

Рефакторинг Java Коллекций в Eclipse Collections

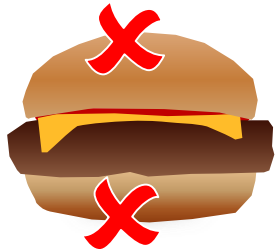
Всё, что вам нужно, и ничего лишнего!

Введение

- Что такое Eclipse Collections?
 - Библиотека коллекций для Java, 100% совместимость с JDK, множество функций, оптимизированное использование памяти
- История
 - Разработка Eclipse Collections началась в компании Goldman Sachs в 2004 году
 - В 2012 году его исходный код был открыт на GitHub как проект под названием [GS Collections](#)
 - Фреймворк GS Collections перешёл в Eclipse Foundation, и был соответственно переименован в [Eclipse Collections](#) в 2015 году
- Выучите Eclipse Collections с помощью [ката](#)
- Принимайте участие в [разработке Eclipse Collections](#)!



Зачем рефакторинг к Eclipse Collections?



Нет методов-"булок"

Ленивые или Энергичные

TestUtils

Тестирование/Отладка

Прямая замена типов из JDK

Наглядный код

Польза!

Легко и просто!

Интуитивные, гуманные API

Богатые API

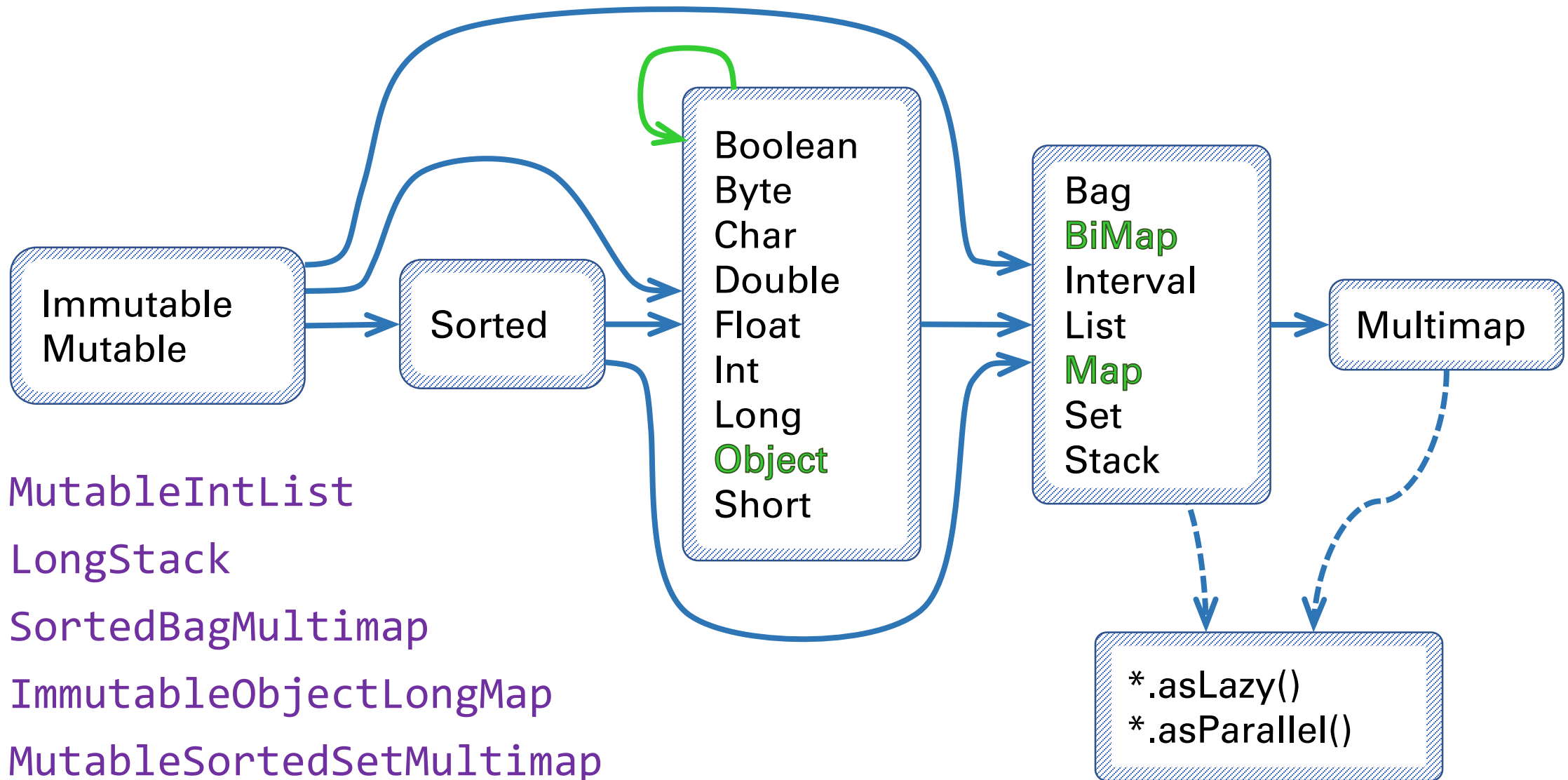
Полностью поддерживаются
примитивные типы

Экономия памяти

Оптимизированная
производительность^[citation needed]

Компактные Словари и Множества

Все типы, которые вам нужны



Создайте их, используя фабричные методы

| Primitive Type | Container Type | Mutability | Multimap Values | Initialized | Lazy |
|----------------|----------------|------------|-----------------|----------------------------|-----------|
| Boolean | Bags | .mutable | .bag | .empty() .of() .with() | .asLazy() |
| Byte | BiMaps | .immutable | .list | .of(one) .with(one) | |
| Char | Lists | .fixedSize | .set | ... | |
| Double | Maps | | .sortedSet | .of(one,...,ten) | |
| Float | Multimaps | | | .with(one,...,ten) | |
| Int | Sets | | | .of(... elements) | |
| Long | SortedBags | | | .with(... elements) | |
| Object | SortedMaps | | | .ofAll(Iterable) | |
| Short | SortedSets | | | .withAll(Iterable) | |
| | Stacks | | | | |

ImmutableLongStack



LongStacks.*immutable*.with(1, 2, 3).asLazy()

LazyLongIterable

Методы по категориям – основные

transform

collect[With]
collect[Boolean,Byte,Char,Double,Float,Int,Long,Short]
collectIf
collectKeysAndValues
collectValues
collectWithIndex
collectWithOccurrences
flatCollect

group

groupBy
groupByEach
groupByUniqueKey
sumBy[Double,Float,Int,Long]
sumOf[Double,Float,Int,Long]
aggregateBy
aggregateInPlaceBy

wrap

asLazy
asParallel
asReversed
asSynchronized
asUnmodifiable

find

detect[With]
detect[With]IfNone
detect[With]Optional
max[By]
min[By]

convert

toArray
toBag
toImmutable
toList
toMap
toReversed
toSet
toSortedBag[By]
toSortedList[By]
toSortedMap
toSortedSet
toSortedSet[By]
toStack
toString

filter

select[With]
selectByOccurrences
selectInstancesOf
reject[With]
partition[With]
partitionWhile

test

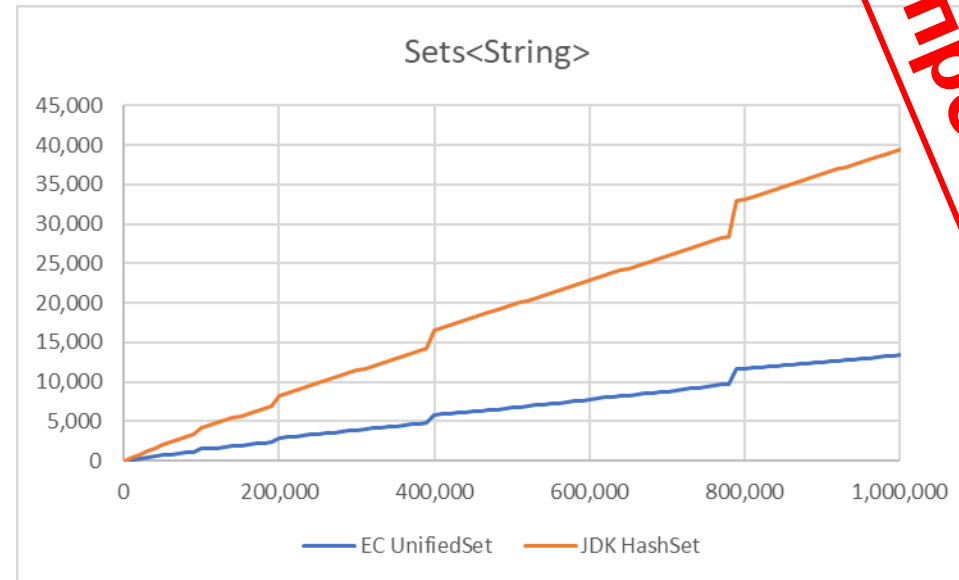
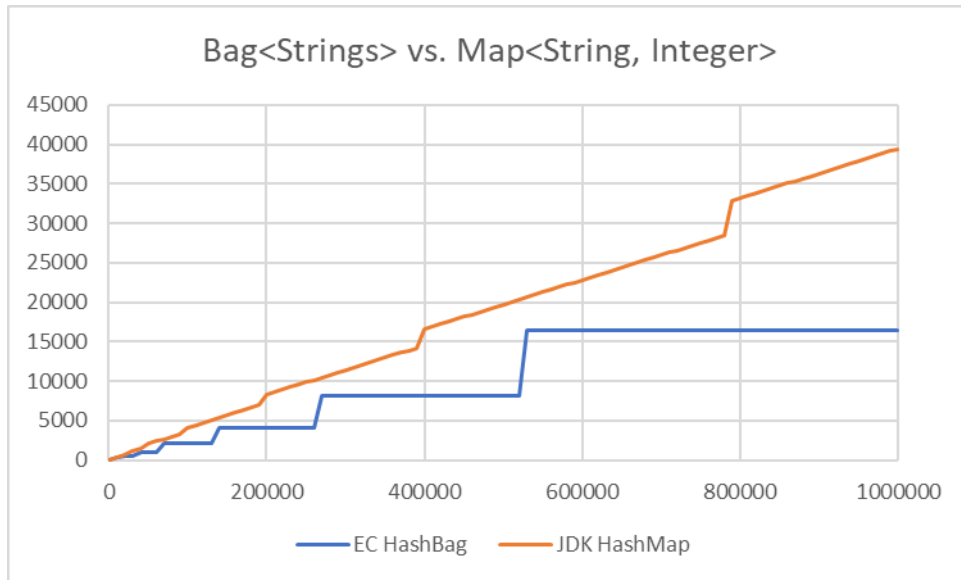
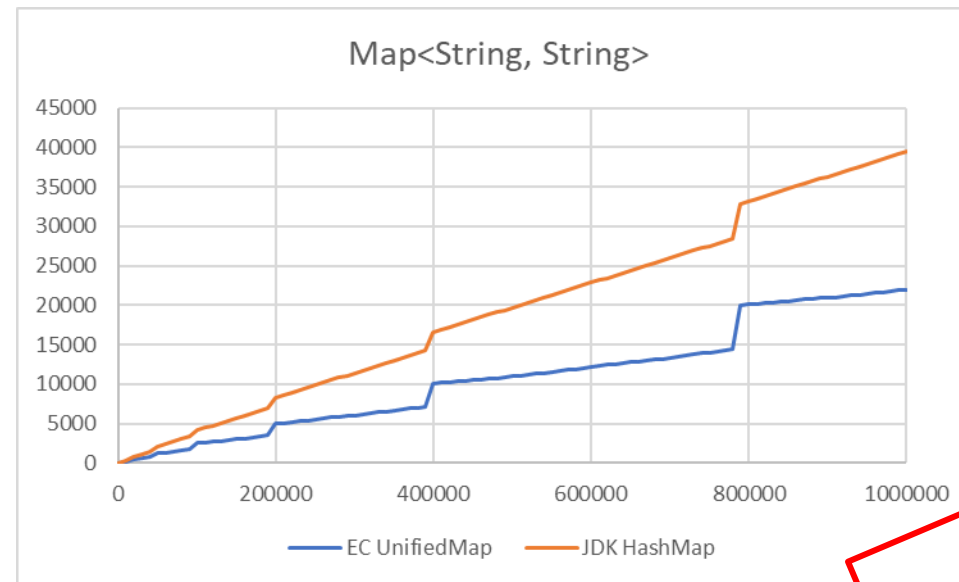
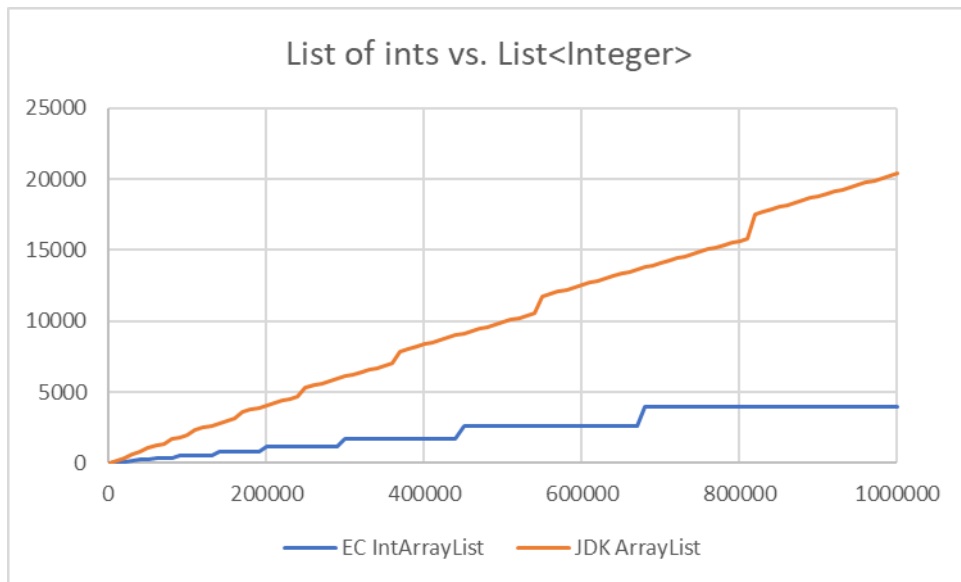
allSatisfy[With]
anySatisfy[With]
noneSatisfy[With]
notEmpty
isEmpty

Вы просите методов? Их есть у меня!



Ну, что ж, за дело!

Сравниваем память (Кбайт, элементов)



Доверяй, но
проверяй!

Сравниваем скорость (Бенчмарк JMH)

