

Nicholas White

Anagram:

Anagram.class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package anagram;
import java.util.Arrays;
```

```
/**
 * Palindrome is a generic class that checks whether an array of elements
 * is a palindrome.
 * It also displays the content of an array of elements.
 * @author marin
 * @param <T>
 */
```

```
public class Anagram<T> {
```

```
    /**
     * Constructor
     */
```

```
    public Anagram() {
    }
```

```
    public static void main(String[] args) {
        new MyIntegerAnagramClient().run();
        new MyCharacterAnagramClient().run();
    }
```

```
    /**
     * Checks whether an array of generic elements is a palindrome.
     * In other words, the array is equivalent to its reverse.
     * @param arr the array to be checked
     * @return true if the checked array is a palindrome, false otherwise
     */
```

```

*/
public boolean isAnagram(String str1,String str2) {
    if ( str1.length() != str2.length() ) {
        return false;
    }
    char[] char1 = str1.toCharArray();
    char[] char2 = str2.toCharArray();
    Arrays.sort(char1);
    Arrays.sort(char2);
    String str1New = new String(char1);
    String str2New = new String(char2);
    if (!str1New.equals(str2New)){
        return false;
    }
    else{
        return true;
    }
}
}

```

```

/**
 * Displays the contents of an array of generic elements.
 * @param arr the array to be displayed
 */
public void display(T[] arr1, T[] arr2) {
    int i = 1;
    for (T arr1 : arr) {
        System.out.print(arr1.toString() + " ");
    }
    System.out.println("");
}
}

```

```

}

```

MyCharacterAnagramClient.class

- \* To change this license header, choose License Headers in Project Properties.
- \* To change this template file, choose Tools | Templates
- \* and open the template in the editor.

```

*/
package anagram;
import java.util.Random;

/**
 *
 * @author Nicholas_White
 */
public class MyCharacterAnagramClient {
    /**
     * Size of the Character array
     */
    int size;

    /**
     * Constructor
     * @param size the fixed size of the Character array
     */
    public void MyCharacterAnagramClient() {
        this.size = size;
    }

    /**
     * Generates a new Character array that is a palindrome.
     * @param size the fixed size of the generated Character array
     * @return the generated Character array
     */
    // */
    // public Character[] generateAnagram(int size) {
    //     Character[] arr = new Character[size];
    //     Random randomGenerator = new Random();
    //     for (int i = 0; i < size; i++) {
    //         int randomInt = randomGenerator.nextInt(26)+97;
    //         arr[i] = (char)(randomInt);
    //         System.out.print(arr[i]);
    //     }
    //     return arr;
    // }
    // }
    // public Character[] generateAnagram2(int size) {
    //     Character[] arr= new generateAnagram(this.size);
    //     Character[] arr2 = new Character[size];
    //     int k=0;
    //     for (int i = arr.length; i > 0; i--) {
    //         arr2[i] = arr[k];

```

```

//      k++;
//      System.out.print(arr[i]);
//  }
//
//  return arr2;
//  }

/**
 * Generates a new Character array that is PROBABLY NOT a palindrome.
 * @param size the fixed size of the generated Character array
 * @return the generated Character array
 */
//  */
//  public Character[] generateNotAnagram(int size) {
//      Character[] arr = new Character[size];
//      Random randomGenerator = new Random();
//      for (int i = 0; i < size; i++) {
//          int randomInt = randomGenerator.nextInt(26)+97;
//          arr[i] = (char)(randomInt);
//          System.out.print(arr[i]);
//      }
//      return arr;

/**
 * Tests the class.
 * Generates a new array of Character, displays its contents, then
 * instantiates an Character-typed Palindrome object called myPal,
 * uses it to display the contents of the array, and finally gets
 * it to check if the array is a palindrome.
 */
public void run() {
    Anagram<Character> myAna = new Anagram<Character>();
    String str1 = "abcd";
    String str2 = "affd";
    if(myAna.isAnagram(str1,str2)){
        System.out.println("Its an anagram");
    }
    else{
        System.out.println("Not an anagram");
    }
}
}

```

```
}
```

MyIntegerAnagramClient.class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anagram;

import java.util.Random;

/**
 * MyIntegerPalindromeClient uses the generic Palindrome class to
 * check whether the array of Integers it generates is indeed a palindrome.
 * @author marin
 */
public class MyIntegerAnagramClient {

    /**
     * Size of the Integer array
     */
    int size;

    /**
     * Constructor
     * @param size the fixed size of the Integer array
     */
    public void MyIntegerAnagramClient() {
        this.size = size;
    }

    /**
     * Generates a new Integer array that is a palindrome.
     * @param size the fixed size of the generated Integer array
     */
    // * @return the generated Integer array
    // */
    // public Integer[] generateAnagram(int size) {
    //     Integer[] arr = new Integer[size];
    //     Random randomGenerator = new Random();
    //     for (int i = 0; i < size; i++) {
```

```

//      int randomInt = randomGenerator.nextInt(10);
//      arr[i] = randomInt;
//      // System.out.print(arr[i]);
//  }
//  return arr;
// }
//
// public Integer[] generateAnagram2(Integer arr[]) {
//     Integer[] arr2 = new Integer[size];
//     for (int i=size; i<0;i--){
//         for (int j=0;j<size;j++){
//             arr2[j]=arr[i];
//         }
//     }
//
//     Random randomGenerator = new Random();
//     for (int i = 0; i < size; i++) {
//         int randomInt = randomGenerator.nextInt(10);
//         arr2[i] = randomInt;
//         System.out.print(arr2[i]);
//     }
//     return arr2;
// }

```

```

/**
 * Generates a new Integer array that is PROBABLY NOT a palindrome.
 * @param size the fixed size of the generated Integer array
 * @return the generated Integer array
 */
// */
// public Integer[] generateNotAnagram(int size) {
//     Random rd = new Random();
//     Integer arr[] = new Integer[size];
//     for (Integer arr1 : arr) {
//         arr1 = new Integer(rd.nextInt(101));
//     }
//     for (int i = 0; i < size; i++) {
//         arr[i] = new rd.nextInt(101);
//     }
//     return arr;
// }

```

/\*\*

```

* Tests the class.
* Generates a new array of Integer, displays its contents, then
* instantiates an Integer-typed Palindrome object called myPal,
* uses it to display the contents of the array, and finally gets
* it to check if the array is a palindrome.
*/
public void run() {
    Anagram<Integer> myAna = new Anagram<Integer>();
    String str1 = "1234";
    String str2 = "4231";
    if(myAna.isAnagram(str1,str2)){
        System.out.println("Its an anagram");
    }
    else{
        System.out.println("Not an anagram");
    }
}

// Integer arr1[] = this.generateAnagram(size);
// for (int i = 0; i < size; i++) {
//     System.out.print(arr1[i]);
// }
// System.out.print(' ');
// Anagram<Integer> myAna2 = new Anagram<Integer>();
// Integer arr2[] = this.generateAnagram2(arr1);
// for (int i = 0; i < size; i++) {
//     System.out.print(arr2[i]);
// }
//
// myAna.display(arr1);
// if (myAna.isAnagram(arr1)) {
//     System.out.println("We have an anagram");
// } else {
//     System.out.println("Nope");
// }
// Integer arr2[] = this.generateNotAnagram(size/2);
// myAna.display(arr2);
// if (myAna.isAnagram(arr2)) {
//     System.out.println("We have an anagram");
// } else {
//     System.out.println("Nope");
// }
}

```

BWR:

```

import java.util.Random;
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hw_one;
/**
 *
 * @author Nicholas_White
 */
public class HW_One {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        new HW_One().run();
    }

    void run() {
        /*Exercise 8 - BLUE-WHITE-RED flag
        Write a program that generates an array of 30 characters that
        is a random sequence of characters 'B', 'W', and 'R'.
        Once the array is ready the program displays the resulting string,
        and then sorts the array so that all 'B's come first,
        all 'R's are at the end, and all 'W's are in between. */
        int stringSize = 30;
        Random randomGenerator = new Random();
        String[] BWR = new String[stringSize];
        for (int i=0; i<stringSize; i++){
            int randomInt = randomGenerator.nextInt(3) + 1;
            switch (randomInt){
                case 1: BWR[i] = "B";
                    break;
                case 2: BWR[i] = "W";
                    break;
                case 3: BWR[i] = "R";
                    break;
            }
        }
    }
}

```



```

        for (int i=0;i<stringSize;i++){
            System.out.print(BWR[i] + ' ');
        }
        System.out.println("");
        int redCount = 0;
        int whiteCount = 0;
        int blueCount = 0;

        for (int i=0;i<stringSize;i++){
            switch(BWR[i]){
                case "R": redCount++;
                    break;
                case "W": whiteCount++;
                    break;
                case "B": blueCount++;
            }
        }
        int finalCount = redCount+blueCount+whiteCount;
        //      System.out.println(redCount+":R "+blueCount+":B "+whiteCount+":W
        "+finalCount);
        String[] Final = new String[finalCount];
        for (int i=0;i<blueCount;i++){
            Final[i] = "B";
        }
        for (int i=0;i<whiteCount;i++){
            Final[i+blueCount] = "W";
        }
        for (int i=0;i<redCount;i++){
            Final[i+whiteCount+blueCount] = "R";
        }
        for (int i=0;i<stringSize;i++){
            System.out.print(Final[i] + ' ');
        }
        System.out.println("");
    }
}

```

Vigenere:  
Vigenere.class

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.

```

```

*/
package vigenere;

/**
 *
 * @author Nicholas_White
 */
class Vigenere {
    public static void main(String args[]){
        new Vigenere().run();
    }

    public void Encrypt(String s1, String s2) {
        String phrase = s1;
        String key = s2;
        phrase = phrase.toLowerCase();
        key = key.toLowerCase();
        for (int i=0; i<key.length();i++){
            if (key.length() < phrase.length()){
                key = key+key.charAt(i);
            }
        }
        Character[] Phrase = new Character[phrase.length()];
        Character[] Key = new Character[key.length()];
        Character[] Alphabet =
        {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
        //    //Character[] Phrase = new Character[phrase.length()];
        for (int i =0;i<phrase.length();i++){
            char phraseChar = phrase.charAt(i);
            //    System.out.print(phrase.substring(i,i+1));
            //System.out.println(phraseChar);
            Phrase[i] = phraseChar;
        }
        for (int i=0;i<phrase.length();i++){
            char keyChar = key.charAt(i);
            Key[i] = keyChar;
        }
        int[] phraseNum = new int[phrase.length()];
        int[] keyNum = new int[key.length()];
        int[] finalNum = new int[phrase.length()];
        char[] codedArray = new char[finalNum.length];
        for (int j=0;j<phrase.length();j++){
            for(int i=0;i<Alphabet.length;i++){
                if (Phrase[j] == Alphabet[i]){

```

```

        phraseNum[j] = i%26;
        // System.out.println(phraseNum);
    }
}
for (int j=0;j<key.length();j++){
    for(int i=0;i<Alphabet.length;i++){
        if (Key[j] == Alphabet[i]){
            keyNum[j] = i%26;
            // System.out.println(keyNum);
        }
    }
}

for (int i=0;i<phraseNum.length;i++){
    finalNum[i] = phraseNum[i] + keyNum[i];
}

for (int i=0;i<finalNum.length;i++){
    codedArray[i] = Alphabet[finalNum[i]%26];
    System.out.print(codedArray[i]);

}

// for (int i=0;i<phrase.length();i++){
//     System.out.print(Key[i]);
// }
}

// String Str = new String("Welcome to Tutorialspoint.com");
// //Character[] Phrase = new Character[Str.length()];
// for (int i=0;i<Str.length();i++){
//     char firstletter = Str.charAt(i);
//     //Phrase[i] = firstletter;
//     System.out.println(firstletter);

void run(){
    new Vigenere().Encrypt("ABC","NYUSH");
}

// Vigenere() {
//     this.Phrase = new Character[];
// }

```

}