HW04 Code

ArrayOps:

findMissingInt:

```
public static int findMissingInt (int [] array) {
    int maxInArray = 0;
    int missingNumber = 0;
    int[] exisitingNumbers = new int[array.length + 1];
    for(int i = 0; i < array.length; i++){
        exisitingNumbers[array[i]] = 1;
    }
    for(int j = 0; j < exisitingNumbers.length; j++){
        if (exisitingNumbers[j] != 1) {
            missingNumber = j;
        }
    }
    return missingNumber;
  }
```

secondMaxValue:

```
public static int secondMaxValue(int [] array) {
        int firstMax = 0;
        int secondMax = 0;
        int firstMaxCounter = 0;
        for(int i = 0; i < array.length; i++){
            if (array[i] >= firstMax) {
                firstMax = array[i];
            }
        }

        for(int i = 0; i < array.length; i++){
            if (array[i] == firstMax) {
                firstMaxCounter++;
            }
            if ((array[i] >= secondMax) && (array[i] < firstMax)) {
                secondMax = array[i];
            }
            if (firstMaxCounter > 1) {
                secondMax = firstMax;
            }
        }
        return secondMax;
    }
```

containsTheSameElements:

```
public static boolean containsTheSameElements(int [] array1,int [] array2){
      int elementExists = 0;
      if (array2.length >= array1.length) {
         for(int i = 0; i < array2.length; i++){
            elementExists = 0;
            for(int j = 0; j < array1.length; j++){
               if (array2[i] == array1[j]) {
                  elementExists = 1;
               }
            }
            if (elementExists != 1) {
               return false;
            }
         }

         return true;
      }else{
         for(int i = 0; i < array1.length; i++){
            elementExists = 0;
            for(int j = 0; j < array2.length; j++){
               if (array1[i] == array2[j]) {
                  elementExists = 1;
               }
            }
            if (elementExists != 1) {
               return false;
            }
         }

         return true;
      }
   }
```

isSorted:

```
public static boolean isSorted(int [] array) {
      boolean isIncreasing = false;
      boolean isDecreasing = false;
      for(int i = 0; i < array.length - 1; i++){
         if (array[i + 1] > array[i]) {
            isIncreasing = true;
         }else if (array[i + 1] < array[i]) {
            isDecreasing = true;
         }
         if (isIncreasing && isDecreasing) {
            return false;
         }
      }
      return true;
   }
```

StringOps:

capVowelsLowRest:

```
public static String capVowelsLowRest (String string) {
        for(int i = 0; i < string.length(); i++){
            Character letter =  string.charAt(i);
            if ((((int) letter) > 64) && (((int) letter) < 91)) {
                if (i == 0) {
                    string = (char)(((int) letter) + 32) +
string.substring(i+1,string.length());
                }else if ( i == string.length()) {
                    string = string.substring(0, i) + (char)(((int) letter) + 32);
                }else{
                    string = string.substring(0, i) + (char)(((int) letter) + 32) +
string.substring(i+1,string.length());
                }
            }
        }
        for(int i = 0; i < string.length(); i++){
            Character letter =  string.charAt(i);
            if ((((int) letter) == 97) || (((int) letter) == 101) || (((int) letter) == 105)
|| (((int) letter) == 111) || (((int) letter) == 117)) {
                if (i == 0) {
                    string = (char)(((int) letter) - 32) +
string.substring(i+1,string.length());
                }else if ( i == string.length()) {
                    string = string.substring(0, i) + (char)(((int) letter) - 32);
                }else{
                    string = string.substring(0, i) + (char)(((int) letter) - 32) +
string.substring(i+1,string.length());
                }
            }
        }
        return string;
    }
```

camelCase:

```
    public static String camelCase (String string) {
        int wordCounter = 0;
        int letterInWord = 0;
        for(int i = 0; i < string.length(); i++){
            Character letter =  string.charAt(i);

            if ((int)(string.charAt(i)) == 32) {
                if(i != 0){
                    if(string.charAt(i-1) != 32){
                        wordCounter++;
                    }
                }
                letterInWord = 0;
                if(i == 0){
                    string = string.substring(i+1,string.length());
                }else{
                    string = string.substring(0, i) +
    string.substring(i+1,string.length());
                }
                i = i-1;
            }else{
                if (wordCounter == 0) {
                    if ((((int) letter) > 64) && (((int) letter) < 91)) {
                        if (i == 0) {
                            string = (char)(((int) letter) + 32) +
    string.substring(i+1,string.length());
                        }else if ( i == string.length()) {
                            string = string.substring(0, i) + (char)(((int) letter) + 32);
                        }else{
                            string = string.substring(0, i) + (char)(((int) letter) + 32) +
    string.substring(i+1,string.length());
                        }
                    }
                }else{
                    if (letterInWord == 0) {
                        if ((((int) letter) > 97) && (((int) letter) < 122)) {
                            if (i == 0) {
                                string = (char)(((int) letter) - 32) +
string.substring(i+1,string.length());
                            }else if ( i == string.length()) {
                                string = string.substring(0, i) + (char)(((int) letter) - 32);
                            }else{
                                string = string.substring(0, i) + (char)(((int) letter) - 32)
+ string.substring(i+1,string.length());
                            }
                        }
                    }else{
                        if ((((int) letter) > 64) && (((int) letter) < 91)) {
                            if (i == 0) {
                                string = (char)(((int) letter) + 32) +
string.substring(i+1,string.length());
                            }else if ( i == string.length()) {
                                string = string.substring(0, i) + (char)(((int) letter) +
32);
                            }else{
                                string = string.substring(0, i) + (char)(((int) letter) + 32)
+ string.substring(i+1,string.length());
                            }
                        }
                    }
                    letterInWord++;
                }


            }

        }
        return string;
    }
```

allIndexOf:

```
public static int[] allIndexOf (String string, char chr) {
        int arrayLength = 0;
        for(int i = 0; i < string.length(); i++){
            Character letter =  string.charAt(i);
            if (letter == chr) {
                arrayLength++;
            }
        }
        int[] indexArray = new int[arrayLength];
        for(int j = 0; j < arrayLength; j++){
            for(int z = 0; z < string.length(); z++){
                Character letter =  string.charAt(z);
                if (letter == chr) {
                    indexArray[j] = z;
                    if (z == 0) {
                        string = (char)(((int) letter) + 32) +
string.substring(z+1,string.length());
                    }else if ( z == string.length()) {
                        string = string.substring(0, z) + (char)(((int) letter) + 32);
                    }else{
                        string = string.substring(0, z) + (char)(((int) letter) + 32) +
string.substring(z+1,string.length());
                    }
                    break;
                }
            }
        }
        return indexArray;
    }
}
```

Coins:

```
public class Coins {
	public static void main(String[] args) {
		int quarters = Integer.valueOf(args[0]) / 25;
		int cents = Integer.valueOf(args[0]) % 25;
		System.out.println("Use " + quarters + " quarters and " + cents + " cents");
	}
}
```

LinearEq:

```
public class LinearEq {
        public static void main(String[] args) {
                double a = Double.valueOf(args[0]);
                double b = Double.valueOf(args[1]);
                double c = Double.valueOf(args[2]);
                double result = (c - b) / a;
                System.out.println(a + " * x + " + b + " = " + c);
                System.out.println("x = " + result);
        }
}
```

```
}
```

Triangle:

```
public class Triangle {
	public static void main(String[] args) {
		int sideOne = Integer.valueOf(args[0]);
		int sideTwo = Integer.valueOf(args[1]);
		int sideThree = Integer.valueOf(args[2]);
		boolean isTriangle = false;
		isTriangle = (((sideOne + sideTwo) > sideThree ) && ((sideOne +
		sideThree) > sideTwo) && ((sideTwo + sideThree) > sideOne));
		System.out.println(sideOne + ", " + sideTwo + ", " + sideThree + ": " +
		isTriangle);

	}
}
```

GenThree:

```
public class GenThree {
	public static void main(String[] args) {
		// Put your code here
		int min = Integer.valueOf(args[0]);
		int max = Integer.valueOf(args[1]);
		int i = 0;
		int[] numberArray = new int[3];
		numberArray[0] = (int)(Math.random() * (max - min) + min);
		numberArray[1] = (int)(Math.random() * (max - min) + min);
		numberArray[2] = (int)(Math.random() * (max - min) + min);
		int minNumber = Math.min(numberArray[0], numberArray[1]);
		minNumber = Math.min(numberArray[2], minNumber);
```

```
                    System.out.println(numberArray[0] + "\n" + numberArray[1] + "\n" +
                    numberArray[2] + "\n" + "The minimal generated number was "+
                    minNumber);
          }
}
```