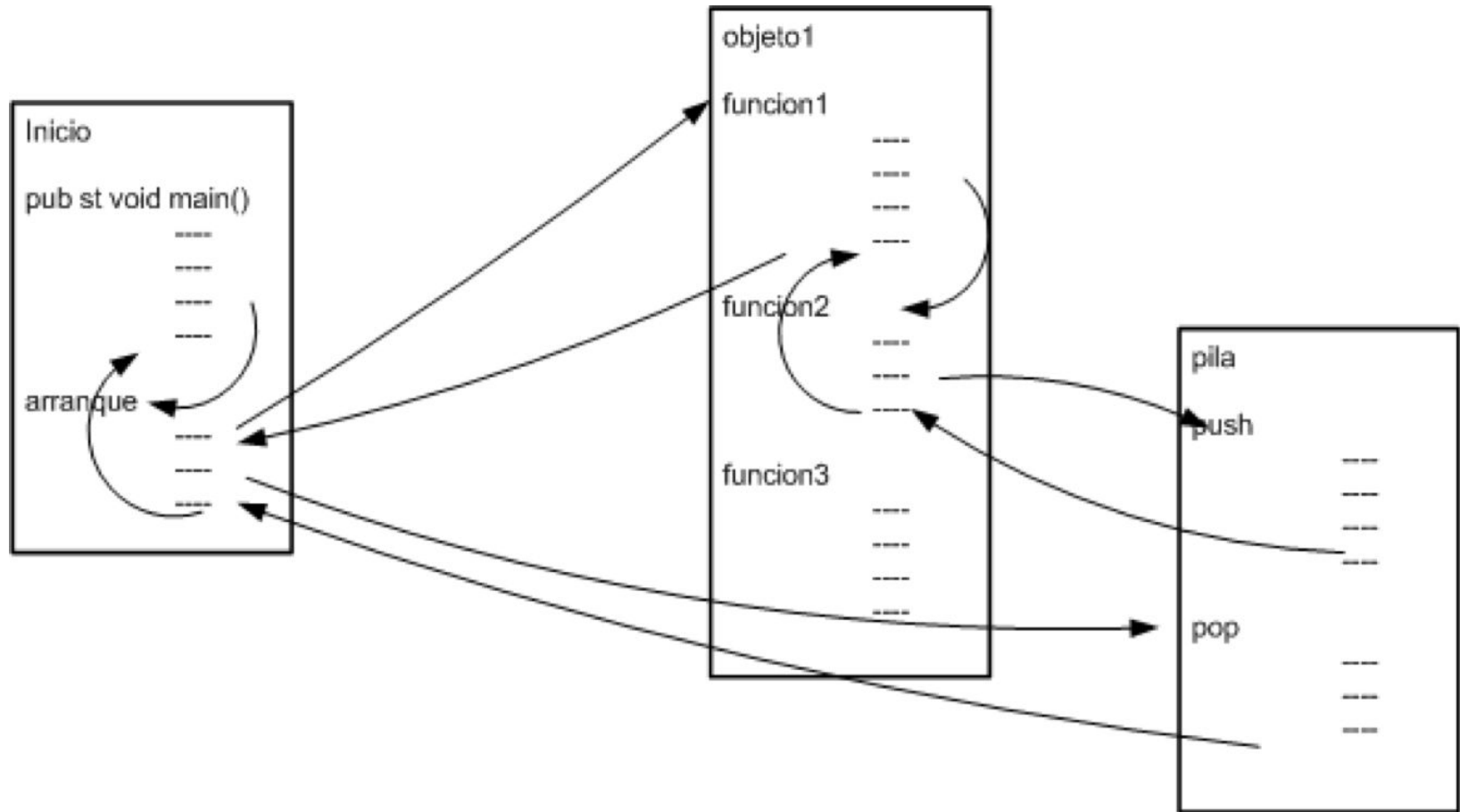


# Clase 5

Hilos

# Único camino de ejecución



# Hilos

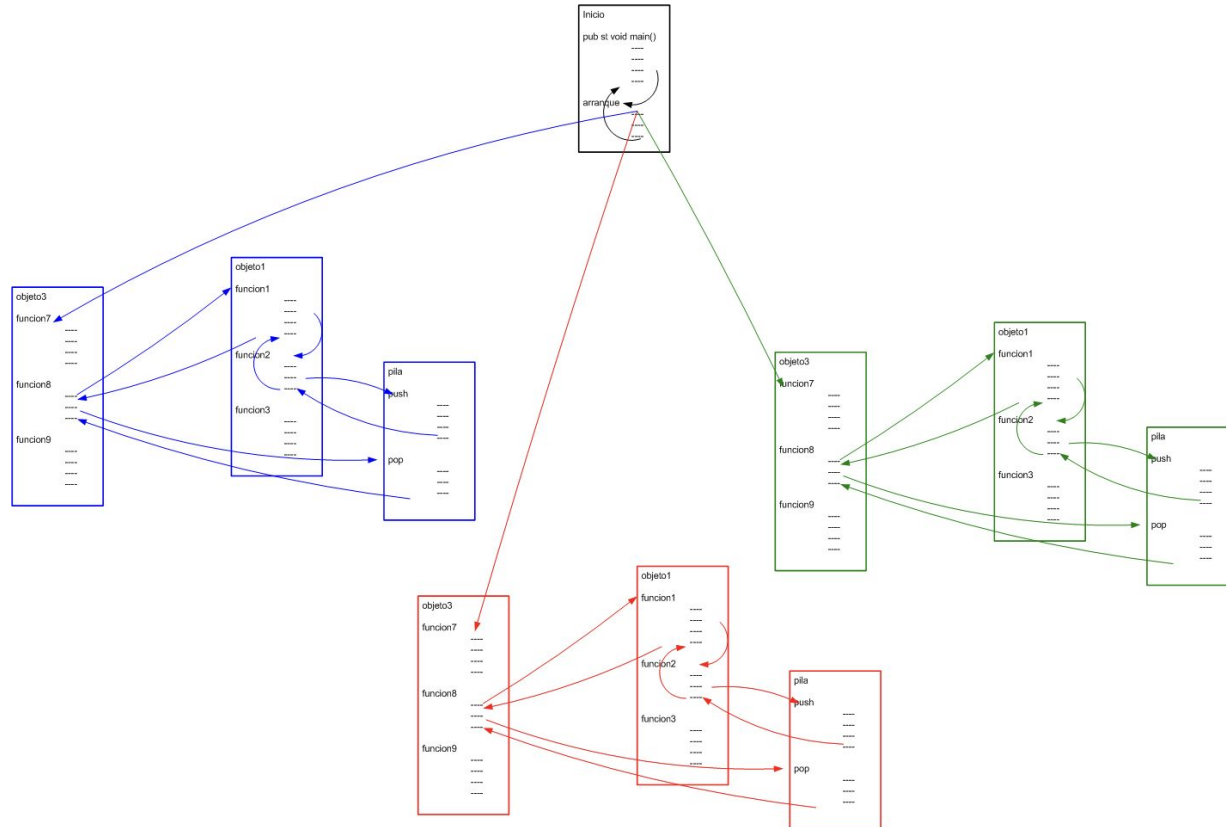
## **Definición**

Bloques de código que se pueden ejecutar simultáneamente dentro de la JVM.

## **Ventajas**

- Comparten memoria y recursos, es más rápido y eficiente.
- La creación de hilos es más rápida y eficiente que los procesos.
- Es más escalable el manejo de hilos que los procesos.
- La sincronización de datos es más sencilla con hilos.
- Se utilizan para ejecutar tareas en segundo plano.

# Hilos



# Definición de hilos

Los hilos se pueden generar de 3 formas

Forma “original”

- Extendiendo la clase Thread
- Implementando la interfaz Runnable

Forma más reciente

- Implementando la interfaz Callable

# Definición de hilos - Extensión de Thread

```
public class Hilo1 extends Thread {
    protected String nombre;
    protected int demora;
    public Hilo1() {}
    public Hilo1(String nombre, int demora) {
        this.nombre = nombre;
        this.demora = demora;
    }

    @Override
    public void run() {
        for(int i=0; i < 10; i++) {
            System.out.println("Hilo "+this.nombre);
            try {
                Thread.sleep(this.demora);
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
        }
    }
}
```

# Ejecución de hilos - Extensión de Thread

```
import hilos.Hilo1;

public class Main {
    public static void main(String[] args) {
        Main m = new Main();
        m.ejecutar();
    }

    public void ejecutar() {
        Hilo1 h1 = new Hilo1("extends 1",500);
        h1.start();
    }
}
```

# Definición de hilos - Implementación Runnable

```
public class Hilo2 implements Runnable{
    protected String nombre;
    protected int demora;
    public Hilo2() {}

    public Hilo2(String nombre, int demora) {
        this.nombre = nombre;
        this.demora = demora;
    }

    @Override
    public void run() {
        for(int i=0; i < 10; i++) {
            System.out.println("Hilo "+this.nombre);
            try {
                Thread.sleep(this.demora);
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
        }
    }
}
```



# Ejecución de hilos - Implementación Runnable

```
import hilos.Hilo2;

public class Main {
    public static void main(String[] args) {
        Main m = new Main();
        m.ejecutar();
    }

    public void ejecutar() {
        Hilo2 h2_implements = new Hilo2("implements 1", 900);
        Thread h2 = new Thread(h2_implements);
        h2.start();
    }
}
```

# Ejecución de hilos

```
import hilos.Hilo1;
import hilos.Hilo2;

public class Main {
    public static void main(String[] args) {
        Main m = new Main();
        m.ejecutar();
    }

    public void ejecutar() {
        Hilo1 h1 = new Hilo1("extends 1",500);
        h1.start();
        Hilo2 h2_implements = new Hilo2("implements 1", 900);
        Thread h2 = new Thread(h2_implements);
        h2.start();
    }
}
```

# Ciclo de vida de un hilo

