Remark

1. Array: group of elements with the same type

```java
import java.util.Arrays;

public class MainArr {
    public static void main(String[] args) {
        int[] ages = {18,20,35,19,15};
        double[] scores = new double[]{2.3, 50.5, 70.8, 88.8, 99,9};
        Group p = new Group("I4", null);
        Group[] groups = new Group[4];
        groups[0] = p;
        groups[1] = new Group("I4A", p);
        groups[2] = new Group("I4B", p);
        groups[3] = new Group("I4C", p);
        System.out.println(Arrays.toString(ages));
        System.out.println(Arrays.toString(scores));
        System.out.println(Arrays.deepToString(groups));

    }
}
class Group{
    String name;
    Group parent;
    public Group(String name, Group parent) {
        this.name = name;
        this.parent = parent;
    }
    @Override
    public String toString() {
        return String.format("%s%s", name, parent==null?"":"(in "+parent+")");
    }
}
```

2. Collections: Group of elements of compatible:

In package java.util has some classes working with collections: LinkedList<E>, ArrayList<E>, Stack<E>, and Vector<E>, HashSet<E>, PriorityQueue<E>, ArrayDeque<E>. Deque, also known as a double-ended queue, is a data structure where we can add and remove the elements from both ends of the queue.

Addition to these classes, there are also some classes that represent associative arrays or dictionary such as: Hashtable<K,V>, HashMap<K,V> where K is key and V is value. They are also be called Key-Value pair arrays because they use the Key to access to value does not index anymore.

Example: We have dictionary French-English and English-French:

Un: equals to One in English
Two: est égal à deux en français

```java
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Hashtable;
import java.util.List;

public class MainColls {
    public static void main(String[] args) {
        var ar = new ArrayList<String>(List.of("Bopha","Sokha"));
        System.out.println(ar.toString());

        var di = new Hashtable<String,String>();
        di.put("Un", "One");
        di.put("Deux", "Two");
        di.put("Trois", "Three");

        System.out.println(di.get("Deux"));

        var enFr = new HashMap<String, String>(){{
            put("Four","Quatre");
            put("Five","Cinq");
            put("Six","Six");
        }};

        System.out.println(enFr.get("Five"));
    }
}
```

TP07.1. Rectangle

We have a class Rectangle as below:

```java
public class Rectangle {
    int width;
    int height;
    public Rectangle(int width, int height) {
        this.width = width;
        this.height = height;
    }
    public int calculatePerimeter(){
        return (width + height) * 2;
    }
    public int calculateSurface(){
        return width * height;
    }
}
```

Create a class RectangleTest that is a Java application that test:

1. Create an instance of class Rectangle (call its constructor)
2. Display perimeter of it (the new created rectangle object)
3. Display surface of it (the new created rectangle object)

TP07.2. SMS Encrypt

Create a Java class represents SMS. SMS class contains:

- Attributes: Subject, From phone number, Receiver number, Type (Text, MMS), content, and status (new, read)
- Constructor (s): suggests 3 constructors

Then create a class named SMSList, that represents list of SMS and manage SMS in and out:

- Attributes: ArrayDeque<SMS>, and static field max_characters_per_sms=160 characters
- Operations/Methods: suggests 3 overloaded methods (add(SMS), add(String msg), add(String msg, String title))

Then write a program that will display a menu:

1. Send new SMS with Encrypted content using password method
2. View SMS detail
3. List SMSes
4. Remove SMSes by index
5. Quit

TP07.3. SMS Decryption

Using previous exercise as references and reuse classes of it. Create new project for SMS Decryption part. This part will be used by another user.

Then, write a program in Java will show a menu:

1. List all SMSes
2. View SMS Detail (decrypt content using password)

3. View readable SMSes (all SMS that can be decrypted using given password)
4. Remove SMSes by index
5. Quit

TP07.CE.1. SMS Tool

Using HashMap or Hashtable to store Username-Password pairs. Example:

sasa: 123

vanny: 123456

bopha: 7273

Create a program that let user to login using their username and password. After logging in, the user can view and send SMS without using password (because they already logging in). Example:

```
Please enter username: bopha
Password: 7273

---------- Welcome to private SMS app -----------
1. List all SMS
2. View SMS Detail
3. Send SMS
4. Remove SMS by index
5. Quit

Choose an option: 3
To username: sasa
Title: Secrets
Content (Enter END to end the content):
There is something secret I want to tell you.
If you wanna know, please reply.
END

:) Your message has been sent to sasa.
```