

Removing Items from an ArrayList

Looping through an ArrayList and removing multiple items

Alwin Tareen

ArrayList Item Removal

- ▶ Care must be taken when looping across an ArrayList and removing items.
- ▶ Consider the following algorithm:

```
Traverse the ArrayList from beginning to end
  if the current number is even:
    remove the number
  end if
```

ArrayList Item Removal

- ▶ Removing single items, or items that are not neighbours from an `ArrayList` is simple.
- ▶ However, care must be taken when removing two items that are adjacent to each other in an `ArrayList`.
- ▶ Consider the following example, in which we attempt to remove all of the even numbers from an `ArrayList`.

Removing Even Integers from an ArrayList

```
import java.util.*;
public class RemoveIntegers
{
    public static void main(String[] args)
    {
        ArrayList<Integer> nums = new ArrayList<Integer>();
        nums.add(9);
        nums.add(3);
        nums.add(5);
        nums.add(2);
        nums.add(6);
        nums.add(4);
        nums.add(8);
        nums.add(1);

        System.out.println(nums);
    }
}
```

Removing Even Integers from an ArrayList

```
// remove even integers, incorrect version
for (int i = 0; i < nums.size(); i++)
{
    if (nums.get(i)%2 == 0)
    {
        nums.remove(i);
    }
}

System.out.println(nums);
}
}
```

Removing Even Integers from an ArrayList

- ▶ The output of the preceding program is:

```
[9, 3, 5, 2, 6, 4, 8, 1]
```

```
[9, 3, 5, 6, 8, 1]
```

- ▶ As you can see from the output, two of the even numbers were not removed.
- ▶ Specifically, the 6 and the 8 remain in the ArrayList.

Removing Even Integers from an ArrayList

- ▶ Let's examine the point where it encounters two even numbers that are adjacent to each other.
- ▶ Assume that the Java program has already advanced to index 3, where it encounters the first even number, which is 2.

number	9	3	5	2	6	4	8	1
index	0	1	2	3	4	5	6	7

- ▶ After calling `num.remove(3)`, the even number 2 is removed.
- ▶ The `ArrayList`'s size is decreased by one, and the loop control variable `i` has advanced to index 4.

Removing Even Integers from an ArrayList

- ▶ Now, the ArrayList appears as follows:

number	9	3	5	6	4	8	1
index	0	1	2	3	4	5	6

- ▶ At this point, the problem occurs. When the number is removed from the ArrayList, the ArrayList is adjusted so that every item below the deleted item is moved up by one index.
- ▶ However, the loop control variable `i` of the `for` loop has also advanced to the next index, which is 4.
- ▶ This causes the number that was adjacent to the deleted number to be skipped.
- ▶ In this example, the number 6 is passed over.

Removing Even Integers from an ArrayList

- ▶ To overcome this problem, simply iterate backwards through the ArrayList.

```
for (int i = nums.size()-1; i >= 0; i--)  
{  
    if (nums.get(i)%2 == 0)  
    {  
        nums.remove(i);  
    }  
}
```

Removing Items from an ArrayList: End of Notes