

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 9. PREDAVANJA (30.11.2006)**

Študijsko leto 2006/07

## Shranjevanje objektov v vektor:

- `boolean add(E) ;`
- `add(int, E) ;`
- `addElement(E) ;`
- `E set (int, E) ;`
- `setElementAt(E, int) ;`
- `boolean addAll(Collection) ;`
- `boolean addAll(int, Collection) ;`

## Branje podatkov iz vektorja:

- `E get(int) ;`
- `E elementAt(int) ;`
- `E firstElement() ;`
- `E lastElement() ;`

Konverzija tipa **ni potrebna**

**Primer branja, če vektor hrani objekte tipa `Delavec`:**

```
Delavec d= (Delavec) v.get(4);  
...  
Delavec d= (Delavec) v.firstElement();
```

**Primer obdelave vseh elementov vektorja objektov tipa `Delavec`:**

```
Delavec d;  
for (int poz=0; poz< v.size(); ++poz)  
{  
    d= (Delavec) v.get(poz);  
    // nadaljna obdelava objekta d  
}
```

## Prepis elementov iz vektorja v tabelo

- uporaba vektorja zahteva dodatno režijo, zato včasih smiselno

Metodi:

- `E[] toArray()` ;
- `E[] toArray(E[])` ;

## Odstranjevanje elementov iz vektorja:

Pri odstranjevanju se kapaciteta ne zmanjšuje, zmanjšuje se samo velikost.

- `E remove(int)` ;
- `boolean remove (E)` ;
- `removeElementAt(int)` ;
- `clear()` ;
- `boolean removeAll(Collection)` ;

## Iskanje elementov v vektorju:

- `int indexOf(E);`
- `int indexOf(E, int);`
- `int lastIndexOf(E);`
- `int lastIndexOf(E, int);`

## Primer: vsa nastopanja objekta `d` v vektorju:

```
...
int poz=0;
while (poz<v.size() && poz<=0)
{
    poz = v.indexOf(d,poz);
        if (poz != -1)
        {
            // obdelava objekta na poziciji poz
            ++poz;
        }
}
```

## Iterator

V vseh razredih, ki implementirajo vmesnik `Collection<E>`, obstaja objekt `Iterator<E>`, ki omogoča sprehajanje po elementih zbirke.

Razred `Vector<E>` implementira vmesnik `Collection<E>`.

`Iterator` ima konstruktor in 3 metode:

- `Iterator<E> () ;`
- `Object next () ;`
- `boolean hasNext () ;`
- `remove () ;`

## Primer: Obhod s pomočjo iteratorja:

```
Delavec d;  
Iterator<Delavec> it=v.iterator();  
while(it.hasNext())  
{  
    d=it.next();  
    //se naprej stavki za obdelavo tega elementa  
}
```

## Primer: Vektor.java

VektorG.java

Podatkovna struktura, ki deluje po principu LIFO

Primer: zlaganje kovancev ...

Razred `Stack<E>` (razširitev razreda `Vector<E>`)

Operacije za delo s skladom:

- `empty ()`
- `peek ()`
- `pop ()`
- `push (E)`
- `search (E)`

Deklaracija razreda `Stack` ...