Iterable/Iterator

Smart måte å "bla" gjennom Collections-objekt (liste) på

Iterator er et interface som har 2 viktige metoder

- hasNext()
- next()

Iterable er et interface som har 1 viktig metode

- iterator() som brukes vanligvis på en collection
- Lager en ny iterator som begynner på begynnelsen

Example lager en iterator:

```
public Iterable<String> getAllWords(){
    return this.wordList; // wordList is ArrayList<String>
}
```

Da kan man bruke den:

```
public void printAllWords(){
    for (String word : this.getAllWords()){
        System.out.println(word);
    }
}
```

Vi kan lage en custom Iterator!

```
public class RangeIterator implements Iterator<Integer> {
    // Et Iterator -objekt kan «bla gjennom» en samling elementer én gang.

    private Range range;
    private int current;

    public RangeIterator(Range range) {
        this.range = range;
        current = range.getStart();
    }

    @Override
    public boolean hasNext() {
        if(range.getStep()>=0) {
            return current<=range.getSlutt();
        }
```

```
            else {
                return current>=range.getSlutt();
            }
        }

        @Override
        public Integer next() {
            int value = current;
            current = current+range.getStep();
            return value;
        }
    }

    public class Range implements Iterable<Integer>{
        // Et Iterable -objekt har en metode for å lage nye Iterator-objekter.
        // Et objekt som er Iterable kan man derfor bla igjennom flere ganger.

        private int start;
        private int slutt;
        private int step;

        public int getStart() { return start; }

        public int getSlutt() { return slutt; }

        public int getStep() { return step; }

        public Range(int start, int slutt, int step){
            this.start = start; this.slutt = slutt; this.step = step;
        }
        public Range(int start, int slutt){
            this(start,slutt,1);
        }
        @Override
        public Iterator<Integer> iterator() {
            return new RangeIterator(this);
        }
        @Override
        public String toString() {
            return "From: "+start+" to: "+slutt+" with steps of "+step;
        }
        public static Iterable<Integer> range(int start, int slutt, int step){
            return new Range(start,slutt,step);
        }
        public static Iterable<Integer> range(int start, int slutt){
            return new Range(start,slutt);
        }
    }
```