Written by Caspian Peavyhouse
CS101-02
12/4/2014

(+) Set
(-)setArray:int[]
(+)Set() (+)makeEmpty() (+)isEmpty():boolean (+)add(int) (+)remove(int) (+)elementOf(int):boolean (+)size():int (+)union(Set):Set (+)intersection(Set):Set (+)setDiference(Set):Set (+)toString():String

Legend

(+) public
(-) private
( ) package
(#) protected

Algorithm for Set:

```
Set()
        setArray <-- new int [0]



makeEmpty()
        this.setArray <--  new int[0]



isEmpty()
        if setArray.size() == 0
                return true
        else
                return false

add(int addThis)
        if ( !this.elementOf(int addThis))
                int [] tempArray <-- new int [this.size() + 1]

                for (int i = 0; i < this.size(); i++)
                        tempArray[i] = this.setArray[i]
                this.setArray <-- tempArray
                this.setArray[size() - 1] <-- addThis

                for (int run = 0; run < size() - 1; run++)
                        for (int j = 0; j < size() - 1; j++)
                if (setArray[j] > setArray[j + 1])
                    int temp <-- setArray[j]
                    this.setArray[j] <-- this.setArray[j + 1]
                    this.setArray[j + 1] <-- temp

remove(int removeThis)
    if ( this.elementOf(removeThis))
        for (int i = 0; i < this.size() - 1; i++)
            if (this.setArray[i] > removeThis)
                this.setArray[i] <-- this.setArray[i + 1]

        int [] tempArray <-- new int [this.size() - 1]
        for (int j = 0; j < tempArray.length; j++)
            tempArray[i] <-- this.setArray[i]

        this.setArray <-- tempArray

elementOf(int checkThis)
    for (int i = 0; i < this.size(); i++)
        if (this.setArray[i] == checkThis)
            return true

     // will only reach this point if the element is not in the array
    return false

size()
    return this.setArray.length


union(Set otherSet)
    Set newSet <-- new Set()
    for (int i = 0; i < this.size(); i++)
        newSet.add(this.setArray[i])

    for (int j = 0; j < otherSet.size(); j++)
        newSet.add(otherSet.setArray[j])

    return newSet


intersection(Set otherSet)
    Set newSet <-- new Set()
    int currentNum
    for (int i = 0; i < this.size(); i++)
        currentNum <-- this.setArray[i]
        if (otherSet.elementOf(currentNum))
```

```
            newSet.add(currentNum)
    return newSet


setDifference(Set otherSet)
    Set newSet <-- new Set()
    int currentNum
    for (int i = 0; i < this.size(); i++)
        currentNum <-- this.setArray[i]
        if  !(otherSet.elementOf(currentNum))
            newSet.add(currentNum)
    return newSet


toString()
    String output <-- new String("{")
    for (int i = 0; i < this.size(); i++)
        if (i == this.size() -1)
            output += "" + this.setArray[i]
        else
            output += "" + this.setArray[i] + ", "
    output += "}"
    return output
```