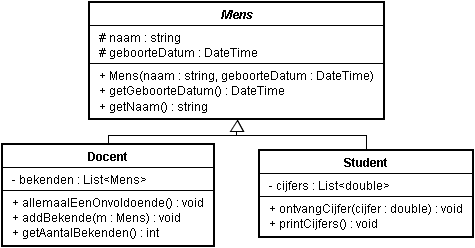
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opleiding**  **Titel Toetseenheid**  **Titel Toets**  **Code**  **Toetsperiode/studiejaar**  **Maximale score** | HBO-ICT  Programmeren Gevorderd  Programmeren Gevorderd Toets  TE.ICT.Vt.P.17.1.PRGV  P3 2017/2018  90 punten | | **Toetsvorm**  **Datum**  **Examinator**  **Nakijker**  **Tijdsduur**  **Cijfer** | Tentamen  -  Jos Foppele  2 uur  (#punten+10)/100 |
| **Toegestane hulpmiddelen** | | Aantekeningen, Presentatie | | |

1. **Docenten zijn net mensen** (15 punten)

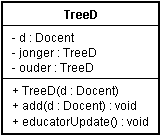
Gegeven is het onderstaande klassendiagram.



Maak de implementatie van de klasse Docent. Voeg hierbij zelf een **constructor** toe. De methode AllemaalEenOnvoldoende() moet alle studenten in de lijst bekenden een onvoldoende (cijfer van 1 t/m 5.4) geven. De methode GetAantalBekenden() is al ingevuld.

1. **Docenten kunnen de boom in** (10+20 punten)

De klasse TreeD wordt gebruikt om docenten uit opgave 1 te ordenen.



1. Implementeer de methode Add(Docent d). Docenten moeten geordend op leeftijd in de boom komen (jongere docenten komen dus in de jonger subboom).

Docenten met dezelfde geboortedatum komen niet voor.

1. Implementeer de methode educatorUpdate(). Bij een EducatorUpdate() gaan alle docenten die jonger zijn dan 35 jaar alle studenten een onvoldoende geven. (Docenten van 35 of ouder doen niets.)

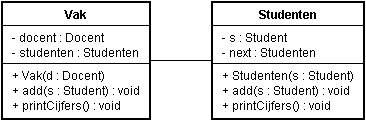
Om de datum van 35 jaar geleden te kunnen krijgen kan de volgende code gebruikt worden: DateTime dt = DateTime.Now.AddYears(-35);

Maak gebruik van de methode AllemaalEenOnvoldoende() uit opgave 1.

1. **Studenten met vakken** (15+15 punten)

Omdat school niet helemaal blij is met het klassendiagram uit opgave 1 is de

onderstaande uitbreiding voorgesteld.



Een vak bestaat uit een docent die het vak geeft en studenten die het vak volgen. Om de

studenten te ordenen is gekozen om een eigen studentenlijst te maken (de klasse

Studenten).

1. Implementeer de methoden Add(Student s) en PrintCijfers() van de klasse Vak. Gebruik hierbij de gegeven code van de rest van de klasse. De methode PrintCijfers() moet PrintCijfers() van de klasse Studenten aanroepen.
2. Implementeer alle methoden van de klasse Studenten. De methode PrintCijfers() moet PrintCijfers() van iedere klasse Student in de lijst aanroepen. Dit moet **recursief**.
3. **Pi berekenen** (15 punten)

Omdat sommige docenten de decimalen van pi niet kunnen onthouden is een methode gewenst die deze decimalen kan uitrekenen.

Hiervoor kan de volgende serie gebruikt worden:

Implementeer nu de methode PiDecimalen(int n) die de decimalen van pi **recursief** berekent met behulp van n termen.