

Oefening 5: Binaire bomen (vervolg)

1.1 Hoeveel is dat alstublieft?

Voeg aan de klasse Boom een methode toe die het aantal knopen in de boom telt:

```
public int aantalKnopen ( ) ;
```

De implementatie hiervan gebeurt best door middel van het aanroepen van een gelijkaardige methode in klasse Knoop, die het aantal knopen telt die er nog onder de knoop in kwestie hangen (de knoop zelf inbegrepen). Dit kan als volgt:

- bij een interne knoop geeft deze methode de vraag door aan zowel zijn linker- als zijn rechterkind en gebruikt de resulterende antwoorden om het gewenste resultaat te berekenen;
- bij een blad produceert deze methode natuurlijk gewoon het getal één.

Gebruik deze methode om na te gaan dat er in de boom die je zonet hebt aangemaakt inderdaad 8 knopen zitten.

Schrijf nu volgens hetzelfde principe een methode `int som()` die de som van alle waardes in een boom berekent. Gebruik deze som methode om te controleren dat de boom die je hebt aangemaakt inderdaad een som van 82 heeft.

1.2 Ordnung muss sein

Een binaire boom is gesorteerd indien volgende eigenschap geldt voor elke knoop: inhoud van linkerkind \leq inhoud van knoop zelf \leq inhoud van rechterkind. Met andere woorden, als je de boom horizontaal overloopt van links naar rechts, mogen de waardes die je tegenkomt niet dalen.

Bovenstaande boom is dus gesorteerd. Vanaf nu beschouwen we enkel nog maar gesorteerde bomen.

Voeg aan de klasse Boom een methode *public boolean bevat (int waarde)* ; toe, die nagaat of een boom een bepaald getal bevat. Omdat de boom gesorteerd is, kan dit erg efficiënt gebeuren!

- in het geval van een interne knoop kijkt de methode na of hijzelf dit getal bevat en, zo nee, wordt deze vraag recursief doorgegeven aan zijn kinderen;
- in het geval van een blad wordt enkel maar nagekeken of hijzelf dit getal bevat.

Test je methode uit door na te gaan of 11 en 12 in de hoger getekende boom zitten.

1.3 Mag het ook ietsje meer zijn ?

Voeg nu ook volgende methode toe aan de klasse boom :

```
public void voegToe(int waarde)
```

waarmee je een bepaalde waarde kan toevoegen aan een gegeven boom. Er is een moeilijke manier om dit te doen en een gemakkelijke. De gemakkelijke manier is dat deze methode een waarde toevoegt

als nieuwe wortel van de boom en dan de bestaande boom als linker- of rechterkind aan deze nieuwe wortel hangt. De moeilijke manier wordt verderop als mogelijke uitbreiding beschreven. Herschrijf nu je *main* methode zodat deze niet langer meer gebruik maakt van setter-methodes maar nu de voegtoe-methode gebruikt om de voorbeeldboom te construeren.

1.4 Zal ik dat nog even voor u inpakken?

Maak een pakket boom en steek die klassen die je hebt aangemaakt in dit pakket.

Schrijf een klasse *BoomTest*, die zelf niet tot dit pakket behoort, en verplaats je *main* methode naar deze klasse. Controleer de toegangsrechten van je methoden/attributen en zorg ervoor dat deze je implementatie zo goed mogelijk encapsuleren.

1.5 Kom erbij!

Veralgemeen je implementatie door, in plaats van enkel gehele getallen, meer soorten van objecten toe te laten in je bomen. Al deze objecten moeten echter verplicht een aantal methodes implementeren anders kunnen ze niet in de boom opgenomen worden. Dit gaan we realiseren door gebruik te maken van een interface *BoomWaarde*, met daarin de verplichte methodes:

- methodes die je toelaten om twee BoomWaardes met elkaar te vergelijken (*lessThan*, *equals*)
- een methode om de *optelWaarde* op te vragen waarvoor de BoomWaarde zal meetellen indien we sommeren over de ganse boom.
- een *toString* om elke boomWaarde zinvol te kunnen afdrukken.

Maak daarna een klasse *BoomGetal* die deze interface zodanig implementeert dat je hiermee de functionaliteit die we al hadden voor gehele getallen kan nabootsen.

Maak dan een klasse *Kerstbal* met daarin twee attributen: een *String* kleur en een *double* diameter. Zorg ervoor dat je deze kerstballen in een Boom kan steken, gesorteerd van klein naar groot, en dat je deze boom dan kan gebruiken om het totale volume aan kerstbal te berekenen (het volume van een bol met diameter d is $(\pi d^3)/6$). Maak een boom met daarin een rode kerstbal van 5 cm diameter, een groene kerstbal van 7 cm en een gele kerstbal van 3 cm.

Hoeveel kubieke centimeter kerstbal heb je nu in totaal ?

Zoek een bepaalde kerstbal in de boom (let wel : 2 kerstballen zijn pas identiek als ze hetzelfde volume en dezelfde kleur hebben)

1.6 Extra:

Schrijf een methode *public Knoop voegToe (BoomWaarde waarde)* die zal toelaten om systematisch een boom op te bouwen te beginnen bij een blad, tot je de volledige boom hebt opgebouwd. Deze methode zal bij bladeren en interne knopen het volgende moeten doen:

- Bij een blad wordt een nieuwe interne knoop gemaakt (op de gepaste wijze) en deze wordt als resultaat van de functie teruggegeven;
- Een interne knoop geeft de vraag om de waarde toe te voegen door aan één van zijn kinderen (op gepaste wijze), doet dan natuurlijk iets met de resultwaarde die terugkomt, en geeft tot slot zichzelf terug.