

BÀI TẬP LAB 7

Người thực hiện: Nguyễn Chí Sơn – MSSV: 22130235

MỤC LỤC

7.1 Bài 1:	4
7.1.1 Yêu cầu:	4
7.2 Bài 2:	4
7.2.1 Yêu cầu:	4
7.2.2 Mã nguồn:	4
7.3 Bài 3:	4
7.3.1 Yêu cầu:	5
7.3.2 Mã nguồn:	5
7.4 Bài 4:	5
7.4.1 Yêu cầu:	5
7.4.2 Mã nguồn:	5
7.5 Bài 5:	6
7.5.1 Yêu cầu:	6
7.5.2 Mã nguồn:	6
7.6 Bài 6:	6
7.6.1 Yêu cầu:	6
7.6.2 Mã nguồn:	6
7.7 Bài 7:	7
7.7.1 Yêu cầu:	7
7.7.2 Mã nguồn:	7
7.8 Bài 8:	7
7.8.1 Yêu cầu:	7
7.8.2 Mã nguồn:	7
7.9 Bài 9:	8
7.9.1 Yêu cầu:	8
7.9.2 Mã nguồn:	8
7.10 Bài 10:	8
7.10.1 Yêu cầu:	8
7.10.2 Mã nguồn:	8

7.11 Bài 11:	8
7.11.1 Yêu cầu:.....	8
7.11.2 Mã nguồn:.....	8
7.12 Bài 12:	9
7.12.1 Yêu cầu:.....	9
7.12.2 Mã nguồn:.....	9
7.13 Bài 13:	9
7.13.1 Yêu cầu:.....	9
7.13.2 Mã nguồn:.....	9
7.14 Bài 14:	10
7.14.1 Yêu cầu:.....	10
7.14.2 Mã nguồn:.....	10
7.15 Bài 15:	10
7.15.1 Yêu cầu:.....	10
7.15.2 Mã nguồn:.....	10
7.16 Bài 16:	11
7.16.1 Yêu cầu:.....	11
7.16.2 Mã nguồn:.....	11
7.17 Bài 17:	11
7.17.1 Yêu cầu:.....	11
7.17.2 Mã nguồn:.....	11
7.18 Bài 18:	12
7.18.1 Yêu cầu:.....	12
7.18.2 Mã nguồn:.....	12
7.19 Bài 19:	12
7.19.1 Yêu cầu:.....	12
7.19.2 Mã nguồn:.....	12
7.20 Bài 20:	12
7.20.1 Yêu cầu:.....	12
7.20.2 Mã nguồn:.....	13
Lap 7 Swing:	13
7.1 Yêu cầu:.....	13
7.2 Mã nguồn:.....	13

7.1 Bài 1:

7.1.1 Yêu cầu:

- Write a Java method to reverse a string.

```
package mainProgram;

public class Ex1 {
    public static String daoNguocChuoai(String s) {
        s.replaceAll("\\s\\s+", " ").trim();
        String[] words = s.split(" ");
        String reversedString = "";
        for (int i = 0; i < words.length; i++) {
            String word = words[i];
            // đảo ngược từ
            String reverseWord = "";
            for (int j = word.length() - 1; j >= 0; j--) {
                reverseWord = reverseWord + word.charAt(j);
            }
            reversedString = reversedString + reverseWord + " ";
        }
        return reversedString;
    }
}
```

7.2 Bài 2:

7.2.1 Yêu cầu:

- Write a Java method to check if a given string is a palindrome or not.

7.2.2 Mã nguồn:

```
package mainProgram;

public class Ex2 {
    public static boolean isCheckPalindrome(String s) {
        s.replaceAll(s, "/" + s + "/").trim();
        boolean dungSai = false;

        for (int i = 0; i < s.length() / 2; i++) {
            if (s.charAt(i) == s.charAt(s.length() - 1 - i)) {
                return dungSai = true;
            }
            return dungSai = false;
        }
        return dungSai;
    }
}
```

7.3 Bài 3:

7.3.1 Yêu cầu:

- Write a Java method to count the number of vowels and consonants in a given string.

7.3.2 Mã nguồn:

```
package mainProgram;

public class Ex3 {

    public static String demNguyenAm(String str) {

        int vcount = 0, ccount = 0;

        str = str.toLowerCase();

        for (int i = 0; i < str.length(); i++) {

            char ch = str.charAt(i);

            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

                vcount++;

            } else if ((ch >= 'a' && ch <= 'z')) {

                ccount++;

            }

        }

        return "Số nguyên âm là: " + vcount + " và " + "Số phụ âm là: " + ccount;

    }

}
```

7.4 Bài 4:

7.4.1 Yêu cầu:

- Write a Java method to find the largest word in a given string.

7.4.2 Mã nguồn:

```
package mainProgram;

public class Ex4 {

    public static String timTuLonNhat(String chuoi) {

        String[] mangTu = chuoi.split(" "); // tách chuỗi thành một mảng
        các từ

        String tuLonNhat = mangTu[0]; // giả sử từ đầu tiên là từ lớn nhất

        // duyệt qua tất cả các từ trong mảng để tìm từ lớn nhất
        for (int i = 1; i < mangTu.length; i++) {
```

```

        if (mangTu[i].length() > tuLonNhat.length()) {
            tuLonNhat = mangTu[i]; // nếu tìm thấy từ dài hơn, gán lại
            từ lớn nhất
        }

        return tuLonNhat;
    }
}

```

7.5 Bài 5:

7.5.1 Yêu cầu:

- Write a Java method to check if two given strings are anagrams or not.

7.5.2 Mã nguồn:

```

package mainProgram;

public class Ex5 {

    public static String reverseString(String s) {

        if (s == null || s.isEmpty()) {
            return "chuỗi rỗng";
        }
        // đảo ngược chuỗi bằng phương thức reverse của lớp StringBuilder
        StringBuilder reversed = new StringBuilder(s).reverse();
        // trả về kết quả sau khi đảo ngược
        return reversed.toString();
    }

    public static boolean daoNguoc(String s1, String s2) {
        return reverseString(s1).equals(s2);
    }

}

```

7.6 Bài 6:

7.6.1 Yêu cầu:

- Write a Java method to remove all the white spaces from a given string.

7.6.2 Mã nguồn:

```

package mainProgram;

public class Ex6 {
    public static String xoaKhoangtrang(String s) {
        return "Chuỗi đã xóa khoảng trắng là:" + s.replaceAll("\\s\\s+", " ")
            .trim();
    }
}

```

7.7 Bài 7:

7.7.1 Yêu cầu:

- Write a Java method to find the second most frequent character in a given string.

7.7.2 Mã nguồn:

```
package mainProgram;

import java.util.HashMap;

public class Ex7 {
    public static char findSecondMostFrequentChar(String str) {
        // Tạo một HashMap để đếm số lần xuất hiện của các ký tự trong chuỗi
        HashMap<Character, Integer> charCount = new HashMap<>();
        for (int i = 0; i < str.length(); i++) {
            char c = str.charAt(i);
            charCount.put(c, charCount.getOrDefault(c, 0) + 1);
        }

        // Tìm giá trị thứ hai lớn nhất trong HashMap
        int max1 = 0, max2 = 0;
        char secondMostFreqChar = '\0';
        for (char c : charCount.keySet()) {
            int count = charCount.get(c);
            if (count > max1) {
                max2 = max1;
                max1 = count;
                secondMostFreqChar = c;
            } else if (count > max2 && count < max1) {
                max2 = count;
                secondMostFreqChar = c;
            }
        }

        return secondMostFreqChar;
    }
}
```

7.8 Bài 8:

7.8.1 Yêu cầu:

- Write a Java method to remove the first and last character of a given string.

7.8.2 Mã nguồn:

```
package mainProgram;

public class Ex8 {
    public static String removeFirstAndLastChar(String str) {
        if (str == null || str.length() < 2) {
            return "";
        }
        return str.substring(1, str.length() - 1);
    }
}
```

```
}
```

```
}
```

7.9 Bài 9:

7.9.1 Yêu cầu:

- Write a Java method to count the number of words in a given string.
lớn nhất.

7.9.2 Mã nguồn:

```
package mainProgram;

public class Ex9 {
    public static int countWords(String str) {
        if (str == null || str.isEmpty()) {
            return 0;
        }
        String[] words = str.split("\\s+");
        return words.length;
    }
}
```

7.10 Bài 10:

7.10.1 Yêu cầu:

- Write a Java method to check if a given string contains only digits.

7.10.2 Mã nguồn:

```
package mainProgram;

public class Ex10 {

    // Trong lớp Character có hỗ trợ phương thức isDigit() để kiểm tra
    // chuỗi có phải là 1 con số hay không
    // nếu đúng thì trả về true và nếu sai thì trả về false
    public static boolean checkDigits(String input) {
        for (int i = 0; i < input.length(); i++) {
            if (!Character.isDigit(input.charAt(i))) {
                return false;
            }
        }
        return true;
    }
}
```

7.11 Bài 11:

7.11.1 Yêu cầu:

- Write a Java method that takes a string as input and counts the number of vowels
present in it.

7.11.2 Mã nguồn:


```

package mainProgram;

public class Ex11 {
    public static int countVowels(String str) {
        int count = 0;
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch ==
'u'
                || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch
== 'U') {
                count++;
            }
        }
        return count;
    }
}

```

7.12 Bài 12:

7.12.1 Yêu cầu:

- Write a Java method to reverse a string without using the reverse() method.

7.12.2 Mã nguồn:

```

package mainProgram;

public class Ex12 {
    public static String daoNguocChuoi(String s) {
        String str = s.replaceAll("\\s\\s+", " ").trim();
        String [] words = str.split(" ");
        String reversedString = "";
        for (int i = 0; i < words.length; i++) {
            String word = words[i];
            // đảo ngược từ
            String reverseWord = "";
            for (int j = word.length() - 1; j >= 0; j--) {
                reverseWord = reverseWord + word.charAt(j);
            }
            reversedString = reversedString + reverseWord + " ";
        }
        return reversedString;
    }
}

```

7.13 Bài 13:

7.13.1 Yêu cầu:

- Write a Java method that takes two strings as input and checks whether they are anagrams of each other.

7.13.2 Mã nguồn:

```

package mainProgram;

public class Ex13 {
    public static boolean isReverse(String str1, String str2) {

```

```

        // Nếu độ dài hai chuỗi khác nhau, chúng không thể là đảo chữ cái
của nhau
        if (str1.length() != str2.length()) {
            return false;
        }
        // So sánh các ký tự của hai chuỗi với nhau
        for (int i = 0; i < str1.length(); i++) {
            if (str1.charAt(i) != str2.charAt(str2.length() - 1 - i)) {
                return false;
            }
        }
        return true;
    }
}

```

7.14 Bài 14:

7.14.1 Yêu cầu:

- Write a Java method to remove all the white spaces from a given string.

7.14.2 Mã nguồn:

```

package mainProgram;

public class Ex14 {
    public static String removeSpaces(String str) {
        return str.replaceAll("\\s+", "");
    }
}

```

7.15 Bài 15:

7.15.1 Yêu cầu:

- Write a Java method that takes a sentence as input and capitalizes the first letter of each word.

7.15.2 Mã nguồn:

```

package mainProgram;

public class Ex15 {
    public static StringBuilder inChuDau(String s) {
        String str = s.replaceAll("\\s+", " ").trim();
        String[] words = str.split(" ");

        StringBuilder result = new StringBuilder();

        // vòng lặp foreach giúp ta gán các từ trong words sang word sau
mỗi lần lặp
        // (gán từng từ một)

        for (String word : words) {
            char firstChar = word.charAt(0);
            String rest = word.substring(1).toLowerCase();

            result.append(Character.toUpperCase(firstChar)).append(rest).append("
");

```

```

    }

    // nếu trường hợp chuỗi người ta có khoảng trắng ở 2 bên ở giữa
    thì phải dùng
    // hàm trim còn nếu không có khoảng trắng thì mình in ra result
    thôi
    return result;
}
}

```

7.16 Bài 16:

7.16.1 Yêu cầu:

- Write a Java method that takes a string as input and checks whether it is a palindrome or not.

7.16.2 Mã nguồn:

```

package mainProgram;

public class Ex16 {

    // Để kiểm tra xem một chuỗi có phải là đối xứng hay không, ta có thể
    so sánh
    // chuỗi đó với chuỗi được lật ngược theo thứ tự ký tự. Ví dụ: Chuỗi
    "racecar"
    // lật ngược lại vẫn là "racecar", vì vậy chuỗi này là đối xứng.

    public static boolean isPalindrome(String str) {
        String reversed = new StringBuilder(str).reverse().toString();
        return str.equals(reversed);
    }

}

```

7.17 Bài 17:

7.17.1 Yêu cầu:

- Write a Java method that takes a string as input and returns all the possible substrings of that string.

7.17.2 Mã nguồn:

```

package mainProgram;

```

```

import java.util.ArrayList;

```

```

public class Ex17 {
    public static ArrayList<String> getAllSubstrings(String str) {
        ArrayList<String> substrings = new ArrayList<String>();
        for (int i = 0; i < str.length(); i++) {
            for (int j = i + 1; j <= str.length(); j++) {
                substrings.add(str.substring(i, j));
            }
        }
    }
}

```

```

    }
    return substrings;
}

```

```

}

```

7.18 Bài 18:

7.18.1 Yêu cầu:

- Write a Java method that takes a string as input and return the characters of that string in reverse order.

7.18.2 Mã nguồn:

```

package mainProgram;

```

```

public class Ex18 {
    public static String daoNguocKyTu(String chuoi) {
        StringBuilder daoNguoc = new StringBuilder();
        for (int i = chuoi.length() - 1; i >= 0; i--) {
            daoNguoc.append(chuoi.charAt(i));
        }
        return daoNguoc.toString();
    }
}

```

```

}

```

7.19 Bài 19:

7.19.1 Yêu cầu:

- Write a Java method that takes two strings as input and concatenates them then return the concatenated string.

7.19.2 Mã nguồn:

```

package mainProgram;

```

```

public class Ex19 {
    public static String noiHaiChuoi(String s1 , String s2) {
        String s3 = s1.concat(s2);
        return "Chuỗi đã nối là: " + s3;
    }
}

```

```

}

```

7.20 Bài 20:

7.20.1 Yêu cầu:

- Write a Java method that takes a string as input and then return the string that

removes all the special characters present in it.substrings of that string.

7.20.2 Mã nguồn:

```
package mainProgram;
```

```
public class Ex20 {  
    public static String removeSpecialCharacters(String s) {  
        String result = "";  
        if (s != null) {  
            // Loại bỏ các ký tự đặc biệt  
            result = s.replaceAll("[^a-zA-Z0-9]", "");  
        }  
        return result;  
    }  
}
```

Lap 7 Swing:

7.1 Yêu cầu:

Viết 1 chương trình trên Java Swing để gọi tới các phương thức ở trên. Trên JFrame cần có:

- textarea để nhập vào chuỗi,
- 1 textarea để trả về kết quả của phương thức.
- Có các button để khi người dùng kích vào thì gọi tới từng phương thức, ví dụ có 20 bài thì sẽ có 20 button. Có button Exit để kết thúc chương trình.

Nếu SV không thể viết được phương thức ở trên, thì trong JFrame vẫn phải có button của bài đó và phương thức đó sẽ trả về giá trị mặc định (nếu kiểu trả về là int thì giá trị mặc định là 0, kiểu trả về là boolean thì giá trị mặc định là true;...).

7.2 Mã nguồn:

```
package mainProgram;
```

```
import java.awt.BorderLayout;  
import java.awt.FlowLayout;  
import java.awt.GridLayout;  
import java.awt.LayoutManager;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.JTextArea;
```

```

public class MainProgram extends JFrame implements ActionListener{
    private JButton
jButtonEx1,jButtonEx2,jButtonEx3,jButtonEx4,jButtonEx5,jButtonEx6,jButtonEx7,jBut
tonEx8,jButtonEx9,jButtonEx10,jButtonEx11,jButtonEx12,jButtonEx13,jButtonEx14,jB
uttonEx15,jButtonEx16,jButtonEx17,jButtonEx18,jButtonEx19,jButtonEx20;

    private JTextArea jTextArea1,jTextArea2,jTextArea3;
    public MainProgram() {
        setLayout(new BorderLayout());
        setTitle("Bài tập String");

        JPanel p1 = new JPanel(new FlowLayout(FlowLayout.CENTER));
        jTextArea1 = new JTextArea(5,20);
        jTextArea2 = new JTextArea(5,20);
        p1.add(jTextArea1);
        p1.add(jTextArea2);

        JPanel p2 = new JPanel(new FlowLayout(FlowLayout.CENTER));
        jTextArea3 = new JTextArea(5,20);
        p2.add(jTextArea3);

        JPanel p3 = new JPanel(new GridLayout(2,10));
        jButtonEx1 = new JButton("Ex1");
        jButtonEx2 = new JButton("Ex2");
        jButtonEx3 = new JButton("Ex3");
        jButtonEx4 = new JButton("Ex4");
        jButtonEx5 = new JButton("Ex5");
        jButtonEx6 = new JButton("Ex6");
        jButtonEx7 = new JButton("Ex7");
        jButtonEx8 = new JButton("Ex8");
        jButtonEx9 = new JButton("Ex9");
        jButtonEx10 = new JButton("Ex10");
        jButtonEx11 = new JButton("Ex11");
        jButtonEx12 = new JButton("Ex12");
        jButtonEx13 = new JButton("Ex13");
        jButtonEx14 = new JButton("Ex14");
        jButtonEx15 = new JButton("Ex15");
        jButtonEx16 = new JButton("Ex16");
        jButtonEx17 = new JButton("Ex17");

```

```
jButtonEx18 = new JButton("Ex18");
jButtonEx19 = new JButton("Ex19");
jButtonEx20 = new JButton("Ex20");
p3.add(jButtonEx1);
p3.add(jButtonEx2);
p3.add(jButtonEx3);
p3.add(jButtonEx4);
p3.add(jButtonEx5);
p3.add(jButtonEx6);
p3.add(jButtonEx7);
p3.add(jButtonEx8);
p3.add(jButtonEx9);
p3.add(jButtonEx10);
p3.add(jButtonEx11);
p3.add(jButtonEx12);
p3.add(jButtonEx13);
p3.add(jButtonEx14);
p3.add(jButtonEx15);
p3.add(jButtonEx16);
p3.add(jButtonEx17);
p3.add(jButtonEx18);
p3.add(jButtonEx19);
p3.add(jButtonEx20);
```

```
jButtonEx1.addActionListener(this);
jButtonEx2.addActionListener(this);
jButtonEx3.addActionListener(this);
jButtonEx4.addActionListener(this);
jButtonEx5.addActionListener(this);
jButtonEx6.addActionListener(this);
jButtonEx7.addActionListener(this);
jButtonEx8.addActionListener(this);
jButtonEx9.addActionListener(this);
jButtonEx10.addActionListener(this);
jButtonEx11.addActionListener(this);
jButtonEx12.addActionListener(this);
jButtonEx13.addActionListener(this);
jButtonEx14.addActionListener(this);
jButtonEx15.addActionListener(this);
jButtonEx16.addActionListener(this);
```

```

jButtonEx17.addActionListener(this);
jButtonEx18.addActionListener(this);
jButtonEx19.addActionListener(this);
jButtonEx20.addActionListener(this);

add(p1, BorderLayout.NORTH);
add(p2, BorderLayout.CENTER);
add(p3, BorderLayout.SOUTH);

setSize(1000,400);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);

}
@Override
public void actionPerformed(ActionEvent e) {
    String str1 = jTextArea1.getText();
    String str2 = jTextArea2.getText();
    if(e.getSource() == jButtonEx1) {
        jTextArea3.setText(Ex1.daoNguocChuo(i(str1)));
    }else if(e.getSource() == jButtonEx2) {
        jTextArea3.setText(Ex2.isCheckPalindrome(str1)+"");
    }else if(e.getSource() == jButtonEx3) {
        jTextArea3.setText(Ex3.demNguyenAm(str1));
    }else if(e.getSource() == jButtonEx4) {
        jTextArea3.setText(Ex4.timTuLonNhat(str1));
    }else if(e.getSource() == jButtonEx5) {
        jTextArea3.setText(Ex5.daoNguoc(str1,str2)+"");
    }else if(e.getSource() == jButtonEx6) {
        jTextArea3.setText(Ex6.xoaKhoangtrang(str1));
    }else if(e.getSource() == jButtonEx7) {
        jTextArea3.setText(Ex7.findSecondMostFrequentChar(str1)+"");
    }else if(e.getSource() == jButtonEx8) {
        jTextArea3.setText(Ex8.removeFirstAndLastChar(str1));
    }else if(e.getSource() == jButtonEx9) {
        jTextArea3.setText(Ex9.countWords(str1)+"");
    }else if(e.getSource() == jButtonEx10) {

```



```
        jTextArea3.setText(Ex10.checkDigits(str1)+"");
    }else if(e.getSource() == jButtonEx11) {
        jTextArea3.setText(Ex11.countVowels(str1)+"");
    }else if(e.getSource() == jButtonEx12) {
        jTextArea3.setText(Ex12.daoNguocChuoi(str1));
    }else if(e.getSource() == jButtonEx13) {
        jTextArea3.setText(Ex13.isReverse(str1,str2)+"");
    }else if(e.getSource() == jButtonEx14) {
        jTextArea3.setText(Ex14.removeSpaces(str1));
    }else if(e.getSource() == jButtonEx15) {
        jTextArea3.setText(Ex15.inChuDau(str1)+"");
    }else if(e.getSource() == jButtonEx16) {
        jTextArea3.setText(Ex16.isPalindrome(str1)+"");
    }else if(e.getSource() == jButtonEx17) {
        jTextArea3.setText(E
```