**Использование памяти в Java**

**Stack**

Выделяется 1МВ на 1 поток (по умолчанию но можно и задать самому). Если программа будет пытаться загрузить больше то выпадет StackOverflowException

Хранятся:

- служебная инфа по методу(его структура),

- литералы String,

- локальные переменные*(переменные методов)*,

- ссылочные локальные переменные (на экземпляры классов(“объекты”), массивы, Enum-ы). При этом размер самой ссылки (и как следствие ссылочной локальной переменой) в Java неизвестен!!

Тела (инструкции) методов **не хранятся**!!(они хранятся в PermGen)

**Служебная часть памяти**

Хранятся:

Служебная информация для запуска Java + Константы public static final

+ Enum-ы

**PermGen**

При окончании выделенной на PermGen памяти вылетает OutOffMemoryError: PermGen

Хранятся:

- объявление классов, всех их полей

- тела (инструкции) методов

- статические переменные классов

**Heap**

Будет получать память увеличивая по необходимости

При окончании выделенной на Heap памяти вылетает OutOffMemoryError: Heap

Хранятся:

- экземпляры классов(“объекты”)

- массивы

- коллекции

- не статические переменные классов *(переменные объектов)*

Все что делается по “…= new …”

**GarbageCollector**  ищет в Heap по локальным полям Stack-а и по статическим полям PermGen-а. Все на что в Heap эти поля не ссылаются – в Heap помечается как пустое. Т.е. (для быстрой работы GarbageCollector) лучше каждый раз создавать новые объекты, чем хранить постоянно старые и пользоваться ими в случае необходимости.