

GOP - Opdracht 1

Een eenvoudige studentenadministratie

In deze opgave maken we een applicatie die gebruik maakt van zelf ontworpen klassen waarbij overerving en dynamische binding centraal staan.

Een studentenadministratie beheert studenten. Een student kan een reguliere student of een deelnemer aan een scholingstraject (scholer) zijn. Van een reguliere student wordt de opleiding vastgelegd waarbij deze is ingeschreven, van een scholer het CPP. We gaan ervan uit dat reguliere studenten en scholers gescheiden groepen zijn, dat wil zeggen een scholer is geen reguliere student en een reguliere student is geen scholer.

Er zijn twee opleidingen, namelijk Informatica en Wiskunde.
Er zijn drie CPP's, namelijk CPP Java, CPP Softwarearchitect en CPP Systeemontwikkelaar.

We maken een applicatie voor het invoeren en tonen van studenten met de volgende mogelijkheden en randvoorwaarden:

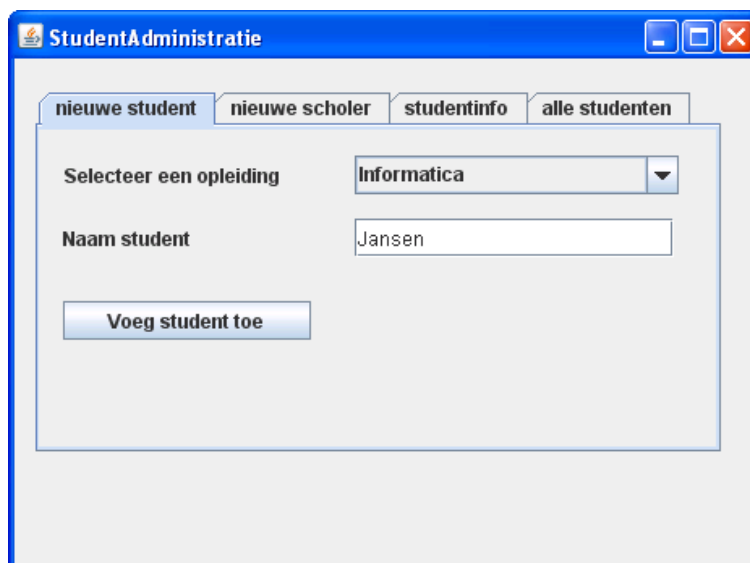
- Van alle studenten wordt de naam ingevoerd en vastgelegd.
- Van reguliere studenten worden de opleiding waarbij ze staan ingeschreven en het aantal behaalde studiepunten vastgelegd. Hoewel de volledige opleiding uit een geheel aantal studiepunten bestaat, kan er per keer ook een gebroken aantal punten behaald worden (bij de OU zijn er vakken van bijvoorbeeld 4.3 studiepunten). Het aantal behaalde studiepunten is dus een gebroken getal.
- Van scholers wordt het CPP en het aantal behaalde modules vastgelegd. Dit laatste is een geheel getal.
- Van iedere opleiding wordt de naam en het totaal aantal studiepunten vastgelegd: Wiskunde 160, Informatica 200. Dit totaal is altijd een geheel getal.
- Van ieder CPP wordt de naam en het aantal modules waaruit het CPP bestaat vastgelegd. (Het CPP Java bestaat uit 6 modules, het CPP Softwarearchitect uit 4 en het CPP Systeemontwikkelaar uit 3)
- Het systeem moet de mogelijkheid hebben de gegevens van alle studenten te tonen: naam, opleiding of CPP, aantal behaalde modules of studiepunten en of de student geslaagd is.

Aanwijzingen

Het gaat in deze opdracht om een goed gebruik van overerving en dynamische binding. De applicatie hoeft niet robuust te zijn; u hoeft bijvoorbeeld niet na te gaan of invoervelden leeg zijn, en of studentnamen uniek zijn, etc.

Als bouwsteen krijgt u een klasse StudentAdminFrame die de grafische interface voor de studentadministratie vormt. De volgende figuren tonen de verschillende delen van de user interface en nog enkele extra aanwijzingen voor de gewenste functionaliteit.

1 Nieuwe student



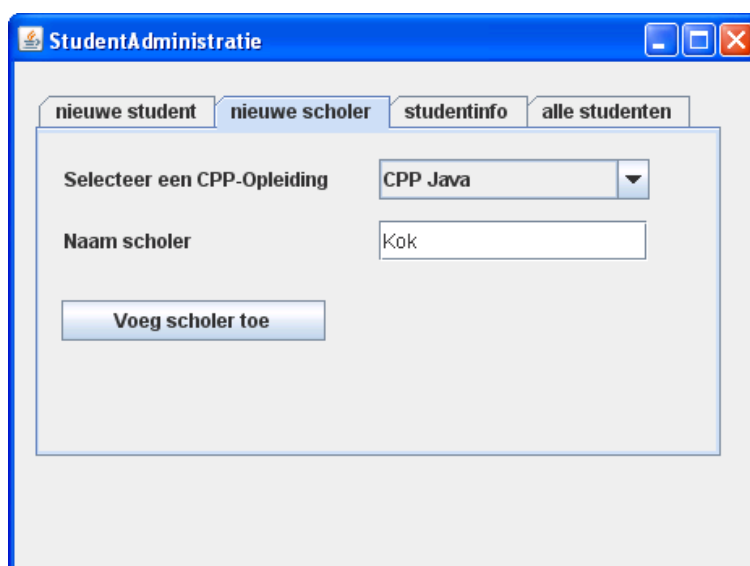
The screenshot shows a Windows-style application window titled 'StudentAdministratie'. It has four tabs: 'nieuwe student', 'nieuwe scholer', 'studentinfo', and 'alle studenten'. The 'nieuwe student' tab is active. Inside this tab, there is a form with two labels: 'Selecteer een opleiding' and 'Naam student'. The 'Selecteer een opleiding' label is next to a dropdown menu currently showing 'Informatica'. The 'Naam student' label is next to a text input field containing 'Jansen'. Below these fields is a button labeled 'Voeg student toe'.

Tabblad nieuwe student dient om een nieuwe reguliere student te kunnen invoeren. Om het niet te ingewikkeld te maken, worden alleen achternamen ingevoerd.

Na klikken op de knop Voeg student toe wordt in dit geval een nieuwe student Jansen met opleiding Informatica toegevoegd aan de studenten. De combobox bevat alle reguliere opleidingen waaruit gekozen kan worden.

In deze niet-robuuste versie hoeft niet gecontroleerd of de student al bestaat en mag dus een eerder toegevoegde student of scholer nog een keer opgenomen worden.

2 Nieuwe scholer

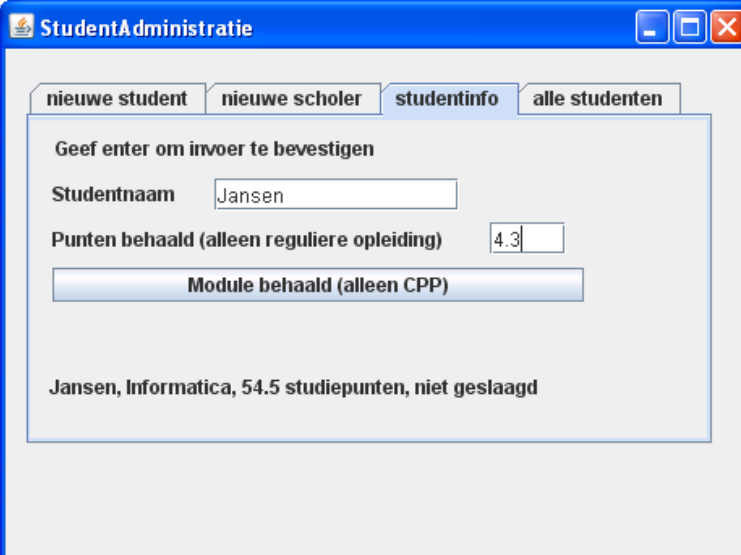


The screenshot shows the same 'StudentAdministratie' application window, but with the 'nieuwe scholer' tab active. The form structure is identical to the previous tab. The 'Selecteer een opleiding' dropdown menu now shows 'CPP Java'. The 'Naam scholer' text input field contains 'Kok'. The button is labeled 'Voeg scholer toe'.

Het tweede tabblad dient om een nieuwe scholer in te voeren.

De functionaliteit is gelijk aan die van nieuwe student maar nu wordt een scholer met zijn bijbehorende CPP-opleiding toegevoegd aan de studenten.

3 Studentinfo



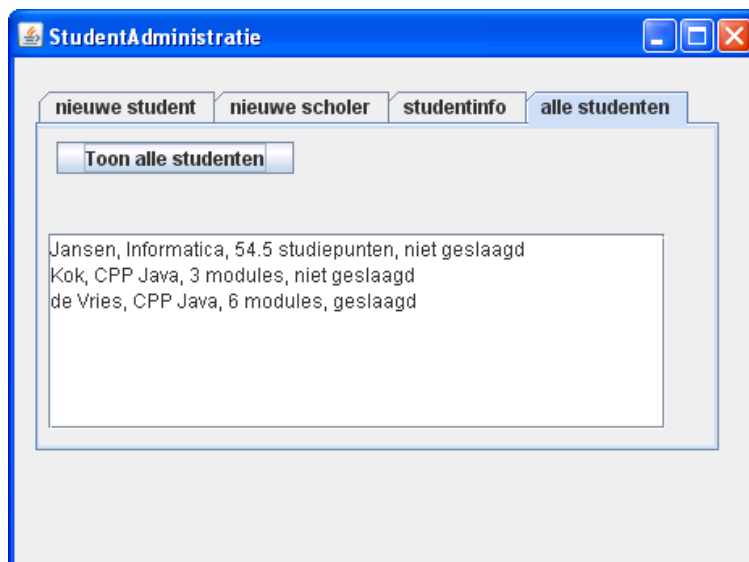
The screenshot shows a window titled "StudentAdministratie" with four tabs: "nieuwe student", "nieuwe scholer", "studentinfo", and "alle studenten". The "studentinfo" tab is selected. Inside the tab, there is a text prompt "Geef enter om invoer te bevestigen". Below this, there are two input fields: "Studentnaam" containing "Jansen" and "Punten behaald (alleen reguliere opleiding)" containing "4.3". Below these fields is a button labeled "Module behaald (alleen CPP)". At the bottom of the tab, the text "Jansen, Informatica, 54.5 studiepunten, niet geslaagd" is displayed.

Op het derde tabblad wordt informatie over een student getoond en kan worden aangegeven dat deze een onderdeel van de studie heeft behaald.

Als de naam van een student ingevoerd is en op enter is gedrukt, verschijnt de informatie van de student onderin het scherm. Deze informatie bestaat uit de naam, de naam van de opleiding of van het CPP, de behaalde studiepunten of het aantal behaalde modules en of een student al dan niet geslaagd is.

Ook kan met dit scherm het aantal studiepunten (door aantal punten op te geven in tekstveld, gevolgd door enter) of het aantal modules (door op knop te drukken wordt het aantal modules met één verhoogd) van de student worden verhoogd. Wanneer dat wordt gedaan, dan moet ook weer de meest actuele informatie van de student worden getoond. Hierbij mag de operatie alleen slagen wanneer deze bij het type student hoort (in deze niet-robuste versie mag een verkeerde operatie eventueel wel tot een exceptie leiden).

4 Alle studenten



Het vierde tabblad toont, bij een druk op de knop, de gegevens van alle studenten. Voor iedere student wordt dezelfde informatie getoond als op het vorige tabblad: naam, de naam van de opleiding of van het CPP, de behaalde studiepunten of het aantal behaalde modules en of een student al dan niet geslaagd is.

Inleveren

U moet het volgende inleveren:

- a Een klassendiagram met alle domeinklassen en hun associaties. In dit diagram hoeft u geen get- en set-methoden op te nemen en dus ook niet de klasse StudentAdminFrame.
- b De applicatie zelf, in de vorm van een gezippt Eclipse-project.

Aanwijzingen

- Denk vooraf na over het ontwerp.
- Denk er aan dat de gui niets anders mag doen dan de gebruikersinvoer inlezen en resultaten tonen; alle functionaliteit moet in domeinklassen zitten.
- Realiseer de functionaliteit bij voorkeur use-case voor use-case, dus scherm voor scherm.
- Test het systeem via de grafische user interface; u hoeft geen JUnit te gebruiken.
- Vergeet de Javadoc niet.
- De applicatie hoeft niet robuust te zijn en dus ook niet te controleren op juiste invoer (e.g. wel wat ingevuld of wel van juiste type). Dat komt in de volgende opdracht.