

Igor Rožanc

Osnove algoritmov in podatkovnih struktur I (OAPS I)

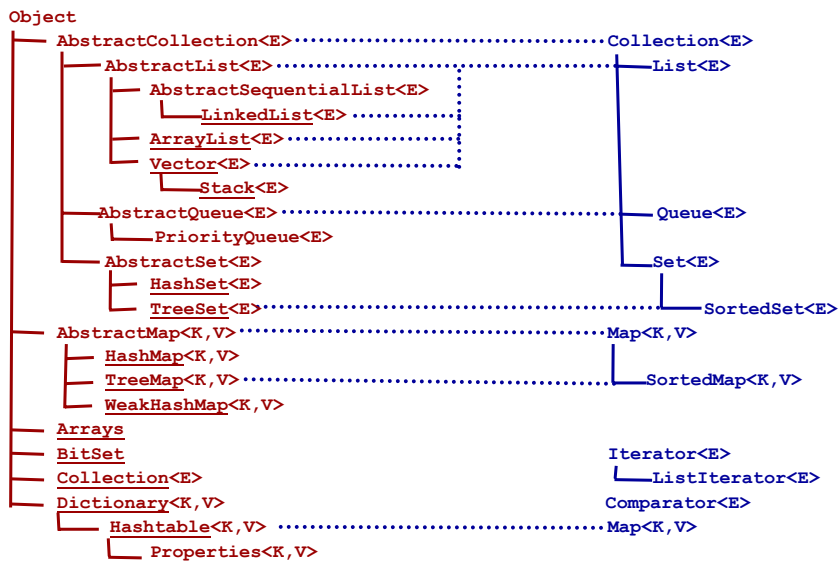
2. letnik, VSP Računalništvo in informatika, vse smeri

PROSOJNICE ZA 11. PREDAVANJA (14.12.2006)

Študijsko leto 2006/07

Collections Framework

62



Definiran v paketu java.util

Zahteva implementacijo naslednjih metod:

- `public boolean add(E);`
- `public boolean addAll(Collection<? extends E>);`
- `public void clear();`
- `public boolean contains(Object);`
- `public boolean containsAll(Collection<?>);`
- `public boolean equals(Object);`
- `public int hashCode();`
- `public boolean isEmpty();`

- `public abstract Iterator<E> iterator();`
- `public boolean remove(Object);`
- `public boolean removeAll(Collection<?>);`
- `public boolean retainAll(Collection<?>);`
- `public abstract int size();`
- `public Object[] toArray();`
- `public E[] toArray(E[]);`
- `public String toString();`

Delna implementacija vmesnika `Collection<E>`

- Implementira vse, kar se da implementirati, ne da bi poznali pomnilniško strukturo za predstavitev zbirke
- Manjka implementacija metod `equals()` in `hashCode()`
 - metode se podedujejo iz razreda `Object`
 - zanašamo se, da te metode redefinirajo podrazredi
- Po drugi strani razred redefinira metodo `toString()`
- Nekatere metode uporabljajo metodi `iterator()` in `size()`, ki sta še vedno abstraktni

Primer 1: metoda `toString()`

Primer 2: metoda `isEmpty()`

Poljubna zbirka objektov, ki lahko vsebuje duplikate

`bag` = "torba" = multiset = množica z duplikati

Prikaz razširitve razreda `AbstractCollection`

`Bag` bo sprva razvit na negeneričen način

Postopen prikaz vsebine razreda `Bag`

Za predstavitev objektov bomo uporabili tabelo:

```
import java.util.*;

public class Bag extends AbstractCollection
{
    private Object[] objects;
    private int size = 0;
    private static final int CAPACITY = 16;
```

Metode:

- `private void resize()`
- `public Bag()`
- `public Bag(int)`
- `public Bag(Object[])`
- `public Bag(Collection)`
- `public boolean add(Object)`
- `public boolean addAll(Collection)`
- `public void clear()`
- `public boolean contains(Object)`
- `public boolean containsAll(Collection)`

- `private static int frequency(Collection, Object)`
- `public boolean equals(Object)`
- `public int hashCode()`
- `public boolean isEmpty()`

Realizacija iteratorja za razred Bag:

- potrebujemo metodo `public Iterator iterator()` ;
- vmesnik predpisuje 3 metode: `hasNext()`, `next()`, `remove()`

1. možnost:

- deklariramo razred, ki implementira vmesnik `Iterator`
- na podlagi tega lahko generiramo objekt tipa `Iterator`

```
private class BagIterator implements Iterator
{   public boolean hasNext()
    public Object next()
    public void remove()
}
public Iterator iterator()
{   return new BagIterator(); }
```

2. možnost:

- anonimni notranji razred (anonymous inner class)

```
public Iterator iterator()
{   return new Iterator()
    {   public boolean hasNext()
        public Object next()
        public void remove()
    }
}
```

Preostale metode:

- `public boolean remove(Object)`
- `public boolean removeAll(Object)`
- `public boolean removeAll(Collection)`
- `public boolean retainAll(Collection)`
- `public int size()`
- `public Object[] toArray()`
- `public Object[] toArray(Object[])`
- `public String toString()`

Primer:

- `razred Bag.java`
- `razred TestBag.java`

Sprememba v generični razred Bag<E> :

- glava razreda
- notranja podatkovna struktura – še vedno tabela objektov:
`E[] contents = (E[])new Object[CAPACITY];`
- uporaba `AbstractCollection<E>`
- spremembe argumentov metod (ne povsod!)
- spremembe iteratorja

Primer:

- razred `BagG.java`
- razred `TestBagG.java`