

# Heap Walker in Java Debugger

Студент: Виталий Бибаев

Руководитель: Егор Ушаков (JetBrains)

# Задача

- Добавить возможность просмотра объектов, находящихся в памяти, при отладке java приложения

# Результаты

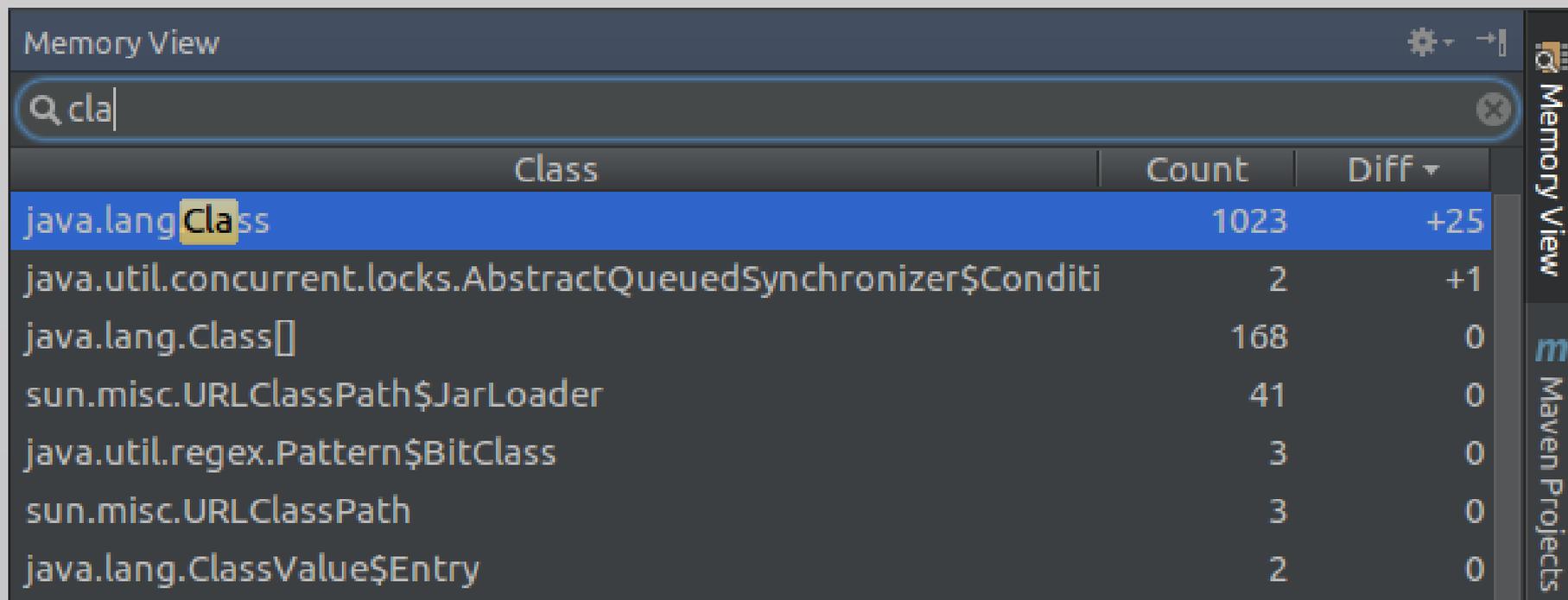
- Исследована возможность добавления функций memory-профайлера в дебаггер IDE
- Выпущен плагин для IntelliJ IDEA и Android Studio

<https://plugins.jetbrains.com/plugin/8537>

# Функции плагина

- Просмотр списка загруженных классов
- Отслеживание изменение количества объектов для каждого класса
- Просмотр объектов какого-либо класса
- Фильтрация объектов по произвольному Java выражению
- Поддержка привычных функций дебаггера: “Add to watches”, “Mark Object”, “Evaluate Expression”

# Результаты



The screenshot shows the 'Memory View' window in IntelliJ IDEA. A search bar at the top contains the text 'cla'. Below the search bar is a table with three columns: 'Class', 'Count', and 'Diff'. The table lists several Java classes, with 'java.lang.Class' being the most prominent result, having a count of 1023 and a difference of +25. Other classes include 'java.util.concurrent.locks.AbstractQueuedSynchronizer\$Condi...', 'java.lang.Class[]', 'sun.misc.URLClassPath\$JarLoader', 'java.util.regex.Pattern\$BitClass', 'sun.misc.URLClassPath', and 'java.lang.ClassValue\$Entry'. On the right side of the window, there are vertical labels for 'Memory View' and 'Maven Projects'.

Class	Count	Diff
java.lang.Class	1023	+25
java.util.concurrent.locks.AbstractQueuedSynchronizer\$Condi...	2	+1
java.lang.Class[]	168	0
sun.misc.URLClassPath\$JarLoader	41	0
java.util.regex.Pattern\$BitClass	3	0
sun.misc.URLClassPath	3	0
java.lang.ClassValue\$Entry	2	0

# Результаты

The screenshot shows the 'Instances of java.lang.String' window in IntelliJ IDEA. The window title is 'Instances of java.lang.String'. The condition for filtering is `charAt(0) == 'a' && charAt(1) == 'l'`. A 'Filter' button is visible next to the condition. A warning message states 'Warning: filtering may have side effects'. The results list shows three instances: `String@1904 = "allocateInstance"`, `String@2189 = "allPermission"`, and `String@2508 = "all_allowed"`. Below these, there is an 'Errors' section with a red exclamation mark icon, containing one error: 'Method threw 'java.lang.StringIndexOutOfBoundsException' exception.'. At the bottom left, it says 'Shown: 3 / 3160' and 'Errors: 5'. A 'Close' button is at the bottom right.

Instances of java.lang.String

Condition: `charAt(0) == 'a' && charAt(1) == 'l'` **Filter**

Warning: filtering may have side effects

- ▶ `String@1904 = "allocateInstance"`
- ▶ `String@2189 = "allPermission"`
- ▶ `String@2508 = "all_allowed"`
- ▼ **Errors**
  - ▶ Method threw 'java.lang.StringIndexOutOfBoundsException' exception.

Shown: 3 / 3160  
Errors: 5

Close

# Что дальше

- Отслеживание новых объектов и где они были созданы
- Оценка размера объектов

# Ссылки

- Описание

<https://blog.jetbrains.com/idea/2016/08/jvm-debugger-memory-view-for-intellij-idea/>

- Плагин

<https://plugins.jetbrains.com/plugin/8537>

Q & A