Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Igor Rožanc

Osnove algoritmov in podatkovnih struktur I

(OAPS I)

2. letnik, VSP Računalništvo in informatika, vse smeri

PROSOJNICE ZA 12. PREDAVANJA (21.12.2006)

Študijsko leto 2006/07

Seznam (`List<E>`)

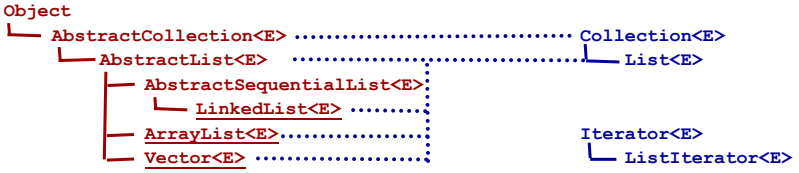
Podatkovna struktura predstavlja splošno zaporedje objektov:

- pojem pozicije
- dodajanje in izločanje objektov v predvidenem času

Primer: kompozicija vagonov ...

`Collection Framework` vsebuje:

- vmesnik **`List<E>`**
- abstraktna razreda: **`AbstractList<E>`**, **`AbstractSequentialList<E>`**
- več realiziranih razredov: **`LinkedList<E>`**, **`ArrayList<E>`**, **`Vector<E>`**
- iterator **`ListIterator<E>`**



Seznam (vmesnik `List<E>`) 73

```
public interface List<E> extends Collection<E>
{
    boolean add(E);
    add (int, E);
    boolean addAll(Collection);
    boolean addAll(int, Collection);
    clear();
    boolean contains(E);
    boolean containsAll(Collection);
    boolean equals(E);
    E get(int);
    int hashCode();
    int indexOf(E);
    boolean isEmpty();
```


Seznam (vmesnik `List<E>`) 74

```
    Iterator<E> iterator();
    int lastIndexOf(E);
    ListIterator listIterator<E>();
    ListIterator listIterator<E>(int);
    boolean remove(E);
    E remove(int);
    boolean removeAll(Collection);
    boolean retainAll(Collection);
    E set(int, E);
    int size();
    List subList(int, int);
    Object[] toArray();
    E[] toArray(E[]);
}
```

Seznam (abstraktni razred `AbstractList<E>`) 75

Dve realizaciji:

			get() set()	add() remove()
AbstractList	ArrayList	Tabela objektov	O(1)	O(n)
AbstractSequentialList	LinkedList	Povezan seznam objektov	O(n)	O(1)

```
public abstract class AbstractList<E> extends AbstractCollection<
      implements List<E>
{  (protected) AbstractList<E>()
   boolean add(E); //opcijsko
   add (int, E); // opcijsko
   boolean addAll(Collection);
   boolean addAll(int, Collection);
   clear();
   //boolean contains(E);
   //boolean containsAll(Collection);
   boolean equals(E);
   (abstract) E get(int);
```


Seznam (abstraktni razred `AbstractList<E>`) 76

```
   int hashCode();
   int indexOf(E);
   //boolean isEmpty();
   Iterator<E> iterator();
   int lastIndexOf(E);
   ListIterator<E> listIterator();
   ListIterator<E> listIterator(int);
   //boolean remove(E);
   E remove(int); // opcijsko
   removeRange(int, int);
   //boolean removeAll(Collection);
   //boolean retainAll(Collection);
   E set(int, E); // opcijsko
   //int size();
   List subList(int, int);
   //Object[] toArray();
   //E[] toArray(E[]);
}
```

Seznam (ab.razred AbstractSequentialList<E>) 77

```
public abstract class AbstractSequentialList<E> extends
    AbstractList<E>
{  (protected) AbstractSequentialList<E>()
   boolean add(E);
   boolean addAll(Collection);
   boolean addAll(int, Collection);
   E get(int);
   Iterator<E> iterator();
   int lastIndexOf(E);
   (abstract) ListIterator<E> listIterator();
   E remove(int);
   E set(int, E);
   (abstract) int size();
}
```


Seznam (razred ListIterator<E>) 78

Dvosmerni iterator

```
public interface ListIterator<E> extends Iterator<E>
{
   boolean hasNext();
   E next();
   remove();
   boolean hasPrevious();
   E previous();
   int nextIndex();
   int previousIndex();
   add(E);
   set(E);
}
```

Seznam (razred ArrayList<E>)


```
public class ArrayList<E> extends AbstractList<E>
      implements List<E>
{
   //vse metode iz vmesnika List ter dodatno:
   ArrayList<E>();
   ArrayList<E>(int);
   ArrayList<E>(Collection);

   ensureCapacity(int);
   trimToSize();
}
```


Seznam (razred LinkedList<E>)


```
public class LinkedList<E> extends AbstractSequentialList<E>
      implements List<E>
{  //vse metode iz vmesnika List ter dodatno:
   addFirst(E);
   addLast(E);
   E getFirst();
   E getLast();
   // manjka metoda get()
   LinkedList<E>();
   LinkedList<E>(Collection);
   boolean remove(E);
   E removeFirst();
   E removeLast();
}
```

Primer: Seznam.java

Vrsta (`Queue<E>`) 81

Podatkovna struktura, ki deluje po principu FIFO

Primer: vrsta pred blagajno v kinu ...

`Collection Framework` vsebuje nekoliko drugačen vmesnik in razred za prioritetno vrsto, zato zastavimo **svojo hierarhijo**

Sprva razvijemo **negenerične vmesnik in razrede**:

Izhajamo iz vmesnika **`Collection`**

- **3 deli:**
  - vmesnik **`Queue`**
  - abstraktni razred **`AbstractQueue`**
  - dve izvedbi: razreda **`ArrayQueue`** in **`LinkedQueue`**

Vrsta (vmesnik `Queue` in razred `AbstractQueue`) 82

- od vmesnika **`Collection`** podeduje 15 metod
- doda 4 operacije za delo z vrsto:
  - `Object dequeue()`
  - `Object enqueue(Object)`
  - `Object getBack()`
  - `Object getFront()`

Prikaz delovanja vrste ...

Deklaracija vmesnika `Queue` ...

Abstraktni razred `AbstractQueue`

- razširitev razreda **`AbstractCollection`**
- implementira vmesnik **`Queue`**

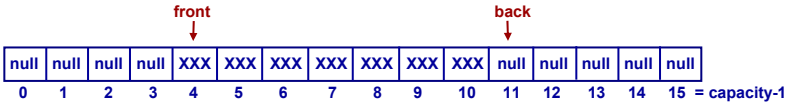
Deklaracija razreda `AbstractCollection`

Vrsta (razreda `ArrayQueue`) 83

Realizacija s tabelo objektov

Razširitev razreda **`AbstractQueue`**

Predstavitev vrste:



Deklaracija razreda `ArrayQueue`

Primer:

- **vmesnik `Queue.java`**
- **abstraktni razred `AbstractQueue.java`**
- **razred `ArrayQueue.java`**
- **razred `TestArrayQueue.java`**

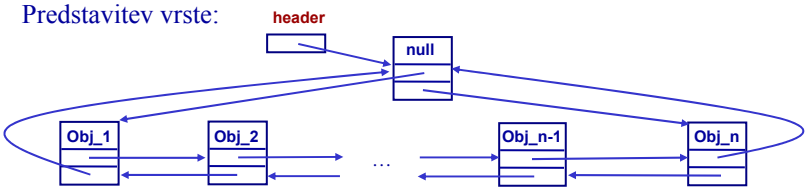
Vrsta (razred `LinkedQueue`) 84

Razširitev razreda **`AbstractQueue`**

Realizacija z **dvosmernimi seznami**: zaporedje elementov, ki so med seboj povezani v obeh smereh

Pojem slepega elementa ...

Predstavitev vrste:



Deklaracija razreda `LinkedQueue`

Primer:

- **razred `LinkedQueue.java`**
- **razred `TestLinkedQueue.java`**

Primer: **QueueG.java**

AbstractQueueG.java

ArrayQueueG.java

TestArrayQueueG.java