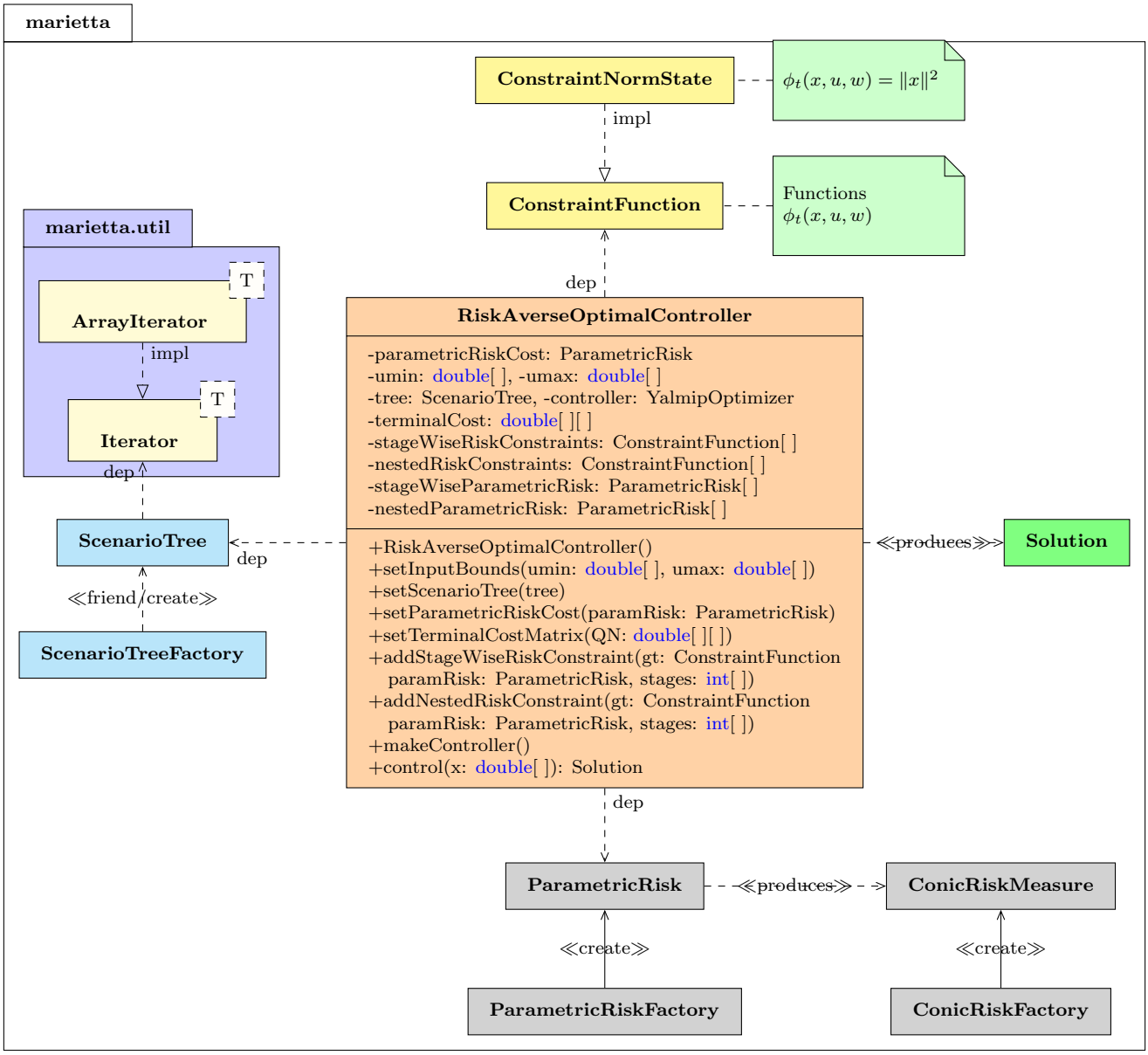

Figure 1: Simple UML — structure of `marietta`.

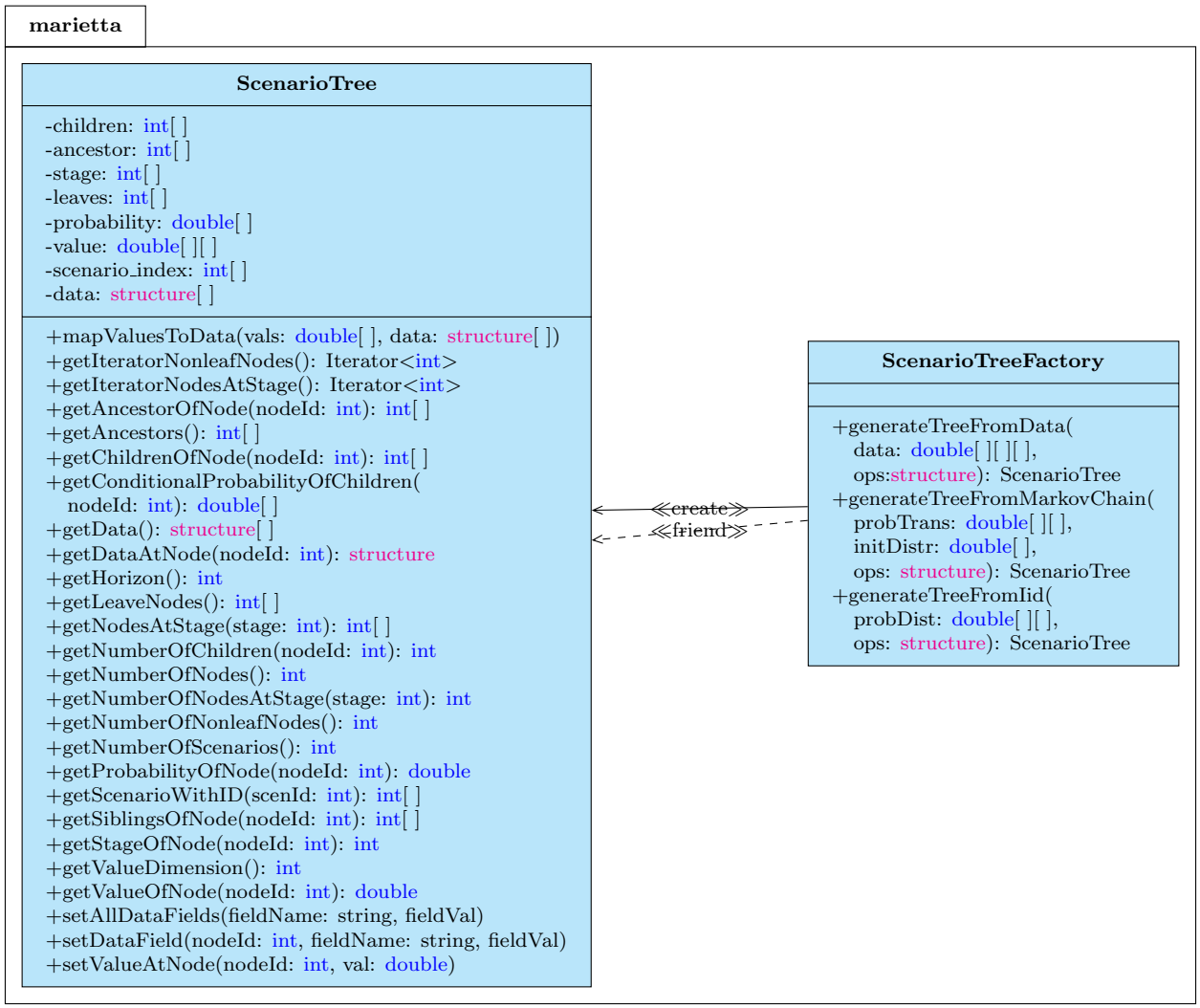

Figure 2: UML of `marietta.ScenarioTree` and the associated factory. `marietta.ScenarioTreeFactory` can be used to make scenario trees from IID and Markov processes as well as raw data.

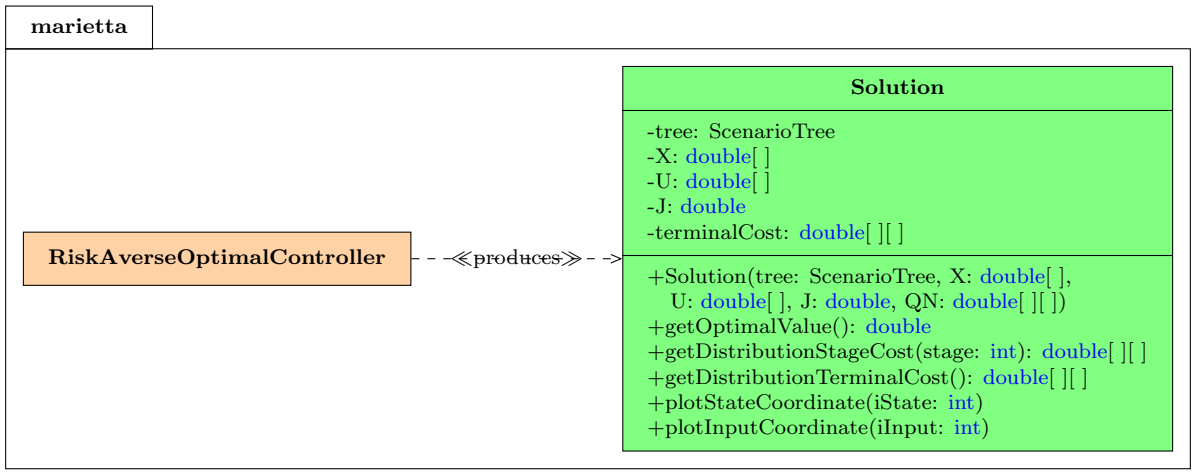

Figure 3: UML for `marietta.Solution`. `RiskAverseOptimalController` produces a solution (given an initial condition x_0 which comprises of (i) a sequence of optimal control laws $u_0^\star, u_1^\star, \ldots, u_{N-1}^\star$, which corresponds to control actions on all non-leaf nodes of the tree, (ii) the corresponding optiamal evolution of the system state. To that corresponds an optimal value $J^\star$. This class allows to plot the solutions.

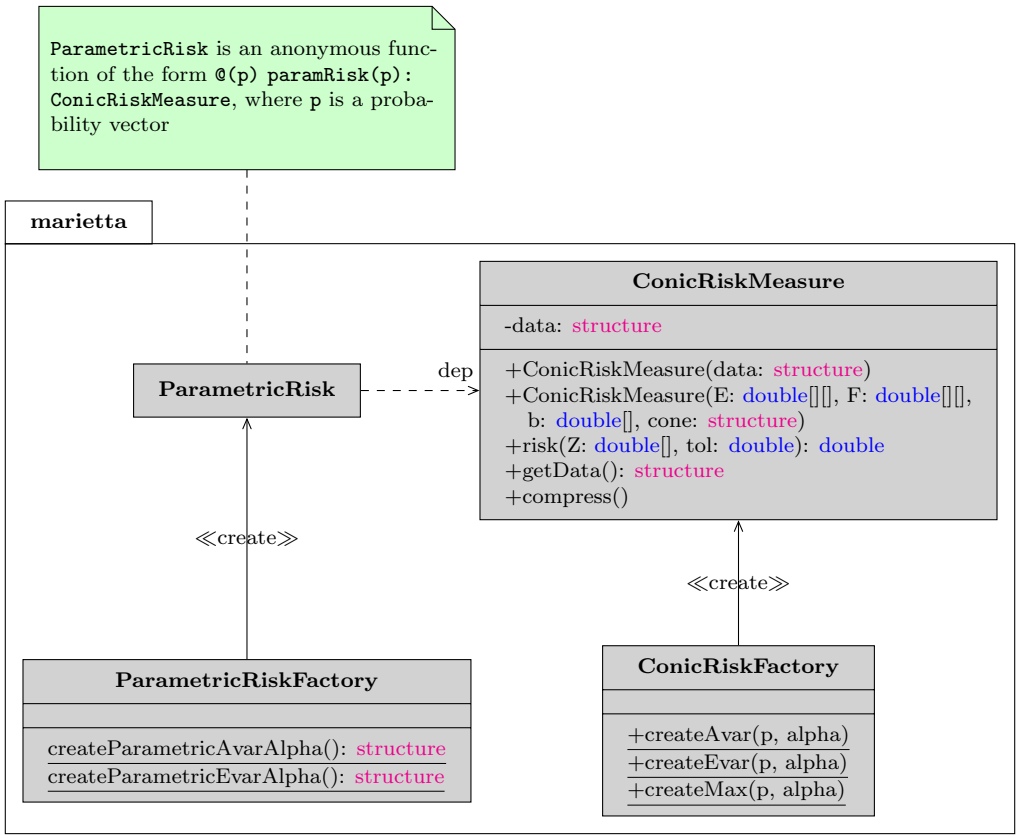

Figure 4: UML for the classes in `marietta` associated with risk measures. Class `ParametricRiskFactory` is used to create parametric risks — conic risk measures which depend parametrically on the the underlying probability vector π.

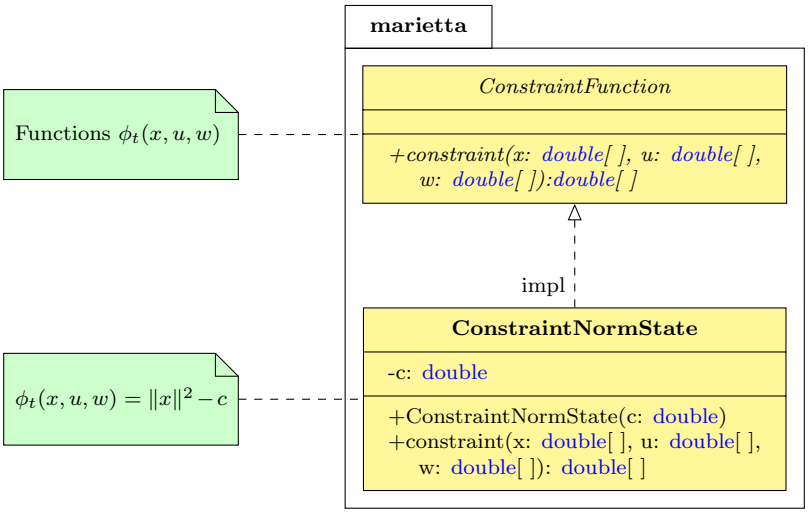

Figure 5: UML of constraint functions. These are functions of the form $\phi_t : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \ni (x, u, w) \mapsto \phi_t(x, u, w) \in \mathbb{R}$. We need to impose constraints of the form $r_t[\phi_t(x, u, w)] \leq 0$ as explained in the paper. An example of such a function is `marietta.ConstraintNormState` which corresponds to $\phi_t(x, u, w) = \|x\|^2 - c$, for some given $c > 0$.

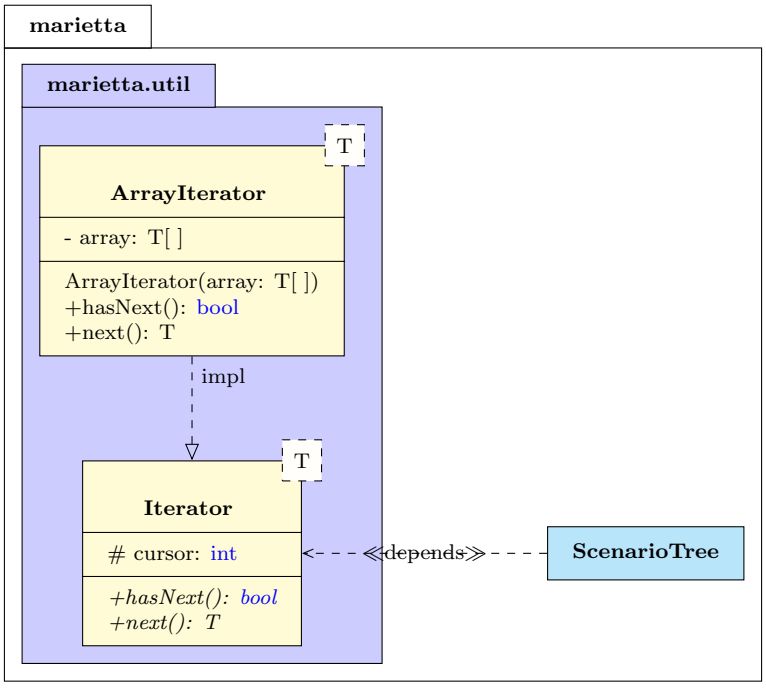

Figure 6: Some utility functions.