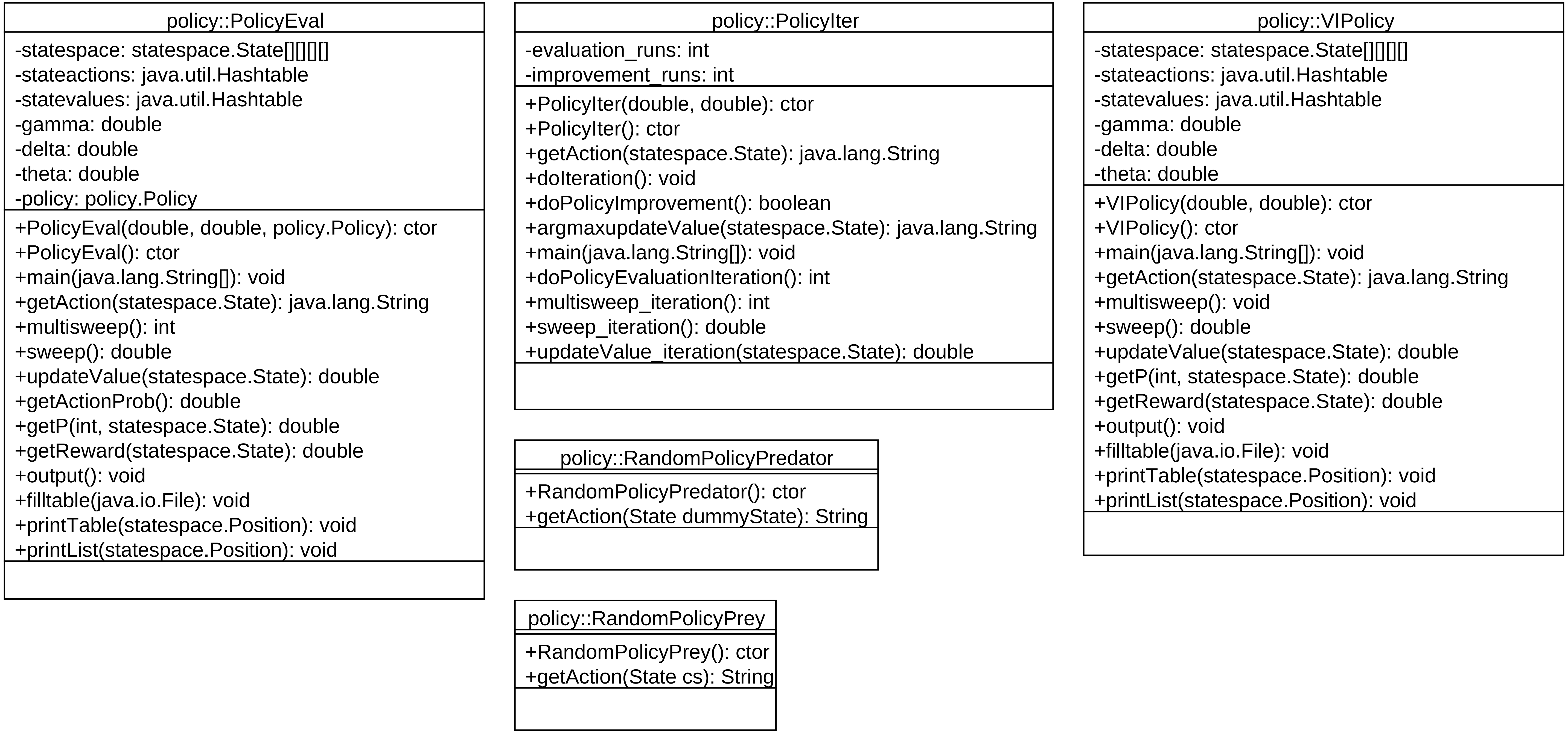

```
policy::PolicyEval
-statespace: statespace.State[][][][]
-stateactions: java.util.Hashtable
-statevalues: java.util.Hashtable
-gamma: double
-delta: double
-theta: double
-policy: policy.Policy
+PolicyEval(double, double, policy.Policy): ctor
+PolicyEval(): ctor
+main(java.lang.String[]): void
+getAction(statespace.State): java.lang.String
+multisweep(): int
+sweep(): double
+updateValue(statespace.State): double
+getActionProb(): double
+getP(int, statespace.State): double
+getReward(statespace.State): double
+output(): void
+filltable(java.io.File): void
+printTable(statespace.Position): void
+printList(statespace.Position): void
```

```
policy::PolicyIter
-evaluation_runs: int
-improvement_runs: int
+PolicyIter(double, double): ctor
+PolicyIter(): ctor
+getAction(statespace.State): java.lang.String
+doIteration(): void
+doPolicyImprovement(): boolean
+argmaxupdateValue(statespace.State): java.lang.String
+main(java.lang.String[]): void
+doPolicyEvaluationIteration(): int
+multisweep_iteration(): int
+sweep_iteration(): double
+updateValue_iteration(statespace.State): double
```

```
policy::RandomPolicyPredator
+RandomPolicyPredator(): ctor
+getAction(State dummyState): String
```

```
policy::RandomPolicyPrey
+RandomPolicyPrey(): ctor
+getAction(State cs): String
```

```
policy::VIPolicy
-statespace: statespace.State[][][][]
-stateactions: java.util.Hashtable
-statevalues: java.util.Hashtable
-gamma: double
-delta: double
-theta: double
+VIPolicy(double, double): ctor
+VIPolicy(): ctor
+main(java.lang.String[]): void
+getAction(statespace.State): java.lang.String
+multisweep(): void
+sweep(): double
+updateValue(statespace.State): double
+getP(int, statespace.State): double
+getReward(statespace.State): double
+output(): void
+filltable(java.io.File): void
+printTable(statespace.Position): void
+printList(statespace.Position): void
```