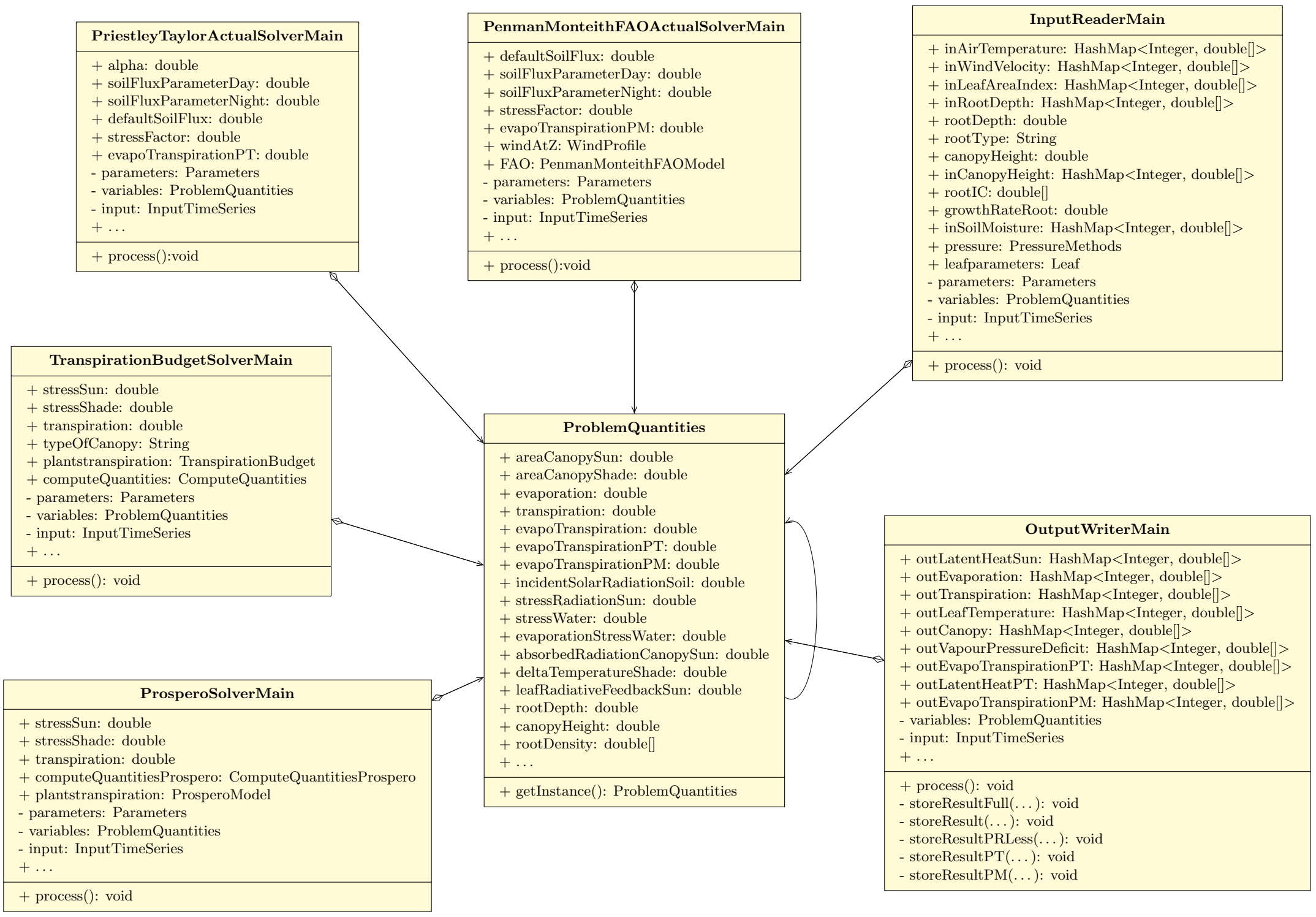
PriestleyTaylorActualSolverMain
+ alpha: double
+ soilFluxParameterDay: double
+ soilFluxParameterNight: double
+ defaultSoilFlux: double
+ stressFactor: double
+ evapoTranspirationPT: double
- parameters: Parameters
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process():void
PenmanMonteithFAOActualSolverMain
+ defaultSoilFlux: double
+ soilFluxParameterDay: double
+ soilFluxParameterNight: double
+ stressFactor: double
+ evapoTranspirationPM: double
+ windAtZ: WindProfile
+ FAO: PenmanMonteithFAOModel
- parameters: Parameters
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process():void
InputReaderMain
+ inAirTemperature: HashMap<Integer, double[]>
+ inWindVelocity: HashMap<Integer, double[]>
+ inLeafAreaIndex: HashMap<Integer, double[]>
+ inRootDepth: HashMap<Integer, double[]>
+ rootDepth: double
+ rootType: String
+ canopyHeight: double
+ inCanopyHeight: HashMap<Integer, double[]>
+ rootIC: double[]
+ growthRateRoot: double
+ inSoilMoisture: HashMap<Integer, double[]>
+ pressure: PressureMethods
+ leafparameters: Leaf
- parameters: Parameters
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process(): void
TranspirationBudgetSolverMain
+ stressSun: double
+ stressShade: double
+ transpiration: double
+ typeOfCanopy: String
+ plantstranspiration: TranspirationBudget
+ computeQuantities: ComputeQuantities
- parameters: Parameters
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process(): void
ProblemQuantities
+ areaCanopySun: double
+ areaCanopyShade: double
+ evaporation: double
+ transpiration: double
+ evapoTranspiration: double
+ evapoTranspirationPT: double
+ evapoTranspirationPM: double
+ incidentSolarRadiationSoil: double
+ stressRadiationSun: double
+ stressWater: double
+ evaporationStressWater: double
+ absorbedRadiationCanopySun: double
+ deltaTemperatureShade: double
+ leafRadiativeFeedbackSun: double
+ rootDepth: double
+ canopyHeight: double
+ rootDensity: double[]
+ ...
+ getInstance(): ProblemQuantities
OutputWriterMain
+ outLatentHeatSun: HashMap<Integer, double[]>
+ outEvaporation: HashMap<Integer, double[]>
+ outTranspiration: HashMap<Integer, double[]>
+ outLeafTemperature: HashMap<Integer, double[]>
+ outCanopy: HashMap<Integer, double[]>
+ outVapourPressureDeficit: HashMap<Integer, double[]>
+ outEvapoTranspirationPT: HashMap<Integer, double[]>
+ outLatentHeatPT: HashMap<Integer, double[]>
+ outEvapoTranspirationPM: HashMap<Integer, double[]>
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process(): void
- storeResultFull(...): void
- storeResult(...): void
- storeResultPRLess(...): void
- storeResultPT(...): void
- storeResultPM(...): void
ProsperoSolverMain
+ stressSun: double
+ stressShade: double
+ transpiration: double
+ computeQuantitiesProspero: ComputeQuantitiesProspero
+ plantstranspiration: ProsperoModel
- parameters: Parameters
- variables: ProblemQuantities
- input: InputTimeSeries
+ ...
+ process(): void