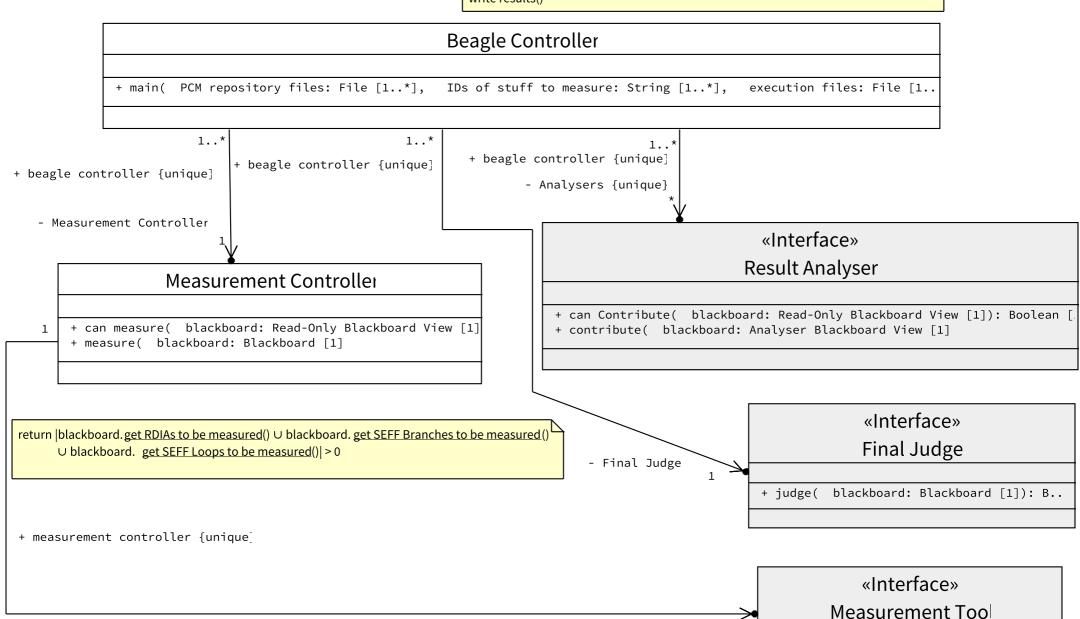
```
prepare input artefacts()
finished := false
readOnlyBlackboard := Read-Only Blackboard View.construct(blackboard)
analyserBlackboard := <u>Analyser Blackboard View.construct(blackboard)</u>
while (¬finished) do
    if Measurement Controller.can measure (readOnlyBlackboard) then
           Measurement Controller. measure()
          analysersIterator := Analysers.iterator()
          while ¬analysersIterator.current(). can Contribute(readOnlyBlackboard) do
               analysersIterator.next()
          od
         if analysersIterator.current(). <a href="mailto:cancontribute">can Contribute</a>(readOnlyBlackboard)
               analysersIterator.current(). contribute(analyserBlackboard)
               finished = Final Judge.judge(blackboard)
         fi
    fi
od
write results()
```



- Measurement Tools {unique}

+ measure(order: Measurement Order [1]