

# Неизменяемые структуры данных в Clojure

Белоглазов Никита

Белорусский Государственный Университет

2013

вектор (vector)

- ▶ ArrayList, java
- ▶ PersistentVector, clojure

отображение (map)

- ▶ HashMap, java
- ▶ PersistentHashMap, clojure

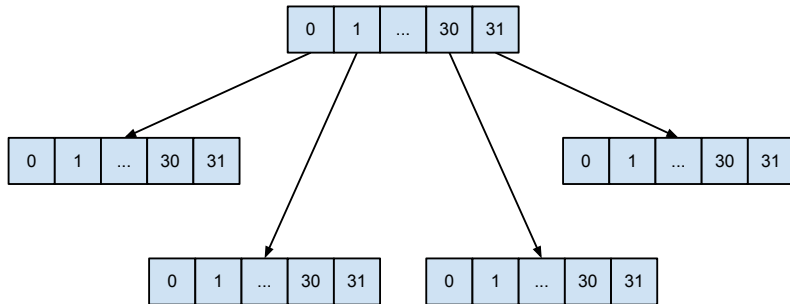
# ArrayList

- ▶ внутри содержит массив фиксированного размера
- ▶ при заполнении массива все элементы копируются в новый массив большего размера

```
public class ArrayList<E> {  
  
    private transient Object[] elementData;  
  
    private int size;  
  
    // constructors, methods  
}
```

# PersistentVector: представление

Вектор является 32-ричным деревом

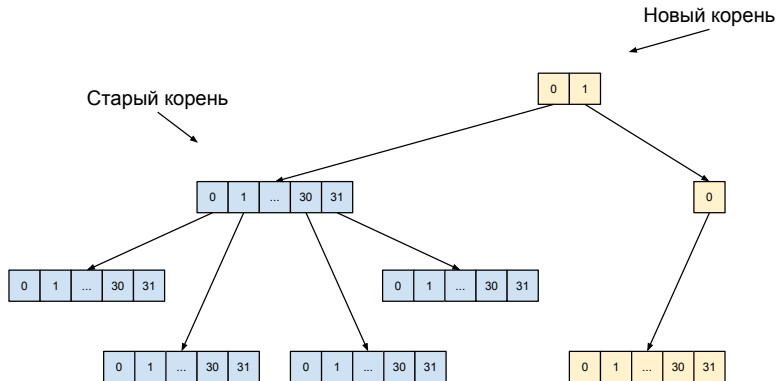


## PersistentVector: индексы

- ▶ битовая запись индекса разбивается на группы по 5 бит
- ▶ каждая группа кодирует число (0-31) - номер ячейки на определённом уровне дерева

[illegible][illegible]

# PersistentVector: добавление элементов



## PersistentVector: coctab

```
1  public class PersistentVector {
2
3      public static class Node {
4          public final AtomicReference<Thread> edit;
5          public final Object[] array;
6      }
7
8      final int cnt;
9      public final int shift;
10     public final Node root;
11     public final Object[] tail;
12     final IPersistentMap _meta;
13 }
```

## PersistentVector: nth

```
1  public Object nth(int i) {
2      Object[] node = arrayFor(i);
3      return node[i & 0x01f];
4  }
5
6  public Object[] arrayFor(int i) {
7      if(i >= 0 && i < cnt) {
8          if(i >= tailoff())
9              return tail;
10         Node node = root;
11         for(int level = shift; level > 0; level -= 5)
12             node = (Node) node.array[(i >>> level) & 0x01f];
13         return node.array;
14     }
15     throw new IndexOutOfBoundsException();
16 }
```



Конец

Спасибо за внимание!

Вопросы?

