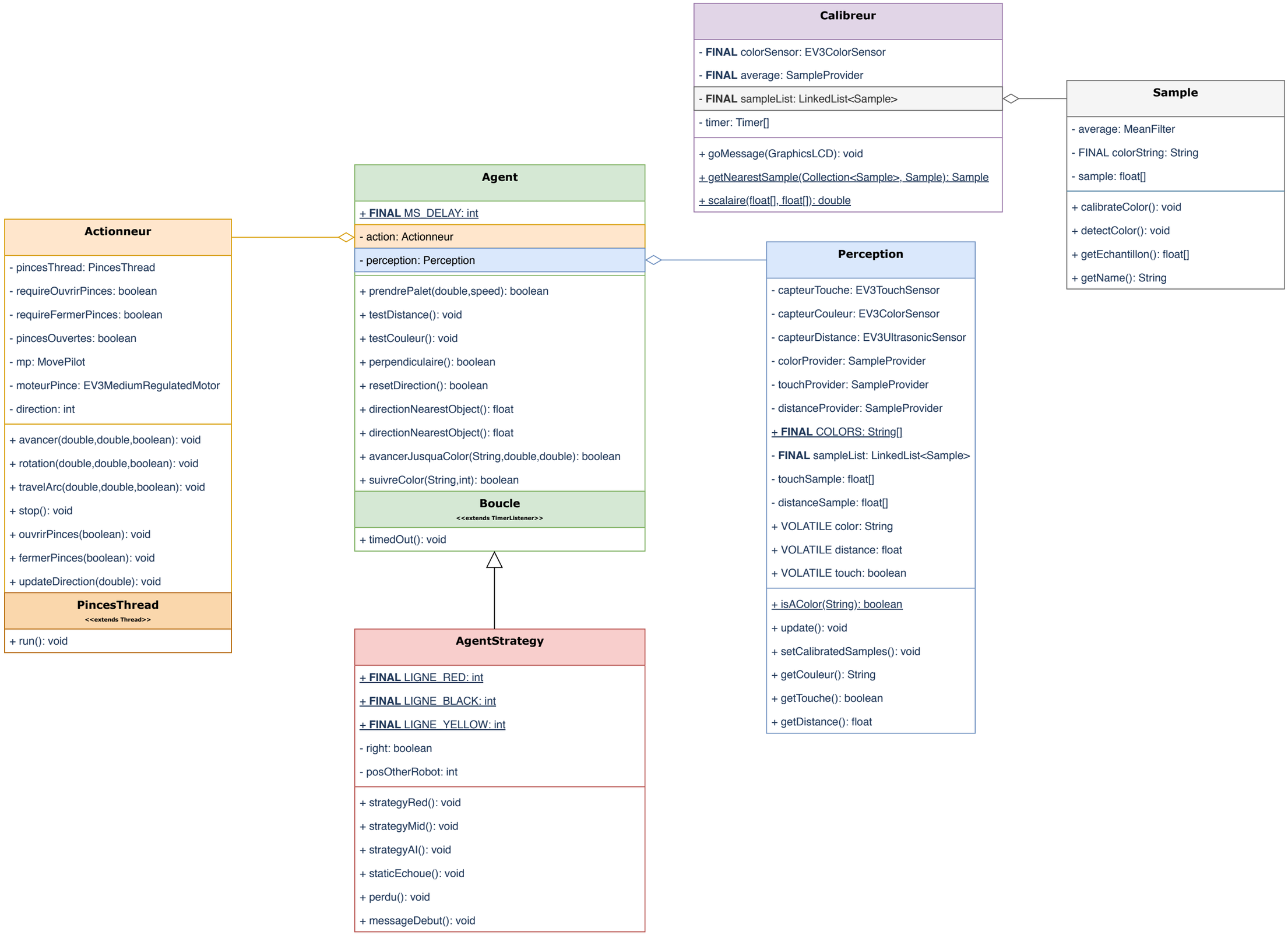
Calibreur
- FINAL colorSensor: EV3ColorSensor
- FINAL average: SampleProvider
- FINAL sampleList: LinkedList<Sample>
- timer: Timer[]
+ goMessage(GraphicsLCD): void
+ getNearestSample(Collection<Sample>, Sample): Sample
+ scalaire(float[], float[]): double
Sample
- average: MeanFilter
- FINAL colorString: String
- sample: float[]
+ calibrateColor(): void
+ detectColor(): void
+ getEchantillon(): float[]
+ getName(): String
Agent
+ FINAL MS_DELAY: int
- action: Actionneur
- perception: Perception
+ prendrePalet(double,speed): boolean
+ testDistance(): void
+ testCouleur(): void
+ perpendiculaire(): boolean
+ resetDirection(): boolean
+ directionNearestObject(): float
+ directionNearestObject(): float
+ avancerJusquaColor(String,double,double): boolean
+ suivreColor(String,int): boolean
Boucle
<<extends TimerListener>>
+ timedOut(): void
Actionneur
- pincesThread: PincesThread
- requireOuvrirPinces: boolean
- requireFermerPinces: boolean
- pincesOuvertes: boolean
- mp: MovePilot
- moteurPince: EV3MediumRegulatedMotor
- direction: int
+ avancer(double,double,boolean): void
+ rotation(double,double,boolean): void
+ travelArc(double,double,boolean): void
+ stop(): void
+ ouvrirPinces(boolean): void
+ fermerPinces(boolean): void
+ updateDirection(double): void
PincesThread
<<extends Thread>>
+ run(): void
Perception
- capteurTouche: EV3TouchSensor
- capteurCouleur: EV3ColorSensor
- capteurDistance: EV3UltrasonicSensor
- colorProvider: SampleProvider
- touchProvider: SampleProvider
- distanceProvider: SampleProvider
+ FINAL COLORS: String[]
- FINAL sampleList: LinkedList<Sample>
- touchSample: float[]
- distanceSample: float[]
+ VOLATILE color: String
+ VOLATILE distance: float
+ VOLATILE touch: boolean
+ isAColor(String): boolean
+ update(): void
+ setCalibratedSamples(): void
+ getCouleur(): String
+ getTouche(): boolean
+ getDistance(): float
AgentStrategy
+ FINAL LIGNE_RED: int
+ FINAL LIGNE_BLACK: int
+ FINAL LIGNE_YELLOW: int
- right: boolean
- posOtherRobot: int
+ strategyRed(): void
+ strategyMid(): void
+ strategyAI(): void
+ staticEchoue(): void
+ perdu(): void
+ messageDebut(): void