

Softwareontwikkeling (E761035)

Cursusomvang *(nominale waarden; effectieve waarden kunnen verschillen per opleiding)*

Studiepunten 6.0

Studietijd 180 u

Aanbodsessies en werkvormen in academiejaar 2024-2025

A (semester 2)

Nederlands

Gent

hoorcollege

werkcollege

Lesgevers in academiejaar 2024-2025

Ongenaë, Veerle

TW05

Verantwoordelijk lesgever

Aangeboden in onderstaande opleidingen in 2024-2025

stptn

aanbodsessie

[Bachelor of Science in de industriële wetenschappen \(afstudeerrichting informatica\)](#)

6

A

[Schakelprogramma tot Master of Science in de industriële wetenschappen: informatica](#)

6

A

[Vorbereidingsprogramma tot Master of Science in de industriële wetenschappen: informatica](#)

6

A

Onderwijstalen

Nederlands

Trefwoorden

UML, Systems analysis, Modelling, C#, Design Patterns, Dependency Injection, Computerwetenschappen (P170), Informatica (P175), Computertechnologie (T120)

Situering

Dit opleidingsonderdeel behandelt een aantal items van gevorderd object-georiënteerd ontwerp.

In het eerste gedeelte van deze cursus wordt het proces behandeld dat doorlopen wordt om software te ontwikkelen. Het doel is de student in staat te stellen om autonoom kleine projecten tot een goed einde te brengen. Hem/haar wordt aangeleerd welke de methodes zijn om programmatuur van goede kwaliteit te maken.

Daarnaast beoogt het om de studenten inzicht te laten verwerven in de beschikbare "design patterns" voor softwareontwerp en voor typische softwareproblemen.

Inhoud

Deel 1: Systeemanalyse en -ontwerp

- Basisprincipes van een goede programmeerpraktijk: kenmerken van goede software en een goed ontwikkelingsproces.
- Hergebruik: hoe bestaande software te hergebruiken, en code te schrijven die hergebruikt kan worden.
- De verschillende fasen van het ontwikkelingsproces.
- Basisprincipes van UML.
- Behoeftanalyse en modellering: methodes om een te ontwikkelen systeem te definiëren.
- Ontwerp en realisatie: het omzetten van een formeel model in code.

Deel 2: Design patterns

Dit deel geeft een overzicht van de meest gebruikte "design patterns" en objectgeoriënteerde ontwerpprincipes:

Strategy, Observer, Decorator, Factory Method, Abstract Factory, Singleton, Command, Adapter, Facade, Template Method, Iterator, Composite, State,

Proxy, MVC, Bridge, Builder, Chain of Responsibility, Flyweight, Interpreter, Mediator, Memento, Prototype, Visitor, inversion of control, dependency injection,

...

Daarnaast worden een aantal gevorderde programmeerconcepten geïntroduceerd: GUI-programmatie, delegates, extensiemethodes, asynchrone methodes, ...

Begincompetenties

Op een doorgedreven niveau kunnen objectgeoriënteerd programmeren

Eindcompetenties

- 1 Principes van softwareontwerp toepassen met het oog op productie, onderhoud en kwaliteit.
- 2 Een relatief complex probleem analyseren, structureren en vertalen naar een objectgeoriënteerd ontwerp.
- 3 Een objectgeoriënteerd ontwerp omzetten naar een werkend computerprogramma in c# en dit programma kritisch testen
- 4 De beschikbare designs patterns voor software-ontwerp toelichten en illustreren met voorbeelden
- 5 Inschatten in welke situatie welk pattern geschikt is
- 6 Programma's ontwikkelen die zinvol gebruik maken van patterns
- 7 Programma's herwerken op basis van patterns

Creditcontractvoorwaarde

Toelating tot dit opleidingsonderdeel via creditcontract is mogelijk na gunstige beoordeling van de competenties

Examencontractvoorwaarde

Dit opleidingsonderdeel kan niet via examencontract gevolgd worden

Didactische werkvormen

Werkcollege, Hoorcollege

Toelichtingen bij de didactische werkvormen

- Hoorcollege (24u)
- Werkcollege (36u): PC-klasoefeningen (zelfstandig werk aan een individuele PC)

Studiemateriaal

Type: Slides

Naam: Kennisclips, slides, voorbeeldprogramma's en oefeningen met oplossingen zijn beschikbaar via het elektronisch leerplatform.

Richtprijs: € 7

Optioneel: ja

Taal : Nederlands

Beschikbaar op Ufora : Ja

Online beschikbaar : Nee

Beschikbaar in de bibliotheek : Nee

Beschikbaar via studentenvereniging : Ja

Type: Software

Naam: Visual Studio .NET

Richtprijs: Gratis of betaald door opleiding

Optioneel: nee

Beschikbaar op Athena : Nee

Online beschikbaar : Ja

Beschikbaar in de bibliotheek : Nee

Beschikbaar via studentenvereniging : Nee

Gebruik en levensduur binnen het opleidingsonderdeel : intensief

Gebruik en levensduur binnen de opleiding : regelmatig

Gebruik en levensduur na de opleiding : niet

Referenties

"Train je hersens in Design Patterns", Eric Freeman & Elisabeth Freeman, Lannoo campus, O'Reilly

"Design Patterns: Elements of Reusable Object-Oriented Software", Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley

"Praktisch UML", 5e editie, Jos Warmer en Anneke Kleppe, ISBN 9789043020558

Vakinhoudelijke studiebegeleiding

Lesgevers zijn ter beschikking voor extra uitleg tijdens de labo's, voor of na de theorielessen en eventueel op andere ogenblikken na afspraak

Evaluatiemomenten

periodegebonden en niet-periodegebonden evaluatie

Evaluatievormen bij periodegebonden evaluatie in de eerste examenperiode

Vaardigheidstest, Schriftelijke evaluatie

Evaluatievormen bij periodegebonden evaluatie in de tweede examenperiode

Vaardigheidstest, Schriftelijke evaluatie

Evaluatievormen bij niet-periodegebonden evaluatie

Vaardigheidstest

Tweede examenkans in geval van niet-periodegebonden evaluatie

Examen in de tweede examenperiode is niet mogelijk

Toelichtingen bij de evaluatievormen

Tijdens het werkcollege worden meerdere testen aan de computer georganiseerd en moeten een aantal opdrachten ingediend worden.

Eindscoreberekening

Examen: 75% (schriftelijk examen (60%) en oefeningen op de computer(40%))

Oefeningen/Labo's: 25% (testen en opdrachten)

In de tweede examenperiode: score = maximum(E; 25% L + 75% E), waarbij L de score van het labo (NPE) is en E de nieuwe score van het examen uit tweede zit