

Eindtoets Object Georiënteerd programmeren (10 EC)

Deze leerlijn bestaat uit de volgende cursussen:

- Associatieve relaties in Java (2 EC)
- Encapsulation en static in Java (2 EC)
- Overerving en Polymorfie (2 EC)
- Abstracte Constructies in Java (2 EC)
- Java Gui en Exceptions (2 EC)

Doelstelling eindtoets:

De student kan:

- classes maken in Java en deze gebruiken in een associatieve relatie;
- het principe van encapsulation uitleggen en dit toepassen in een Java applicatie;
- het principe van static uitleggen en dit toepassen in een Java applicatie;
- gebruik maken van ArrayLists in Java-applicaties, waaronder het gebruik van ArrayLists met als datatype eigen gedefinieerde classes;
- overerving toepassen in een Java-applicatie rekening houdende met de juiste visibility modifiers;
- polymorfie middels overloading en overriding gebruiken in een Java-applicatie;
- abstracte classes vormgeven, inclusief het gebruik van abstracte methods en deze toepassen in een Java-applicatie;
- interfaces ontwerpen en gebruiken in een Java-applicatie, inclusief het gebruiken van een interface als datatype voor een verzameling;
- een JavaFX fxml applicatie maken volgens het MVC-patroon;
- exceptionhandling toepassen in een Java-applicatie;
- unittesting gebruiken in een Java-applicatie.

Opdracht eindtoets:

Je ontwikkelt een applicatie met een onderwerp naar eigen keuze. De applicatie wordt ontwikkeld in een Java-programma. Bij het ontwikkelen van de applicatie toont de student aan in voldoende mate bovenstaande kennis en vaardigheden te beheersen.

Op te leveren producten:

- **Plan van aanpak**

Je stelt een plan van aanpak op waarin je beschrijft wat het onderwerp is van de applicatie die je gaat bouwen. Je beschrijft tevens welke technieken je gaat toepassen en hoe je de beoordelingscriteria gaat verwerken in de applicatie. Indien gewenst bespreek je je plan van aanpak met je (e-)docent.

- **Applicatie**

Nadat je plan van aanpak gereed is start je met het bouwen van de applicatie. Naast het ontwikkelen van de applicatie schrijf je een verantwoording waarin je een overzicht maakt van alle toegepaste technieken. Je vermeldt bij iedere techniek op welk regelnummer van de code dit gevonden kan worden. Voorzie tevens ieder Java file bovenaan van Javadoc-commentaar, waarin je je naam, alsmede de naam van de leerlijn en de datum vermeldt.

- **Uploaden**

Het plan van aanpak, het volledig afgeronde Java-project en het document met de verantwoording upload je in een zipbestand (max. 50MB) via de EdHub.

Beoordelingscriteria

Voor ieder criterium kun je een cijfer krijgen tussen de 1 en 10, afhankelijk van het niveau van de uitwerking. Een 8 wordt gezien als zeer goed. Een cijfer boven de 8 wordt alleen gegeven wanneer er boven het gevraagde niveau opgeleverd wordt.

	Object Georiënteerd Programmeren	Weging	Score
1	De applicatie compileert en runt in eerste aanleg zonder foutmelding. Wanneer hieraan niet voldaan wordt, kan er geen voldoende gegeven worden voor dit onderdeel.	15%	
2	Er wordt in de applicatie correct gebruik gemaakt van tenminste 6 classes. Van deze classes hebben tenminste 3 classes een associatieve relatie met elkaar. Wanneer er gebruik wordt gemaakt van meer classes, kan er een hoger cijfer dan 8 gegeven worden. Wanneer er minder dan 6 classes worden gebruikt of het gebruik van associatieve relaties is foutief, dan kan er geen voldoende gegeven worden voor dit onderdeel.	15%	
3	Encapsulation wordt correct toegepast. Wanneer de encapsulation voor 100% correct is, kan een hoger cijfer dan 8 gegeven worden. Wanneer er tenminste 1 public attribuut is, kan er geen voldoende gegeven worden voor dit onderdeel.	10%	
4	Tenminste 1 arraylist wordt correct gebruikt. Wanneer een eigen class als datatype voor een arraylist wordt gebruikt kan een cijfer hoger dan 8 gegeven worden. Wanneer de arraylist ontbreekt, of foutief gebruikt wordt, kan er geen voldoende gegeven worden voor dit onderdeel.	10%	
5	Overerving wordt minimaal 1 keer toegepast, hierbij wordt correct overriding toegepast. Voor het gebruik maken van een abstracte classe of een interface kan een cijfer hoger dan 8 gegeven worden. Wanneer overerving ontbreekt, of verkeerd wordt toegepast, kan er geen voldoende gegeven worden voor dit onderdeel.	15%	
6	Er wordt correct gebruik gemaakt van een MVC-patroon middels een javafx fxml applicatie. Wanneer het MVC-patroon volledig correct wordt gebruikt, kan er een cijfer hoger dan 8 gegeven worden . Wanneer Model View Controller correct wordt doorgevoerd, bevat een controller object, enkel instanties van domein classes en bewerkingen daarmee. Zoniet, dan kan er geen voldoende gegeven worden voor dit onderdeel.	15%	
7	Er wordt minstens 1 keer correct gebruik gemaakt van exception handling. Wanneer er consequent gebruik wordt gemaakt van exception handling, kan een cijfer hoger dan 8 gegeven worden. Wanneer er geen of foutief gebruik gemaakt wordt van exception handling, kan er geen voldoende gegeven worden voor dit onderdeel.	10%	
8	Tenminste twee classes worden getest middels Unittesting. Wanneer alle domein classes met unittesting getest worden, kan een cijfer hoger dan 8 gegeven worden. Wanneer unittesting ontbreekt, of niet werkt, kan er geen voldoende gegeven worden voor dit onderdeel.	10%	
	Totaal	100%	

Verandering

Verduidelijking: Wanneer er domeincode in de controller wordt opgenomen, kan er geen voldoende gegeven worden voor dit onderdeel. -> Wanneer Model View Controller correct wordt doorgevoerd, bevat een controller object, enkel instanties van domein classes en bewerkingen daarmee. Zoniet, dan kan er geen voldoende gegeven worden voor dit onderdeel.