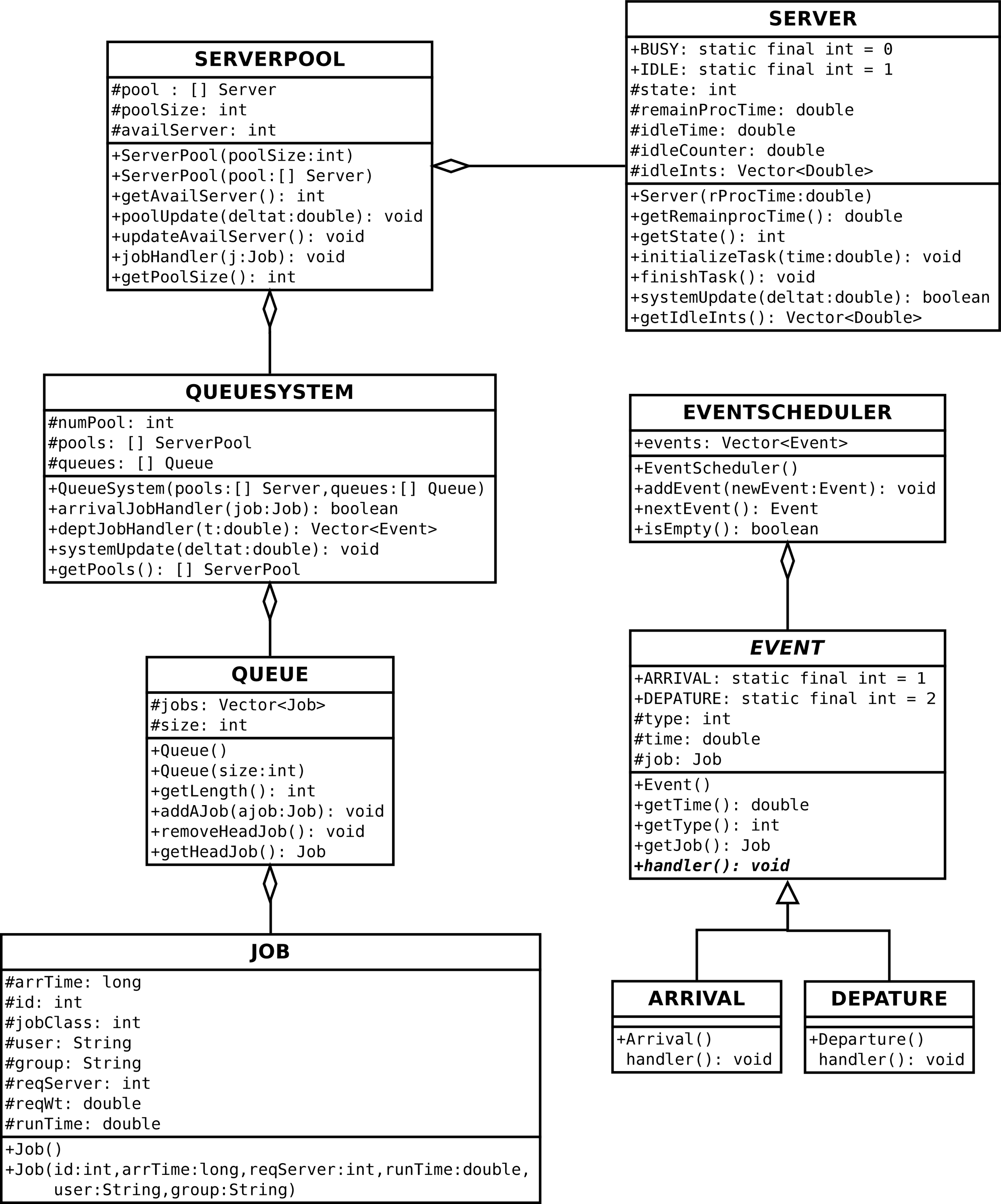
SERVERPOOL
#pool : [] Server
#poolSize: int
#availServer: int
+ServerPool(poolSize:int)
+ServerPool(pool:[] Server)
+getAvailServer(): int
+poolUpdate(deltat:double): void
+updateAvailServer(): void
+jobHandler(j:Job): void
+getPoolSize(): int
SERVER
+BUSY: static final int = 0
+IDLE: static final int = 1
#state: int
#remainProcTime: double
#idleTime: double
#idleCounter: double
#idleInts: Vector<Double>
+Server(rProcTime:double)
+getRemainprocTime(): double
+getState(): int
+initializeTask(time:double): void
+finishTask(): void
+systemUpdate(deltat:double): boolean
+getIdleInts(): Vector<Double>
QUEUESYSTEM
#numPool: int
#pools: [] ServerPool
#queues: [] Queue
+QueueSystem(pools:[] Server,queues:[] Queue)
+arrivalJobHandler(job:Job): boolean
+deptJobHandler(t:double): Vector<Event>
+systemUpdate(deltat:double): void
+getPools(): [] ServerPool
EVENTSCHEDULER
+events: Vector<Event>
+EventScheduler()
+addEvent(newEvent:Event): void
+nextEvent(): Event
+isEmpty(): boolean
QUEUE
#jobs: Vector<Job>
#size: int
+Queue()
+Queue(size:int)
+getLength(): int
+addAJob(ajob:Job): void
+removeHeadJob(): void
+getHeadJob(): Job
EVENT
+ARRIVAL: static final int = 1
+DEPATURE: static final int = 2
#type: int
#time: double
#job: Job
+Event()
+getTime(): double
+getType(): int
+getJob(): Job
+handler(): void
JOB
#arrTime: long
#id: int
#jobClass: int
#user: String
#group: String
#reqServer: int
#reqWt: double
#runTime: double
+Job()
+Job(id:int,arrTime:long,reqServer:int,runTime:double,
user:String,group:String)
ARRIVAL
+Arrival()
handler(): void
DEPATURE
+Departure()
handler(): void