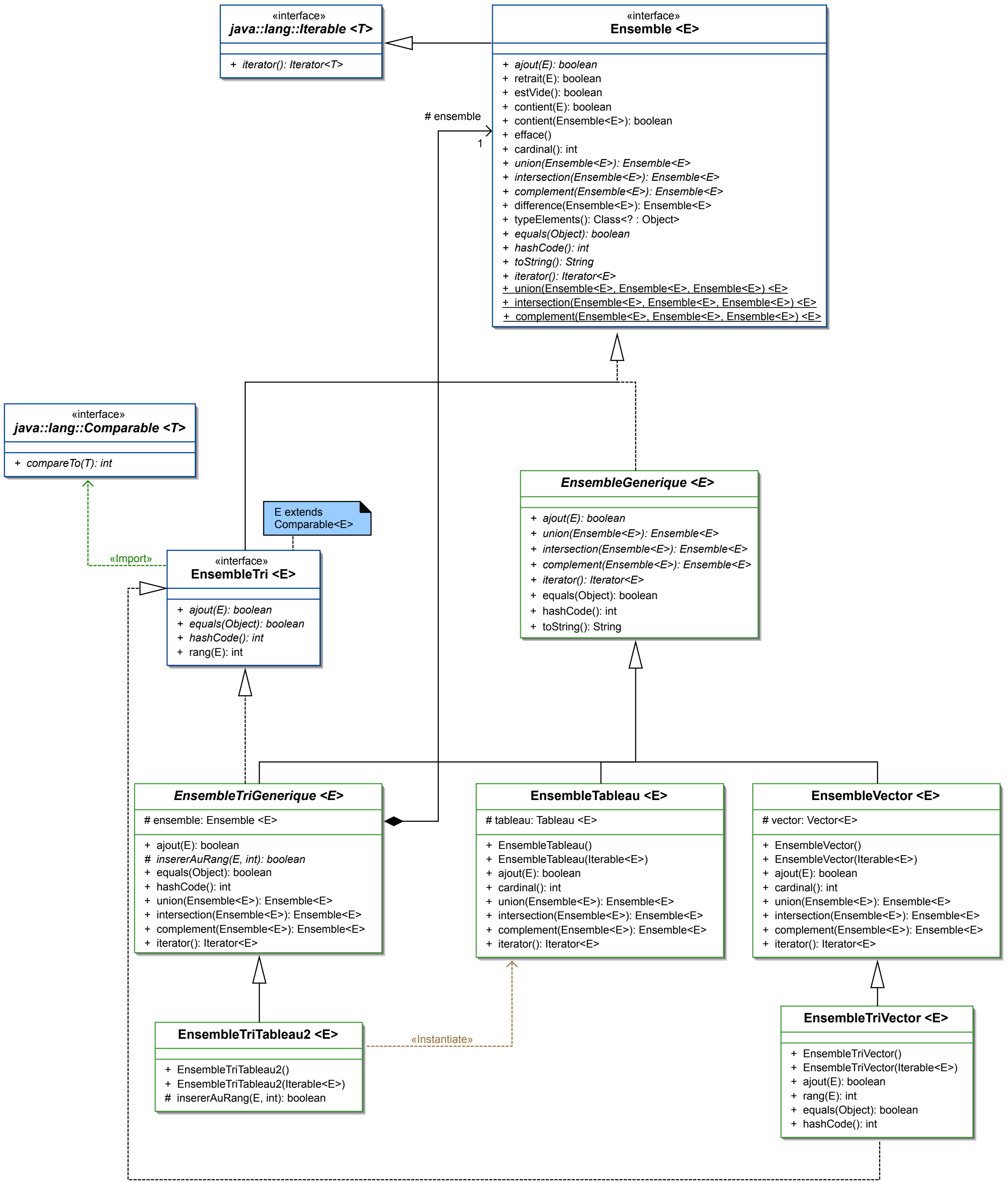
«interface»
java::lang::Iterable <T>
+ iterator(): Iterator<T>
«interface»
Ensemble <E>
+ ajout(E): boolean
+ retrait(E): boolean
+ estVide(): boolean
+ contient(E): boolean
+ contient(Ensemble<E>): boolean
+ efface()
+ cardinal(): int
+ union(Ensemble<E>): Ensemble<E>
+ intersection(Ensemble<E>): Ensemble<E>
+ complement(Ensemble<E>): Ensemble<E>
+ difference(Ensemble<E>): Ensemble<E>
+ typeElements(): Class<? : Object>
+ equals(Object): boolean
+ hashCode(): int
+ toString(): String
+ iterator(): Iterator<E>
+ union(Ensemble<E>, Ensemble<E>, Ensemble<E>) <E>
+ intersection(Ensemble<E>, Ensemble<E>, Ensemble<E>) <E>
+ complement(Ensemble<E>, Ensemble<E>, Ensemble<E>) <E>
# ensemble
1
«interface»
java::lang::Comparable <T>
+ compareTo(T): int
«Import»
E extends Comparable<E>
«interface»
EnsembleTri <E>
+ ajout(E): boolean
+ equals(Object): boolean
+ hashCode(): int
+ rang(E): int
EnsembleGenerique <E>
+ ajout(E): boolean
+ union(Ensemble<E>): Ensemble<E>
+ intersection(Ensemble<E>): Ensemble<E>
+ complement(Ensemble<E>): Ensemble<E>
+ iterator(): Iterator<E>
+ equals(Object): boolean
+ hashCode(): int
+ toString(): String
EnsembleTriGenerique <E>
ensemble: Ensemble <E>
+ ajout(E): boolean
# insererAuRang(E, int): boolean
+ equals(Object): boolean
+ hashCode(): int
+ union(Ensemble<E>): Ensemble<E>
+ intersection(Ensemble<E>): Ensemble<E>
+ complement(Ensemble<E>): Ensemble<E>
+ iterator(): Iterator<E>
EnsembleTableau <E>
tableau: Tableau <E>
+ EnsembleTableau()
+ EnsembleTableau(Iterable<E>)
+ ajout(E): boolean
+ cardinal(): int
+ union(Ensemble<E>): Ensemble<E>
+ intersection(Ensemble<E>): Ensemble<E>
+ complement(Ensemble<E>): Ensemble<E>
+ iterator(): Iterator<E>
EnsembleVector <E>
vector: Vector<E>
+ EnsembleVector()
+ EnsembleVector(Iterable<E>)
+ ajout(E): boolean
+ cardinal(): int
+ union(Ensemble<E>): Ensemble<E>
+ intersection(Ensemble<E>): Ensemble<E>
+ complement(Ensemble<E>): Ensemble<E>
+ iterator(): Iterator<E>
EnsembleTriTableau2 <E>
+ EnsembleTriTableau2()
+ EnsembleTriTableau2(Iterable<E>)
# insererAuRang(E, int): boolean
«Instantiate»
EnsembleTriVector <E>
+ EnsembleTriVector()
+ EnsembleTriVector(Iterable<E>)
+ ajout(E): boolean
+ rang(E): int
+ equals(Object): boolean
+ hashCode(): int