

Bijkomende oefening bomen

29 maart 2018

In deze oefening krijg je een klasse `Vak` die sorteerbaar is op studiepunten en vervolgens op naam. Het vak 'BOP' komt bijvoorbeeld voor het vak 'OOP' en na het vak 'Web1' omwille van verschil in studiepunten. Het vak 'Algo' komt voor het vak 'Web2' omdat ze beide evenveel studiepunten hebben maar 'Algo' alfabetisch voor 'Web2' komt.

Doel van deze oefening

Je hebt verschillende boomstructuren leren opzetten. Elke structuur heeft zijn kenmerken, i.h.b. wat zoeken naar grootste/kleinste waarde betreft. In deze oefening wordt je gevraagd na te denken welke structuur het beste past in de gegeven situatie.

We maken in deze oefening een brug naar de opleidingsonderdelen BOP en Web 2, in het bijzonder naar de lijst- en db-klassen die daar in bod komen. We laten je inzien dat de klassen `BinarySearchTree` uit labo 4 en `BinaryMinHeap` uit labo 6 gelijkaardige db-klassen zijn.

Studeren naar gelang het aantal studiepunten

De examens naderen, je hebt nog veel studeerwerk. Je plant om zoveel mogelijk vakken te studeren en studeert dus eerst de vakken met het minst aantal studiepunten.

1. Welke datastructuur gebruik je om de vakken op te slaan zodat je
 - snel het vak met het minst aantal studiepunten terugvindt
 - snel dat vak kan verwijderen
 - snel opnieuw het vak met het minst aantal studiepunten terugvindt
2. Maak een nieuw Java Project ‘VakkenStuderen’ aan. Organiseer de klassen nu zoals je leerde in de lessen BOP en Web2.
 - (a) Maak een package `domain.model` waar je de gegeven klasse `Vak` in plaatst.
 - (b) Maak een package `domain.db` en de klasse `VakkenLijst`. Deze klasse heeft als instantieveranderlijke de lijst met vakken, bewaard in de door jou gekozen datastructuur. Waarschijnlijk heb je gekozen voor een boomstructuur. Kopieer de overeenstemmende methodes uit labo 3, 4 of 6 naar de nieuwe klasse. De klasse die je nu aanmaakt is dus een lijst-klasse die je kent uit BOP of de db-klasse die je kent uit Web 2.
 - (c) Maak een package `ui` aan die de user interface klasse bevat. Zet de gegeven klasse `VakkenMain` in deze package. In dit project is de user interface een main methode die gegevens wegschrijft naar het scherm met het commando `System.out.println()`.
3. Schrijf code in de main methode zodat de gegeven vakken opgeslagen/toegevoegd worden in een instantie van de db-klasse.
4. Schrijf naar het scherm welk vak je eerst gaat studeren (i.e. het vak met het minst aantal studiepunten).

Voorbeeld van uitvoer:

Dit vak moet je eerst studeren: Algo met 3 studiepunten.
5. Als je klaar bent met Algo, voeg je in de main een lijn code toe die dit vak uit de lijst met te studeren vakken verwijdert.

6. Laat de db-klasse berekenen welk vak je vervolgens moet studeren. Schrijf dit naar het scherm zoals in oef.4.
7. Je bent vergeten dat je het vak Computersystemen (3 studiepunten) ook nog moet studeren. Voeg dit vak toe aan de lijst via de main-methode.
8. Schrijf naar het scherm welk vak het meest aantal studiepunten heeft.
9. (*moeilijker*) Schrijf alle resterende vakken uit waarbij de vakken vermeld worden in stijgende orde van aantal studiepunten.
10. Maak een nieuwe db- en main-klasse aan. Kies nu voor een andere datastructuur waarbij je
 - snel een overzicht van vakken kan gegeven, geordend naargelang het aantal studiepunten
 - zowel het vak met het grootste en het vak met het minste aantal studiepunten relatief snel kan vinden.
11. Laat de nieuwe main klasse naar het scherm schrijven welk vak respectievelijk minst en meest aantal studiepunten heeft.
12. Maak oefening 9 opnieuw.