## EXERCISE 1

Write a program in Java which creates and executes two threads: one of the threads must write the text "**Hello**" 3 times over the screen, while the other must write the text "**Bye**" 3 times over the screen.

Restrictions:

- In this exercise you must not use vectors, lists or any other similar kind of structures.
- Each thread must be a different object from the same clase.

Hint:

- In order to make the threads write a different text there are several aproaches. In this exercise it is adviced to use the constructor to receive an identifier and depending on its value perform a different action.

## EXERCISE 2

Modify the previous exercise to make the threads inherit from the class **java.awt.Frame**

## EXERCISE 3

Write a program in Java according to the following instructions:

- The program must create 5 threads with a vector, list or any other similiar structure. The threads must receive a different identifier starting from 0.
- The main thread must write the following text "**The main thread has finished**"
- Each of the threads created must perform the following actions:
    1. Print the following text "**Hello, this is the thread *n* and I am going to sleep *x* milliseconds**".
        - *n* represents the identifier of the thread
        - *x* represents a random number between 500 and 10.000.
    2. Sleep the time specified
    3. Print the following message "**Hello, this is the thread *n* and I am awaken.**".
        - *n* represents the identifier of the thread

## EXERCISE 4

Copy and paste the following code:

```
public class Exercise04_ThreadPriorities {

        static class MyThread extends Thread {
                String name;

                public MyThread(String name) {
                        this.name = name;
                }

                @Override
                public void run() {
                        for (int i = 0; i < 10000; i++) {
                                System.out.println(this.name + "\t" + i);
                                //yield();
                        }
                }
        }

        public static void main(String[] args) {
                MyThread t1 = new MyThread("T 1:");
                MyThread t2 = new MyThread("T 2:");

                t1.start();
                t2.start();
        }

}
```

Answer the following questions:

1. Which is the output of the program? Is the output of the program the expected result?

2. Uncomment the commented line and execute it again.

   Do you see any significative difference? Which is the purppose of the method "**yield**"?

3. Comment again the line which calls the method "**yield**". Modify the code to assign the maximum priority to the thread "**t1**" (you may have to search for it on Internet) and run again the program.

   Do you see any significative difference?

4. Now assign the highest priority to the thread "**t2**" and the lowest to the thread "**t1**".
   Do you see any significative difference?

## EXERCISE 5

Estudy the code of the file "**Exercise05_WaitingWindow**". This program contains the code of a program which generates a user interface with a button and a combo box.

When the program is run if someone clics on the button the interface remains frozen for 5 seconds. In this moment a message is displayed on the consoled to inform the user that the window is going to be frozen for 5 seconds. After the 5 seconds a message is displayed to inform the user that the waiting time is over.

Make sure that you understand how the program works before continuing the exercise.

Modify the code to allow the user to interact with the window during the 5 seconds. This means that the user must be allowed to clic on the combo and that the interface must response properly.

Hint:

- It is possible to create a class to "make the button sleep" for 5 seconds
- The method "setEnable" might be useful to check the status of the button