Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ивановский государственный энергетический университет имени В. И. Ленина»

Кафедра программного обеспечения компьютерных систем

Отчет

Дисциплина «Профилирование и отладка параллельных программ»

Выполнил: студент группы 2-46М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бакалдин А.А.

(подпись) .

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мочалов А.С.

(подпись) .

Иваново, 2021

# Задание

Проанализировать на предмет проблем алгоритм поиска грубых ошибок:

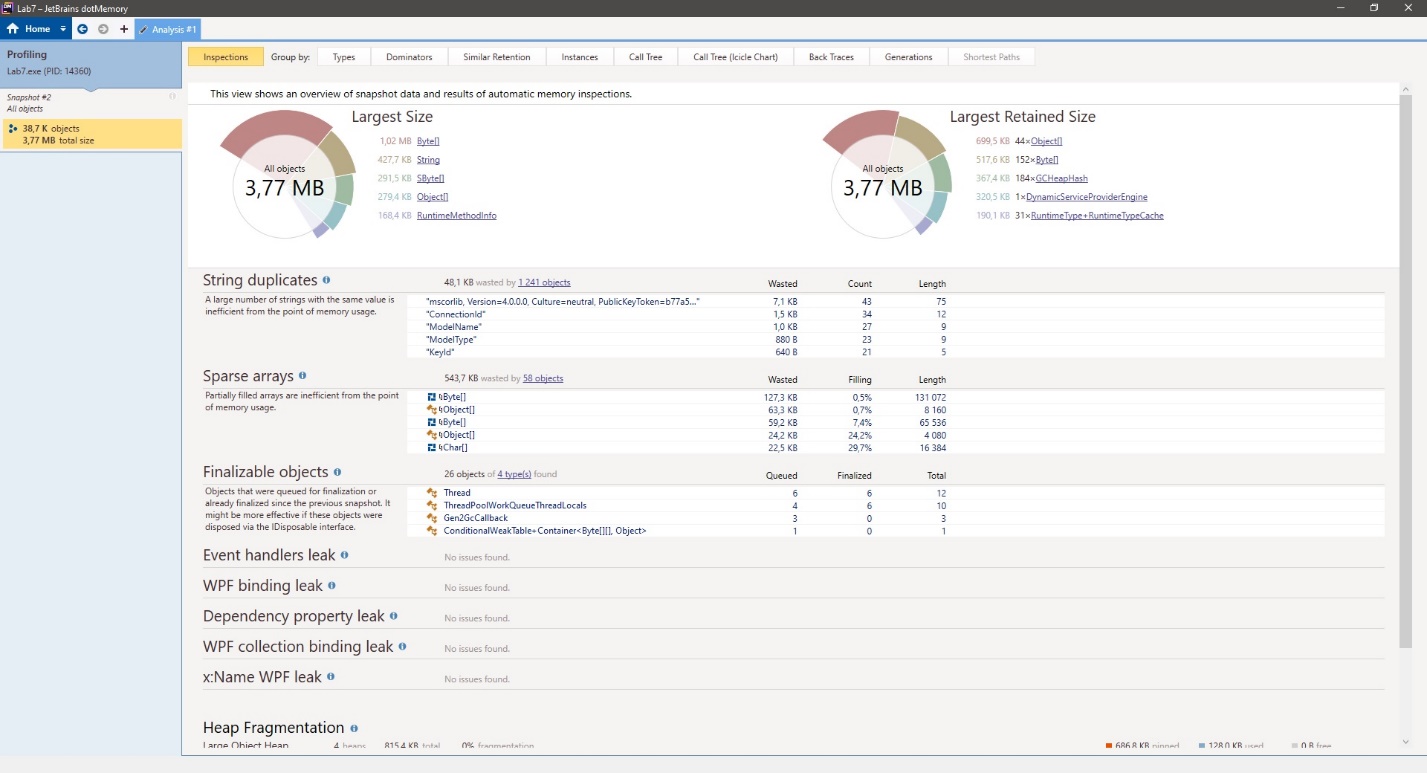
1. Проанализировать приложение профилятором.
2. Выявить источник проблемы.
3. Порекомендовать конкретные исправления.
4. Подготовить краткий отчет в электронном виде. С картинками, графиками, ссылками на проблемные места в коде и с конкретными рекомендациями по исправлению.

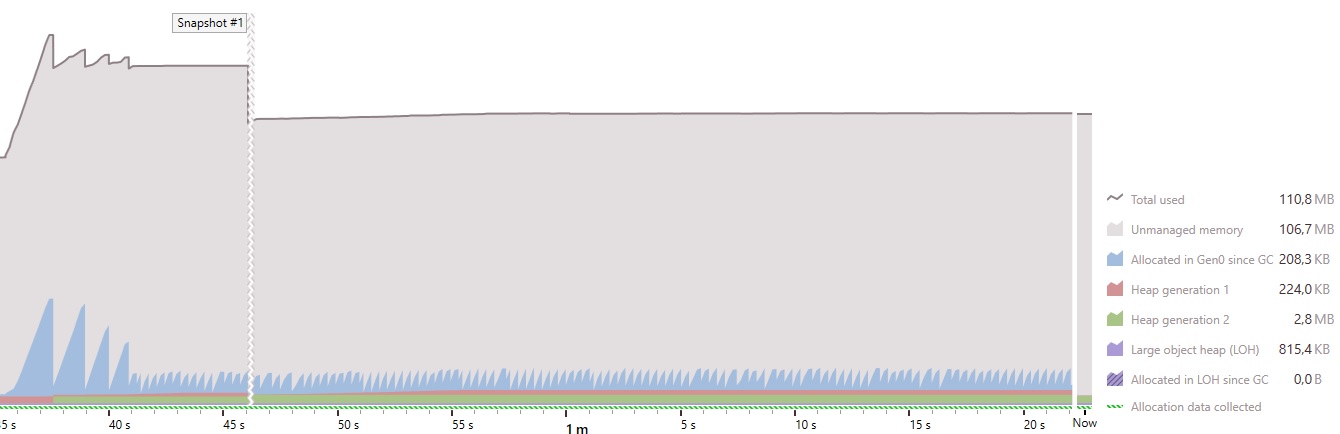
**Выполнение работы**

В качестве профилировщика используется JetBrains dotMemory.

Запустим приложение с присоединенным профилировщиком.

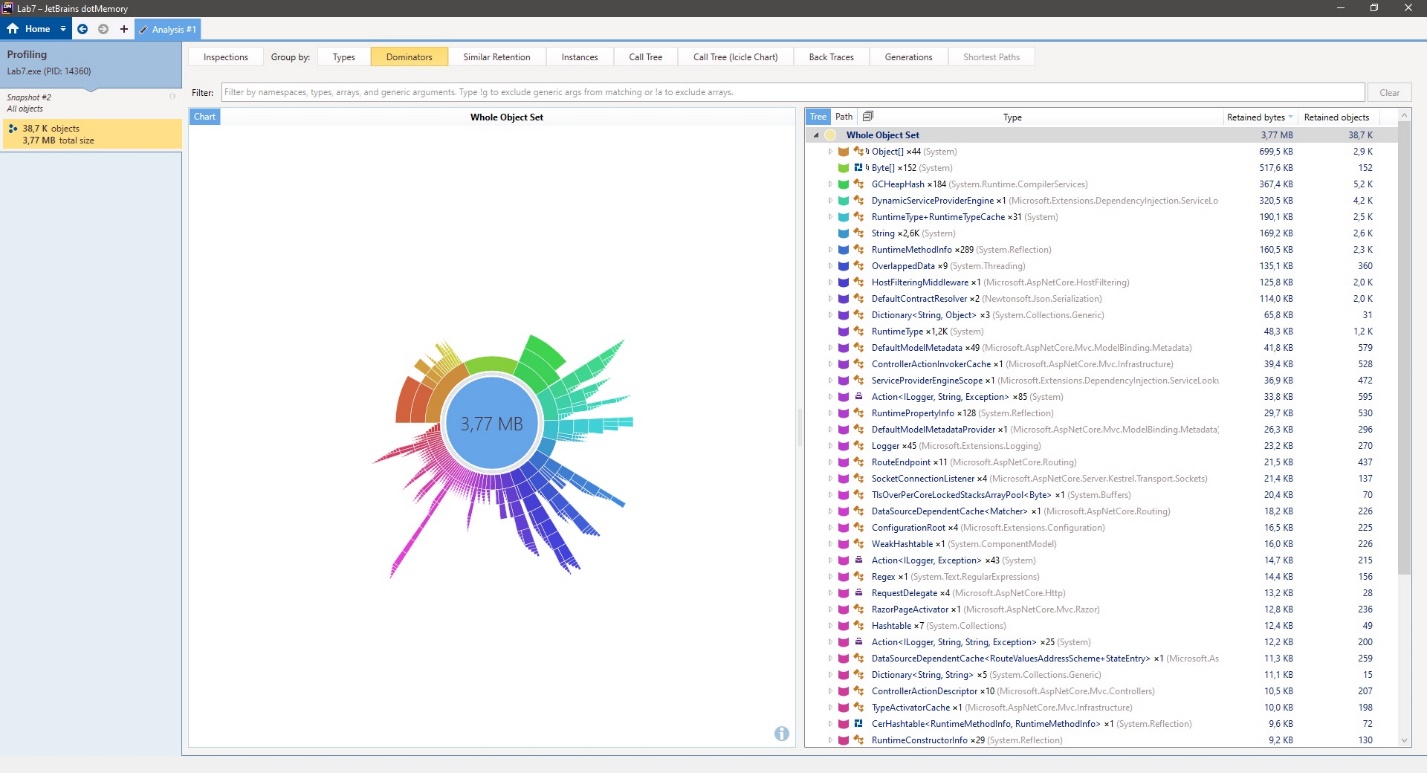
Запустим процесс расчета поиска грубых ошибок (для построения дерева решений использовалась библиотека TreeCollections).

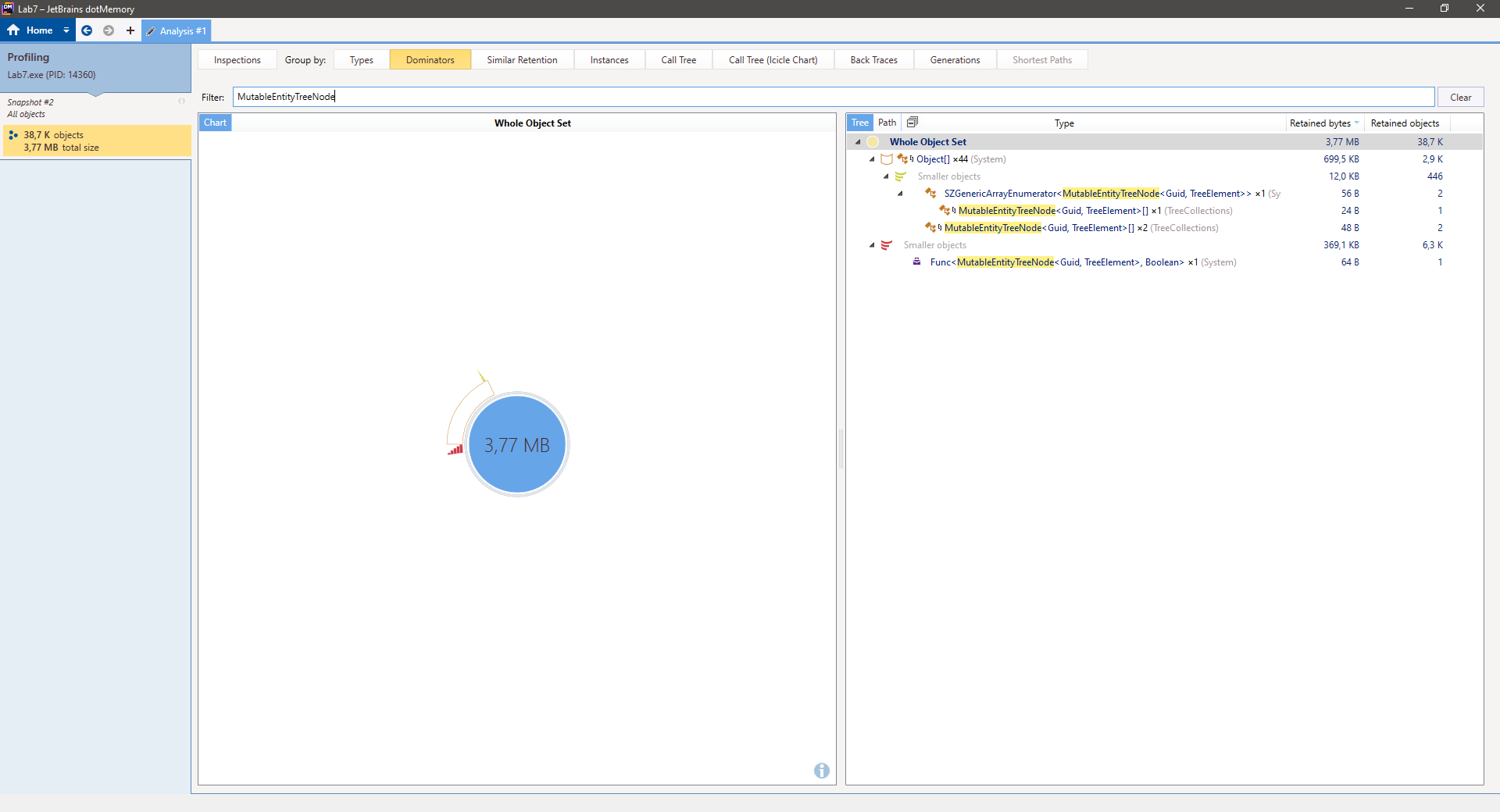




Как видно из графика, память используется достаточно оптимально, при расчете глр на каждой итерации построения дерева происходит заполнение памяти, но после происходит очистка. Общее время выполнения на модели с 9 ошибками 4 минуты 45 секунд. Расходуется не более 110 мегабайт. Примерно 96% всей памяти составляет unmanaged memory, это значит, что память используется максимально эффективно и часто очищается.

Теперь посмотрим заполнение памяти деревом решений:





Общий расход памяти всего 3.77 мегабайт. Общий расход памяти на дерево решений 370 килобайт, что так же показывает оптимальность выбранного решения для построения дерева – дерево использует только 10% от общей расходуемой памяти.

**Вывод**

Как видно из диаграмм памяти, нет смысла оптимизировать расчет поиска грубых ошибок, относительные затраты по памяти достаточно малы. Основной расход памяти происходит в процессе выделения пула асинхронных потоков. Это значит, что используемая библиотека для дерева достаточно хорошо оптимизирована.