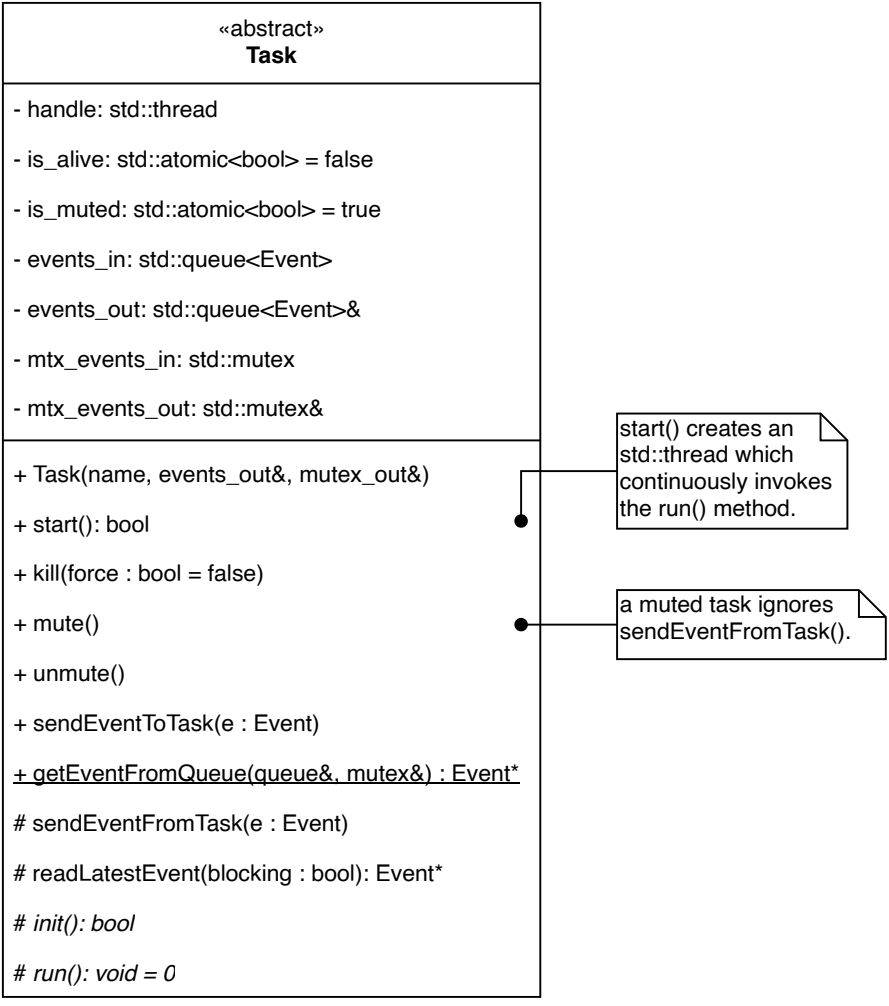
«abstract»
Task
- handle: std::thread
- is_alive: std::atomic<bool> = false
- is_muted: std::atomic<bool> = true
- events_in: std::queue<Event>
- events_out: std::queue<Event>&
- mtx_events_in: std::mutex
- mtx_events_out: std::mutex&
+ Task(name, events_out&, mutex_out&)
+ start(): bool
+ kill(force : bool = false)
+ mute()
+ unmute()
+ sendEventToTask(e : Event)
+ getEventFromQueue(queue&, mutex&) : Event*
# sendEventFromTask(e : Event)
# readLatestEvent(blocking : bool): Event*
# init(): bool
# run(): void = 0
start() creates an std::thread which continuously invokes the run() method.
a muted task ignores sendEventFromTask().