

Verantwoording van de te bezoeken les

Toelichting m.b.t. constructeur leeromgeving:

→ Zie het losse lesvoorbereidingsformulier.

Toelichting m.b.t. de rol van vakinhoudelijk begeleider:

- *Waar in de les motiveert en enthousiasmeert de kandidaat de studenten?*
- *Waar in de les geeft de kandidaat ruimte voor interactie?*
- *Hoe creëert de kandidaat een prettig en veilig leerklimaat in de les?*
- *Hoe zet je studenten aan tot leren?*
- *Hoe begeleid/coach je studenten in de les in hun leerproces?*

Persoonlijk denk ik dat ik met mijn enthousiaste manier van vertellen en door gebruik te maken van voorbeelden uit de praktijk de studenten voor mijn verhaal kan enthousiasmeren. Er is in mijn lessen in principe altijd ruimte om vragen te stellen, maar ik geef ook vaak de ruimte hiertoe. Ik sta studenten met vragen altijd te woord, blijf geduldig en positief. Hierdoor ontstaat volgens mij een prettig leerklimaat. De studenten worden door afwisseling in leervormen geprikkeld om aan de slag te gaan met de stof. Verder benadruk ik vooral de eigen verantwoordelijkheid die ze hebben. Door actief te vragen naar hun voortgang en hulp te bieden als ze vast lopen begeleid ik de studenten.

Toelichting m.b.t. de rol van examiner

(ook in de les ben je bezig met formatief en/of summatief toetsen)

- *Hoe dragen de (activerende) werkvormen bij aan het behalen van de doelen van de module (bijvoorbeeld de toets)?*
- *Worden de criteria (of een gedeelte daarvan) van de toets besproken in deze les?*
- *Worden studenten betrokken bij het beoordelen van hun eigen ontwikkeling en het geven van feedback in deze les?*
- *Waar in de les worden vragen gesteld die de ontwikkeling van de student toetsen?*

De studenten moeten programmeren. Tijdens de OO-toets, een belangrijke summatieve toets in dit vak, zullen ze nagenoeg dezelfde soort code moeten opzetten. De toetscriteria komen niet expliciet aan bod in deze les, maar worden wellicht terloops genoemd. In de oefeningen met unit-testen zullen studenten zelf moeten ondervinden of ze de uitleg begrepen hebben en of ze de opdracht kunnen uitvoeren.

Toelichting m.b.t. bewust docentschap:

- *Persoonlijke leerdoelen met betrekking tot het lesgeven.*

Omdat ik het nog steeds lastig vind om activerende werkvormen te kiezen en in te zetten heb ik in deze les geprobeerd, naast de door de vakeigenaar al bedachte werkvormen, er een aantal toe te voegen: de groepsgewijze definitiebepaling en de unit-test-oefening (FizzBuzz). Zelf ben ik erg benieuwd naar het effect van deze oefeningen.

Ook ben ik erg benieuwd of het lukt de studenten betrokken te houden bij de nieuwe stof, terwijl meer dan de helft met zijn hoofd waarschijnlijk bij de herkansing van de toets het 9^e uur gaat zijn. Misschien dat er ook weerstand ontstaat als het te lang duurt voordat we het daarover gaan hebben. Afhankelijk van de situatie zal ik dan proberen flexibel om te gaan met mijn les.

Lesvoorbereiding



Naam student: Ruben Steins	Vak: SE2 (Software Engineering 2)
Naam SP-docent: Marie-Louise van Lieshout	Praktijkles / theorieles: Mix
School: Fontys Hogescholen ICT	Datum lesuitvoering: 16-03-2016
Klas: S25	Duur van de les: 3x45 min

Lesopdracht

Welk onderwerp komt er aan de orde

Unit testen en inheritance.

Unit testen is het testen van bepaalde stukken code (vaak individuele methodes) met test code. Anders gezegd, je programmeert een test-methode die een methode van je applicatie uitvoert en controleert of de uitvoer en de verwachte gevolgen plaatsgevonden hebben

Bij het object georiënteerd programmeren wordt het concept van inheritance (overerving) gebruikt om gedeelde functionaliteit over klassen te bundelen. Omdat hierdoor code minder gedupliceerd wordt, maakt het correct gebruik van deze techniek je applicaties eenvoudiger en herbruikbaar.

Als derde onderwerp blikken we terug op de Use-Case toets van vorige week en geef ik de studenten die moeten herkansen later op de middag nog wat laatste tips mee.

Doelstellingen

Concreet (in waarneembaar leerling-gedrag, b.v. opschrijven, noemen uitrekenen, tekenen, etc.)

1. De student weet wat het belang en de functie van unit testen is.
2. De student kan een project aanmaken in Visual Studio waarmee unit tests uitgevoerd kunnen worden.
3. De student kan unit tests opzetten en uitvoeren, en zinnige invoer bepalen voor de te testen methodes.
4. De student kan is bekend met de term code coverage en weet hoe dit kan worden gebruiken om te bepalen welke delen van de applicatie (on)voldoende getest zijn.

Beginsituatie

Klas, niveau, aantal leerlingen, apparatuur, etc.
Voorkennis/vaardigheden/houdingen

Klas S25, 2^e semester, 22 studenten. Gebruik maken van eigen laptops met Visual Studio Enterprise of Ultimate. Deze software is al aanwezig voor aanvang de les.

Voorkennis: basis programmeerervaring, mede uit de eerder lessen in dit vak. Weinig tot geen kennis van (gestructureerd) unit-testen. Een aantal studenten zal een beeld bij inheritance hebben, een aantal niet. Het is een vrij abstract concept; zal enige tijd duren voor het landt.

Houding: over het algemeen redelijk positief. Meerwaarde moet snel worden aangetoond, anders verslapt de aandacht. Vanwege noodzakelijk gebruik van de laptop is het erg verleidelijk om te gaan gamen. Wellicht is er enige onrust door de herkansing van de use-case toets na het 3^e lesuur.

Lesfasering (tijd: ± 65 min)	Leerinhouden (lesstof in hoofdzaken)	Leeractiviteiten (zie doelformulering en wat <u>doet</u> de leerling tijdens de les)
Oriënteren: (± 5 min) - onderwerp - lijn van de les - doelen - nut	<ul style="list-style-type: none">• Terugblik op vorige week: klassendiagram• Uitleg over lesopzet:<ul style="list-style-type: none">○ Assessor aanwezig○ Onderwerpen○ Herkansing toets	
Verkennen: (± 10 min) Beginsituatie peilen b.v. door middel van vragen stellen	<ul style="list-style-type: none">• Wat zijn unit-tests?	<ul style="list-style-type: none">• Per tafel proberen een definitie van unit-testen vast stellen.
Verwerven: (± 15 min) Aanbrengen van kennis en vaardigheden d.m.v. bepaalde didactische werkvormen, b.v. uitleggen, demonstreren, onderwijsleergesprek	<ul style="list-style-type: none">• Wat is het belang van unit-tests• Opzetten project Visual Studio• Opzetten eerste test• Berekenen Code Coverage	<ul style="list-style-type: none">• Hoorcollege ca. 10 minuten• Demo ca. 10 minuten
Verwerken: (± 2 x 30 min) Leerlingen gaan oefenen met de verworven kennis en vaardigheden	<ul style="list-style-type: none">• Je weet wat het belang en de functie van unit testen is.• Je kunt een project aanmaken in Visual Studio waarmee unit tests uitgevoerd kunnen worden.• Je kunt unit tests opzetten en uitvoeren, en zinnige invoer bepalen voor de te testen methodes.	<ul style="list-style-type: none">• Oefening opzetten Unit-Test project en schrijven eerste test → FizzBuzz oefening.• Per proftaakgroep: schrijf een (aantal) unit-tests voor de FizzBuzz code (geprojecteerd). Vertel dan welke test je hebt gemaakt, welk scenario wordt getest.• Als laatste gaan we code coverage van de tests van de hele klas doen.• Daarna: eventueel MSDN walkthrough• Ontwikkelopdracht: Unit-testen

<p>Evaluëren: (± 10 + 5 min) Controle of doelen bereikt zijn, eventueel samenvatten van de les en peilen wat de leerlingen er van vonden</p>	<ul style="list-style-type: none"> • Je bent bekend met de term code coverage en hoe je dit kunt gebruiken om te bepalen welke delen van je applicatie (on)voldoende getest zijn 	<ul style="list-style-type: none"> • Na eerste 30 minuten: klassikaal code coverage bepalen. • Samenvatting unit-tests. • Zijn er nog vragen? <p>Daarna: zelf werken</p>
--	---	---

<p>Leerkrachtactiviteiten (didactische werkvormen)</p>	<p>Leer- en hulpmiddelen</p>
<ul style="list-style-type: none"> • Hoorcollege + demo 	<ul style="list-style-type: none"> • Powerpoint presentatie • Laptop met ontwikkelomgeving
<ul style="list-style-type: none"> • Werkcollege (FizzBuzz oefening) 	<ul style="list-style-type: none"> • Code voorbeeld op eigen laptop • Online bronnen • Oefening in ELO • Eventueel CodeHunt?

Terugblik op de les

Wat wilde ik zelf leren, oefenen?
Wat ging goed, wat minder goed?
Hoe zou ik het de volgende keer aanpakken?
