

Programme Java semblable a celui pour les sequences en MPD
=====

----- Fichier OpUnaire.java -----
interface OpUnaire {
 int evaluer(int v);
}

----- Fichier OpBinaire.java -----
interface OpBinaire {
 int evaluer(int v1, int v2);
}

----- Fichier PlusN.java -----
class PlusN implements OpUnaire {
 private int n;

 public PlusN(int n)
 { this.n = n; }

 public int evaluer(int v)
 { return(v + n); }
}

----- Fichier Plus.java -----
class Plus implements OpBinaire {
 public int evaluer(int v1, int v2)
 { return(v1 + v2); }
}

```
----- Fichier Sequence.java -----
class Sequence {
    private int[] elems;
    private int lng;      // long est un mot reserve en Java :(

    Sequence( int[] elems )
    { this.elems = elems; lng = elems.length; }

    public int longueur()
    { return( lng ); }

    public int element( int i )
    { return( elems[i] ); }

    public void obtenirElements( int lesElements[] )
    {
        for ( int i = 0; i < lng; i++ ) {
            lesElements[i] = elems[i];
        }
    }

    public Sequence appliquer( OpUnaire f )
    {
        int []es = new int[lng];
        for ( int i = 0; i < lng; i++ ) {
            es[i] = f.evaluer( elems[i] );
        }
        return( new Sequence(es) );
    }

    public int reduire( OpBinaire o )
    {
        int r = elems[0];
        for ( int i = 1; i < lng; i++ ) {
            r = o.evaluer( r, elems[i] );
        }
        return( r );
    }

    public Sequence tousLesPrefixes( OpBinaire o )
    {
        int []es = new int[lng];
        es[0] = elems[0];
        for ( int i = 1; i < lng; i++ ) {
            es[i] = o.evaluer( es[i-1], elems[i] );
        }
        return( new Sequence(es) );
    }
}
```

```
----- Fichier SequenceTest.java -----
import junit.framework.*; import junit.extensions.*;

public class SequenceTest extends TestCase {
    public SequenceTest( String name ) { super(name); }

    private Sequence s; private OpUnaire f; private OpBinaire o;

    protected void setUp() {
        s = new Sequence( new int[]{10, 20, 30} );
        f = new PlusN(1);
        o = new Plus();
    }

    public void testBase() {
        assertEquals( 3, s.longueur() );
        ...

        int[] r = new int[3]; s.obtenirElements( r );
        assertEquals( 10, r[0] );
        ...
    }

    public void testAppliquer() {
        Sequence r = s.appliquer( f );
        assertEquals( 11, r.element(0) );
        assertEquals( 21, r.element(1) );
        assertEquals( 31, r.element(2) );
    }

    public void testReduire() {
        assertEquals( 60, s.reduire( o ) );
    }

    public void testTousLesPrefixes() {
        Sequence r = s.tousLesPrefixes( o );
        assertEquals( 10, r.element(0) );
        assertEquals( 30, r.element(1) );
        assertEquals( 60, r.element(2) );
    }

    public static Test suite() {
        return new TestSuite(SequenceTest.class);
    }

    public static void main( String[] args ) {
        junit.textui.TestRunner.run( suite() );
    }
}
```