**Contents: pages 1,2 – Join interface and code implementations of Fold and join operations. Page 3 Methods used for testing the implementations**

**Stian Alexander Johansen**

**public** **interface** Join<T,S> {

**public** S join(T t, S s);

}

**public** **static**<T,S> S fold( ArrayList<T> collection,

Join<T,S> joiner,

S currentVal

)

{

**if**(collection.size() == 0) **return** currentVal;

**return** ( joiner.join(collection.remove(0),

*fold*(collection,joiner,currentVal)

);

}

**private** **static** **class** IntSumOfTwo **implements** Join<Integer, Integer>{

@Override

**public** Integer join(Integer t, Integer s) {

**if**(t == **null**) t = 0;

**if**(s == **null**) s = 0;

**return** t+s;

}

}

**private** **static** **class** IntLargestOfTwo **implements** Join<Integer, Integer>{

@Override

**public** Integer join(Integer t, Integer s) {

**if** (t == **null**) t = 0;

**if** (s == **null**) s = 0;

**return** t>s?t:s;

}

}

**private** **static** **class** IntAndStringLargestOfTwo **implements** Join<String, Integer>{

@Override

**public** Integer join(String t, Integer s) {

**if** (t == **null**) t = "";

**if** (s == **null**) s = 0;

**return** t.length()>s?t.length():s;

}

}

**private** **static** **class** IntListAndIntInsertInto

**implements** Join<Integer,ArrayList<Integer>>{

@Override

**public** ArrayList<Integer> join(Integer t, ArrayList<Integer> s) {

//REQUIRES: s has to be ordered {min .. max}

**if** (t == **null**) t = 0;

**if** (s == **null**) s = **new** ArrayList<Integer>();

**for**(**int** i=0;i<s.size();i++){

**if**(s.get(i)>t){

s.add(i,t);

**return** s;

}

}

**if**(s.size() == 0) s.add(t);

**return** s;

}

}

**private** **static** String testAddIntegers(ArrayList<Integer> intList){

**return**(""+Fold.*fold*(intList, **new** IntSumOfTwo(),0));

}

**private** **static** String testIntLargestOfTwo(ArrayList<Integer> intList){

**return**(""+Fold.*fold*(intList, **new** IntLargestOfTwo(), 0));

}

**private** **static** String testIntAndStringLargestOfTwo(

ArrayList<String> strList){

**return**(""+Fold.*fold*(strList,**new** IntAndStringLargestOfTwo(), 0));

}

**private** **static** String testIntListAndIntInsertInto(

ArrayList<Integer> intList){

**return** (""+

Fold.*fold*(intList,

**new** IntListAndIntInsertInto(),

**new** ArrayList<Integer>()

));

}

**OUTPUT (from printing the results from the methods used for testing)**

**List{0..5}**

**Adding integers 0..5:15**

Largest Int in a list 0..5:5

Largest String In List 'uno','quattro','dos':7

Int Insert into List And Result:[0, 1, 2, 3, 4, 5]

**List{0..50}**

**Adding integers 0..50:1275**

Largest Int in a list 0..50:50

Largest String In List 'uno','uno','dos':3

Int Insert into List And Result:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]

**List{0}**

**Adding integers 0..0:0**

Largest Int in a list 0..0:0

Largest String In List 'uno','uno','dos':3

Int Insert into List And Result:[0]