Probleem beschrijving Project1

# ExamenVragenGedeelte

Elk examen heeft een unieke naam en bestaat uit een aantal vragen, die elke keer dat het examen wordt afgenomen hetzelfde zijn. Er is gebruik gemaakt van minimaal één soort vraag (open, meerkeuze of ja/nee).

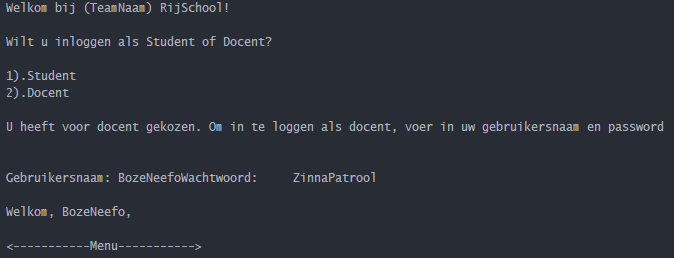
* Bv: PraktijkExamenVragen moet een naam krijgen (PraktijkExamen)
* Elke examen heeft een aantal vragen.
* Minimaal 1 soort vraag, (Meerkeuze,ja/nee etc)
* Vragen mogen hardcodded worden.
* Minimaal 2 examens
* Of het student slaagt of niet hangt af aan de minimum aantal goed aantwoorden. Bijv 60% goed is een voldoende setStudentGeslaagd = true.
* Resultaten van elke poging wordt opgenomen en bewaart met juiste student\_id
* Er moet kunnen bewijzen welke examens er in de programma zijn
* Examens moeten onderscheiden kunnen worden van vragen

# StudentenGedeelte

* Student moet zich kunnen inschrijven. (AddStudent methode aan het begin.)
* Dit betekent dat 4 of 5 studenten hardcoded moeten zijn in het begin van het programma
* Elk student moet een unieke nummer krijgen
* De student kan de toets makken zo vaak hij wil.
* Na het maken van een examen krijgt de student direct te weten of hij geslaagd is of niet en hoe goed of slecht hij de toets heeft gemaakt.
* Er moet kunnen bewijst worden welke studenten allemaal ingeschreven zijn. Lijst met naam en studentnummer
* Is er een student geslaagd voor een bepaald examen? (dus voor TheorieExamen is 20130767 ,20130888 en 20136666 voor de examen geslaagd.)
* Welke examens heeft dit specifieke student gehaald?
* Welke student heeft van alle studenten de meeste examens gehaald? Als het meerdere studenten zijn worden ze allemaal getoond.

We gaan het een beetje veranderen

Welcome Screen ziet er zo uit:



private static void pressAnyKeyToContinue()  
{

try {System.*in*.read();  
 }  
 catch (Exception e){  
 }  
}

# Feedback van M.Barbouch :

* De Classen PraktijkexamenToets en TheorieExamenToets moet onderscheiden van toetsvragen.
* Er moet een duidelijke planning komen Wie doet wat (Doen we op papier en in github).
* UML-diagram moet eerst gemaakt worden. Nu zijn we aan het coderen zonder een duidelijk beeld.
* Meer OOP gericht werken.
* Faissal toevoegen aan onze Repository op Github.
* Iedereen ongeveer hetzelfde hoeveelheid commits en bijdrage(~”Ongeveer”~).
* 1 branche per persoon zodat we beter inzicht hebben over commits en input.

# Wat hebben we fout?

## Wat we nu hebben:

Toetsvraag (Super) ---- : 🡪 TheorieExamen (Subclass)  
 : 🡪 PraktijkExamen (Subclass)

## Hoe het moet zijn:

Toetsvraag (Super) ---- : 🡪 MeerkeuzeVraag\_(Subclass)  
 : 🡪Ja-Nee-Vragen\_(Subclass)  
 : 🡪Open-Vragen\_(subclass)

Meerkeuzevragen ----: 🡪 TheorieExamen ArrayList<MeerTkeuzeVraag>

# Wat hebben we goed?

Student Class is goed onderscheiden van de rest.  
Menu Class is ook goed onderscheiden van de rest.

+ voerJeNaamIn()

+ beginScherm()

+ examenLijst()

+studentenLijst()

+studentInschrijven()

+studentVerwijderen()

+examenAfnemen()