ACTIVITAT
Objectius: - Saber com definir objectes Singleton i com ignorar aquest patró
Instruccions: - Es tracta d'un treball individual, no s'admet cap tipus de còpia. - Responeu a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai. - Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.
Criteris d'avaluació: - Cada pregunta té el mateix pes sobre 90% - Les metodologies de treball, organització personal i participació conten un 10%
Entrega: - Aquest document amb les explicacions i captures necessàries i els arxius adjunts necessaris del codi que es demana - El nom dels arxius adjunts a entregar seràn: nomicognom-nomicognom.zip

Noms i Cognoms: Patricio André Rojas Condori

Materials:

Necessiteu un entorn de desenvolupament en JAVA
Feu servir Google per buscar els tutorials que us serveixin millor
Creeu els arxius a la carpeta 'src' del projecte i executeu amb els scripts './build.sh' i '.\build.ps1'

Tasques:

- Preparació - Crea un arxiu 'Main.java' amb un menú per cridar cada un dels altres arxius amb funció 'main' d'aquesta activitat. Aquí tens un exemple que hauras d'adaptar al què demana l'enunciat de cada exercici:

```
import java.io.IOException;
import java.util.*;

public class Main {

  static Scanner in = new Scanner(System.in); // System.in és global

  // Main
  public static void main(String[] args) throws InterruptedException, IOException {
    boolean running = true;
    while (running) {
      String menu = "Escull una opció:";
      menu = menu + "\n 0) PR430Main";
      menu = menu + "\n 1) PR431Main";
      // Adapta aquí les altres classes de l'exercici (PR432Main…)
      menu = menu + "\n 100) Sortir";
      System.out.println(menu);

      int opcio = Integer.valueOf(llegirLinia("Opció:"));
      try {
        switch (opcio) {
          case 0: PR430Main.main(args); break;
          case 1: PR431Main.main(args); break;
          // Adapta aquí les altres classes de l'exercici (PR432Main…)
          case 100: running = false; break;
          default: break;
        }
      } catch (Exception e) {
          System.out.println(e);
      }
    }
    in.close();
  }

  static public String llegirLinia (String text) {
    System.out.print(text);
    return in.nextLine();
  }
}
```

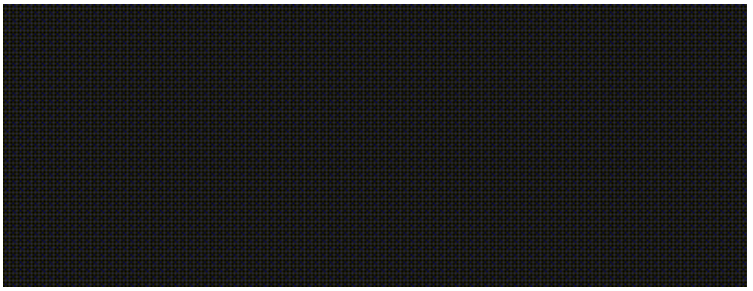
El nostre arxiu Main.java es el següent:

```java

public class Main {


    static Scanner in = new Scanner(System.in); // System.in és global


    // Main
    public static void main(String[] args) throws InterruptedException, IOException {
      boolean running = true;
      while (running) {
        String menu = "Escull una opció:";
        menu = menu + "\n 1) PR430Main";
        menu = menu + "\n 2) PR431Main";
        // Adapta aquí les altres classes de l'exercici (PR432Main…)
        menu = menu + "\n 3) Sortir";
        System.out.println(menu);


        int opcio = Integer.valueOf(llegirLinia("Opció:"));
        try {
          switch (opcio) {
            case 1: PR430Main.main(args); break;
            case 2: PR431Main.main(args); break;
            // Adapta aquí les altres classes de l'exercici (PR432Main…)
            case 3: running = false; break;
            default: break;
          }
        } catch (Exception e) {
            System.out.println(e);
        }
      }
      in.close();
    }
```

- Exercici 0 - Crea un programa "PR430Main.java" que instancii 3 objectes amb dades diferents de la classe "PR430Objecte.java" amb 3 segons de diferència.

L'objecte "PR430Objecte" ha de tenir les variables privades 'nom', 'cognom', 'edat' com a privades NO estàtiques i ha de seguir el model Singleton.

Mostra les dades de cada instància al final (caldrà sobreescriure toString)



Primer de tot creem la classe de la qual farem els objectes, incloses les funcions requerides com els constructors i el toString() amb les pertinents modificacions. Veiem com el constructor esta en privat pero pot ser accés mitjançant una funció que, ens retorna la primera instancia en tots casos, incloent un delay de 3 segons.

```java
final class PR430Objecte {
    private static PR430Objecte instance;
    private String nom;
    private String cognom;
    private int edat;

    private PR430Objecte(String nom, String cognom, int edat) {
        this.nom = nom;
        this.cognom = cognom;
        this.edat = edat;

    }

    public static PR430Objecte getInstance(String nom, String cognom, int edat) {
        // Creacion del delay
        try {
            Thread.sleep(3000);
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        }

        if (instance == null) {
            instance = new PR430Objecte(nom, cognom, edat);
        }
        return instance;
    }

    @Override
    public String toString() {
        return "Nom: " + nom + "    Cognom: " + cognom + "    edat: " + edat;
    }

}
```

MS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash

Aquesta és la funció Main(), on podem veure com s'instancien els objectes amb diferents dades i seguidament es mostren per consola. Com estem fent servir el model Singleton, tot i que en principi em instanciat diferents objectes el programa ens retorna únicament el primer.

```
Run | Debug
public static void main(String[] args) {
    PR430Objecte[] arrayObjectes = new PR430Objecte[3];

    System.out.println("Inicialitzant 0...");
    PR430Objecte pepe = PR430Objecte.getInstance(nom:"Pepe",cognom:"Ruiz", edat
    arrayObjectes[0] = pepe;

    System.out.println("Inicialitzant 1...");
    PR430Objecte maria = PR430Objecte.getInstance(nom:"Maria",cognom:"Antonieta
    arrayObjectes[1] = maria;

    System.out.println("Inicialitzant 2...");
    PR430Objecte jhon = PR430Objecte.getInstance(nom:"Victor",cognom:"Rojas", e
    arrayObjectes[2] = jhon;

    for (PR430Objecte ob: arrayObjectes) {
        System.out.println(ob.toString());
    }
}
```

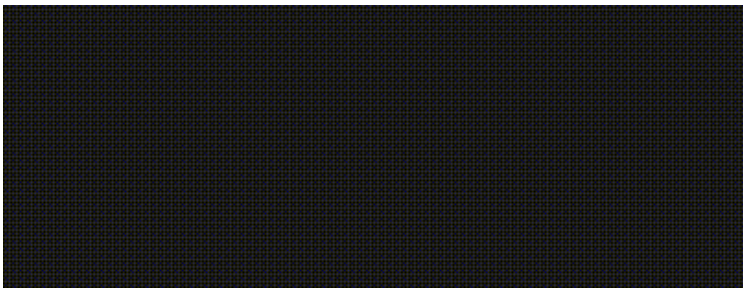
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

Escull una opció:
 1) PR430Main
 2) PR431Main
 3) Sortir
Opció:1
Inicialitzant 0...
Inicialitzant 1...
Inicialitzant 2...
Nom: Pepe   Cognom: Ruiz   edat: 20
Nom: Pepe   Cognom: Ruiz   edat: 20
Nom: Pepe   Cognom: Ruiz   edat: 20
Escull una opció:
 1) PR430Main
 2) PR431Main
 3) Sortir
Opció:
```

- Exercici 1 - Crea un programa "PR431Main.java" que instancii 3 objectes amb dades diferents de la classe "PR431Objecte.java" amb 3 segons de diferència i **aconseguint 3 instàncies diferents**, ignorant el fet que es tracta d'un objecte que implementa el model 'Singleton'.

L'objecte "PR431Objecte" ha de tenir les variables privades 'nom', 'cognom', 'edat' com a privades NO estàtiques i ha de seguir el model Singleton (com l'exercici anterior).

Mostra les dades de cada instància al final (caldrà sobreescriure toString)

Pots crear una funció 'getNewDestroyedInstance' que retorni una instància 'hackejada' de Singleton, per no anar repetint codi.



Molt semblant al primer, però amb la diferència que creem una nova funció amb el nom de **getNewDestroyedInstance()**

```
static PR431Objecte getNewDestroyedInstance (String nom, String cognom, int edat) {

    PR431Objecte result = null;
    try {
        Constructor<?>[] constructors = PR431Objecte.class.getDeclaredConstructors();
        for (Constructor<?> constructor : constructors) {
            //Below code will destroy the singleton pattern
            constructor.setAccessible(true);
            result = (PR431Objecte) constructor.newInstance(nom, cognom, edat);
            break;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    // Creacion del delay
    try {
        Thread.sleep(3000);
    } catch (InterruptedException ex) {
        ex.printStackTrace();
    }
    return result;
}
```

Aquest mètode el que fa és fer el constructor de l'objecte accessible per després crear un nou objecte i retornar-lo, cosa que seria impossible fer amb la funció anterior de **getInstance()**.