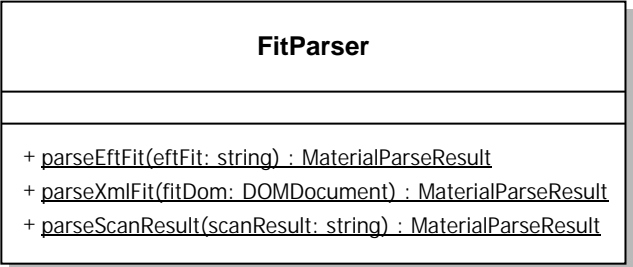
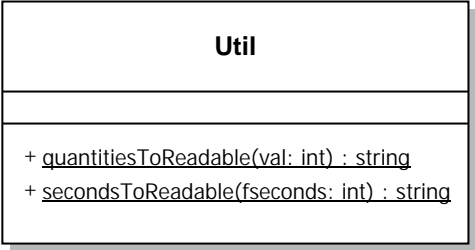
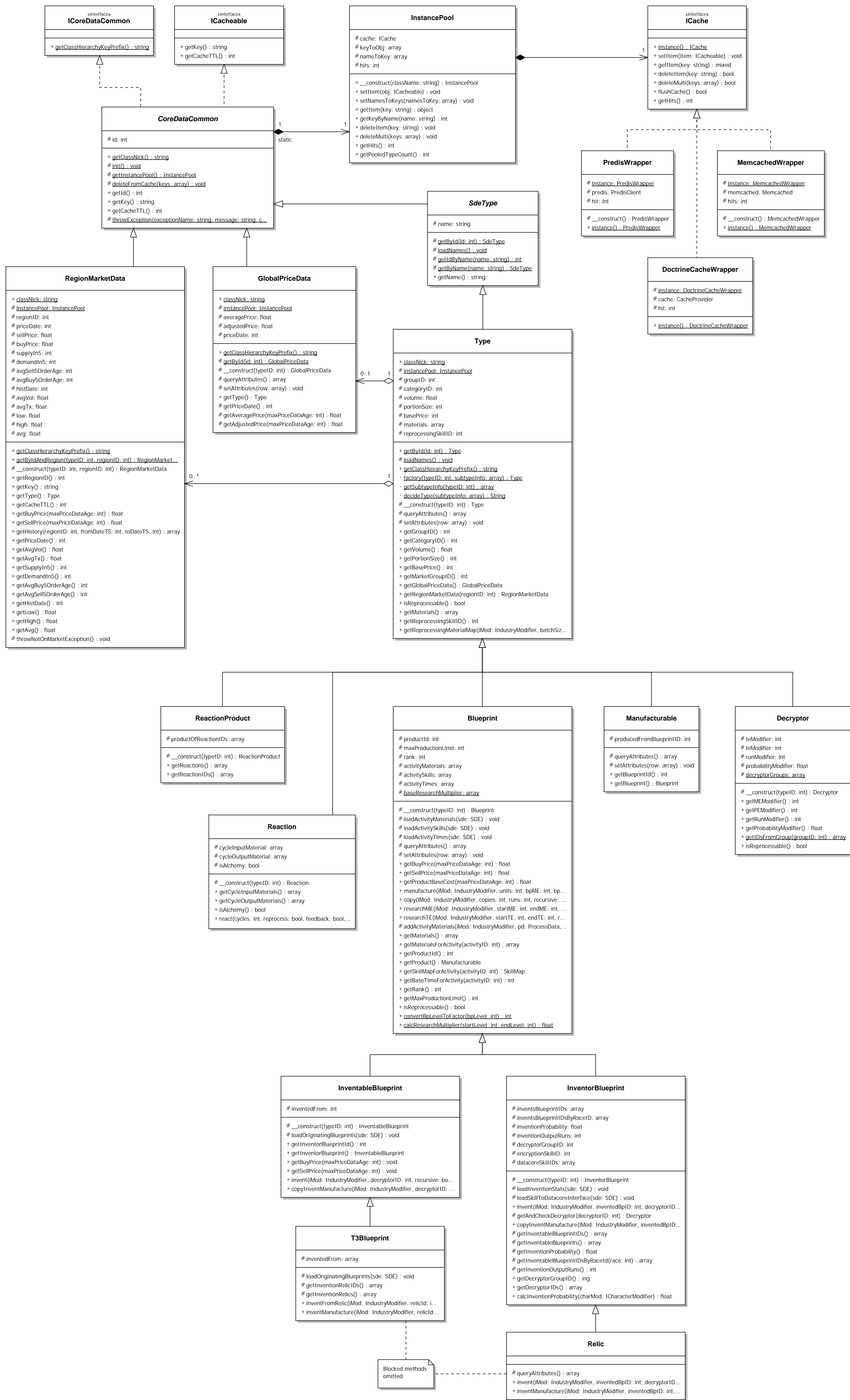


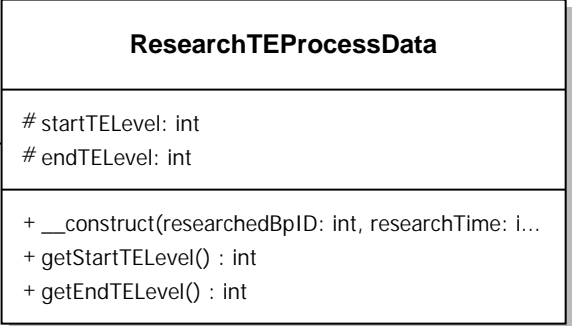
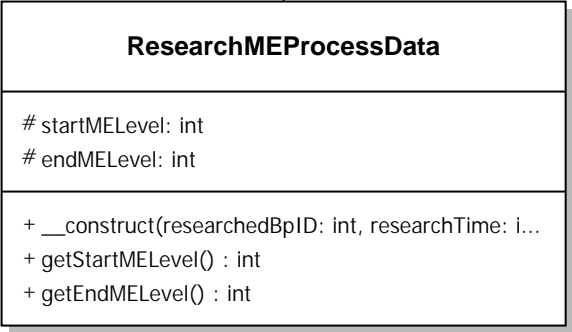
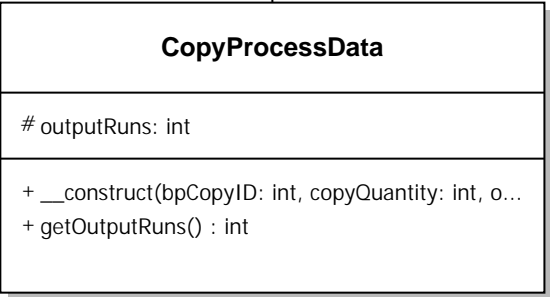
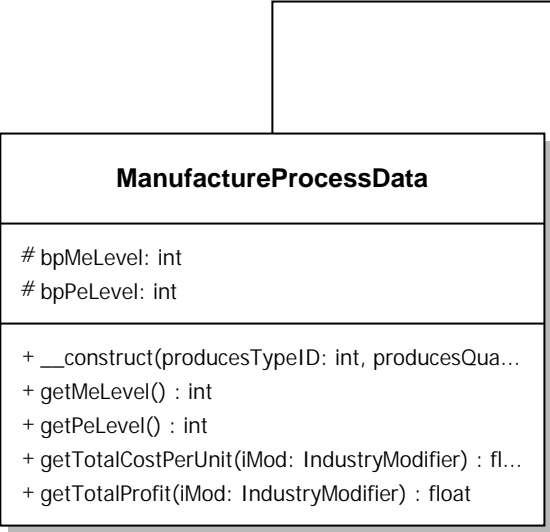
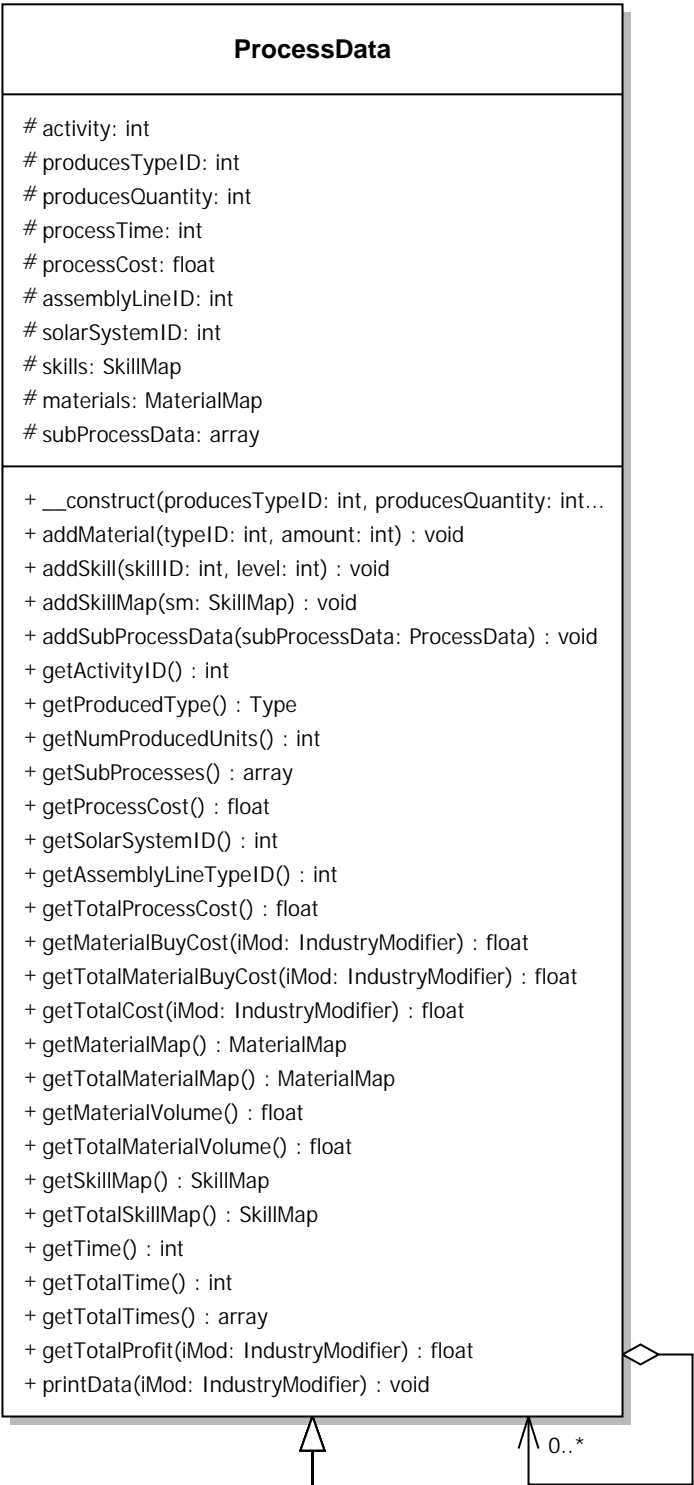
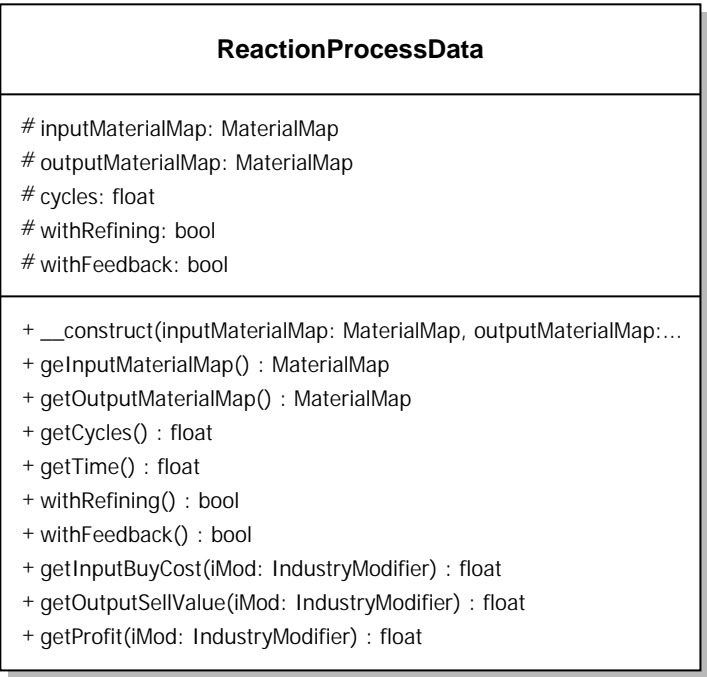


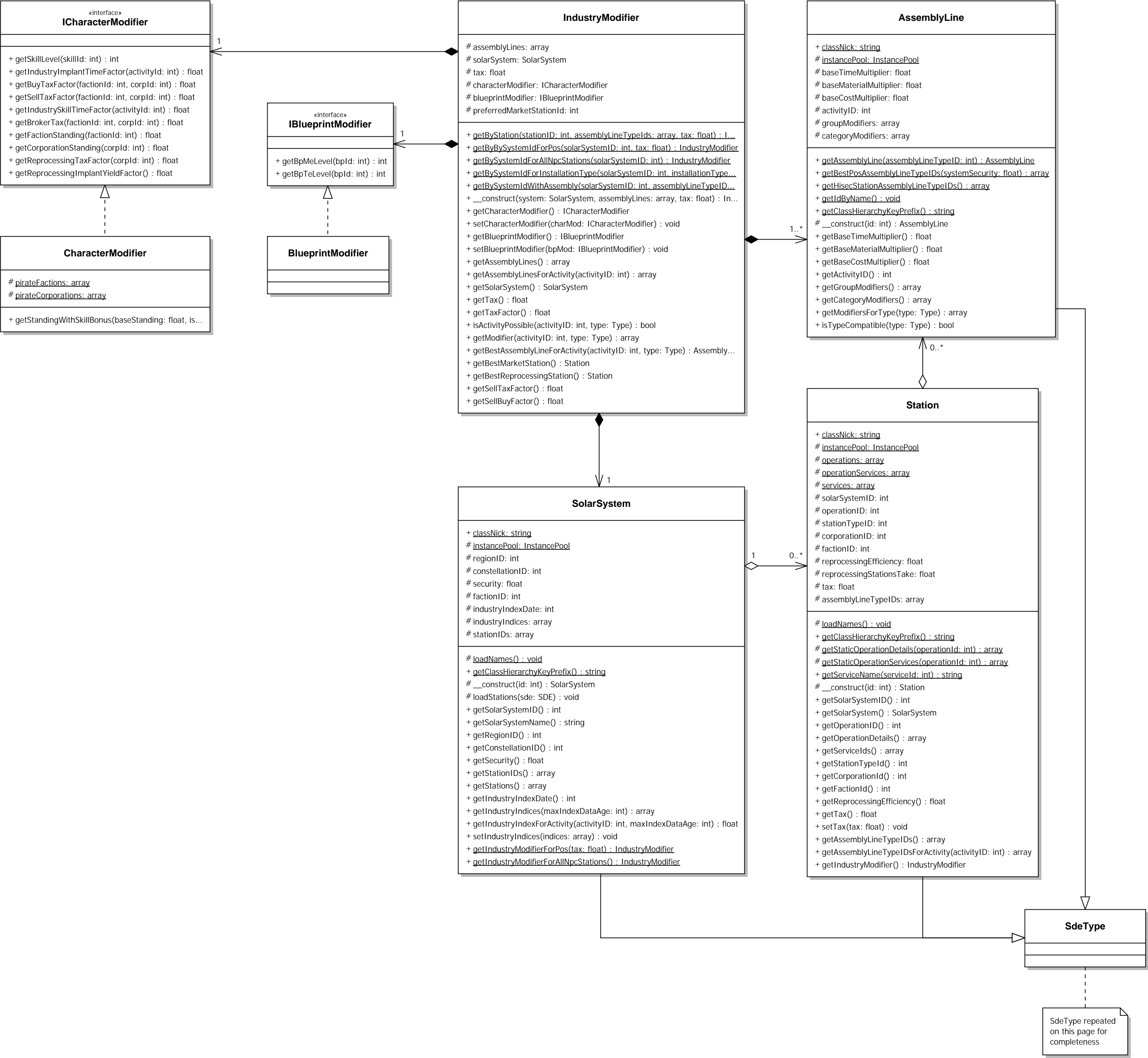
trivial getters and  
setters omitted.

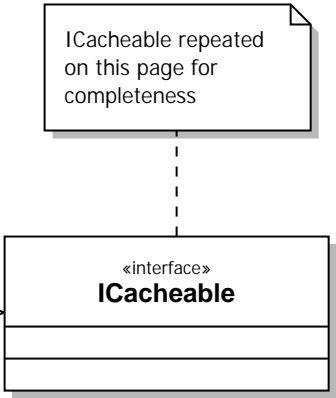
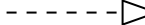
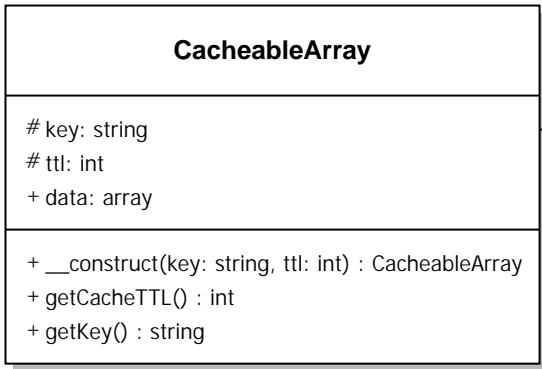
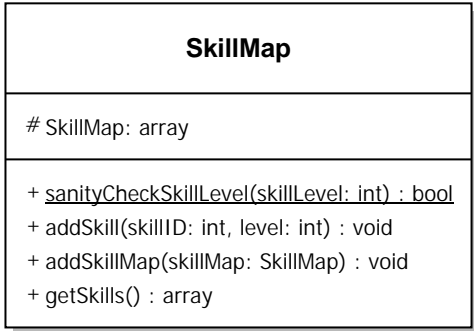
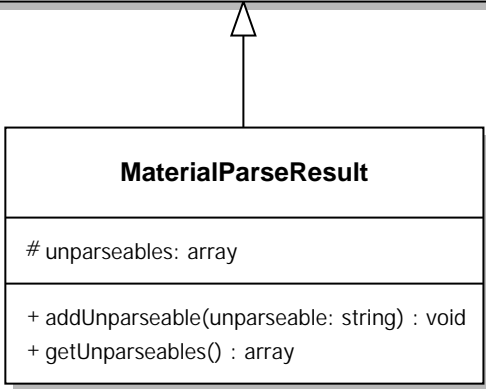
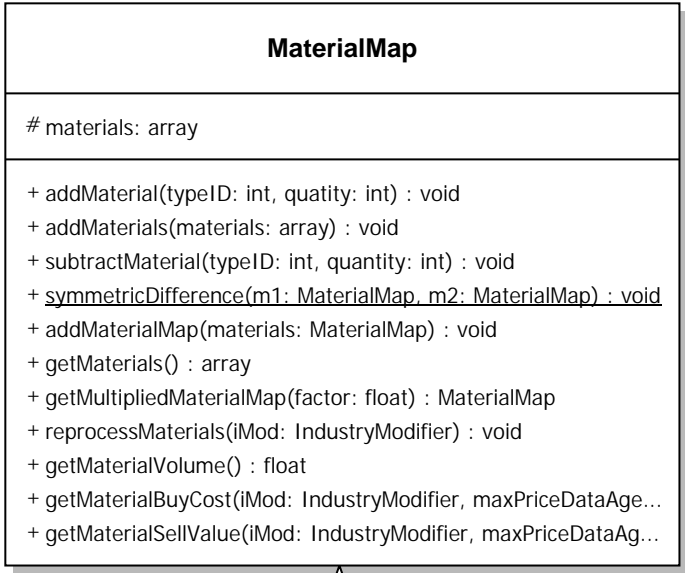


Exception classes  
omitted.









EmdrConsumer
# <u>instance</u> : EmdrConsumer # trackedTypeIDs: array # trackedMarketRegionIDs: array # regions: array # sde: SDE # cache: ICache # emdrPriceUpdateClass: string # emdrHistoryUpdateClass: string
+ <u>instance()</u> : 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... # <u>cmp(a: array, b: array)</u> : 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

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

