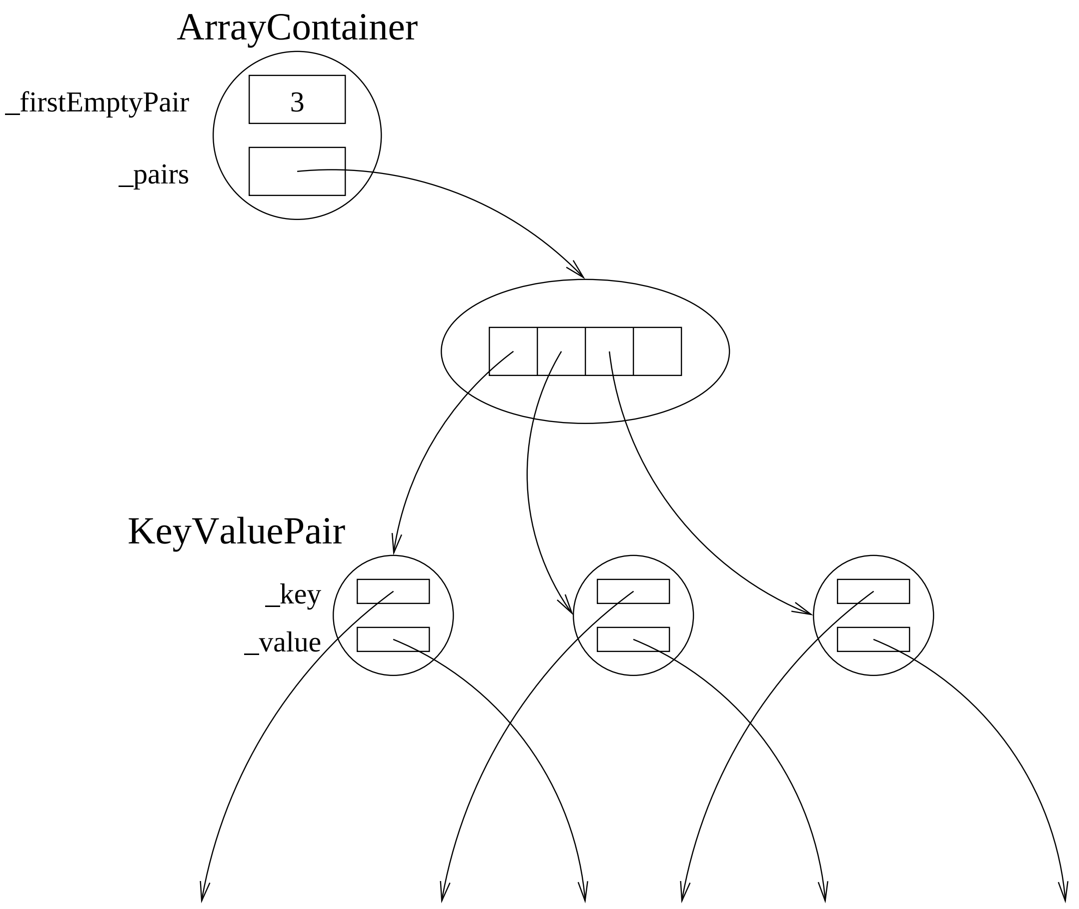
ArrayContainer

```
public class ArrayContainer implements IContainer {
    private int            _firstEmptyPair = 0;
    private KeyValuePair[] _pairs = new KeyValuePair[1];

    public Object find(Object key)
    {
        int i;

        for (i = 0; i < _firstEmptyPair; i++)
            if (_pairs[i].getKey().equals(key))
                return _pairs[i].getValue();

        return null;
    }
    . . .
```

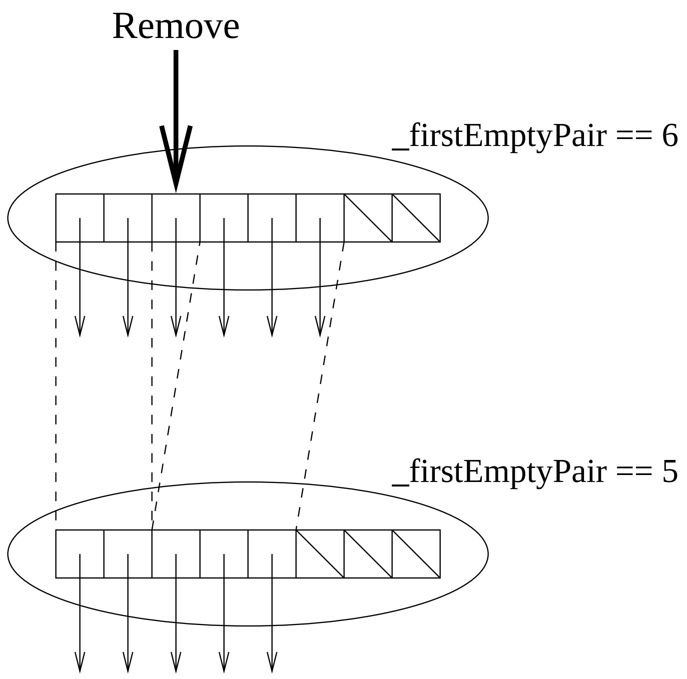
`ArrayContainer` (cont.)



ArrayContainer **(cont.)**

```
public Object remove(Object key)
{
    int i;

    for (i = 0; i < _firstEmptyPair; i++)
        if (_pairs[i].getKey().equals(key)) {
            Object value = _pairs[i].getValue();

            for (_firstEmptyPair--; i < _firstEmptyPair; i++)
                _pairs[i] = _pairs[i + 1];
            _pairs[i] = null;

            return value;
        }
    return null;
}
```

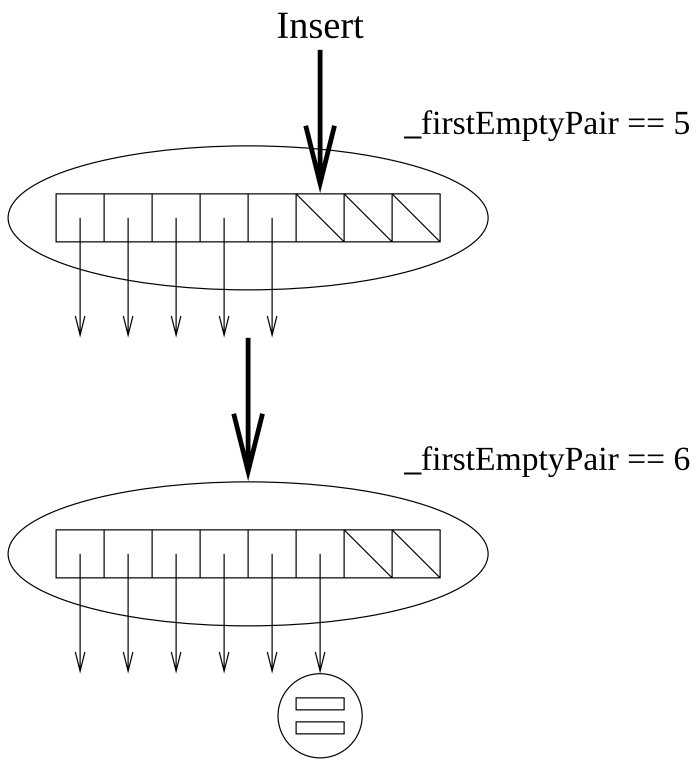
ArrayContainer (cont.)



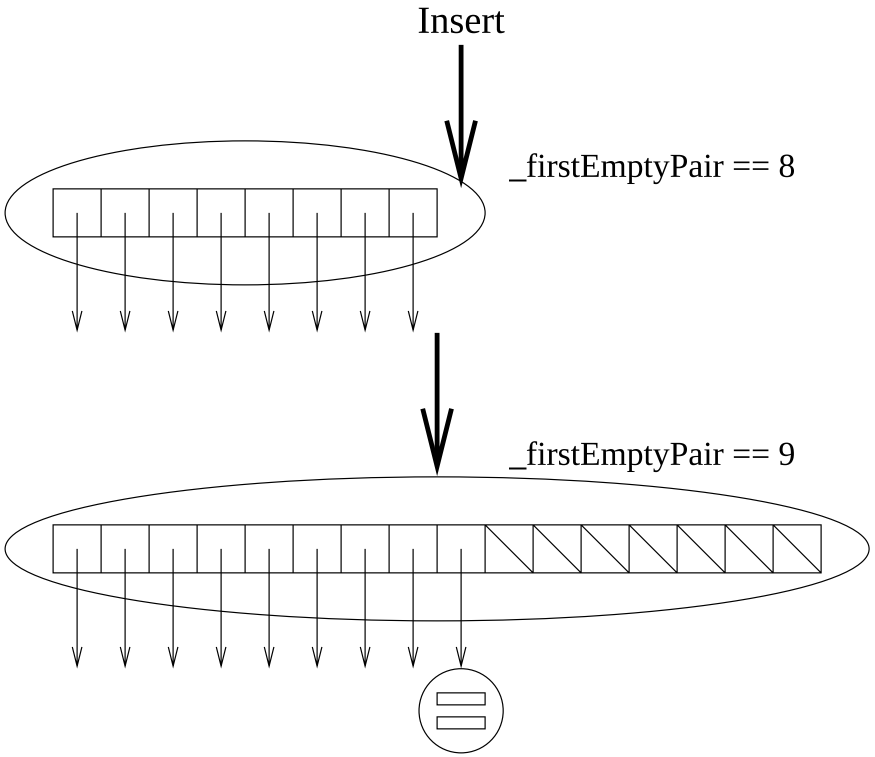
ArrayContainer (cont.)

```
public void insert(Object key, Object value)
{
    if (_firstEmptyPair == _pairs.length) {
        int i;

        KeyValuePair[] newPairs =
            new KeyValuePair[2*_pairs.length];

        for (i = 0; i < _pairs.length; i++)
            newPairs[i] = _pairs[i];

        _pairs = newPairs;
    }
    _pairs[_firstEmptyPair] = new KeyValuePair(key, value);
    _firstEmptyPair++;
}
```

`ArrayContainer` **(cont.)**



`ArrayContainer` (cont.)



ArrayContainer **(cont.)**

```
public Enumeration enumeration()
{
    return new Enumeration() {
        private int _nextPair = 0;

        public boolean hasMoreElements()
        {
            return _nextPair < _pairs.length;
        }

        public Object nextElement()
        {
            return _pairs[_nextPair++];
        }
    };
}
```