COMPONENTES DEL GRUPO
Joaquín Gálvez Díaz (Portavoz)
Jorge Urbelz Alonso-Cortés

HILO GENERADOR

```
arrayValoresCompartido: arrayList de enteros privado volatil;

Constructor HiloGenerador(arrayValoresCompartido: arrayList de enteros)
begin
      arrayValoresCompartido <- arrayValoresCompartido;
end

Procedimiento run()
begin
      r: random;

      r <- Creacion Random;

      for i in 0..9 hacer
            Añadir a la coleccion arrayValoresCompartido(valor entero
random 0-100, valor entero random 1-3, valor entero random 0-100,
                                                valor entero random 1-3,
valor entero random 0-100, valor entero random 1-3,
                                                valor entero random 0-100,
valor entero random 1-3, valor entero random 0-100,
                                                valor entero random 1-3,
valor entero random 0-100);
      fin_for
end
```

HILO CONSUMIDOR

```
id: entero privado;
arrayValoresCompartido: arrayList de enteros privado volatil;
arrayResultadosCompartido: array de enteros privado volatil;

Constructor HiloConsumidor(id: entero, arrayValoresCompartido: arrayList
de enteros, arrayResultadosCompartido: array de enteros)
begin
      id <- id;
      arrayValoresCompartido <- arrayValoresCompartido;
      arrayResultadosCompartido <- arrayResultadosCompartido;
end

Procedimiento run()
begin
      resultado, steps, operador, operando, i: entero;

      resultado <- Obtener valor de arrayValoresCompartido[id*11];
      steps <- 0;
```

```
      i <- id*11 + 1;

      mientras steps < 5 hacer
            operador <- Obtener valor de arrayValoresCompartido[i];
            i <- i + 1;
            operando <- Obtener valor de arrayValoresCompartido[i];
            i <- i + 1;

            switch(operador)
                  caso 1:
                        resultado <- resultado + operando;
                        romper;
                  caso 2:
                        resultado <- resultado - operando;
                        romper;

                  caso 3:
                        resultado <- resultado * operando;
                        romper;
            fin_switch

            steps <- steps + 1;

      fin_mientras

      Imprimir("Hilo" + id + " : " + resultado + "\n");
      arrayResultadosCompartido[id] <- resultado;

end



***HILO SUMADOR***

arrayResultadosCompartido: array de enteros privado volatil;

Constructor HiloSumador(arrayResultadosCompartido: array de enteros)
begin
      arrayResultadosCompartido <- arrayResultadosCompartido;

end

Procedimiento run()
begin
      total: entero;

      total <- 0;

      for valor:entero in arrayResultadosCompartido hacer
            total <- total + valor;

      fin_for

      Imprimir("Total: " + total);
```

```
end



***PROGRAMA PRINCIPAL***

arrayValoresCompartido: arrayList de enteros;
arrayResultadosCompartido: array de enteros;

g: HiloGenerador;
c0: HiloConsumidor;
c1: HiloConsumidor;
c2: HiloConsumidor;
c3: HiloConsumidor;
c4: HiloConsumidor;
c5: HiloConsumidor;
c6: HiloConsumidor;
c7: HiloConsumidor;
c8: HiloConsumidor;
c9: HiloConsumidor;
s: HiloSumador;

tg: Hilo;
tc0: Hilo;
tc1: Hilo;
tc2: Hilo;
tc3: Hilo;
tc4: Hilo;
tc5: Hilo;
tc6: Hilo;
tc7: Hilo;
tc8: Hilo;
tc9: Hilo;
ts: Hilo;

arrayValoresCompartido <- Creacion ArrayList(110);
arrayResultadosCompartido <- {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

g <- Creacion HiloGenerador(arrayValoresCompartido);
c0<- Creacion HiloConsumidor(0, arrayValoresCompartido,
arrayResultadosCompartido);
c1<- Creacion HiloConsumidor(1, arrayValoresCompartido,
arrayResultadosCompartido);
c2<- Creacion HiloConsumidor(2, arrayValoresCompartido,
arrayResultadosCompartido);
c3<- Creacion HiloConsumidor(3, arrayValoresCompartido,
arrayResultadosCompartido);
c4<- Creacion HiloConsumidor(4, arrayValoresCompartido,
arrayResultadosCompartido);
c5<- Creacion HiloConsumidor(5, arrayValoresCompartido,
arrayResultadosCompartido);
c6<- Creacion HiloConsumidor(6, arrayValoresCompartido,
arrayResultadosCompartido);
```

```
c7<- Creacion HiloConsumidor(7, arrayValoresCompartido,
arrayResultadosCompartido);
c8<- Creacion HiloConsumidor(8, arrayValoresCompartido,
arrayResultadosCompartido);
c9<- Creacion HiloConsumidor(9, arrayValoresCompartido,
arrayResultadosCompartido);
s<- Creacion HiloSumador(arrayResultadosCompartido);

tg <- Creacion Hilo(g);
tc0 <- Creacion Hilo(c0);
tc1 <- Creacion Hilo(c1);
tc2 <- Creacion Hilo(c2);
tc3 <- Creacion Hilo(c3);
tc4 <- Creacion Hilo(c4);
tc5 <- Creacion Hilo(c5);
tc6 <- Creacion Hilo(c6);
tc7 <- Creacion Hilo(c7);
tc8 <- Creacion Hilo(c8);
tc9 <- Creacion Hilo(c9);
ts <- Creacion Hilo(s);

tg.iniciar();

intentar
      tg.unir();

fin_intentar

atrapar(e: ExcepcionInterrumpida)
      e.imprimirTrazaPila();

fin_atrapar

tc0.iniciar();
tc1.iniciar();
tc2.iniciar();
tc3.iniciar();
tc4.iniciar();
tc5.iniciar();
tc6.iniciar();
tc7.iniciar();
tc8.iniciar();
tc9.iniciar();

intentar
      tc0.unir();
      tc1.unir();
      tc2.unir();
      tc3.unir();
      tc4.unir();
      tc5.unir();
      tc6.unir();
      tc7.unir();
      tc8.unir();
```

```
      tc9.unir();

fin_intentar

atrapar(e: ExcepcionInterrumpida)
      e.imprimirTrazaPila();

fin_atrapar

ts.iniciar();
```

RECURSOS NO COMPARTIBLES Y CONDICIONES DE SINCRONIZACIÓN
El arrayValoresCompartido es un recurso no compartible, puesto que el HiloGenerador tiene que introducir los valores aleatorios antes de que los
HiloConsumidor lean para empezar los cálculos(condición de sincronización).

El arrayResultadosCompartido también lo es, porque el HiloSumador no puede leer hasta que todos los HiloConsumidor hayan introducido el valor calculado
(condición de sincronización).