

PROJECT 3

CHARSET CLASS

**CSCI 145
JAVA LANGUAGE**

MAI PHAM

**DEVELOPMENT ENVIRONMENT
ECLIPSE**

TABLE OF CONTENTS

 **Project Note**

 **Output**

 **Source Code**

PROJECT NOTE

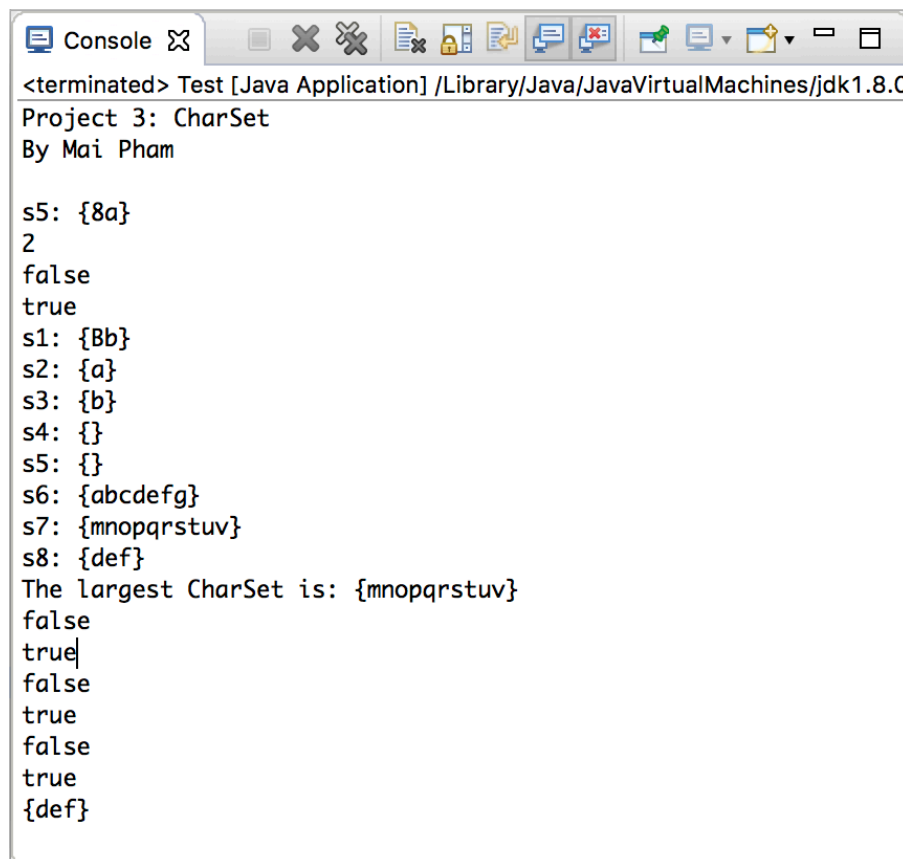
Objective – Create a CharSet class that hold a boolean array of 128 true or false.

Summary – The project this time isn't hard and very straightforward. Just follow the list of methods that need to implement. There is also a provided driver to check if the class is working properly or not. However, the given driver doesn't test some methods so I just added more test cases to it.

Extra credits – There are two extra credits in this project. I did the extra credit 2. I think the extra credit requirement is to implement the Comparable interface and override the compareTo method to determine the length of two sets. And we have to test this feature by calling through the Compare3 class from lab 7 instead of using the compareTo directly. My program does success apply this feature.

Conclusion – As a conclusion, my project is completed and successful run both the original and the extra credit part.

OUTPUT FILE



```
<terminated> Test [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.C
Project 3: CharSet
By Mai Pham

s5: {8a}
2
false
true
s1: {Bb}
s2: {a}
s3: {b}
s4: {}
s5: {}
s6: {abcdefg}
s7: {mnopqrstuv}
s8: {def}
The largest CharSet is: {mnopqrstuv}
false
true
false
true
false
true
{def}
```

SOURCE CODE

CharSet

```
public class CharSet implements Comparable<CharSet>
{
    private final int MAX_SIZE = 128;
    private boolean [] arr = new boolean[MAX_SIZE];

    public CharSet()
    {
    }
    public CharSet(char c)
    {
        arr[c] = true;
    }
    public CharSet(String c)
    {
        for (int i = 0; i < c.length(); i++)
            arr[c.charAt(i)] = true;
    }
    public CharSet union(CharSet a)
    {
        CharSet c = new CharSet();
        for (int i = 32; i < MAX_SIZE; i++)
        {
            if (a.arr[i] || arr[i])
                c.arr[i] = true;
        }
        return c;
    }
    public CharSet intersection(CharSet a)
    {
        CharSet c = new CharSet();
        for (int i = 32; i < MAX_SIZE; i++)
        {
            if (a.arr[i] && arr[i])
                c.arr[i] = true;
        }
        return c;
    }
    public boolean insert(char c)
    {
        if (c < MAX_SIZE && c > 31)
        {
            arr[c] = true;
            return true;
        }
        return false;
    }
    public boolean remove(char c)
    {
        if (c < MAX_SIZE && c > 31)
        {
            arr[c] = false;
            return true;
        }
    }
}
```

```
    }  
    return false;  
}  
public boolean isElement(char c)  
{  
    if (arr[c])  
        return true;  
    return false;  
}  
public int size()  
{  
    int count = 0;  
    for (int i = 32; i < MAX_SIZE; i++)  
        if (arr[i])  
            count++;  
    return count;  
}  
public String toString()  
{  
    String s = "{";  
    for (int i = 32; i < MAX_SIZE; i++)  
        if (arr[i])  
            s+= (char)i;  
    s+="}";  
    return s;  
}  
public CharSet clone()  
{  
    CharSet c = new CharSet();  
    for (int i = 32; i < MAX_SIZE; i++)  
    {  
        if (arr[i])  
            c.arr[i] = true;  
    }  
    return c;  
}  
public boolean equal(CharSet a)  
{  
    boolean b = false;  
  
    for (int i = 32; i < MAX_SIZE; i++)  
    {  
        if (arr[i] && a.arr[i] || !arr[i] && !a.arr[i])  
            b = true;  
        else  
        {  
            b = false;  
            break;  
        }  
    }  
    return b;  
}  
public boolean subset(CharSet a)  
{  
    boolean b = false;  
    for (int i = 32; i < MAX_SIZE; i++)  
    {
```

```
        if (arr[i])
            if (a.arr[i])
                b = true;
            else
            {
                b = false;
                break;
            }
        }
        return b;
    }
    public boolean superset(CharSet a)
    {
        boolean b = false;
        for (int i = 32; i < MAX_SIZE; i++)
        {
            if (a.arr[i])
                if(arr[i])
                    b = true;
                else
                {
                    b = false;
                    break;
                }
        }
        return b;
    }
    public void clear()
    {
        for (int i = 32; i < MAX_SIZE; i++)
            arr[i] = false;
    }
    //EXTRA CREDIT 2
    @Override
    public int compareTo(CharSet a)
    {
        int set1 = 0, set2 = 0;
        for (int i = 32; i < MAX_SIZE; i++)
        {
            if (arr[i])
                set1++;
            if (a.arr[i])
                set2++;
        }
        if (set1 == set2)
            return 0;
        else if (set1 < set2)
            return -1;
        else
            return 1;
    }
}
```

Test

```
public class Test
{
```

```

public static void main(String[] args)
{
    System.out.println("Project 3: CharSet");
    System.out.println("By Mai Pham\n");

    //ORIGINAL TEST CASES BY PROF.T.VO
    CharSet s1, s2, s3, s4, s5;
    String values = "abc";
    s1 = new CharSet();
    s2 = new CharSet('a');
    s1.insert('b');
    s3 = new CharSet(values);
    s3 = s1.clone();
    s4 = s1.union(s2);

    // s1 is an empty set, {}
    // s2 is now {a}
    // s1 is now {b}
    // s3 is now {abc}
    // s3 is now {b}
    // s4 is now {ab} and s1 is

    still {b}
    s1.insert('B');
    s4.insert('8');
    s4.remove('b');
    s5 = s4;
    System.out.println("s5: " + s5);
    System.out.println(s5.size());
    System.out.println(s3.subset(s4));
    System.out.println(s2.subset(s4));
    s5.clear();

    // s1 is now {Bb}
    // s4 is now {8ab}
    // s4 is now {8a}
    // s5 references s4
    // s5: {8a}
    // output 2
    // output false
    // output true
    // s4 and s5 both reference

    same empty set, {}
    char c = (char)227;
    s1.insert(c);

    // character for PI
    // invalid element so

    ignore, s1 is still {b}
    s1.remove(c);

    // invalid element so

    ignore, s1 is still {b}
    System.out.println("s1: " + s1);
    System.out.println("s2: " + s2);
    System.out.println("s3: " + s3);
    System.out.println("s4: " + s4);
    System.out.println("s5: " + s5);

    // s1: {Bb}
    // s2: {a}
    // s3: {b}
    // s4: {}
    // s5: {}

    //EXTRA CREDIT 2 & additional test cases
    CharSet s6 = new CharSet("abcdefg");
    CharSet s7 = new CharSet("mnopqstruv");
    CharSet s8 = new CharSet("def");
    System.out.println("s6: " + s6);
    System.out.println("s7: " + s7);
    System.out.println("s8: " + s8);
    System.out.println("The largest CharSet is: " + Compare3.largest(s6, s7,
s8));

    System.out.println(s2.equal(s3));
    System.out.println(s4.equal(s5));
    System.out.println(s3.superset(s1));
    System.out.println(s6.superset(s8));

    // output false
    // output true
    // output false
    // output true

    System.out.println(s6.isElement('t'));
    System.out.println(s7.isElement('t'));
    System.out.println(s6.intersection(s8));

    // output false
    // output true
    // output {def}
}
}

```

Compare3

```
public class Compare3
{
    public static CharSet largest(CharSet obj1, CharSet obj2, CharSet obj3)
    {
        if (obj1.compareTo(obj2) > 0 && obj1.compareTo(obj3) > 0)
            return obj1;
        else if (obj2.compareTo(obj1) > 0 && obj2.compareTo(obj3) > 0)
            return obj2;
        else if (obj3.compareTo(obj1) > 0 && obj3.compareTo(obj2) > 0)
            return obj3;
        return obj3;
    }
}
```