```
Work management
                                                        Calls flush() after managing each communication phase.
                   Manager
- workers : Matrix<Worker>
- comMan : CommunicationManager
                                                       Memory model
+Manager(n : int, p : int, numPhases : int,
                                                                 CommunicationManager
  initialMemory : Matrix<PrivateMemory>,
  workerClass : Class<? extends Worker>) : Manager
                                                       privateMemories : Matrix<PrivateMemory>
-startWorkerExecution(phase : int,
                                                      +MemoryController(p : int, privateMemories
      phaseType : WorkerPhase) : List<...>
                                                        : Matrix<PrivateMemorv>) : CommunicationManager
-checkForWorkerFailure(...) : void
                                                      +broadcastRow(i : int, j : int, value : Number)
+doWork() : void
                                                      +broadcastCol(i : int, j : int, value : Number)
+getResult(label : String) : Matrix<Number>
                                                      +receiveRowBroadcast(i : int, j : int, label : String)
                                                      +receiveColBroadcast(i : int, j : int,label : String)
   Uses the factory to create the workers
                                                      +sendData(sendI : int, sendJ : int,
                                                           reciveI : int, receiveJ : int, value : Number)
                                                      +receiveData(i : int, j : int, label : String)
                                                      +flush()
             WorkerFactory
+WorkerFactory(workerClass
                                                                                            Modifies worker's
                                                          Defines message-passing
  : Class<? extends Worker>) : WorkerFactory
                                                                                            private memory
                                                          interface using the
                                                                                            during execution
~init(comMan : communicationManager) : void
                                                          communication manager's
                                                                                            of flush()
~createWorker(i : int, j : int,
                                                          communication methods
   memory : PrivateMemory, ...) : Worker
                                                                              PrivateMemory
  Instantiate objects of some
  specified class that implements
                                                                    +set(label : String, n : Number,
  the Worker interface
                                                                       i : int = 0, j : int = 0) : void
                                                                    +get(label : String,
                                                                       i : int = 0, j : int = 0) : Number
                  Worker
 -memory : PrivateMemory
                                                                     Implements the algorithm by overrding
 -comMan : communicationManager
                                                                     the four phases, having access to:
 +Worker(...) : Worker
                                                                     * this.i, this.j, this.n, this.p
 +initialisation(): void
                                                                     * read(...)
 +communicationBefore(phase : int) : void
                                                                     * write(...)
 +computation(phase : int) : void
                                                                     * send(...)
 +communicationAfter(phase : int) : void
                                                                     * receive(....)
 #read(label : String) : Number
                                                                     * broadcastRow(...)
 #write(label : String, n : Number) : void
                                                                     * broadcastCol(...)
 #send(i : int, j : int, n : Number) : void
                                                                     * receiveRowBroadcast(...)
 #receive(label : String) : void
                                                                     * receiveColBroadcast(...)
 #rowBroadcast(n : Number) : void.
 #colBroadcast(n : Number) : void
 #receiveRowBroadcast(label : String) : void
                                                            FoxOtto
                                                                               GeneralisedFoxOtto
 #receiveColBroadcast(label : String) : void
```

Parallel system simulation