

# Tree

## De interface Tree

De nodes van een boomstructuur zullen ergens in opgeslagen moeten worden om ermee te kunnen werken. Die collectie moet tevens gebruikt kunnen worden om via een `Treeliterator` te doorlopen. De interface `Tree` is daarvoor bedoeld. Om elementen in de `Tree` op te kunnen slaan, en om ze met een `Treeliterator` te kunnen doorlopen, **Tree is dus een subinterface van `Treeliterable`**.

Om nodes toe te voegen zouden we `Tree` ook subinterface van `Collection` kunnen maken. Het is echter de vraag of er een is-een relatie bestaat tussen een `Tree` en een `Collection`. Lees hiervoor ook de documentatie bij `java.util.Collection`. Omdat we (voorlopig) alleen nodes willen kunnen toevoegen, kunnen we in dit geval beter kiezen voor een heeft-een relatie.

Om elementen toe te kunnen voegen krijgt `Tree` de methode **`add(E node)`**. Later zullen we deze methode in een implementerende class uitwerken, door zo'n node toe te voegen aan een interne collectie.

*Bespreek de opdrachten van dit hoofdstuk met uw docent.*