Exempel IntContainer – del 5 (del 3, men med ArrayList)

Specifikation av klassen IntContainer (utan maxstorlek, då behövs ej metoden full):

```
/** Skapar en ”heltalsbehållare” */
IntContainer();

/** Lägger in heltalet n i behållaren */
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);
```

```
// Implementation 5 (Implementation 3, men med ArrayList)
import java.util.ArrayList;
public class IntContainer {
      private ArrayList<Integer> v;  // Behållaren,
                                     // osorterad

      /** Skapar en ”heltalsbehållare” */
      public IntContainer() {
            v = new ArrayList<Integer>();
      }

      /** Lägger in heltalet n i behållaren */
      public void insertInt(int n) {
            v.add(Integer.valueOf(n));    //Se *) nedan
      }

      /** Tar bort (ett) heltal n från behållaren,
             om det ej finns något händer inget */
      public void removeInt(int n) {
            int pos = 0;
            while (pos < v.size() &&
                   v.get(pos).intValue() != n) {
                  pos++;
            }
            if (pos < v.size()) {
                  v.remove(pos);
            }
      }

      /** Kontrollerar om heltalet n finns i behållaren */
      public boolean exist(int n) {
            int pos = 0;
            while (pos < v.size() &&
                   v.get(pos).intValue() != n) {
                  pos++;
            }
            return pos < v.size();
      }
}
```

*) Sedan Java 9 används den statiska metoden valueOf för att skapa objekt av alla wrapper-klasser (Integer, Double, etc):

```
public static Integer valueOf(int i);
public static Double valueOf(double d);
```

Exempel IntContainer – del 6 (del 5, men med autoboxing)

Specifikation av klassen IntContainer (utan maxstorlek, då behövs ej metoden full):

```
/** Skapar en ”heltalsbehållare” */
IntContainer();

/** Lägger in heltalet n i behållaren */
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);
```

```
// Implementation 6 (Implementation 5, men med autoboxing)
import java.util.ArrayList;
public class IntContainer {
      private ArrayList<Integer> v;  // Behållaren,
                                     // osorterad

      /** Skapar en ”heltalsbehållare” */
      public IntContainer() {
            v = new ArrayList<Integer>();
      }

      /** Lägger in heltalet n i behållaren */
      public void insertInt(int n) {
            v.add(n);
      }

      /** Tar bort (ett) heltal n från behållaren,
             om det ej finns något händer inget */
      public void removeInt(int n) {
            int pos = 0;
            while (pos < v.size() && v.get(pos) != n) {
                  pos++;
            }
            if (pos < v.size()) {
                  v.remove(pos);
            }
      }

      /** Kontrollerar om heltalet n finns i behållaren */
      public boolean exist(int n) {
            int pos = 0;
            while (pos < v.size() && v.get(pos) != n) {
                  pos++;
            }
            return pos < v.size();
      }
}
```