

# Автоматизированный поиск утечек памяти в .NET приложениях

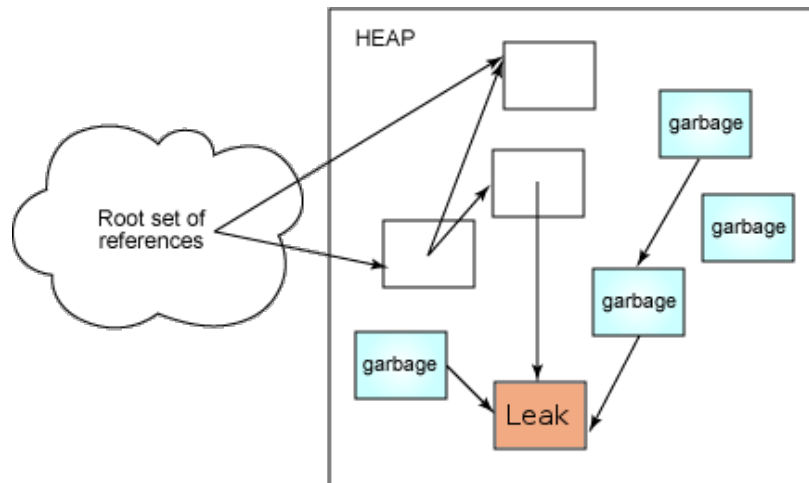
Опейкин Александр

Кафедра Математических и Информационных Технологий  
Санкт-Петербургский Академический Университет  
Санкт-Петербург

3 июня, 2013

Руководитель: Белов Р. В., JetBrains

# Память в .NET



## Утечка памяти

Память, которая должна быть освобождена, по мнению разработчика, но которая не может быть удалена сборщиком мусора на основе анализа достижимости.

Поиск утечек памяти:

- 1 ручной анализ кода;
- 2 статистически (наибольший размер);
- 3 сравнение снимков памяти (наибольший прирост).

## Цель:

Для профилировщика *dotMemory* разработать ряд анализов позволяющих обнаруживать наиболее типичные утечки памяти в .NET приложениях или локализовывать их для быстрого поиска.

## Задачи:

- 1 анализ типичных ошибок средств C#;
- 2 анализ типичных утечек памяти в WPF;
- 3 общий анализ памяти .NET приложений.

# Существующие решения

В .NET профилировщиках (*SciTech*, *Ants*, *YourKit*) существуют анализы:

- разреженные массивы;
- дубликаты строк;
- одинаковые поля;
- ...
- прочее анализы неэффективного использования памяти.

# Анализ типичных ошибок средств C#

## EventHandler

Система, позволяющая объекту **B** подписаться на события объекта **A**

```
var B = new TypeB();  
var A = new TypeA();  
A.SomeEvent += B.SomeCallback;
```

# Анализ типичных ошибок средств C#

## EventHandler

Система, позволяющая объекту **B** подписаться на события объекта **A**

```
var B = new TypeB();  
var A = new TypeA();  
A.SomeEvent += B.SomeCallback;
```

Утечка, если не

```
A.SomeEvent -= B.SomeCallback;
```

## Windows Presentation Foundation

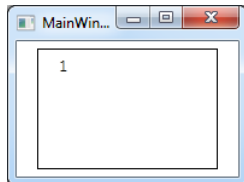
Графическая (презентационная) подсистема в составе .NET Framework

Особенности WPF:

- декларативный XML-подобный язык разметки;
- возможность привязки данных;
- стили;
- шаблоны;
- ...

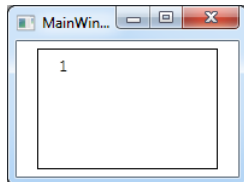


# Привязка данных в WPF



```
<StackPanel x:Name="_myStackPanel">  
    <Label Content="{Binding ElementName=_myStackPanel, Path=Margin}" />  
</StackPanel>
```

# Привязка данных в WPF

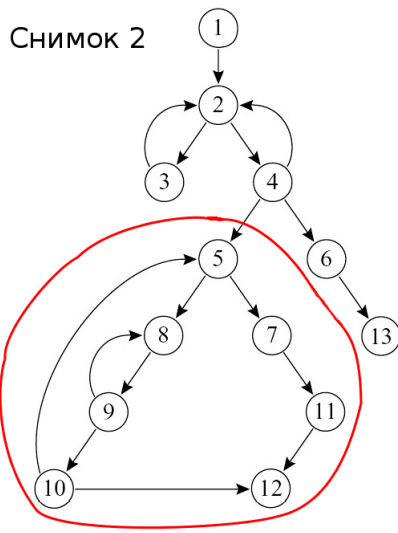
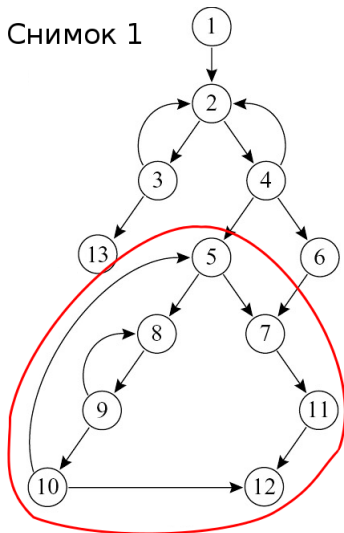


```
<StackPanel x:Name="_myStackPanel">  
    <Label Content="{Binding ElementName=_myStackPanel, Path=Margin}" />  
</StackPanel>
```

Утечка:

```
<StackPanel x:Name="_myStackPanel">  
    <Label Content="{Binding ElementName=_myStackPanel, Path=Children.Count}" />  
</StackPanel>
```

# Общий анализ .NET приложений



# Стабильные подграфы

Имея два (или более) снимка памяти можно найти **стабильные** подграфы.

## Стабильный подграф:

- на подграф не появилось новых ссылок;
- ссылки внутри подграфа - неизменны;
- новых ссылок из подграфа нет.

# Стабильные подграфы в dotMemory

| Тип                       | Стабильные | Всего |
|---------------------------|------------|-------|
| ObservableCollection<...> | 2          | 34    |
| MetadataAssembly          | 945        | 975   |
| ContentPresenter          | 4233       | 16051 |
| EffectiveValueEntry[]     | 3872       | 80153 |

Были реализованные следующие анализы:

- 1 EventHandlers;
- 2 привязка данных в WPF;
- 3 WPF DependencyProperty AddValueChanged;
- 4 WPF DependencyProperty EffectiveValueEntry;
- 5 стабильные подграфы;
- 6 разрастающиеся коллекции.

Спасибо за внимание!