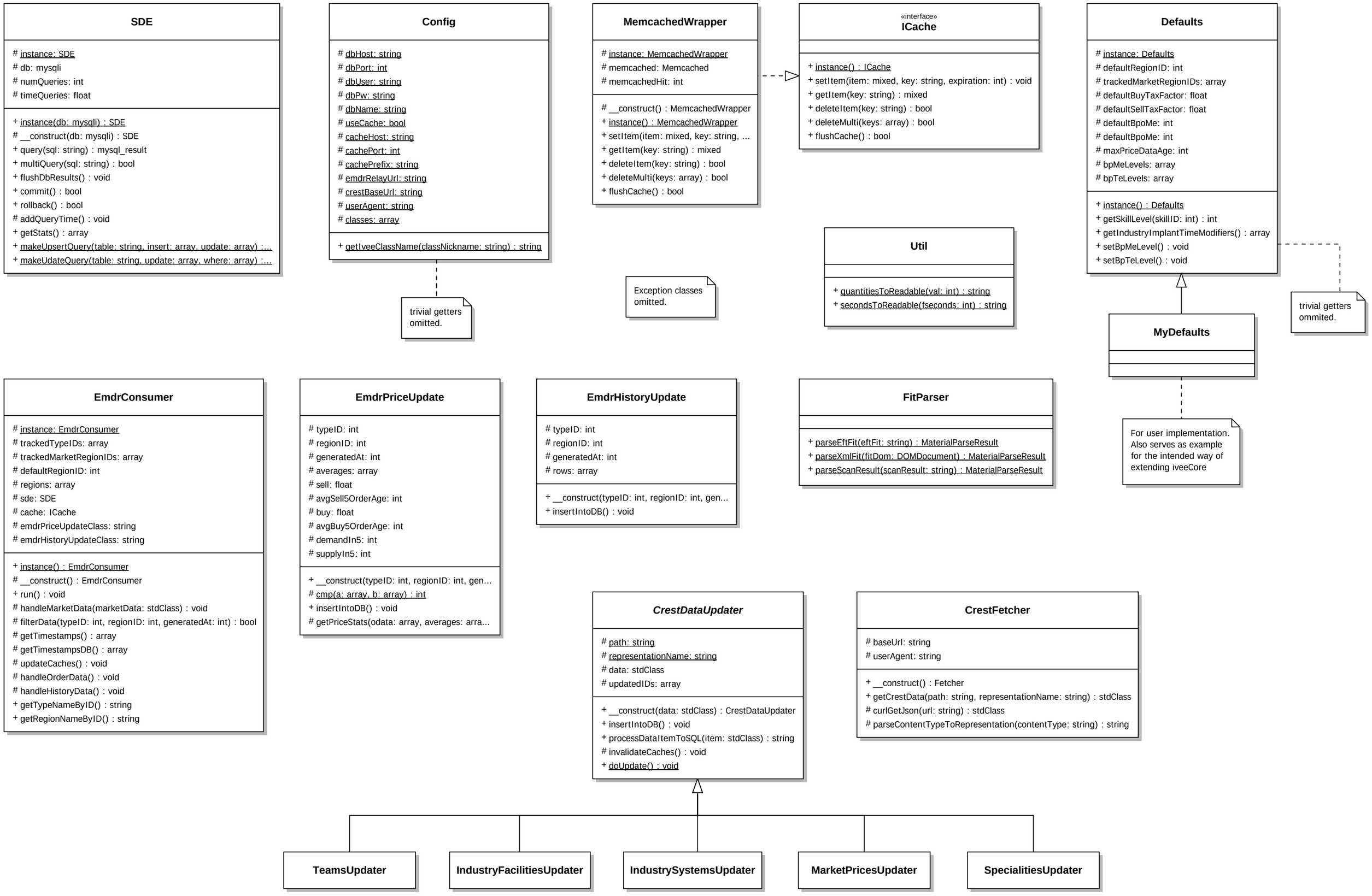
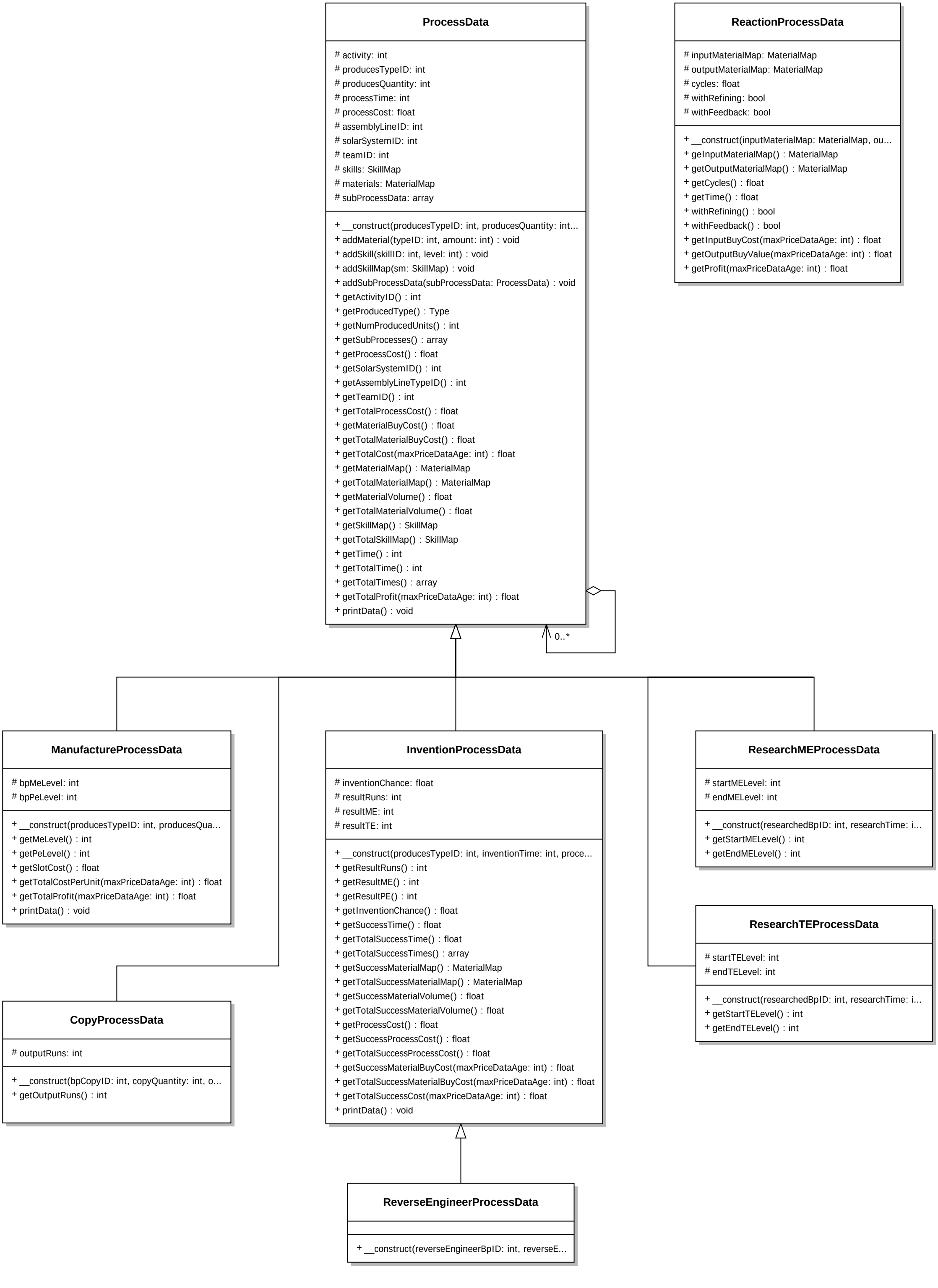
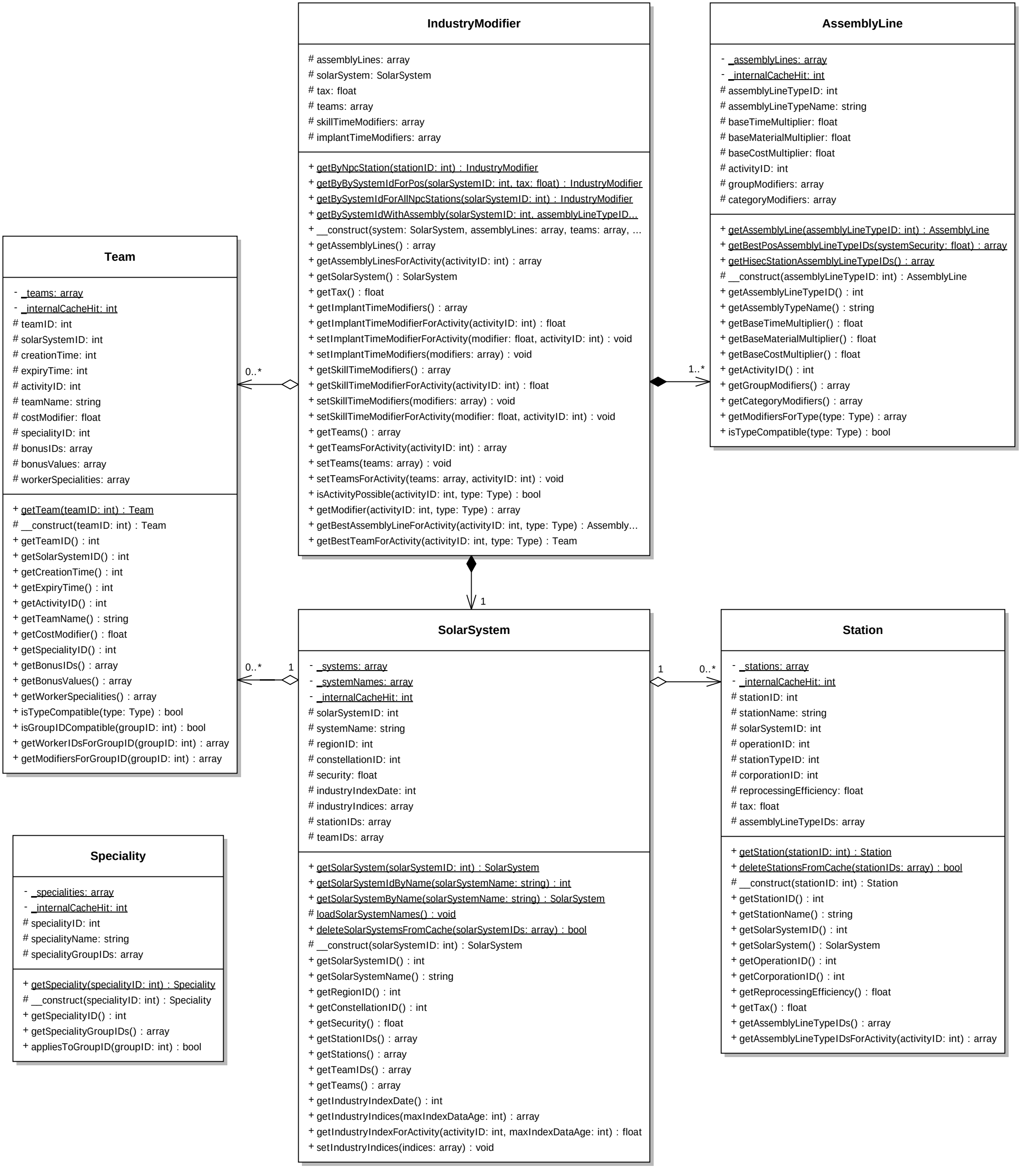
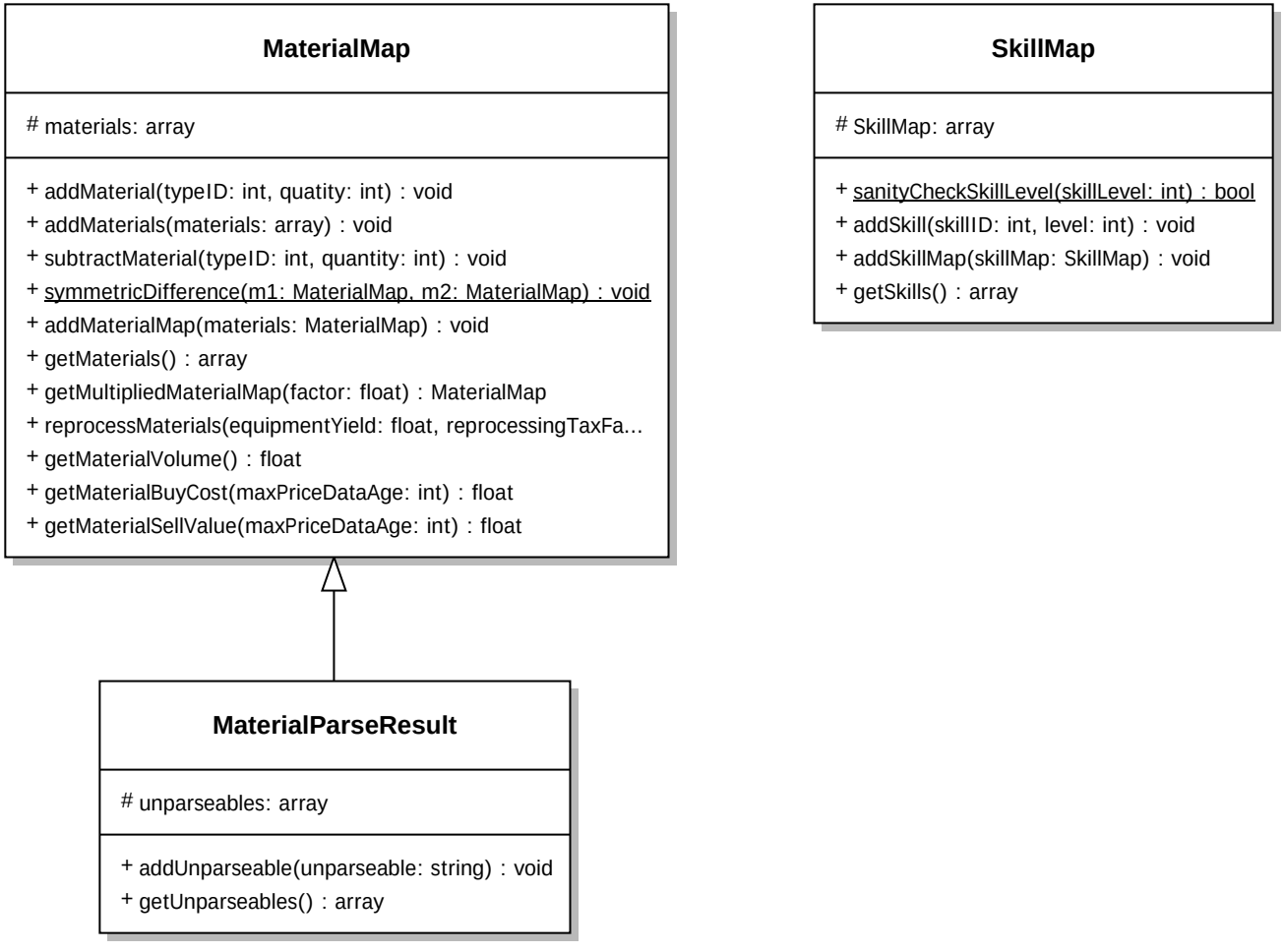

SDE
instance: SDE
db: mysqli
numQueries: int
timeQueries: float
+ instance(db: mysqli) : SDE
# __construct(db: mysqli) : SDE
+ query(sql: string) : mysql_result
+ multiQuery(sql: string) : bool
+ flushDbResults() : void
+ commit() : bool
+ rollback() : bool
# addQueryTime() : void
+ getStats() : array
+ makeUpsertQuery(table: string, insert: array, update: array) :...
+ makeUdateQuery(table: string, update: array, where: array) :...
Config
dbHost: string
dbPort: int
dbUser: string
dbPw: string
dbName: string
useCache: bool
cacheHost: string
cachePort: int
cachePrefix: string
emdrRelayUrl: string
crestBaseUrl: string
userAgent: string
classes: array
+ getIveeClassName(classNickname: string) : string
trivial getters omitted.
MemcachedWrapper
instance: MemcachedWrapper
memcached: Memcached
memcachedHit: int
__construct() : MemcachedWrapper
+ instance() : MemcachedWrapper
+ setItem(item: mixed, key: string, ...
+ getItem(key: string) : mixed
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : bool
«interface»
ICache
+ instance() : ICache
+ setItem(item: mixed, key: string, expiration: int) : void
+ getItem(key: string) : mixed
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : bool
Exception classes omitted.
Util
+ quantitiesToReadable(val: int) : string
+ secondsToReadable(fseconds: int) : string
Defaults
instance: Defaults
defaultRegionID: int
trackedMarketRegionIDs: array
defaultBuyTaxFactor: float
defaultSellTaxFactor: float
defaultBpoMe: int
defaultBpoMe: int
maxPriceDataAge: int
bpMeLevels: array
bpTeLevels: array
+ instance() : Defaults
+ getSkillLevel(skillID: int) : int
+ getIndustryImplantTimeModifiers() : array
+ setBpMeLevel() : void
+ setBpTeLevel() : void
trivial getters ommited.
MyDefaults
For user implementation. Also serves as example for the intended way of extending iveeCore
EmdrConsumer
instance: EmdrConsumer
trackedTypeIDs: array
trackedMarketRegionIDs: array
defaultRegionID: int
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
FitParser
+ parseEftFit(eftFit: string) : MaterialParseResult
+ parseXmlFit(fitDom: DOMDocument) : MaterialParseResult
+ parseScanResult(scanResult: string) : MaterialParseResult
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
TeamsUpdater
IndustryFacilitiesUpdater
IndustrySystemsUpdater
MarketPricesUpdater
SpecialitiesUpdater


Type
- _types: array
- _typeNames: array
- _internalCacheHit: int
typeID: int
groupID: int
categoryID: int
typeName: String
volume: float
portionSize: int
basePrice: int
typeMaterials: array
reprocessingSkillID: int
crestAveragePrice: float
crestAdjustedPrice: float
crestPriceDate: int
+ getType(typeID: int) : Type
+ getTypeIdByName(typeName: string) : int
# loadTypeNames() : void
+ deleteTypesFromCache(typeIDs: array) : bool
+ factory(typeID: int, subtypeInfo: array) : Type
# getSubtypeInfo(typeID: int) : array
# decideType(subtypeInfo: array) : String
# __construct(typeID: int) : Type
# queryAttributes() : array
# setAttributes(row: array) : void
+ getTypeID() : int
+ getGroupID() : int
+ getCategoryID() : int
+ getName() : String
+ getVolume() : float
+ getPortionSize() : int
+ getBasePrice() : int
+ isReprocessable() : bool
+ getTypeMaterials() : array
+ getReprocessingSkillID() : int
+ getCrestPriceDate() : int
+ getCrestAveragePrice(maxPriceDataAge: int) : float
+ getCrestAdjustedPrice(maxPriceDataAge: int) : float
+ getReprocessingMaterialMap(batchSize: int, equipmentYield: f...
+ calcReprocessingTaxFactor(standings: float) : float
Sellable
marketGroupID: 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
__construct(typeID: int) : Sellable
queryAttributes() : array
setAttributes(row: array) : void
+ getMarketGroupID() : int
+ onMarket() : bool
+ 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
Manufacturable
producedFromBlueprintID: int
queryAttributes() : array
setAttributes(row: array) : void
+ getBlueprint() : Blueprint
ReactionProduct
productOfReactionIDs: array
__construct(typeID: int) : ReactionProduct
+ getReactions() : array
+ getReactionIDs() : array
Blueprint
productTypeID: int
maxProductionLimit: int
rank: int
activityMaterials: array
activitySkills: array
activityTimes: array
baseResearchCostMultiplier: array
__construct(typeID: int) : Blueprint
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...
+ getMaterials() : array
+ getMaterialsForActivity(activityID: int) : array
+ getProductTypeID() : int
+ getProduct() : Manufacturable
+ getSkillMapForActivity(activityID: int) : SkillMap
+ getBaseTimeForActivity(activityID: int) : int
+ getRank() : int
+ getMaxProductionLimit() : int
+ isReprocessable() : bool
+ calcBaseResearchTime(startLevel: int, endLevel: int) : int
+ calcResearchCostMultiplier(startLevel: int, endLevel: int) : float
Relic
activityMaterials: array
activitySkills: array
activityTimes: array
reverseEngineersBlueprintIDs: array
reverseEngineersBlueprintIDsByRaceID: array
decryptorGroupID: int
reverseEngineerProbability: float
reverseEngineerOutputRuns: int
encryptionSkillID: int
datacoreSkillIDs: array
__constructor(typeID: int) : Relic
+ reverseEngineer(iMod: IndustryModifier, reverseEngineerBpI...
+ getReverseEngineerBlueprintIDs() : array
+ getReverseEngineerBlueprintIDsByRaceID(raceID: int) : array
+ getReverseEngineerOutputRuns() : int
+ getReverseEngineeringProbability() : float
+ getMaterialsForActivity(activityID: int) : array
+ getSkillMapForActivity(activityID: int) : SkillMap
+ getBaseTimeForActivity(activityID: int) : int
+ calcReverseEngineeringProbability() : float
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
inventedFromBlueprintID: int
queryAttributes() : array
setAttributes(row: array) : void
+ getInventorBlueprint() : InventableBlueprint
+ getBuyPrice(maxPriceDataAge: int) : void
+ getSellPrice(maxPriceDataAge: int) : void
+ invent(iMod: IndustryModifier, decryptorID: int, recursive: bo...
+ copyInventManufacture(iMod: IndustryModifier, decryptorID: ...
InventorBlueprint
inventsBlueprintIDsChance: array
decryptorGroupID: int
encryptionSkillID: int
datacoreSkillIDs: array
__construct(typeID: int) : InventorBlueprint
+ invent(iMod: IndustryModifier, inventedBpID: int, decryptorID...
# getAndCheckDecryptor(decryptorID: int) : Decryptor
+ copyInventManufacture(iMod: IndustryModifier, inventedBpID...
# getInventableBlueprintIDs() : array
+ getDecryptorIDs() : array
+ calcInventionChance(inventedBpID: int, metaLevel: int) : float
REBlueprint
raceDecryptor: array
reverseEngineeredFromRelicIDs: array
reverseEngineeringDecryptorID: array
__construct(typeID: int) : REBlueprint
+ getReverseEngineeringRelicIDs() : array
+ getReverseEngineeringDecryptorID() : int


ProcessData
activity: int
producesTypeID: int
producesQuantity: int
processTime: int
processCost: float
assemblyLineID: int
solarSystemID: int
teamID: 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
+ getTeamID() : int
+ getTotalProcessCost() : float
+ getMaterialBuyCost() : float
+ getTotalMaterialBuyCost() : float
+ getTotalCost(maxPriceDataAge: int) : float
+ getMaterialMap() : MaterialMap
+ getTotalMaterialMap() : MaterialMap
+ getMaterialVolume() : float
+ getTotalMaterialVolume() : float
+ getSkillMap() : SkillMap
+ getTotalSkillMap() : SkillMap
+ getTime() : int
+ getTotalTime() : int
+ getTotalTimes() : array
+ getTotalProfit(maxPriceDataAge: int) : float
+ printData() : void
0..*
ReactionProcessData
inputMaterialMap: MaterialMap
outputMaterialMap: MaterialMap
cycles: float
withRefining: bool
withFeedback: bool
+ __construct(inputMaterialMap: MaterialMap, ou...
+ geInputMaterialMap() : MaterialMap
+ getOutputMaterialMap() : MaterialMap
+ getCycles() : float
+ getTime() : float
+ withRefining() : bool
+ withFeedback() : bool
+ getInputBuyCost(maxPriceDataAge: int) : float
+ getOutputBuyValue(maxPriceDataAge: int) : float
+ getProfit(maxPriceDataAge: int) : float
ManufactureProcessData
bpMeLevel: int
bpPeLevel: int
+ __construct(producesTypeID: int, producesQua...
+ getMeLevel() : int
+ getPeLevel() : int
+ getSlotCost() : float
+ getTotalCostPerUnit(maxPriceDataAge: int) : float
+ getTotalProfit(maxPriceDataAge: int) : float
+ printData() : void
InventionProcessData
inventionChance: float
resultRuns: int
resultME: int
resultTE: int
+ __construct(producesTypeID: int, inventionTime: int, proce...
+ getResultRuns() : int
+ getResultME() : int
+ getResultPE() : int
+ getInventionChance() : float
+ getSuccessTime() : float
+ getTotalSuccessTime() : float
+ getTotalSuccessTimes() : array
+ getSuccessMaterialMap() : MaterialMap
+ getTotalSuccessMaterialMap() : MaterialMap
+ getSuccessMaterialVolume() : float
+ getTotalSuccessMaterialVolume() : float
+ getProcessCost() : float
+ getSuccessProcessCost() : float
+ getTotalSuccessProcessCost() : float
+ getSuccessMaterialBuyCost(maxPriceDataAge: int) : float
+ getTotalSuccessMaterialBuyCost(maxPriceDataAge: int) : float
+ getTotalSuccessCost(maxPriceDataAge: int) : float
+ printData() : void
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
ReverseEngineerProcessData
+ __construct(reverseEngineerBpID: int, reverseE...


IndustryModifier
assemblyLines: array
solarSystem: SolarSystem
tax: float
teams: array
skillTimeModifiers: array
implantTimeModifiers: array
+ getByNpcStation(stationID: int) : IndustryModifier
+ getBySystemIdForPos(solarSystemID: int, tax: float) : IndustryModifier
+ getBySystemIdForAllNpcStations(solarSystemID: int) : IndustryModifier
+ getBySystemIdWithAssembly(solarSystemID: int, assemblyLineTypeID...
+ __construct(system: SolarSystem, assemblyLines: array, teams: array, ...
+ getAssemblyLines() : array
+ getAssemblyLinesForActivity(activityID: int) : array
+ getSolarSystem() : SolarSystem
+ getTax() : float
+ getImplantTimeModifiers() : array
+ getImplantTimeModifierForActivity(activityID: int) : float
+ setImplantTimeModifierForActivity(modifier: float, activityID: int) : void
+ setImplantTimeModifiers(modifiers: array) : void
+ getSkillTimeModifiers() : array
+ getSkillTimeModifierForActivity(activityID: int) : float
+ setSkillTimeModifiers(modifiers: array) : void
+ setSkillTimeModifierForActivity(modifier: float, activityID: int) : void
+ getTeams() : array
+ getTeamsForActivity(activityID: int) : array
+ setTeams(teams: array) : void
+ setTeamsForActivity(teams: array, activityID: int) : void
+ isActivityPossible(activityID: int, type: Type) : bool
+ getModifier(activityID: int, type: Type) : array
+ getBestAssemblyLineForActivity(activityID: int, type: Type) : Assembly...
+ getBestTeamForActivity(activityID: int, type: Type) : Team
AssemblyLine
- _assemblyLines: array
- _internalCacheHit: int
assemblyLineTypeID: int
assemblyLineTypeName: string
baseTimeMultiplier: float
baseMaterialMultiplier: float
baseCostMultiplier: float
activityID: int
groupModifiers: array
categoryModifiers: array
+ getAssemblyLine(assemblyLineTypeID: int) : AssemblyLine
+ getBestPosAssemblyLineTypeIDs(systemSecurity: float) : array
+ getHisecStationAssemblyLineTypeIDs() : array
# __construct(assemblyLineTypeID: int) : AssemblyLine
+ getAssemblyLineTypeID() : int
+ getAssemblyTypeName() : string
+ getBaseTimeMultiplier() : float
+ getBaseMaterialMultiplier() : float
+ getBaseCostMultiplier() : float
+ getActivityID() : int
+ getGroupModifiers() : array
+ getCategoryModifiers() : array
+ getModifiersForType(type: Type) : array
+ isTypeCompatible(type: Type) : bool
1..*
Team
- _teams: array
- _internalCacheHit: int
teamID: int
solarSystemID: int
creationTime: int
expiryTime: int
activityID: int
teamName: string
costModifier: float
specialityID: int
bonusIDs: array
bonusValues: array
workerSpecialities: array
+ getTeam(teamID: int) : Team
# __construct(teamID: int) : Team
+ getTeamID() : int
+ getSolarSystemID() : int
+ getCreationTime() : int
+ getExpiryTime() : int
+ getActivityID() : int
+ getTeamName() : string
+ getCostModifier() : float
+ getSpecialityID() : int
+ getBonusIDs() : array
+ getBonusValues() : array
+ getWorkerSpecialities() : array
+ isTypeCompatible(type: Type) : bool
+ isGroupIDCompatible(groupID: int) : bool
+ getWorkerIDsForGroupID(groupID: int) : array
+ getModifiersForGroupID(groupID: int) : array
0..*
1
SolarSystem
- _systems: array
- _systemNames: array
- _internalCacheHit: int
solarSystemID: int
systemName: string
regionID: int
constellationID: int
security: float
industryIndexDate: int
industryIndices: array
stationIDs: array
teamIDs: array
+ getSolarSystem(solarSystemID: int) : SolarSystem
+ getSolarSystemIdByName(solarSystemName: string) : int
+ getSolarSystemByName(solarSystemName: string) : SolarSystem
# loadSolarSystemNames() : void
+ deleteSolarSystemsFromCache(solarSystemIDs: array) : bool
# __construct(solarSystemID: int) : SolarSystem
+ getSolarSystemID() : int
+ getSolarSystemName() : string
+ getRegionID() : int
+ getConstellationID() : int
+ getSecurity() : float
+ getStationIDs() : array
+ getStations() : array
+ getTeamIDs() : array
+ getTeams() : array
+ getIndustryIndexDate() : int
+ getIndustryIndices(maxIndexDataAge: int) : array
+ getIndustryIndexForActivity(activityID: int, maxIndexDataAge: int) : float
+ setIndustryIndices(indices: array) : void
1
0..*
Station
- _stations: array
- _internalCacheHit: int
stationID: int
stationName: string
solarSystemID: int
operationID: int
stationTypeID: int
corporationID: int
reprocessingEfficiency: float
tax: float
assemblyLineTypeIDs: array
+ getStation(stationID: int) : Station
+ deleteStationsFromCache(stationIDs: array) : bool
# __construct(stationID: int) : Station
+ getStationID() : int
+ getStationName() : string
+ getSolarSystemID() : int
+ getSolarSystem() : SolarSystem
+ getOperationID() : int
+ getCorporationID() : int
+ getReprocessingEfficiency() : float
+ getTax() : float
+ getAssemblyLineTypeIDs() : array
+ getAssemblyLineTypeIDsForActivity(activityID: int) : array
Speciality
- _specialities: array
- _internalCacheHit: int
specialityID: int
specialityName: string
specialityGroupIDs: array
+ getSpeciality(specialityID: int) : Speciality
# __construct(specialityID: int) : Speciality
+ getSpecialityID() : int
+ getSpecialityGroupIDs() : array
+ appliesToGroupID(groupID: int) : bool


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(equipmentYield: float, reprocessingTaxFa...
+ getMaterialVolume() : float
+ getMaterialBuyCost(maxPriceDataAge: int) : float
+ getMaterialSellValue(maxPriceDataAge: int) : float
SkillMap
SkillMap: array
+ sanityCheckSkillLevel(skillLevel: int) : bool
+ addSkill(skillID: int, level: int) : void
+ addSkillMap(skillMap: SkillMap) : void
+ getSkills() : array
MaterialParseResult
unparseables: array
+ addUnparseable(unparseable: string) : void
+ getUnparseables() : array