Advanced Programme Design Coursework 4, Stian A. Johansen

Myset Iterator (implemented in the Myset.java class)

```
    @Override
    public Iterator<E> iterator() {
        return new MyIterator();
    }

    private class MyIterator implements Iterator<E> {
        private int index = 0;

        @Override
        public boolean hasNext() {
            if (index < els.size()) {
                return true;
            } else {
                return false;
            }
        }

        @Override
        public E next() {
            if (hasNext()) {
                return els.get(index++);
            } else {
                throw new NoSuchElementException();
            }
        }

        @Override
        public void remove() {
            throw new UnsupportedOperationException("Not supported yet.");
        }
    }
```

Filter.java

```
package johansen.stian.apd.cw4;

import java.util.Iterator;
import java.util.NoSuchElementException;

public class Filter<E> implements Iterator<E> {
    private E nextElement;
    private Iterator<E> iterator;
    private Checker<E> checker;

    public Filter(Iterator<E> iterator, Checker<E> checker) {
        this.iterator = iterator;
        this.checker = checker;
        getNextElement();
    }

    private void getNextElement() {
        if (iterator.hasNext()) {
            nextElement = iterator.next();
            if (!checker.check(nextElement)) {
                getNextElement();
            }
        } else {
            nextElement = null;
        }
    }

    @Override
    public boolean hasNext() {
        return nextElement != null;
    }

    @Override
    public E next() {
        if (hasNext()) {
            E elm = nextElement;
            getNextElement();
            return elm;
        } else {
            throw new NoSuchElementException();
        }
    }

    @Override
    public void remove() {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

Usages of Filter.java
Using the examples for the coursework documents – Odd numbers and strings longer than an Integer.

Checker classes:
Odd number checker (implemented in the UseIntSets2 class):

```
private static class Odd implements Checker<Integer>{
        @Override
        public boolean check(Integer obj) {
            return (obj.intValue() % 2 != 0);
        }
 }
```

LongerThan checker (implemented in the UseStringSets2 class):

```
private static class LongerThan implements Checker<String>{
 private int value;
     private LongerThan(int n){
         value = n;
     }
        @Override
        public boolean check(String obj) {
            return (obj.length() > value);
        }
 }
```

Test 1 using UseStringSets2 (using value from the coursework document):

```
: i twas brillig and the slithy toves did gyre gimble in wabe
: p
{twas,brillig,and,the,slithy,toves,did,gyre,gimble,in,wabe}
: q

Enter a string length: 3
Set elements length greater than 3:

  twas
  brillig
  slithy
  toves
  gyre
  gimble
  wabe
```

Test 2 using UseStringSets2 (using a value that nothing matches):

```
: i a ab abc
: p
{a,ab,abc}
: q

Enter a string length: 5
Set elements length greater than 5:
```

Test 3 using UseStringSets2 (with no input):
: p
{}
: q

Enter a string length: 3
Set elements length greater than 3:

Test 4 using UseStringSets2 (with removing an element before iteration):
: i Abe Gabriel George Bean
: p
{Abe,Gabriel,George,Bean}
: r George
: p
{Abe,Gabriel,Bean}
: q

Enter a string length: 3
Set elements length greater than 3:

Gabriel
Bean

Test 1 using UseIntSets2:
: i 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
: p
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}
: q
Odd set elements are:

1
3
5
7
9
11
13
15

Test 2 using UseIntSets2 (with no Odd numbers present):
: i 2 4 6 8 10
: p
{2,4,6,8,10}
: q
Odd set elements are:

Test 3 using UseIntSets2 (with no input to test):
: p
{}
: q
Odd set elements are: