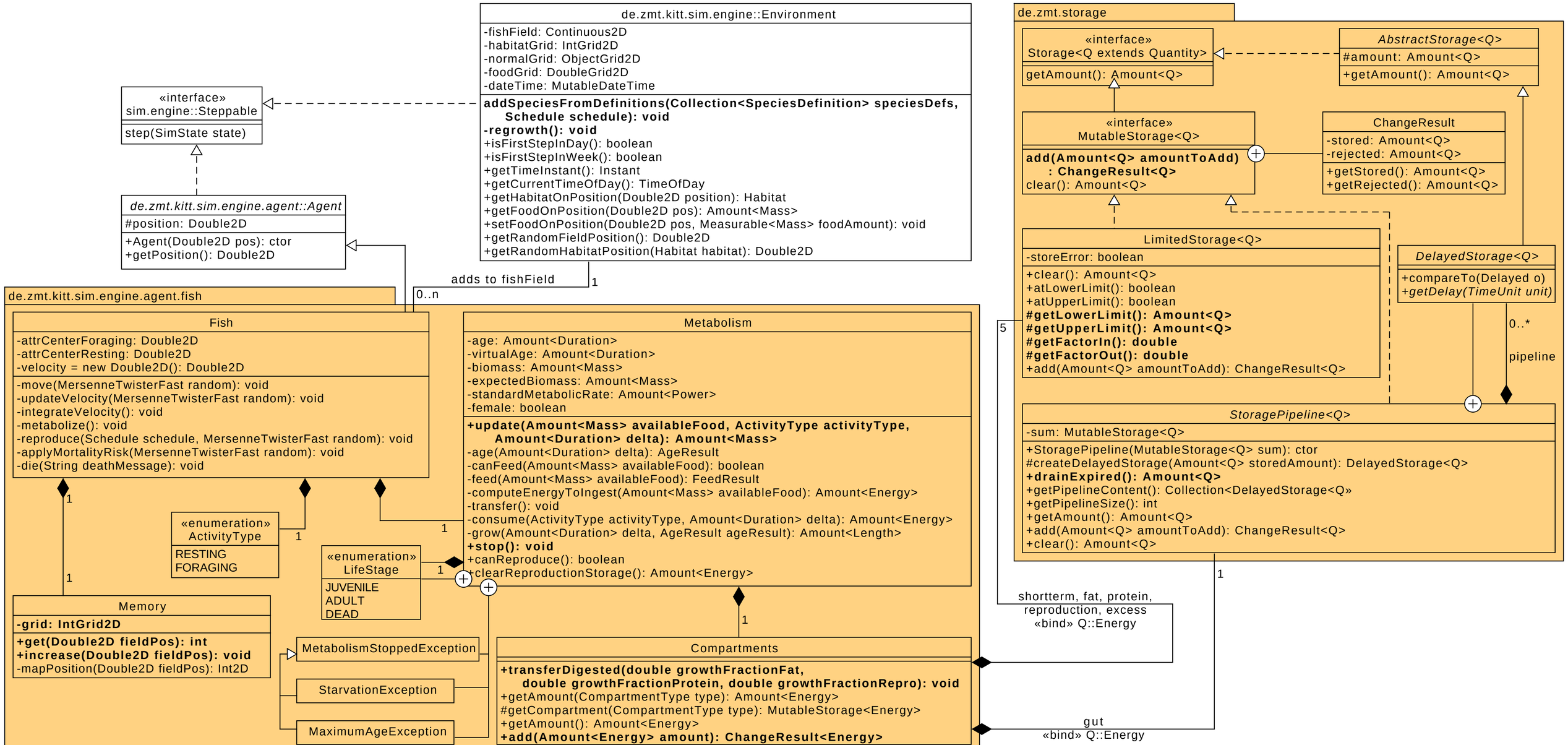
de.zmt.kitt.sim.engine::Environment
-fishField: Continuous2D
-habitatGrid: IntGrid2D
-normalGrid: ObjectGrid2D
-foodGrid: DoubleGrid2D
-dateTime: MutableDateTime
addSpeciesFromDefinitions(Collection<SpeciesDefinition> speciesDefs, Schedule schedule): void
-regrowth(): void
+isFirstStepInDay(): boolean
+isFirstStepInWeek(): boolean
+getTimeInstant(): Instant
+getCurrentTimeOfDay(): TimeOfDay
+getHabitatOnPosition(Double2D position): Habitat
+getFoodOnPosition(Double2D pos): Amount<Mass>
+setFoodOnPosition(Double2D pos, Measurable<Mass> foodAmount): void
+getRandomFieldPosition(): Double2D
+getRandomHabitatPosition(Habitat habitat): Double2D
«interface»
sim.engine::Steppable
step(SimState state)
de.zmt.kitt.sim.engine.agent::Agent
#position: Double2D
+Agent(Double2D pos): ctor
+getPosition(): Double2D
adds to fishField
1
0..n
de.zmt.kitt.sim.engine.agent.fish
Fish
-attrCenterForaging: Double2D
-attrCenterResting: Double2D
-velocity = new Double2D(): Double2D
-move(MersenneTwisterFast random): void
-updateVelocity(MersenneTwisterFast random): void
-integrateVelocity(): void
-metabolize(): void
-reproduce(Schedule schedule, MersenneTwisterFast random): void
-applyMortalityRisk(MersenneTwisterFast random): void
-die(String deathMessage): void
Metabolism
-age: Amount<Duration>
-virtualAge: Amount<Duration>
-biomass: Amount<Mass>
-expectedBiomass: Amount<Mass>
-standardMetabolicRate: Amount<Power>
-female: boolean
+update(Amount<Mass> availableFood, ActivityType activityType, Amount<Duration> delta): Amount<Mass>
-age(Amount<Duration> delta): AgeResult
-canFeed(Amount<Mass> availableFood): boolean
-feed(Amount<Mass> availableFood): FeedResult
-computeEnergyToIngest(Amount<Mass> availableFood): Amount<Energy>
-transfer(): void
-consume(ActivityType activityType, Amount<Duration> delta): Amount<Energy>
-grow(Amount<Duration> delta, AgeResult ageResult): Amount<Length>
+stop(): void
+canReproduce(): boolean
+clearReproductionStorage(): Amount<Energy>
«enumeration»
ActivityType
RESTING
FORAGING
«enumeration»
LifeStage
JUVENILE
ADULT
DEAD
Memory
-grid: IntGrid2D
+get(Double2D fieldPos): int
+increase(Double2D fieldPos): void
-mapPosition(Double2D fieldPos): Int2D
MetabolismStoppedException
StarvationException
MaximumAgeException
Compartments
+transferDigested(double growthFractionFat, double growthFractionProtein, double growthFractionRepro): void
+getAmount(CompartmentType type): Amount<Energy>
#getCompartment(CompartmentType type): MutableStorage<Energy>
+getAmount(): Amount<Energy>
+add(Amount<Energy> amount): ChangeResult<Energy>
de.zmt.storage
«interface»
Storage<Q extends Quantity>
getAmount(): Amount<Q>
AbstractStorage<Q>
#amount: Amount<Q>
+getAmount(): Amount<Q>
«interface»
MutableStorage<Q>
add(Amount<Q> amountToAdd) : ChangeResult<Q>
clear(): Amount<Q>
ChangeResult
-stored: Amount<Q>
-rejected: Amount<Q>
+getStored(): Amount<Q>
+getRejected(): Amount<Q>
LimitedStorage<Q>
-storeError: boolean
+clear(): Amount<Q>
+atLowerLimit(): boolean
+atUpperLimit(): boolean
#getLowerLimit(): Amount<Q>
#getUpperLimit(): Amount<Q>
#getFactorIn(): double
#getFactorOut(): double
+add(Amount<Q> amountToAdd): ChangeResult<Q>
DelayedStorage<Q>
+compareTo(Delayed o)
+getDelay(TimeUnit unit)
0..*
pipeline
StoragePipeline<Q>
-sum: MutableStorage<Q>
+StoragePipeline(MutableStorage<Q> sum): ctor
#createDelayedStorage(Amount<Q> storedAmount): DelayedStorage<Q>
+drainExpired(): Amount<Q>
+getPipelineContent(): Collection<DelayedStorage<Q»
+getPipelineSize(): int
+getAmount(): Amount<Q>
+add(Amount<Q> amountToAdd): ChangeResult<Q>
+clear(): Amount<Q>
5
shortterm, fat, protein, reproduction, excess
«bind» Q::Energy
gut
«bind» Q::Energy