

Name: Shangirne Kharbanda

Registration Number: 20BAI1154

JAVA LAB 7

THREADS AND ARRAYLIST EXERCISES

1. Create a Secure Password using Multithreading. Create three thread class using any concept like Fibonacci, Armstrong number, Random number. Get the input from user and run the all the three algorithm parallely and combine the all the result is a secured password.

armstrong.java

```
package Threading;

public class armstrong extends Thread {
    int num;
    String result="";
    armstrong(int n)
    {
        num = n;
    }
    public void run()
    {
        int a,r,ans=0;
        a=num;
        while(a!=0)
        {
            r=a%10;
            ans+=Math.pow(r, 3);
            a/=10;
        }
        if(ans==num)
        {
            result += "It is an Armstrong number";
        }
        else
        {
            result += "Not an Armstrong number";
        }
    }
    public String isArmstrong()
    {
        return result;
    }
}
```

fibonacci.java

```
package Threading;

public class fibonacci extends Thread{
    int num;
    fibonacci(int n)
    {
        num = n;
    }
    public void run()
    {
        int a=0;
        int b=1;
        int c=0;
        for(int i=0; i<num; i++){
            System.out.print(c+" ");
            a=b;
            b=c;
            c=a+b;
        }
    }
}
```

random.java

```
package Threading;

public class random extends Thread {
    int num;
    String result = "";
    random(int n) {
        num = n;
    }
    public void run() {
        if(num%2==0)
        {
            result+="It is an even number";
        }
        else
        {
            result+="It is an odd number";
        }
    }
    public String isOddeven() {
        return result;
    }
}
```

threads.java

```
package Threading;
import java.util.Scanner;

public class threads {
    public static void main(String[] args) throws InterruptedException {
        Scanner sc = new Scanner(System.in);
        int num;
```

```

String pwd = "whatsupppp@2221";
System.out.print("Enter number : ");
num = sc.nextInt();
sc.nextLine();
armstrong a = new armstrong(num);
a.start();
a.join();
String a2 = a.isArmstrong();
random o = new random(num);
o.start();
o.join();
String a3 = o.isOddeven();
String p;
System.out.print("Enter password: ");
p = sc.nextLine();
if (pwd.equals(p)) {
    System.out.println("The entered number is: " + num);
    fibonacci f = new fibonacci(num);
    f.start();
    f.join();
    System.out.println();
    System.out.println(a2);
    System.out.println(a3);
} else {
    System.out.println("Incorrect password");
}
sc.close();
}
}

```

```

Enter number : 6
Enter password: Shangirne451
Incorrect password

Process finished with exit code 0
|

```

```

Enter number : 6
Enter password: whatsupppp@2221
The entered number is: 6
0 1 1 2 3 5
Not an Armstrong number
It is an even number

Process finished with exit code 0

```

2. Create a program that has two arraylists of Strings (you can decide if you define these in your code directly or if the user should enter the strings).

Now create a new **arraylist** from these two in the following way:

- take as first value for the new list the first value from the first list
- take as next value for the new list the first value from the second list
- take as next value for the new list the second value from the first list ...

until all values from both lists are in the new list. (start with two lists of the same size => if you have a solution, extend it by handling lists of different sizes => as soon as one source list has no value anymore, just add all remaining values from the other list)

Example: Given two lists of same size, list 1 being ["a", "b", "c"] list 2 being ["x", "y", "z"]

resulting list: ["a","x","b","y","c","z"]

Example: Given two lists of different size list 1 being ["a", "b", "c", "x"] list 2 being ["y", "z"] ["a","y","b","z","c","x"].

quesonarraylist.java

```
import java.util.*;

public class quesonarraylist {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        ArrayList<String> l1=new ArrayList<>();
        ArrayList<String> l2=new ArrayList<>();
        ArrayList<String> l3=new ArrayList<>();
        int n1,n2;
        System.out.print("Enter number of elements in list 1: ");
        n1=in.nextInt();
        System.out.print("Enter number of elements in list 2: ");
        n2=in.nextInt();
        in.nextLine();
        System.out.println("Enter elements of list 1: ");
        for(int i=0;i<n1;i++) {
            String temp;
            System.out.print("Element " + (i+1) + ": ");
            temp=in.nextLine();
            l1.add(temp);
        }
        System.out.println("Enter elements of list 2: ");
        for(int i=0;i<n2;i++) {
            String temp;
            System.out.print("Element " + (i+1) + ": ");
            temp=in.nextLine();
            l2.add(temp);
        }
        if(l1.size() == l2.size()) {
```

```

        for(int i=0;i<l1.size();i++) {
            String temp;
            temp=l1.get(i);
            l3.add(temp);
            temp=l2.get(i);
            l3.add(temp); }
        System.out.println(l3);
    }
    else if(l1.size() > l2.size()) {
        for(int i=0;i<l1.size();i++) {
            String temp;
            for(int j=0;j<l2.size();j++) {
                temp=l1.get(i);
                l3.add(temp);
                temp=l2.get(j);
                l3.add(temp); }
        }
        Set<String> l4 = new LinkedHashSet<String>(l3);
        System.out.println(l4);
    }
    in.close();
}
}

```

```

Enter number of elements in list 1: 5
Enter number of elements in list 2: 5
Enter elements of list 1:
Element 1: x
Element 2: y
Element 3: u
Element 4: i
Element 5: o
Enter elements of list 2:
Element 1: r
Element 2: t
Element 3: q
Element 4: a
Element 5: b
[x, r, y, t, u, q, i, a, o, b]

```

Process finished with exit code 0

```
Enter number of elements in list 1: 6
Enter number of elements in list 2: 4
Enter elements of list 1:
Element 1: x
Element 2: y
Element 3: u
Element 4: i
Element 5: o
Element 6: t
Enter elements of list 2:
Element 1: w
Element 2: q
Element 3: s
Element 4: d
[x, w, q, s, d, y, u, i, o, t]

Process finished with exit code 0
|
```

3. Guess character

Write a program that creates an [ArrayList](#) of characters (choose as many and which you like).

Now let the user enter characters to guess which characters are in the [arraylist](#).

Let him continue guessing until he has entered all characters in the list (bonus point: print at the end how many guesses he took).

Example: Let the [arraylist](#) in my code contains the characters a,d,n

Guess a character

a

Correct

Guess a character

z

Wrong

Guess a character

d

Correct

Guess a character

n

Correct

You guessed all characters in 4 tries

guessCharacter.java

```
import java.util.*;
public class guessCharacter {
    public static void main(String[] args) {
        ArrayList<String> l = new ArrayList<String>();
        l.add("A");
        l.add("B");
        l.add("C");
        Scanner in = new Scanner(System.in);
        String ch;
        int count = 0;
        while (!l.isEmpty()) {
            System.out.print("Guess the character: ");
            ch = in.nextLine();
            if (ch.equals(l.get(0))) {
                l.remove(0);
                System.out.println("Correct");
            } else {
                System.out.println("Incorrect");
            }
            count++;
        }
        System.out.println("You guessed all the characters in " + count + " tries");
        in.close();
    }
}
```

```
Guess the character: F  
Incorrect  
Guess the character: H  
Incorrect  
Guess the character: A  
Correct  
Guess the character: R  
Incorrect  
Guess the character: B  
Correct  
Guess the character: T  
Incorrect  
Guess the character: C  
Correct  
You guessed all the characters in 7 tries  
  
Process finished with exit code 0
```