

# Неизменяемые структуры данных (Persistent Data Structures)

## Участники

Варнакин Илья (19227)

Ижицкий Руслан (19226)

## Описание проекта

Неизменяемые структуры данных – структуры, сохраняющие предыдущую свою версию при модификации. Структуры являются фактически неизменяемыми, так как все операции не меняют существующие данные, а возвращают новую структуру с обновленными данными.

Результатом проекта является библиотека со следующими структурами данных в persistent-вариантах:

- Массив (константное время доступа, переменная длина)
- Двусвязный список
- Ассоциативный массив (на основе бинарного дерева)

Массив использует технологию path copying, аналогичную той, которая применяется в языке Clojure. Для константного доступа к элементам можно задать достаточно большое разветвление дерева (внутри используется так называемый m-ary tree), что позволит приблизиться к константному доступу. Например, в Clojure используется разветвление в 32, что обеспечивает доступ к элементам за  $O(\log_{32} N)$ .

Двусвязный список построен на реализации персистентного массива. Каждый новый элемент (вне зависимости от того, в какую часть списка он добавляется) добавляется в конец массива, затем в максимум трех листьях (элемент до добавленного в списке, сам элемент и элемент, идущий в списке после добавленного) изменяются ссылки (индексы соответствующих элементов в массиве) на соответствующие элементы. Для константного доступа к началу и концу списка дополнительно поддерживаются два индекса, соответственно означающие индексы (в дереве) начала и конца списка. При удалении элементов не из конца дерева, в нем образуются пропуски, которые сохраняются в список, отслеживающий все пропуски. Этих пропусков нет в списке, так что будет логичным заполнять эти пропуски в массиве, что не приведет к росту используемой памяти массива при ограниченной памяти списка. Таким образом при следующем добавлении элемента в список он будет добавлен не в конец массива, а в первый доступный пропуск.

Если принять  $O(\log_{BF} N)$  за  $O(1)$  при большом BF (фактор разветвления дерева), то асимптотики операций на двусвязном списке равны асимптотикам на идеальном

двусвязном списке:  $O(N)$  за произвольные доступ, вставку и удаление элементов,  $O(1)$  за доступ, вставку и удаление для начала и конца списка.

Ассоциативный массив использует персистентный массив с фиксированными максимальным размером и глубиной, в качестве индекса используется хэш ключа по модулю максимального размера. В каждом листе дерева находится список, содержащий все ключи и значения, имеющие хэш, соответствующий этому ключу. Этот список также полностью копируется при каждом его изменении.

Реализация структур приводится на языке Java с использованием API существующих структур данных в этом языке.

## Требования

- Все структуры данных сделаны на основе path copying.
- Обеспечивается произвольная вложенность данных, не отказываясь при этом полностью от типизации посредством generic/template; это продемонстрировано в тесте на массиве.
- Реализован универсальный undo-redo механизм для перечисленных структур с поддержкой каскадности (для вложенных структур). Операции undo возвращают версию назад по указателю на предыдущие версии для каждого элемента, операции redo с учётом стека операций undo возвращают отмененную версию.
- Реализована поддержка транзакционной памяти для массива – одна распределенная версия на все потоки.
- Расширено экономичное использование памяти на операцию преобразования одной структуры к другой: списка в массив (по порядку истории добавления элементов в список) и массива в список.

## Структура проекта

Класс Node содержит внутреннее представление узла для персистентного массива и списка. Класс NodeMap содержит внутреннее представление узла для персистентного ассоциативного массива.

Класс PersistentArray содержит персистентный массив. PersistentArrayHistory является подклассом PersistentArray и добавляет методы для версионирования.

PersistentArrayParallel является распределенной версией PersistentArray.

Классы PersistentLinkedList, PersistentLinkedListHistory и PersistentLinkedListParallel аналогичны, только они содержат персистентный двусвязный список. Класс

PersistentTreeMap содержит персистентный ассоциативный массив.

## Календарный план

Этап	Участинки	Функциональность
------	-----------	------------------

Изучение материалов	Варнакин Ижицкий	-
Реализация базовых требований	Варнакин	Ассоциативный массив
	Ижицкий	Массив Двусвязный список
Реализация дополнительных требований	Ижицкий	Массив Двусвязный список

## Ссылки

Код:

[Репозиторий с кодом](#)

Источники:

[Clojure Vectors](#)

[Лекция курса "Параллелизм и транзакции \(часть 2\)"](#)