

Exempel IntContainer – del 1+2

Specifikation av klassen IntContainer:

```
/** Skapar en "heltalsbehållare" med plats för  
    maximalt size heltal */  
IntContainer(int size);  
  
/** Läger in heltalet n i behållaren,  
    som inte får vara full */  
void insertInt(int n);  
  
/** Tar bort (ett) heltal n från behållaren,  
    om det ej finns något händer inget */  
void removeInt(int n);  
  
/** Kontrollerar om heltalet n finns i behållaren */  
boolean exist(int n);  
  
/** Kontrollerar om behållaren är full */  
boolean full();
```

```
// Implementation 1  
public class IntContainer {  
    private int[] v;          // Behållaren, osorterad  
    private int nbr;          // Antal tal just nu  
  
    /** Skapar en "heltalsbehållare" med plats för  
        maximalt size heltal */  
    public IntContainer(int size) {  
        v = new int[size];  
        nbr = 0;              // Så är det redan  
    }  
  
    /** Läger in heltalet n i behållaren,  
        som inte får vara full */  
    public void insertInt(int n) {  
        v[nbr] = n;  
        nbr++;  
    }  
  
    /** Tar bort (ett) heltal n från behållaren,  
        om det ej finns något händer inget */  
    public void removeInt(int n) {  
        int pos = 0;  
        while (pos < nbr && v[pos] != n) {  
            pos++;  
        }  
        if (pos < nbr) {  
            v[pos] = v[nbr-1];  
            nbr--;  
        }  
    }  
  
    /** Kontrollerar om heltalet n finns i behållaren */  
    public boolean exist(int n) {  
        int pos = 0;  
        while (pos < nbr && v[pos] != n) {  
            pos++;  
        }  
        return pos < nbr;  
    }  
  
    /** Kontrollerar om behållaren är full */  
    public boolean full() {  
        return nbr == v.length;  
    }  
}
```

```
//Implementation 2
public class IntContainer {
    private int[] v;          // Behållaren, sorterad
    private int nbr;          // Antal tal just nu
    private int size;         // Storlek

    /** Skapar en "heltalsbehållare" med plats för
        maximalt size heltal */
    public IntContainer(int size) {
        v = new int[size];
        this.size = size;      // size == v.length
        nbr = 0;               // Så är det redan
    }

    /** Letar upp index för heltalet n i v */
    private int find(int n) {
        int low = 0, high = nbr-1;
        while (low <= high) {
            int mid = (low+high)/2;
            if (n == v[mid]) {
                return mid;    //found
            } else if (n < v[mid]) {
                high = mid-1;
            } else {
                low = mid+1;
            }
        }
        return low;           //om n ej finns så anger platsen
    }

    /** Lägger in heltalet n i behållaren,
        som inte får vara full */
    public void insertInt(int n) {
        int pos = find(n);
        for (int k=nbr; k>pos; k--) {
            v[k] = v[k-1];
        }
        v[pos] = n;
        nbr++;
    }
}
```

```
/** Tar bort (ett) heltal n från behållaren,
    om det ej finns något händer inget */
public void removeInt(int n) {
    int pos = find(n);
    if (pos < nbr && v[pos] == n) {
        for (int k=pos; k+1<nbr; k++) {
            v[k] = v[k+1];
        }
        nbr--;
    }
}

/** Kontrollerar om heltalet n finns i behållaren */
public boolean exist(int n) {
    int pos = find(n);
    if (pos < nbr && v[pos] == n) {
        return true;
    } else {
        return false;
    }
}

/** Kontrollerar om behållaren är full */
public boolean full() {
    return nbr == size;
}
}
```

Exempel på användning av klassen IntContainer:

```
// Skapa en börs med plats för max 50 mynt
IntContainer wallet = new IntContainer(50);
...
if (!wallet.full()) {           // Lägg en enkrona i börsen
    wallet.insertInt(100);      // om det finns plats
}
...
if (wallet.exist(100)) {
    // Det finns en enkrona i börsen
}
...
int coins = 0;
while (wallet.exist(100)) {
    wallet.removeInt(100);
    coins++;
}
// Nu har alla enkronor (coins styck) tagits upp ur börsen
```