощи dotPeek.

## Решение:

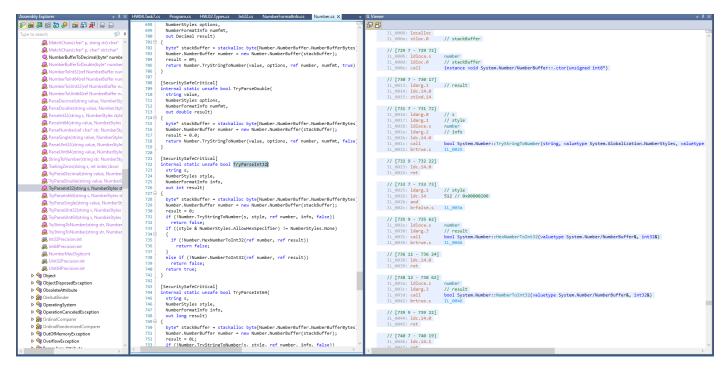
- 1. Ищем структуру int32 (путь mscorlib.System.int32)
- 2. Ищем метод Int.TryParse



3. Видим в IL, что в первую очередь в методе Int.TryParse происходит обращение к свойству currentInfo, из которого получаем объект класса NumberFormatInfo

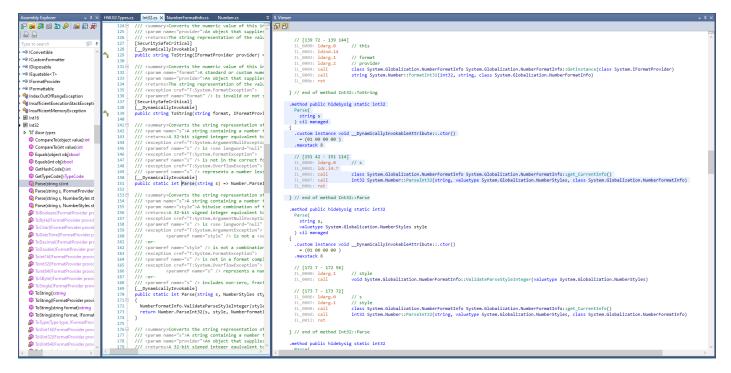


4. Видим в IL, что далее в методе Int.TryParse происходит вызов метода System.Number::TryParseInt32, в который передаётся в том числе полученный ранее NumberFormatInfo.

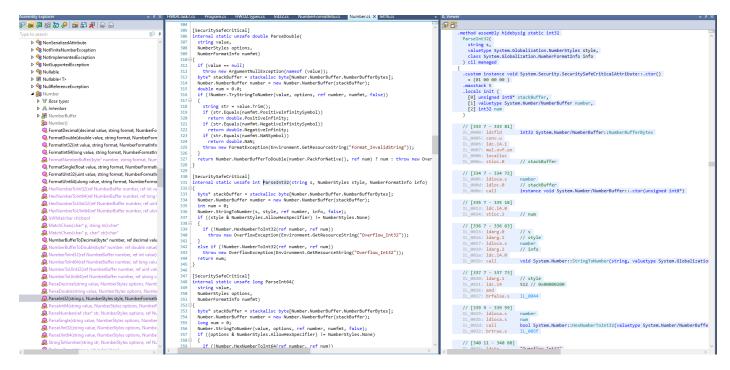


Анализируя код на языке С# можно сделать вывод, что полученные (в качестве параметров) данные используются для определения возможности преобразования строки в число типа int32 и в случае, если преобразование возможно, будут переданы true и результат преобразования в параметре result, а если — нет, то будет передано false.

5. Ищем метод Int.Parse



- 6. Видим, что в Int.Parse происходит такое же обращение к свойству currentInfo, из которого получаем объект класса NumberFormatInfo
- 7. Видим в IL, что далее в методе Int.Parse происходит вызов метода System.Number::ParseInt32, в который передаётся в том числе полученный ранее NumberFormatInfo.



Анализируя код на языке С# можно сделать вывод, что полученные (в качестве параметров) данные используются для определения возможности преобразования строки в число типа int32 и в случае, если преобразование возможно, то будут генерироваться исключения, что приведёт к завершению работы программы, если эти исключения не будут обрабатываться в блоке catch.

Выводы: метод TryParse не генерирует исключения и возвращает bool несмотря на возможность или невозможность преобразования, принятой в качестве параметра, строки в int32, а Parse генерирует исключения в случает невозможности преобразования, принятой в качестве параметра, строки.