Iterator

```java
public interface Iterator {
        public Object getNext();
        public boolean hasNext();
}
public interface IterableCollection {
        public Iterator createIterator();
}
import java.util.ArrayList;
public class Collection implements IterableCollection {
        public ArrayList<Object> collection = new ArrayList<>();
        public void addToCollection(Object obj) {
                collection.add(obj);
        }
        public int size() {
                return collection.size();
        }
        public Object getFromCollection(int i) {
                return collection.get(i);
        }
        @Override
        public Iterator createIterator() {
                return new ConcreteIterator(this);
        }
}
public class ConcreteIterator implements Iterator {
        int i;
        Collection collection;
        public ConcreteIterator(Collection collection) {
                this.collection = collection;
        }
        @Override
        public Object getNext() {
                if(hasNext()) {
                        return collection.getFromCollection(i++);
                }
                return null;
        }
        @Override
        public boolean hasNext() {
                return i < collection.size();
        }
}
public class Main {
        public static void main(String[] args) {
                new Main().go();
        }
        void go() {
                Collection collection = new Collection();
                collection.addToCollection("Noumekpo");
                collection.addToCollection("Germain");
                collection.addToCollection(22);
                for(Iterator iter = collection.createIterator(); iter.hasNext();) {
                        Object obj = iter.getNext();
                        System.out.println("[Collection Elemenent]: " + obj);
                }
        }
}
```