Certamen Final

Todas las preguntas tienen igual puntaje.

1.- Responda brevemente:

a) Mencione dos casos en que se use palabra reservada super.

Para invocar desde un constructor la ejecución de otro constructor.
Para invocar la implementación en la clase base de un método sobremontado en la clase derivada.

b) ¿Qué diferencia existe entre crear un nuevo proceso y crear un nuevo hilo?

En ambos casos se crean dos flujos de ejecución que corren en forma concurrente (una CPU) o paralela (2 ó más CPUs). La diferencia principal es que los hilos comparten los espacios de memoria de datos mientras que los procesos no. Por ello los hilos no requieren de mecanismos de compartición de memoria.

Además los hilos son mucho más rápidos de crear que los procesos dado que no se requiere hacer una copia de los espacios de datos.

c) ¿Por qué los nombres de los programas en Java deben tomar el nombre de la clase que en él se define?

Se hace para permitir a la máquina virtual saber dónde se encuentra el código que implementa las clases que son usadas en otros archivos. A partir del nombre de la clase, la máquina virtual busca la implementación en el propio archivo, si no está, la busca en un archivo externo con ese nombre.

d) ¿Cuándo usted debe crear un paquete (package) con los programas que usted ha escrito?

Se crea un paquete cuando queremos distinguir los nombres que hemos dado a nuestras clases de aquellas presentes en otros paquetes, ya sea de las clases estándares de Java u otros módulos del sistema.

e) ¿Cuándo se usa la palabra reservada static?

Declaramos un miembro dato o función como static cuando representa un dato cuyo valor es compartido entre todas las instancias de la clase. Un método es static cuando corresponde a una operación que usa sólo datos estáticos u otros métodos estáticos.

2.- Escriba un programa en Java que solicite un número vía una ventana de diálogo - JOptionPane - y escriba en consola si el número es par o impar.
Muestre los pasos para compilar y ejecutar su programa.

```
import javax.swing.*;

public class p2
{
  public static void main(String[] args)
  {
    String input = JOptionPane.showInputDialog("Ingrese su numero");
    int number = Integer.parseInt(input);
```

```
    if (number%2==0)
      System.out.println(number + " es par.");
    else
      System.out.println(number + " es impar.");
    System.exit(0);
  }
}
```

El programa debe llamarse p2.java

Para compilar:
% javac p2.java
para ejecutarlo
% java p2

3.- Modifique el programa adjunto para que el programa haga lo mismo pero la clase SimpleThread no herede de Thread.

```
public class SimpleThread extends Thread {
  private int countDown = 5;
  private int threadNumber;
  private static int threadCount = 0;
  public SimpleThread() {
    threadNumber = ++threadCount;
    System.out.println("Making " + threadNumber);
  }
  public void run() {
    while(true) {
      System.out.println("Thread " +
        threadNumber + "(" + countDown + ")");
      if(--countDown == 0) return;
    }
  }
  public static void main(String[] args) {
    for(int i = 0; i < 5; i++)
      new SimpleThread().start();
    System.out.println("All Threads Started");
  }
}
```

```
public class SimpleThread implements Runnable {
  private int countDown = 5;
  private int threadNumber;
  private static int threadCount = 0;
  public SimpleThread() {
    threadNumber = ++threadCount;
    System.out.println("Making " + threadNumber);
  }
  public void run() {
    while(true) {
      System.out.println("Thread " +
        threadNumber + "(" + countDown + ")");
      if(--countDown == 0) return;
    }
  }
  public static void main(String[] args) {
    for(int i = 0; i < 5; i++)
      new Thread(new SimpleThread()).start();
    System.out.println("All Threads Started");
  }
}
```

4.- Haga un programa que muestre una ventana que incluya un label en que se muestra el valor de un contador, el cual incrementa su valor cada segundo. No se preocupe por la forma en que el programa termina.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class Contador extends JFrame implements ActionListener{
  int contador;
  JLabel label;
  Timer t;

  public Contador() {
    setTitle("Contador de Segundos");
    setSize( 100,50);
    contador = 0;
    label = new JLabel(" "+contador);  // " "+contador para transformarlo a String
    addWindowListener( new WindowAdapter() {
        public void windowClosing( WindowEvent e) {
          System. exit( 0);
        }
```

```
      }
                  );
    getContentPane().add(label);
    t = new Timer(1000, this);
    setVisible(true);
    t.start();
  }

  public void actionPerformed(ActionEvent event){
    contador++;
    label.setText(" "+contador);
  }

  public static void main( String[] args) {
    new Contador();
  }
}
```