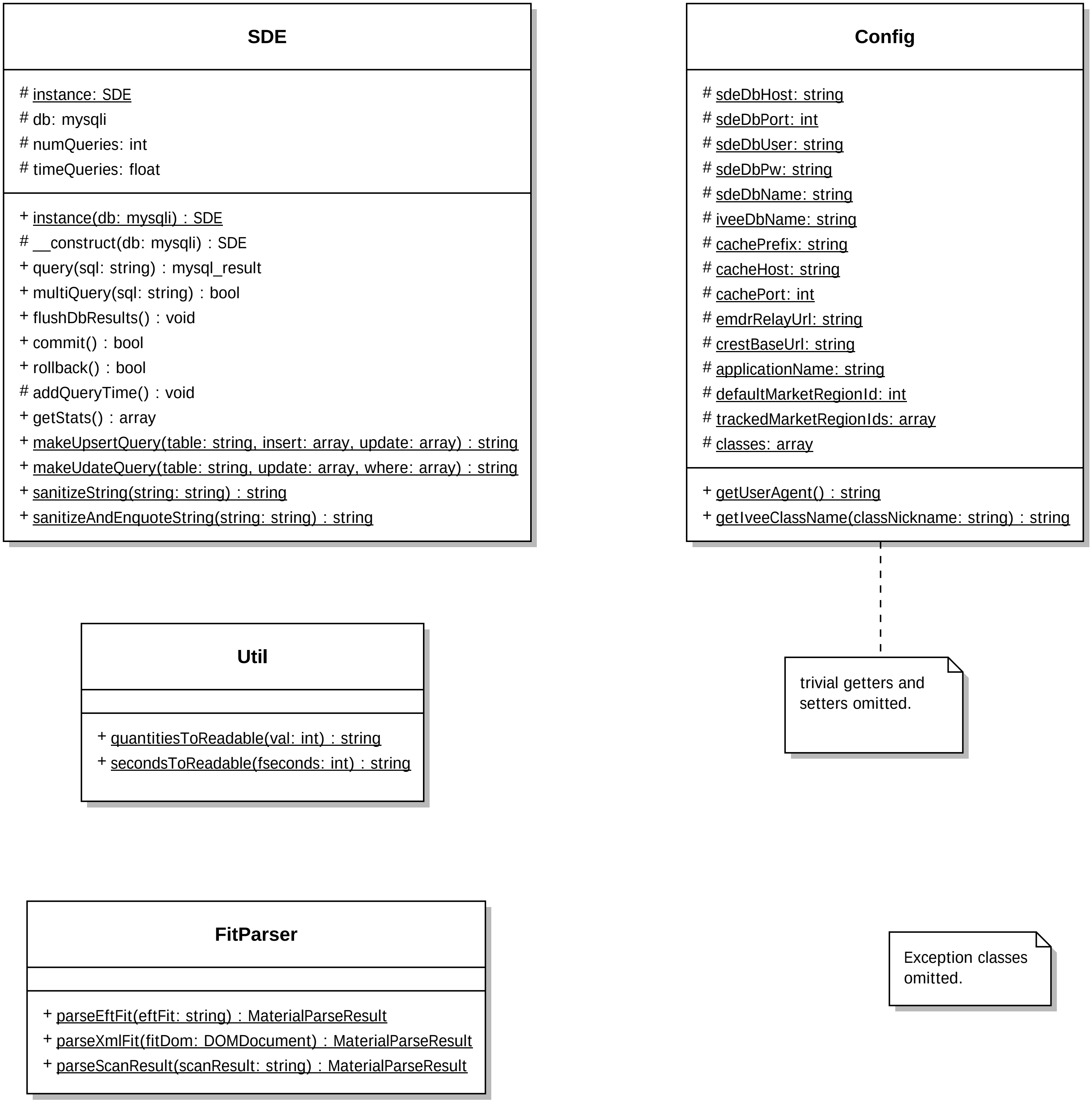


SDE

```
# instance: SDE
# db: mysqli
# numQueries: int
# timeQueries: float
```

```
+ instance(db: mysqli) : SDE
# __construct(db: mysqli) : SDE
+ query(sql: string) : mysql_result
+ multiQuery(sql: string) : bool
+ flushDbResults() : void
+ commit() : bool
+ rollback() : bool
# addQueryTime() : void
+ getStats() : array
+ makeUpsertQuery(table: string, insert: array, update: array) : string
+ makeUdateQuery(table: string, update: array, where: array) : string
+ sanitizeString(string: string) : string
+ sanitizeAndEnquoteString(string: string) : string
```

Config

```
# sdeDbHost: string
# sdeDbPort: int
# sdeDbUser: string
# sdeDbPw: string
# sdeDbName: string
# iveeDbName: string
# cachePrefix: string
# cacheHost: string
# cachePort: int
# emdrRelayUrl: string
# crestBaseUrl: string
# applicationName: string
# defaultMarketRegionId: int
# trackedMarketRegionIds: array
# classes: array
```

```
+ getUserAgent() : string
+ getIveeClassName(classNickname: string) : string
```

trivial getters and setters omitted.

Util

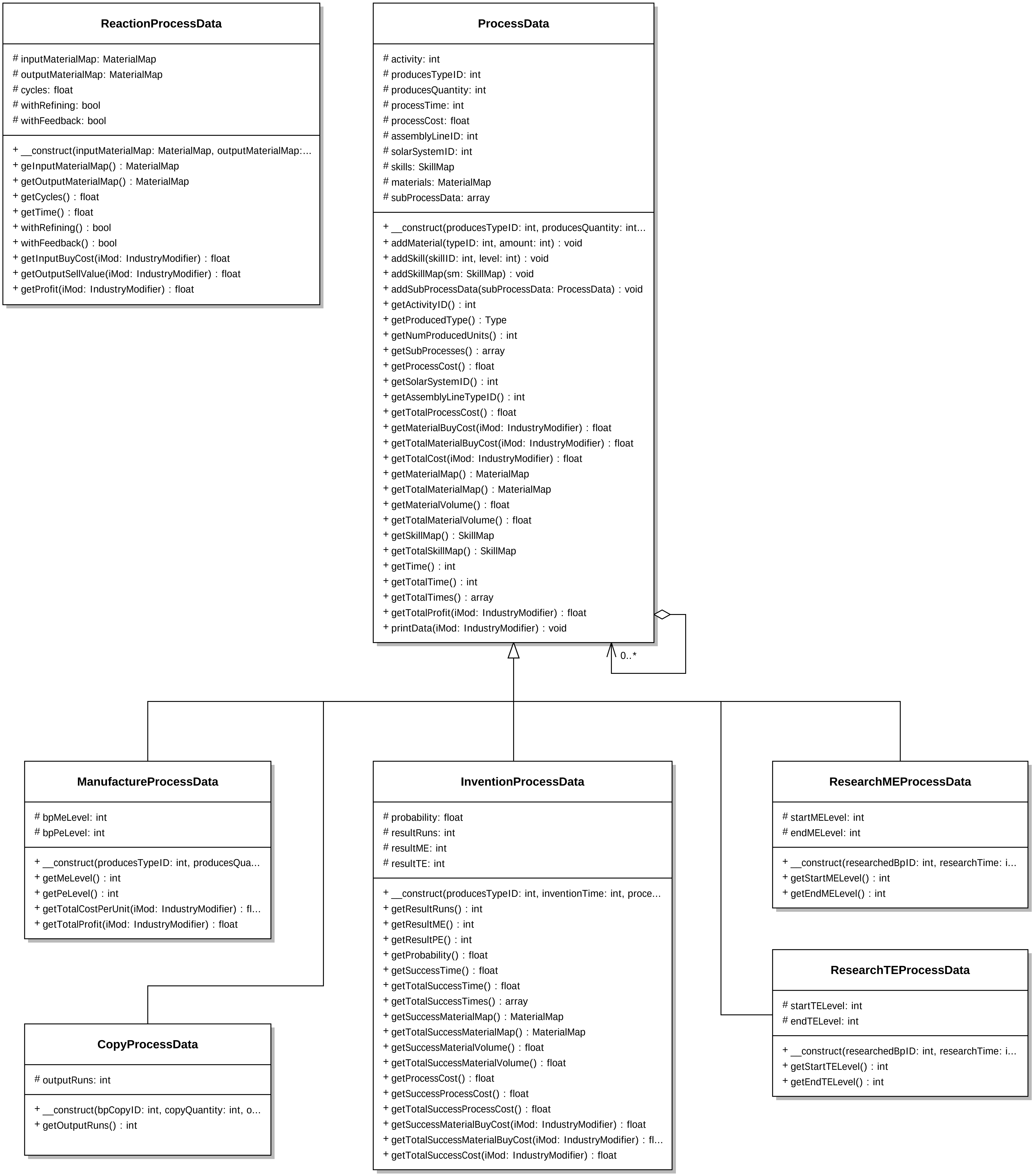
```
+ quantitiesToReadable(val: int) : string
+ secondsToReadable(fseconds: int) : string
```

FitParser

```
+ parseEftFit(eftFit: string) : MaterialParseResult
+ parseXmlFit(fitDom: DOMDocument) : MaterialParseResult
+ parseScanResult(scanResult: string) : MaterialParseResult
```

Exception classes omitted.

«interface»
ICoreDataCommon
+ getClassHierarchyKeyPrefix() : string
«interface»
ICacheable
+ getKey() : string
+ getCacheTTL() : int
InstancePool
# cache: ICache
# keyToObj: array
# nameToKey: array
# hits: int
+ __construct(className: string) : InstancePool
+ setItem(obj: ICacheable) : void
+ setNamesToKeys(namesToKey: array) : void
+ getItem(key: string) : object
+ getKeyByName(name: string) : int
+ deleteItem(key: string) : void
+ deleteMulti(keys: array) : void
+ getHits() : int
+ getPooledTypeCount() : int
1
«interface»
ICache
+ instance() : ICache
+ setItem(item: ICacheable) : void
+ getItem(key: string) : mixed
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : bool
+ getHits() : int
CoreDataCommon
# id: int
+ getClassNick() : string
+ init() : void
# getInstancePool() : InstancePool
# deleteFromCache(keys: array) : void
+ getId() : int
+ getKey() : string
+ getCacheTTL() : int
# throwException(exceptionName: string, message: string, c...
1
static
1
PredisWrapper
# instance: PredisWrapper
# predis: PredisClient
# hit: int
# __construct() : PredisWrapper
+ instance() : PredisWrapper
MemcachedWrapper
# instance: MemcachedWrapper
# memcached: Memcached
# hits: int
# __construct() : MemcachedWrapper
+ instance() : MemcachedWrapper
SdeType
# name: string
# getById(id: int) : SdeType
# loadNames() : void
# getIdByName(name: string) : int
# getByName(name: string) : SdeType
+ getName() : string
DoctrineCacheWrapper
# instance: DoctrineCacheWrapper
# cache: CacheProvider
# hit: int
+ instance() : DoctrineCacheWrapper
RegionMarketData
+ classNick: string
# instancePool: InstancePool
# regionID: int
# priceDate: int
# sellPrice: float
# buyPrice: float
# supplyIn5: int
# demandIn5: int
# avgSell5OrderAge: int
# avgBuy5OrderAge: int
# histDate: int
# avgVol: float
# avgTx: float
# low: float
# high: float
# avg: float
+ getClassHierarchyKeyPrefix() : string
+ getByIdAndRegion(typeID: int, regionID: int) : RegionMarket...
# __construct(typeID: int, regionID: int) : RegionMarketData
+ getRegionID() : int
+ getKey() : string
+ getType() : Type
+ getCacheTTL() : int
+ getBuyPrice(maxPriceDataAge: int) : float
+ getSellPrice(maxPriceDataAge: int) : float
+ getHistory(regionID: int, fromDateTS: int, toDateTS: int) : array
+ getPriceDate() : int
+ getAvgVol() : float
+ getAvgTx() : float
+ getSupplyIn5() : int
+ getDemandIn5() : int
+ getAvgBuy5OrderAge() : int
+ getAvgSell5OrderAge() : int
+ getHistDate() : int
+ getLow() : float
+ getHigh() : float
+ getAvg() : float
# throwNotOnMarketException() : void
GlobalPriceData
+ classNick: string
# instancePool: InstancePool
# averagePrice: float
# adjustedPrice: float
# priceDate: int
+ getClassHierarchyKeyPrefix() : string
+ getById(id: int) : GlobalPriceData
# __construct(typeID: int) : GlobalPriceData
# queryAttributes() : array
# setAttributes(row: array) : void
+ getType() : Type
+ getPriceDate() : int
+ getAveragePrice(maxPriceDataAge: int) : float
+ getAdjustedPrice(maxPriceDataAge: int) : float
0..1
1
Type
+ classNick: string
# instancePool: InstancePool
# groupID: int
# categoryID: int
# volume: float
# portionSize: int
# basePrice: int
# materials: array
# reprocessingSkillID: int
+ getById(id: int) : Type
# loadNames() : void
+ getClassHierarchyKeyPrefix() : string
- factory(typeID: int, subtypeInfo: array) : Type
- getSubtypeInfo(typeID: int) : array
- decideType(subtypeInfo: array) : String
# __construct(typeID: int) : Type
# queryAttributes() : array
# setAttributes(row: array) : void
+ getGroupID() : int
+ getCategoryID() : int
+ getVolume() : float
+ getPortionSize() : int
+ getBasePrice() : int
+ getMarketGroupID() : int
+ getGlobalPriceData() : GlobalPriceData
+ getRegionMarketData(regionID: int) : RegionMarketData
+ isReprocessable() : bool
+ getMaterials() : array
+ getReprocessingSkillID() : int
+ getReprocessingMaterialMap(iMod: IndustryModifier, batchSiz...
0..*
1
ReactionProduct
# productOfReactionIDs: array
# __construct(typeID: int) : ReactionProduct
+ getReactions() : array
+ getReactionIDs() : array
Blueprint
# productId: int
# maxProductionLimit: int
# rank: int
# activityMaterials: array
# activitySkills: array
# activityTimes: array
# baseResearchMultiplier: array
# __construct(typeID: int) : Blueprint
# loadActivityMaterials(sde: SDE) : void
# loadActivitySkills(sde: SDE) : void
# loadActivityTimes(sde: SDE) : void
# queryAttributes() : array
# setAttributes(row: array) : void
+ getBuyPrice(maxPriceDataAge: int) : float
+ getSellPrice(maxPriceDataAge: int) : float
+ getProductBaseCost(maxPriceDataAge: int) : float
+ manufacture(iMod: IndustryModifier, units: int, bpME: int, bp...
+ copy(iMod: IndustryModifier, copies: int, runs: int, recursive: ...
+ researchME(iMod: IndustryModifier, startME: int, endME: int, ...
+ researchTE(iMod: IndustryModifier, startTE: int, endTE: int, r...
# addActivityMaterials(iMod: IndustryModifier, pd: ProcessData,...
+ getMaterials() : array
+ getMaterialsForActivity(activityID: int) : array
+ getProductId() : int
+ getProduct() : Manufacturable
+ getSkillMapForActivity(activityID: int) : SkillMap
+ getBaseTimeForActivity(activityID: int) : int
+ getRank() : int
+ getMaxProductionLimit() : int
+ isReprocessable() : bool
+ convertBpLevelToFactor(bpLevel: int) : int
+ calcResearchMultiplier(startLevel: int, endLevel: int) : float
Manufacturable
# producedFromBlueprintID: int
# queryAttributes() : array
# setAttributes(row: array) : void
+ getBlueprintId() : int
+ getBlueprint() : Blueprint
Decryptor
# teModifier: int
# teModifier: int
# runModifier: int
# probabilityModifier: float
# decryptorGroups: array
# __construct(typeID: int) : Decryptor
+ getMEModifier() : int
+ getPEModifier() : int
+ getRunModifier() : int
+ getProbabilityModifier() : float
+ getIDsFromGroup(groupID: int) : array
+ isReprocessable() : bool
Reaction
# cycleInputMaterial: array
# cycleOutputMaterial: array
# isAlchemy: bool
# __construct(typeID: int) : Reaction
+ getCycleInputMaterials() : array
+ getCycleOutputMaterials() : array
+ isAlchemy() : bool
+ react(cycles: int, reprocess: bool, feedback: bool,...
InventableBlueprint
# inventedFrom: int
# __construct(typeID: int) : InventableBlueprint
# loadOriginatingBlueprints(sde: SDE) : void
+ getInventorBlueprintId() : int
+ getInventorBlueprint() : InventableBlueprint
+ getBuyPrice(maxPriceDataAge: int) : void
+ getSellPrice(maxPriceDataAge: int) : void
+ invent(iMod: IndustryModifier, decryptorID: int, recursive: bo...
+ copyInventManufacture(iMod: IndustryModifier, decryptorID: ...
InventorBlueprint
# inventsBlueprintIDs: array
# inventsBlueprintIDsByRaceID: array
# inventionProbability: float
# inventionOutputRuns: int
# decryptorGroupID: int
# encryptionSkillID: int
# datacoreSkillIDs: array
# __construct(typeID: int) : InventorBlueprint
# loadInventionStats(sde: SDE) : void
# loadSkillToDatacoreInterface(sde: SDE) : void
+ invent(iMod: IndustryModifier, inventedBpID: int, decryptorID...
# getAndCheckDecryptor(decryptorID: int) : Decryptor
+ copyInventManufacture(iMod: IndustryModifier, inventedBpID...
+ getInventableBlueprintIDs() : array
+ getInventableBlueprints() : array
+ getInventionProbability() : float
+ getInventableBlueprintIDsByRaceId(race: int) : array
+ getInventionOutputRuns() : int
+ getDecryptorGroupID() : ing
+ getDecryptorIDs() : array
+ calcInventionProbability(charMod: ICharacterModifier) : float
T3Blueprint
# inventedFrom: array
# loadOriginatingBlueprints(sde: SDE) : void
# getInventionRelicIDs() : array
+ getInventionRelics() : array
+ inventFromRelic(iMod: IndustryModifier, relicId: i...
+ inventManufacture(iMod: IndustryModifier, relicId...
Relic
# queryAttributes() : array
+ invent(iMod: IndustryModifier, inventedBpID: int, decryptorID...
+ inventManufacture(iMod: IndustryModifier, inventedBpID: int,...
Blocked methods
omitted



ReactionProcessData

# inputMaterialMap: MaterialMap
# outputMaterialMap: MaterialMap
# cycles: float
# withRefining: bool
# withFeedback: bool

+ __construct(inputMaterialMap: MaterialMap, outputMaterialMap:...
+ geInputMaterialMap() : MaterialMap
+ getOutputMaterialMap() : MaterialMap
+ getCycles() : float
+ getTime() : float
+ withRefining() : bool
+ withFeedback() : bool
+ getInputBuyCost(iMod: IndustryModifier) : float
+ getOutputSellValue(iMod: IndustryModifier) : float
+ getProfit(iMod: IndustryModifier) : float

ProcessData

# activity: int
# producesTypeID: int
# producesQuantity: int
# processTime: int
# processCost: float
# assemblyLineID: int
# solarSystemID: int
# skills: SkillMap
# materials: MaterialMap
# subProcessData: array

+ __construct(producesTypeID: int, producesQuantity: int...
+ addMaterial(typeID: int, amount: int) : void
+ addSkill(skillID: int, level: int) : void
+ addSkillMap(sm: SkillMap) : void
+ addSubProcessData(subProcessData: ProcessData) : void
+ getActivityID() : int
+ getProducedType() : Type
+ getNumProducedUnits() : int
+ getSubProcesses() : array
+ getProcessCost() : float
+ getSolarSystemID() : int
+ getAssemblyLineTypeID() : int
+ getTotalProcessCost() : float
+ getMaterialBuyCost(iMod: IndustryModifier) : float
+ getTotalMaterialBuyCost(iMod: IndustryModifier) : float
+ getTotalCost(iMod: IndustryModifier) : float
+ getMaterialMap() : MaterialMap
+ getTotalMaterialMap() : MaterialMap
+ getMaterialVolume() : float
+ getTotalMaterialVolume() : float
+ getSkillMap() : SkillMap
+ getTotalSkillMap() : SkillMap
+ getTime() : int
+ getTotalTime() : int
+ getTotalTimes() : array
+ getTotalProfit(iMod: IndustryModifier) : float
+ printData(iMod: IndustryModifier) : void

ManufactureProcessData

# bpMeLevel: int
# bpPeLevel: int

+ __construct(producesTypeID: int, producesQua...
+ getMeLevel() : int
+ getPeLevel() : int
+ getTotalCostPerUnit(iMod: IndustryModifier) : fl...
+ getTotalProfit(iMod: IndustryModifier) : float

InventionProcessData

# probability: float
# resultRuns: int
# resultME: int
# resultTE: int

+ __construct(producesTypeID: int, inventionTime: int, proce...
+ getResultRuns() : int
+ getResultME() : int
+ getResultPE() : int
+ getProbability() : float
+ getSuccessTime() : float
+ getTotalSuccessTime() : float
+ getTotalSuccessTimes() : array
+ getSuccessMaterialMap() : MaterialMap
+ getTotalSuccessMaterialMap() : MaterialMap
+ getSuccessMaterialVolume() : float
+ getTotalSuccessMaterialVolume() : float
+ getProcessCost() : float
+ getSuccessProcessCost() : float
+ getTotalSuccessProcessCost() : float
+ getSuccessMaterialBuyCost(iMod: IndustryModifier) : float
+ getTotalSuccessMaterialBuyCost(iMod: IndustryModifier) : fl...
+ getTotalSuccessCost(iMod: IndustryModifier) : float

ResearchMEProcessData

# startMELevel: int
# endMELevel: int

+ __construct(researchedBpID: int, researchTime: i...
+ getStartMELevel() : int
+ getEndMELevel() : int

ResearchTEProcessData

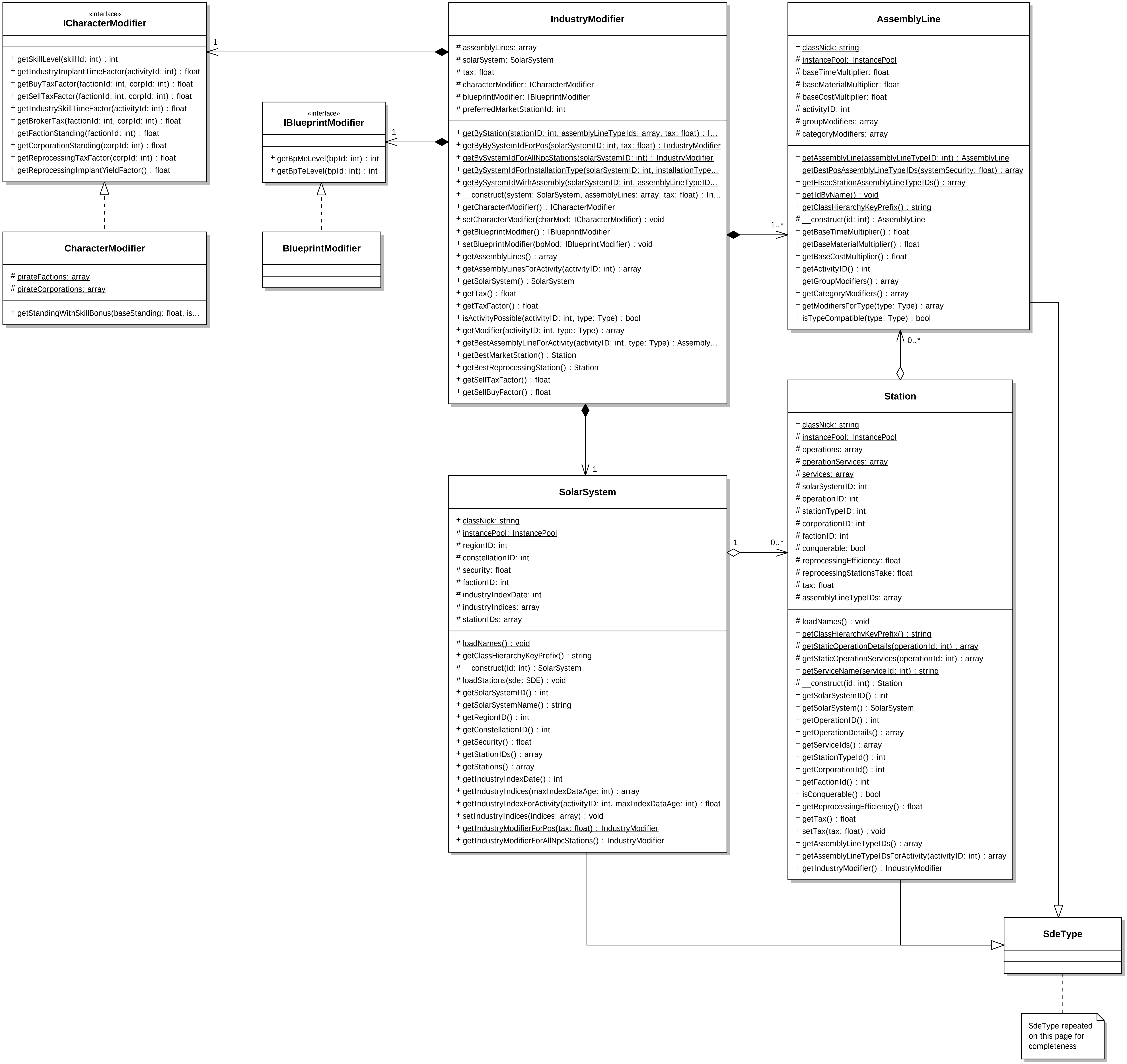
# startTELevel: int
# endTELevel: int

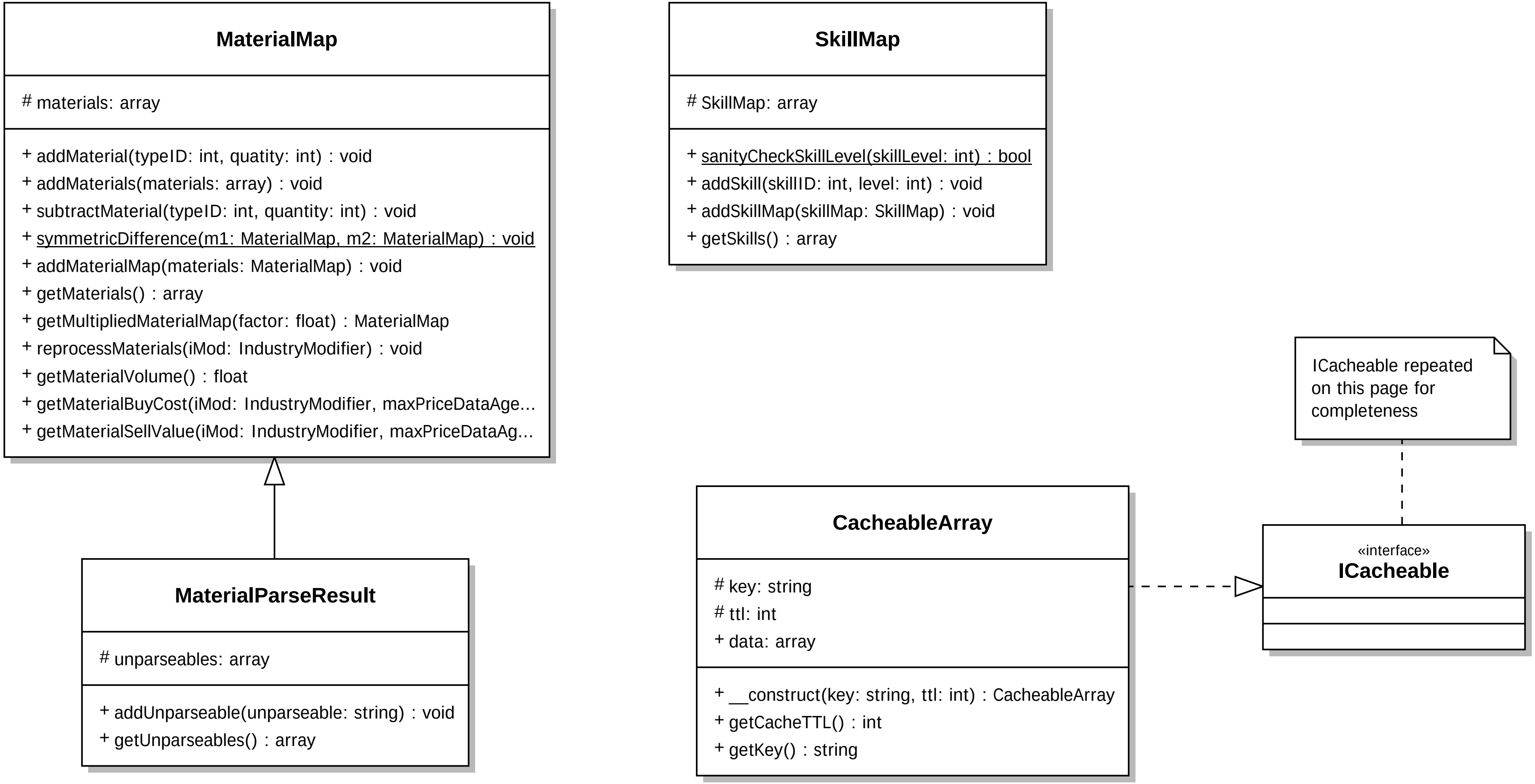
+ __construct(researchedBpID: int, researchTime: i...
+ getStartTELevel() : int
+ getEndTELevel() : int

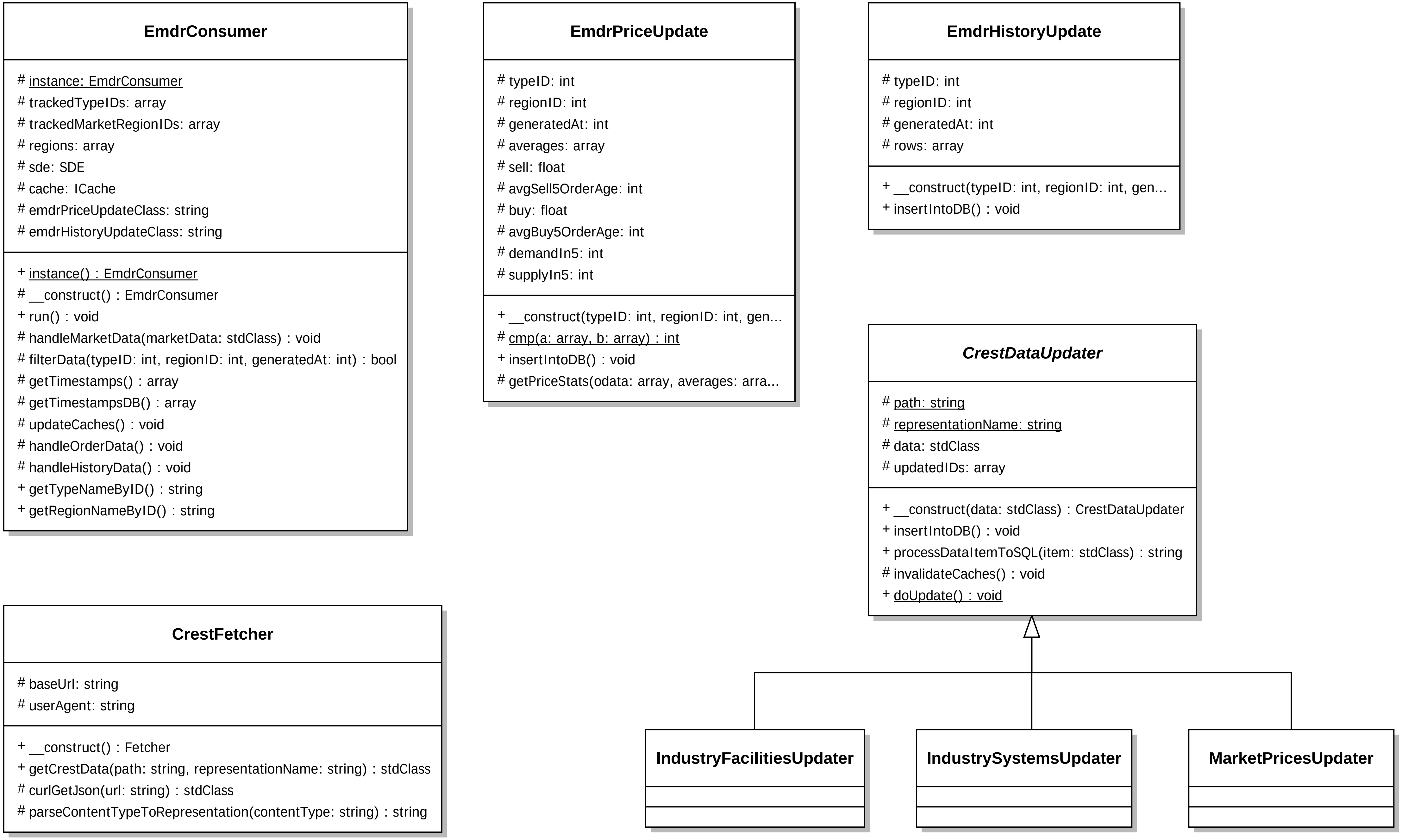
CopyProcessData

# outputRuns: int

+ __construct(bpCopyID: int, copyQuantity: int, o...
+ getOutputRuns() : int


«interface»
ICharacterModifier
+ getSkillLevel(skillId: int) : int
+ getIndustryImplantTimeFactor(activityId: int) : float
+ getBuyTaxFactor(factionId: int, corpId: int) : float
+ getSellTaxFactor(factionId: int, corpId: int) : float
+ getIndustrySkillTimeFactor(activityId: int) : float
+ getBrokerTax(factionId: int, corpId: int) : float
+ getFactionStanding(factionId: int) : float
+ getCorporationStanding(corpId: int) : float
+ getReprocessingTaxFactor(corpId: int) : float
+ getReprocessingImplantYieldFactor() : float
1
CharacterModifier
# pirateFactions: array
# pirateCorporations: array
+ getStandingWithSkillBonus(baseStanding: float, is...
«interface»
IBlueprintModifier
+ getBpMeLevel(bpId: int) : int
+ getBpTeLevel(bpId: int) : int
1
BlueprintModifier
IndustryModifier
# assemblyLines: array
# solarSystem: SolarSystem
# tax: float
# characterModifier: ICharacterModifier
# blueprintModifier: IBlueprintModifier
# preferredMarketStationId: int
+ getByStation(stationID: int, assemblyLineTypeIds: array, tax: float) : I...
+ getByBySystemIdForPos(solarSystemID: int, tax: float) : IndustryModifier
+ getBySystemIdForAllNpcStations(solarSystemID: int) : IndustryModifier
+ getBySystemIdForInstallationType(solarSystemID: int, installationType...
+ getBySystemIdWithAssembly(solarSystemID: int, assemblyLineTypeID...
+ __construct(system: SolarSystem, assemblyLines: array, tax: float) : In...
+ getCharacterModifier() : ICharacterModifier
+ setCharacterModifier(charMod: ICharacterModifier) : void
+ getBlueprintModifier() : IBlueprintModifier
+ setBlueprintModifier(bpMod: IBlueprintModifier) : void
+ getAssemblyLines() : array
+ getAssemblyLinesForActivity(activityID: int) : array
+ getSolarSystem() : SolarSystem
+ getTax() : float
+ getTaxFactor() : float
+ isActivityPossible(activityID: int, type: Type) : bool
+ getModifier(activityID: int, type: Type) : array
+ getBestAssemblyLineForActivity(activityID: int, type: Type) : Assembly...
+ getBestMarketStation() : Station
+ getBestReprocessingStation() : Station
+ getSellTaxFactor() : float
+ getSellBuyFactor() : float
1..*
AssemblyLine
+ classNick: string
# instancePool: InstancePool
# baseTimeMultiplier: float
# baseMaterialMultiplier: float
# baseCostMultiplier: float
# activityID: int
# groupModifiers: array
# categoryModifiers: array
+ getAssemblyLine(assemblyLineTypeID: int) : AssemblyLine
+ getBestPosAssemblyLineTypeIDs(systemSecurity: float) : array
+ getHisecStationAssemblyLineTypeIDs() : array
+ getIdByName() : void
+ getClassHierarchyKeyPrefix() : string
# __construct(id: int) : AssemblyLine
+ getBaseTimeMultiplier() : float
+ getBaseMaterialMultiplier() : float
+ getBaseCostMultiplier() : float
+ getActivityID() : int
+ getGroupModifiers() : array
+ getCategoryModifiers() : array
+ getModifiersForType(type: Type) : array
+ isTypeCompatible(type: Type) : bool
0..*
1
SolarSystem
+ classNick: string
# instancePool: InstancePool
# regionID: int
# constellationID: int
# security: float
# factionID: int
# industryIndexDate: int
# industryIndices: array
# stationIDs: array
# loadNames() : void
+ getClassHierarchyKeyPrefix() : string
# __construct(id: int) : SolarSystem
# loadStations(sde: SDE) : void
+ getSolarSystemID() : int
+ getSolarSystemName() : string
+ getRegionID() : int
+ getConstellationID() : int
+ getSecurity() : float
+ getStationIDs() : array
+ getStations() : array
+ getIndustryIndexDate() : int
+ getIndustryIndices(maxIndexDataAge: int) : array
+ getIndustryIndexForActivity(activityID: int, maxIndexDataAge: int) : float
+ setIndustryIndices(indices: array) : void
+ getIndustryModifierForPos(tax: float) : IndustryModifier
+ getIndustryModifierForAllNpcStations() : IndustryModifier
1
0..*
Station
+ classNick: string
# instancePool: InstancePool
# operations: array
# operationServices: array
# services: array
# solarSystemID: int
# operationID: int
# stationTypeID: int
# corporationID: int
# factionID: int
# conquerable: bool
# reprocessingEfficiency: float
# reprocessingStationsTake: float
# tax: float
# assemblyLineTypeIDs: array
# loadNames() : void
+ getClassHierarchyKeyPrefix() : string
# getStaticOperationDetails(operationId: int) : array
# getStaticOperationServices(operationId: int) : array
+ getServiceName(serviceId: int) : string
# __construct(id: int) : Station
+ getSolarSystemID() : int
+ getSolarSystem() : SolarSystem
+ getOperationID() : int
+ getOperationDetails() : array
+ getServiceIds() : array
+ getStationTypeId() : int
+ getCorporationId() : int
+ getFactionId() : int
+ isConquerable() : bool
+ getReprocessingEfficiency() : float
+ getTax() : float
+ setTax(tax: float) : void
+ getAssemblyLineTypeIDs() : array
+ getAssemblyLineTypeIDsForActivity(activityID: int) : array
+ getIndustryModifier() : IndustryModifier
SdeType
SdeType repeated on this page for completeness


MaterialMap
# materials: array
+ addMaterial(typeID: int, quatity: int) : void
+ addMaterials(materials: array) : void
+ subtractMaterial(typeID: int, quantity: int) : void
+ symmetricDifference(m1: MaterialMap, m2: MaterialMap) : void
+ addMaterialMap(materials: MaterialMap) : void
+ getMaterials() : array
+ getMultipliedMaterialMap(factor: float) : MaterialMap
+ reprocessMaterials(iMod: IndustryModifier) : void
+ getMaterialVolume() : float
+ getMaterialBuyCost(iMod: IndustryModifier, maxPriceDataAge…
+ getMaterialSellValue(iMod: IndustryModifier, maxPriceDataAg…
MaterialParseResult
# unparseables: array
+ addUnparseable(unparseable: string) : void
+ getUnparseables() : array
SkillMap
# SkillMap: array
+ sanityCheckSkillLevel(skillLevel: int) : bool
+ addSkill(skillID: int, level: int) : void
+ addSkillMap(skillMap: SkillMap) : void
+ getSkills() : array
ICacheable repeated on this page for completeness
CacheableArray
# key: string
# ttl: int
+ data: array
+ __construct(key: string, ttl: int) : CacheableArray
+ getCacheTTL() : int
+ getKey() : string
«interface»
ICacheable



EmdrConsumer

# instance: EmdrConsumer
# trackedTypeIDs: array
# trackedMarketRegionIDs: array
# regions: array
# sde: SDE
# cache: ICache
# emdrPriceUpdateClass: string
# emdrHistoryUpdateClass: string

+ instance() : EmdrConsumer
# __construct() : EmdrConsumer
+ run() : void
# handleMarketData(marketData: stdClass) : void
# filterData(typeID: int, regionID: int, generatedAt: int) : bool
# getTimestamps() : array
# getTimestampsDB() : array
# updateCaches() : void
# handleOrderData() : void
# handleHistoryData() : void
+ getTypeNameByID() : string
+ getRegionNameByID() : string

EmdrPriceUpdate

# typeID: int
# regionID: int
# generatedAt: int
# averages: array
# sell: float
# avgSell5OrderAge: int
# buy: float
# avgBuy5OrderAge: int
# demandIn5: int
# supplyIn5: int

+ __construct(typeID: int, regionID: int, gen...
# cmp(a: array, b: array) : int
+ insertIntoDB() : void
# getPriceStats(odata: array, averages: arra...

EmdrHistoryUpdate

# typeID: int
# regionID: int
# generatedAt: int
# rows: array

+ __construct(typeID: int, regionID: int, gen...
+ insertIntoDB() : void

CrestDataUpdater

# path: string
# representationName: string
# data: stdClass
# updatedIDs: array

+ __construct(data: stdClass) : CrestDataUpdater
+ insertIntoDB() : void
+ processDataItemToSQL(item: stdClass) : string
# invalidateCaches() : void
+ doUpdate() : void

CrestFetcher

# baseUrl: string
# userAgent: string

+ __construct() : Fetcher
+ getCrestData(path: string, representationName: string) : stdClass
# curlGetJson(url: string) : stdClass
# parseContentTypeToRepresentation(contentType: string) : string

IndustryFacilitiesUpdater

IndustrySystemsUpdater

MarketPricesUpdater