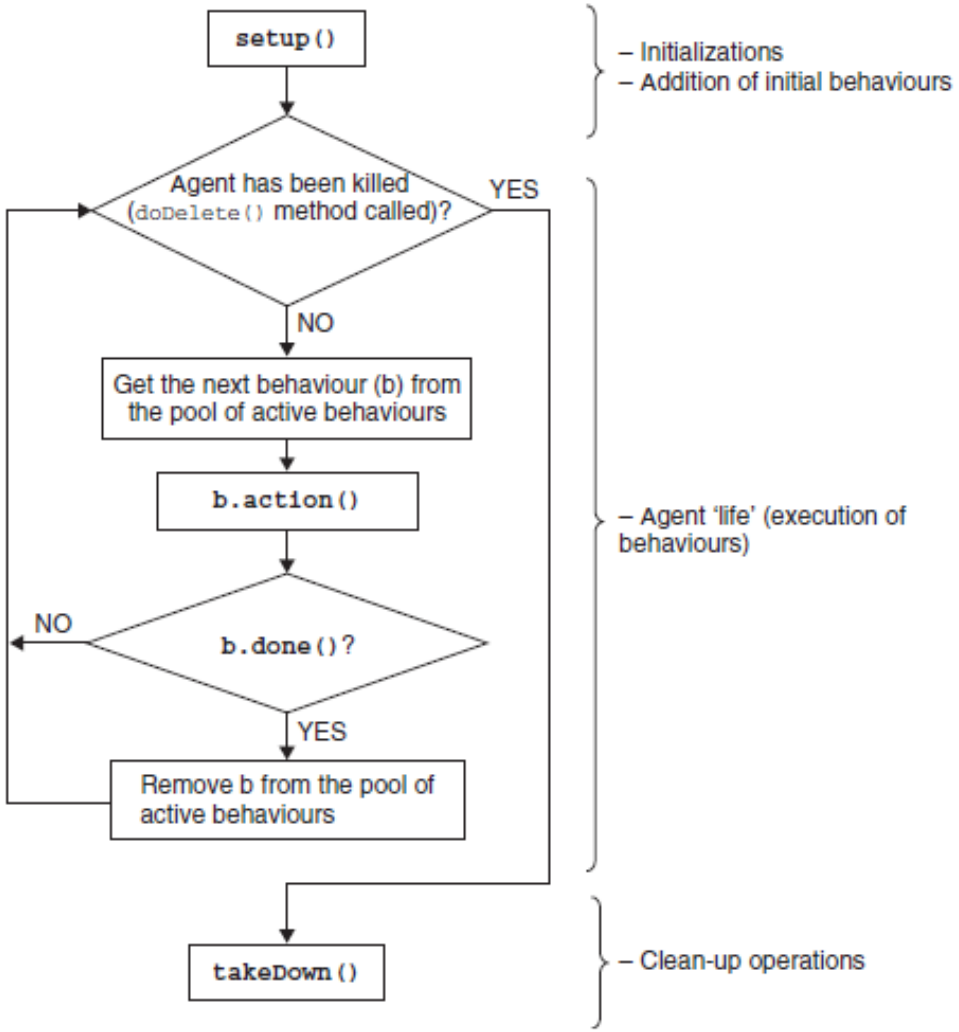
Behaviours

Jade: behaviour (1)

Un behaviour rappresenta un task che l'agente esegue.

- è un oggetto della classe jade.core.behaviours.Behaviour;
- è aggiunto ad un agente grazie al metodo *addBehaviour()*;
- il metodo *action()* implementa le operazioni da eseguire nel behaviour;
- il metodo *done()* ritorna un booleano che indica se il behaviour è completato

Jade: agenti e behaviour



Jade: scheduling dei behaviour

- thread singolo per ogni agente;
- ogni behaviour è eseguito nello stesso thread (nessun problema di sincronizzazione)

La schedulazione dei behaviour è **non-preemptive**: quando un behaviour è schedulato per l'esecuzione, il metodo *action()* esegue fino al termine.

Jade: OneShotBehaviour

```
public class MyOneShotBehaviour extends OneShotBehaviour {
  public void action() {
    // perform operation X
  }
}
```

X viene eseguita una sola volta.

done() ritorna true.

Jade: CyclicBehaviour

```
public class MyCyclicBehaviour extends CyclicBehaviour {
  public void action() {
    // perform operation Y
  }
}
```

Y viene eseguita continuamente e **per sempre**.

done() ritorna false.

Jade: WakerBehaviour

```
this.myAgent.addBehaviour(new WakerBehaviour(this.myAgent, 1000){
    public void onWake(){
        // Perform operation X
    }
});
```

X viene eseguita una sola volta, dopo il *timeout* in ms.

done() ritorna true (solamente dopo il timeout).

Jade: TickerBehaviour

```
public class MyAgent extends Agent {
  protected void setup() {
    addBehaviour(new TickerBehaviour(this, 10000) {
      protected void onTick() {
        // perform operation Y
      }
    } );
  }
}
```

Y viene eseguita periodicamente (period in ms).

done() ritorna false.