

Визуализация кода встроенных языков в Microsoft Visual Studio IDE

Ершов Александр

СПбГУ

2014

Встроенные языки

SQL в C#

```
string t = "";
string s = "select * from "
if (check)
    t = "table1 ";
else
    t = "table2 ";
var command =
    new SqlCommand(s + t, sqlconn);
```

Проблема

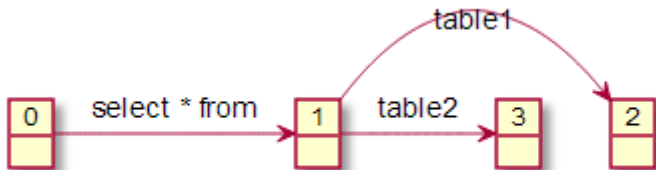
- Компилятор не проводит статического анализа кода встроенного языка
- УС ReSharper plugin проводит статический анализ
- Строка, формируемая динамически может иметь сложную структуру

Задача

Добавить в YC ReSharper plugin возможность графического представления кода встроенного языка в виде графа используя библиотеку GraphX и ReSharper SDK

Пример

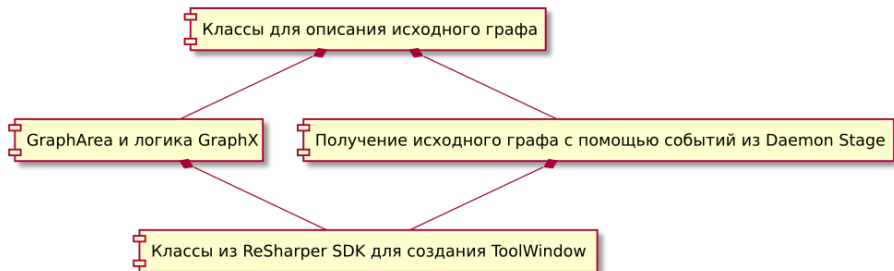
```
string t1 = "table1"  
string t2 = "table2"  
string s = "select * from "  
execute(s + if check then t1 else t2)
```



Существующие решения

- Graphviz - ПО для визуализации графов
- Alvor - плагин для Eclipse, который статически проверяет встроенный SQL в Java

Архитектура



Особенности реализации

- Документация по ReSharper SDK устарела
- Большинство дуг в графе параллельны
- Доступа к экземпляру класса, где анализируется код и создается граф, нет

Результаты

В плагин добавлена возможность визуализации кода встроенного языка в виде графа. Проблема отображения параллельных дуг - они рисуются как прямые линии и не входят в вершину.

Пример работы

The image displays a screenshot of a development environment, likely Visual Studio, showing a C# code file named `SqIDemo2.cs` and its corresponding control flow graph (CFG).

Code Snippet:

```
class SqIDemo2
{
    public void SelectByName(int cond)
    {
        var baseQuery = "drop procedure";
        string tableName;
        switch (cond)
        {
            case 1:
                tableName = "some_else_table";
                break;
            case 2:
                tableName = "some_else_table_2";
                break;
            case 3:
                tableName = "some_else_table_3";
                break;
            default:
                tableName = "default_table d";
                break;
        }
    }
}
```

Control Flow Graph:

The CFG illustrates the execution flow of the `SelectByName` method. It starts at a node labeled `0`, which leads to a `drop procedure` node. From there, the flow branches based on the `cond` parameter value:

- Branch 1 (labeled `1`) leads to the `some_else_table_2` node.
- Branch 2 (labeled `2`) leads to the `some_else_table` node.
- Branch 3 (labeled `3`) leads to the `some_else_table_3` node.
- The `default` branch (labeled `0`) leads to the `default_table d` node.

The graph shows that the `drop procedure` statement is executed before the `switch` statement, and the flow then branches to the appropriate `tableName` assignment based on the `cond` value.

Дальнейшее развитие

Добавить возможность по нажатию на ребро попадать в участок кода. В каждом ребре графа хранится координата сегмента строки, отображаемого им, во внешнем коде. Это позволит более удобно ориентироваться по коду встроенного языка.