An ArrayList Implementation

A version of the author's code

The MyArrayList class

```
public class MyArrayList<AnyType> implements
Iterable<AnyType>
{

Fields

Public and Private Methods
}
```

Fields

```
private static final int DEFAULT_CAPACITY = 10;

public AnyType [ ] theItems;
private int theSize;
```

Constructor

```
public MyArrayList( )
{
    clear( );
}
```

```
public void clear( )
{
    theSize = 0;
    ensureCapacity( DEFAULT_CAPACITY );
}
```

```
public void ensureCapacity( int newCapacity )
{
    if( newCapacity < theSize )
        return;

    AnyType [ ] old = theItems;
    theItems = (AnyType []) new Object[ newCapacity ];
    for( int i = 0; i < size( ); i++ )
        theItems[ i ] = old[ i ];
}
```

```
public int size( )
{
    return theSize;
}
```

```
public boolean isEmpty( )
{
    return size( ) == 0;
}
```

```
    public AnyType get( int idx )
    {
        if( idx < 0 || idx >= size( ) )
            throw new ArrayIndexOutOfBoundsException( "Index "
+ idx + "; size " + size( ) );
        return theItems[ idx ];
    }
    public AnyType set( int idx, AnyType newVal )
    {
        if( idx < 0 || idx >= size( ) )
            throw new ArrayIndexOutOfBoundsException( "Index "
+ idx + "; size " + size( ) );
        AnyType old = theItems[ idx ];
        theItems[ idx ] = newVal;

        return old;
    }
```

```
public void add( int idx, AnyType x )
{
    if( theItems.length == size( ) )
        ensureCapacity( size( ) * 2 + 1 );

    for( int i = theSize; i > idx; i-- )
        theItems[ i ] = theItems[ i - 1 ];

    theItems[ idx ] = x;
    theSize++;
}
```

```
public boolean add( AnyType x )
{
    add( size( ), x );
    return true;
}
```

```
public AnyType remove( int idx )
{
    AnyType removedItem = theItems[ idx ];

    for( int i = idx; i < size( ) - 1; i++ )
        theItems[ i ] = theItems[ i + 1 ];
    theSize--;

    return removedItem;
}
```

```
public java.util.Iterator<AnyType> iterator( )
{
    return new ArrayListIterator<AnyType>( this );
}
```

The ArrayListIterator Class

```
class ArrayListIterator<AnyType> implements java.util.
Iterator<AnyType>
{

    Field

    Methods


}
```

Fields

```
private int current ;
private boolean okToRemove;
private MyArrayList<AnyType> theList;
```

Constructor

```
public ArrayListIterator( MyArrayList<AnyType> list )
{
    theList = list;
    current = 0;
    okToRemove = false;
}
```

Methods

```
public boolean hasNext( )
{
    return current < theList.size( );
}
```

```
public AnyType next( )
{
    if( !hasNext( ) )
        throw new java.util.NoSuchElementException( );

    okToRemove = true;
    return theList.theItems[ current++ ];
}
```

```
public void remove( )
{
    if( !okToRemove )
        throw new IllegalStateException( );

    theList.remove( --current );
    okToRemove = false;
}
```

Code to Test the ArrayList

```
class TestArrayList
{
    public static void main( String [ ] args )
    {

Code


    }
}
```

Code

```
MyArrayList<Integer> lst = new MyArrayList<Integer>( );

for( int i = 0; i < 10; i++ )
    lst.add( i );
for( int i = 20; i < 30; i++ )
    lst.add( 0, i );

lst.remove( 0 );
lst.remove( lst.size( ) - 1 );
```

Code

```
java.util.Iterator<Integer> itr = lst.iterator();

while (itr.hasNext()) {
    if (  (itr.next() % 2) == 0 ) itr.remove();
}
```