

Informaticawetenschappen''

3de graad D-finaliteit

III-Inf''-d

BRUSSEL

D/2024/13.758/203

Versie januari 2024

1 Inleiding

De uitrol van de modernisering secundair onderwijs gaat gepaard met een nieuwe generatie leerplannen. Leerplannen geven richting en laten ruimte. Ze faciliteren de inhoudelijke dynamiek en de continuïteit in een school en lerarenteam. Ze garanderen binnen het kader dat door de Vlaamse regering werd vastgelegd voldoende vrijheid voor schoolbesturen om het eigen pedagogisch project vorm te geven vanuit de eigen schoolcontext. Leerplannen zijn ingebed in het vormingsconcept van de katholieke dialoogschool. Ze versterken het eigenaarschap van scholen die d.m.v. eigen beleidskeuzes de vorming van leerlingen gestalte geven. Leerplannen laten ruimte voor het vakinhoudelijk en pedagogisch-didactisch meesterschap van de leraar, maar bieden ondersteuning waar nodig.

1.1 Het leerplanconcept: vijf uitgangspunten

Leerplannen vertrekken vanuit het **vormingsconcept** van de katholieke dialoogschool. Ze laten toe om optimaal aan te sluiten bij het pedagogisch project van de school en de beleidsbeslissingen die de school neemt vanuit haar eigen visie op onderwijs (taalbeleid, evaluatiebeleid, zorgbeleid, ICT-beleid, kwaliteitsontwikkeling, keuze voor vakken en lesuren ...).

Leerplannen ondersteunen **kwaliteitsontwikkeling**: het leerplanconcept spoort met kwaliteitsverwachtingen van het Referentiekader onderwijskwaliteit (ROK). Kwaliteitsontwikkeling volgt dan als vanzelfsprekend uit keuzes die de school maakt bij de implementatie van leerplannen.

Leerplannen faciliteren een **gerichte studiekeuze**. De leerplandoelen sluiten aan bij de verwachte competenties van leerlingen in een bepaald structuuronderdeel. De feedback en evaluatie bij de realisatie ervan beïnvloeden op een positieve manier de keuze van leerlingen na elke graad.

Leerplannen gaan uit van de **professionaliteit** van de leraar en het **eigenaarschap** van de school en het lerarenteam. Ze bieden voldoende ruimte voor eigen inhoudelijke keuzes en een eigen didactische aanpak van de leraar, het lerarenteam en de school.

Leerplannen borgen de **samenhang** in de vorming. Die samenhang betreft de verticale samenhang (de plaats van het leerplan in de opbouw van het curriculum) en de horizontale samenhang tussen vakken binnen structuuronderdelen en over structuuronderdelen heen. Leerplannen geven expliciet aan voor welke leerplandoelen van andere leerplannen in de school verdere afstemming mogelijk is. Op die manier faciliteren en stimuleren de leerplannen leraren om over de vakken heen samen te werken en van elkaar te leren. Een verwijzing van een leraar naar de lessen van een collega laat leerlingen niet alleen aanvoelen dat de verschillende vakken onderling samenhangen en dat ze over dezelfde werkelijkheid gaan, maar versterkt ook de mogelijkheden tot transfer.

1.2 De vormingscirkel – de opdracht van secundair onderwijs

De leerplannen vertrekken vanuit een gedeelde inspiratie die door middel van een vormingscirkel voorgesteld wordt. We ‘lezen’ de cirkel van buiten naar binnen.

- Een lerarenteam werkt in een katholieke dialoogschool die onderwijs verstrekt vanuit een **specifieke traditie**. Vanuit het eigen pedagogisch project kiezen leraren voor wat voor hen en hun school goed



onderwijs is. Ze wijzen leerlingen daarbij de weg en gebruiken daarvoor **wegwijzers**. Die zijn een inspiratiebron voor leraren en zorgen voor een Bijbelse 'drive' in hun onderwijs.

- De kwetsbaarheid van leerlingen ernstig nemen betekent dat elke leerling **beloftevol** is en alle leerkansen verdient. Die leerling is **uniek als persoon** maar ook **verbonden** met de klas, de school en de bredere samenleving. Scholen zijn **gastvrije plaatsen** waar leerlingen en leraren elkaar ontmoeten in diverse contexten. De leraar vormt zijn leerlingen vanuit een **genereuze** attitude, hij geeft om zijn leerlingen en hij houdt van zijn vak. Hij durft af en toe de gebaande paden verlaten en stimuleert de **verbeelding en creativiteit** van leerlingen. Zo zaait hij door zijn onderwijs de kiemen van een hoopvolle, **meer duurzame en meer rechtvaardige wereld**.
- Leraren vormen leerlingen door middel van leerinhouden die we groeperen in negen **vormingscomponenten**. De aaneengesloten cirkel van vormingscomponenten wijst erop dat vorming een geheel is en zich niet in schijfjes laat verdelen. Je kan onmogelijk over taal spreken zonder over cultuur bezig te zijn; wetenschap en techniek hebben een band met economie, wiskunde, geschiedenis ... Dwarsverbanden doorheen de vakken zijn belangrijk. De vormingscirkel vormt dan ook een dynamisch geheel van elkaar voortdurend beïnvloedende en versterkende componenten.
- Vorming is voor een leraar nooit te herleiden tot een cognitieve overdracht van inhouden. Zijn meesterschap en passie brengt een leraar ertoe om voor iedere leerling de juiste woorden en gebaren te zoeken om **de wereld te ontsluiten**. Hij introduceert leerlingen in de wereld waarvan hij houdt. Een leraar zorgt er bijvoorbeeld voor dat leerlingen kunnen worden gegrepen door de cultuur van het Frans of door het ambacht van een metselaar. Hij initieert leerlingen in een wereld en probeert hen zover te brengen dat ze er hun eigen weg in kunnen vinden.
- Een leraar vormt leerlingen als **individuele leraar**, maar werkt ook binnen **lerarenteams** en binnen een **beleid van de school**. Het Gemeenschappelijk funderend leerplan helpt daartoe. Het zorgt voor het fundament van heel de vorming dat gerealiseerd wordt in vakken, in projecten, in schoolbrede initiatieven of in een specifieke schoolcultuur.
- De uiteindelijke bedoeling is om **alle leerlingen** kwaliteitsvol te vormen. Leerlingen zijn dan ook het hart van de vormingscirkel, zij zijn het op wie we inzetten. Zij dragen onze hoop mee: de nieuwe generatie die een meer duurzame en meer rechtvaardige wereld zal creëren.



1.3 Ruimte voor leraren(teams) en scholen

De leraar als professional, als meester in zijn vak krijgt vrijheid om samen met zijn collega's vanuit de leerplannen aan de slag te gaan. Hij kan eigen accenten leggen en differentiëren vanuit zijn passie, expertise, het pedagogisch project van de school en de beginsituatie van zijn leerlingen.

De leerplandoelen zijn noch chronologisch, noch hiërarchisch geordend. Ze laten ruimte aan het lerarenteam en de individuele leraar om te bepalen welke leerplandoelen op welk moment worden samengenomen, om didactische werkvormen te kiezen, contexten te bepalen, eigen leerlijnen op te bouwen, vakoverschrijdend te werken, flexibel om te gaan met een indicatie van onderwijstijd.

1.4 Differentiatie

Om optimale leerkansen te bieden is [differentiëren](#) van belang in alle leerlingengroepen. Leerlingen voor wie dit leerplan is bestemd, behoren immers wel tot dezelfde doelgroep, maar bevinden zich niet noodzakelijk in dezelfde beginsituatie. Zij hebben een niet te onderschatten – maar soms sterk verschillende – bagage mee vanuit de onderliggende graad, de thuissituatie en vormen van informeel leren. Het is belangrijk om zicht te krijgen op die aanwezige kennis en vaardigheden en vanuit dat gegeven, soms gedifferentieerd, verder te bouwen. Positief en planmatig omgaan met verschillen tussen leerlingen verhoogt de motivatie, het welbevinden en de leerwinst voor elke leerling.

De leerplannen bieden kansen om te differentiëren door te verdiepen en te verbreden en door de leeromgeving aan te passen. Ze nodigen ook uit om te differentiëren in evaluatie.

Differentiatie door te verdiepen en te verbreden

Sommige leerlingen denken meer conceptueel en abstract. Andere leerlingen komen vanuit een meer concrete benadering sneller tot inzichtelijk denken. Variëren in abstractie spreekt leerlingen aan op hun capaciteiten en daagt hen uit om van daaruit te groeien.

Daarnaast bieden leerplannen kansen om de complexiteit van leerinhouden aan te passen. Dat kan door een complexere situatie te schetsen, een minder ingewikkelde bewerking of handeling voor te stellen, of door meer kennis of vaardigheden aan te bieden om leerlingen uit te dagen.

De ene context kan betekenisvol zijn voor een leerlingengroep, terwijl een andere context dan weer betekenisvoller kan zijn voor een andere leerlingengroep. Leerinhouden in verschillende contexten aanbrengen biedt kansen om leerlingen aan te spreken op hun interesses en daagt hen tegelijk uit om andere interesses te verkennen en zo hun horizon te verruimen.

In 'extra' wenken bij de leerplandoelen en in beperkte mate ook via keuzeleerplandoelen bieden we je inspiratie om te differentiëren door te verdiepen en te verbreden.

Differentiatie door de leeromgeving aan te passen

Doordachte variatie in werkvormen (groepswork, individueel, auditief, visueel, actief ...) vergroot de kans dat leerdoelen worden gerealiseerd door alle leerlingen. Het helpt hen bovendien ontdekken welke manieren van leren en informatie verwerken best bij hen passen.

De ene leerling kan snel of zelfstandig werken, de andere heeft meer tijd of begeleiding nodig. Variëren in de mate van ondersteuning, gericht aanbieden van hulpmiddelen (voorbeelden, schrijfkaders, stappenplannen ...) en meer of minder tijd geven, daagt leerlingen uit op hun niveau en tempo.

Leerlingen op hun niveau en vanuit eigen interesses laten werken kan door te differentiëren in product, bijvoorbeeld door leerlingen te laten kiezen tussen opdrachten die leiden tot verschillende eindproducten.

Het samenstellen van groepen kan een effectieve manier zijn om te differentiëren. Rekening houden met verschil in leerdoelen en leerlingenkenmerken laat leerlingen toe van en met elkaar te leren.

Technologie kan al die vormen van differentiatie ondersteunen. Zo kunnen leerlingen op hun maat werken met digitale leermiddelen zoals educatieve software of online oefenprogramma's.

Differentiatie in evaluatie

Tenslotte laten de leerplannen toe te differentiëren in [evaluatie](#) en feedback. Evalueren is beoordelen om te waarderen, krachtiger te maken en te sturen.

Na de afronding van een lessenreeks of na een langere periode gaan leraren door middel van summatieve evaluatie na waar leerlingen staan. De keuze van een evaluatie- en feedbackvorm is afhankelijk van de vooropgestelde doelen.



Formatieve evaluatie is geïntegreerd in het leerproces en gaat uit van een actieve betrokkenheid van leraar en leerling. Het zet leerlingen aan het denken over hun vorderingen en laat leraren toe om tijdens het leerproces effectieve feedback te geven. Door middel van formatieve evaluatie krijgen leraren een goed zicht op het leerproces van leerlingen zodat ze het verder gericht en waar nodig kunnen bijsturen. Het is bovendien een rijke bron voor leraren om te reflecteren over de eigen onderwijspraktijk en de eigen pedagogisch-didactische aanpak bij te sturen.

1.5 Opbouw van leerplannen

Elk leerplan is opgebouwd volgens een vaste structuur. Alle onderdelen maken inherent deel uit van het leerplan. Schoolbesturen van Katholiek Onderwijs Vlaanderen die de leerplannen gebruiken, verbinden zich tot de realisatie van het gehele leerplan.

De **inleiding** licht het leerplanconcept toe en gaat dieper in op de visie op vorming, de ruimte voor leraren(teams) en scholen en de mogelijkheden tot differentiatie.

De **situering** geeft aan waarop het leerplan is gebaseerd en beschrijft de samenhang binnen de graad en met de onderliggende graad, en de plaats in de lessentabel.

In de **pedagogisch-didactische duiding** komen de inbedding in het vormingsconcept, de krachtlijnen, de opbouw, de leerlijnen, de aandachtspunten met o.m. nieuwe accenten van het leerplan aan bod.

De **leerplandoelen** zijn helder geformuleerd en geven aan wat van leerlingen wordt verwacht. Waar relevant geeft een opsomming of een afbakening (★) aan wat bij de realisatie van het leerplandoel aan bod moet komen. Ook pop-ups bevatten informatie die noodzakelijk is bij de realisatie van het leerplandoel. De leerplandoelen zijn gebaseerd op de minimumdoelen van de basisvorming, de specifieke minimumdoelen of de doelen die leiden naar een beroepskwalificatie. Indien een leerplandoel verder gaat, vind je een '+' bij het nummer van het leerplandoel. Al die leerplandoelen zijn verplicht te realiseren. In een aantal gevallen zijn keuzedoelen opgenomen; die leerplandoelen zijn weergegeven in een grijze kleur en het nummer van het leerplandoel wordt voorafgegaan door 'K'.

De leerplandoelen zijn ingedeeld in een aantal rubrieken. Bovenaan elke rubriek vind je de relevante minimumdoelen van de basisvorming, de specifieke minimumdoelen en/of doelen die leiden naar een of meer beroepskwalificaties, afhankelijk van de finaliteit. Als leraar hoef je je die taal niet eigen te maken. Het volstaat dat je de leerplandoelen realiseert zoals opgenomen in het leerplan.

Waar relevant wordt de samenhang met andere leerplannen in dezelfde graad aangegeven, evenals de samenhang met de onderliggende graad.

'Duiding' bij een leerplandoel bevat een noodzakelijke toelichting bij het doel. In pedagogisch-didactische wenken vinden leraren inspiratie om met het leerplandoel aan de slag te gaan. Een rubriek 'extra' bij een leerplandoel biedt leraren inspiratie om verder te gaan dan wat het leerplandoel minimaal vraagt.

De **basisuitrusting** geeft aan welke materiële uitrusting vereist is om de leerplandoelen te kunnen realiseren.

Het **glossarium** bevat een overzicht van handelingswerkwoorden die in alle leerplannen van de graad als synoniem van elkaar worden gebruikt of meer toelichting nodig hebben.

De **concordantie** geeft aan welke leerplandoelen gerelateerd zijn aan bepaalde minimumdoelen, specifieke minimumdoelen of doelen die leiden naar een of meer beroepskwalificaties.

2 Situering

2.1 Samenhang in de derde graad

2.1.1 Samenhang met het leerplan Informatica- en communicatiewetenschappen binnen de finaliteit

In de D-finaliteit zijn er twee studierichtingen met een focus op Informaticawetenschappen nl. Informatica- en communicatiewetenschappen (III-InCo-d) in het studiedomein STEM en Bedrijfsondersteunende informaticawetenschappen in het studiedomein Economie & organisatie. In die studierichtingen wordt Informaticawetenschappen theoretisch benaderd met een aantal toepassingen, met het oog op verder studeren in het hoger onderwijs. In beide studierichtingen vormt softwareontwikkeling de hoofdbrok van het lessenpakket. De context van de toepassingen, gelinkt aan het studiedomein van de studierichting, en de bijkomende onderwerpen maken er twee totaal verschillende studierichtingen van.

2.1.2 Samenhang over de finaliteiten heen

In de studierichting Applicatie- en databeheer (III-ApDa-da) in de D/A-finaliteit van het studiedomein STEM en het studiedomein Economie & organisatie vormt Informaticawetenschappen het grootste onderdeel van het specifieke gedeelte. De benadering ervan is veel praktischer dan in de studierichtingen van de D-finaliteit. Informaticawetenschappen wordt in de studierichting Applicatie- en databeheer meer rechtstreeks gelinkt aan de realiteit op de werkvloer en het verder studeren voor professionele bachelor.

2.2 Plaats in de lessentabel

Het leerplan is gebaseerd op specifieke minimumdoelen. Het is gericht op 12 graaduren en is bestemd voor de studierichting Bedrijfsondersteunende informaticawetenschappen.

Het grootste deel van de lestijd wordt besteed aan softwareontwikkeling in combinatie met algoritmieken. De leerplandoelen van die onderdelen kunnen niet los van elkaar worden behandeld. Het onderdeel relationele databanken wordt in functie van het programmatorisch benaderen van databanken aangebracht.

Het geheel van de algemene en specifieke vorming in elke studierichting vind je terug op de [PRO-pagina](#) met alle vakken en leerplannen die gelden per studierichting.

3 Pedagogisch-didactische duiding

3.1 Informaticawetenschappen'' en het vormingsconcept

Het leerplan Informaticawetenschappen'' is ingebed in het vormingsconcept van de katholieke dialoogschool. In het leerplan ligt de nadruk op de economische, wiskundige, natuurwetenschappelijke en technische vorming. De wegwijzers duurzaamheid en creativiteit maken er inherent deel van uit.

In het vak Informaticawetenschappen gaan de leerlingen zelf algoritmen ontwerpen en programmeren om concrete problemen op te lossen. Dit hoort tot de natuurwetenschappelijke en technische vorming. Op het moment dat ze numerieke methodes implementeren in hun oplossing, passen ze ook de wiskundige vorming toe.



Het uiteindelijke doel is om zelf software te ontwikkelen. Dit houdt in: (bedrijfs)economische problemen analyseren, een ontwerp realiseren, dit implementeren en debuggen.

Software ontwikkelen is een creatief proces. Onderliggend is er kennis van algoritmen, algoritmische technieken, datastructuren, programmeertalen ... nodig. Die kennis vormt het fundament waarop verder wordt gebouwd. Het vraagt creativiteit van de leerling om de bouwstenen te combineren tot goed werkende software. De creativiteit in de software is vaak een maatstaf voor het inzicht dat de leerlingen hebben verworven.

De digitale wereld verandert continu, er worden voortdurend nieuwe digitale ontwikkelingen op de markt gebracht. Van leerlingen uit de richting Bedrijfsondersteunende informaticawetenschappen wordt verwacht dat ze de softwarematige evoluties actief opvolgen. Tevens moeten ze zich steeds bewust zijn van het gegeven dat al die ontwikkelingen grondstoffen nodig hebben, grondstoffen onder de vorm van bv. energie ... Grondstoffen die eindig zijn, die nog maar beperkt in de wereld aanwezig zijn. Ze moeten zich realiseren dat het geheugen misschien wel oneindig kan worden uitgebreid maar dat dit niet duurzaam is en een nodeloze verspilling van grondstoffen met zich meebrengt. In hun ontwikkelingen moeten ze streven naar duurzame oplossingen door die oplossing naar voor te schuiven die het minst geheugen en processortijd gebruikt.

Uit die vormingscomponenten en wegwijzers zijn de krachtlijnen van het leerplan ontstaan.

3.2 Krachtlijnen

Inzicht verwerven in algoritmes, algoritmische technieken, datastructuren en numerieke methodes

Een probleem heeft meestal meerdere oplossingen. Er bestaan verschillende algoritmes om bv. te sorteren. Afhankelijk van de hoeveelheid gegevens, de gebruikte datastructuur waarmee je werkt zal het ene algoritme, die ene algoritmische techniek meer zijn aangewezen dan de andere. Leerlingen moeten bewust zijn van de verscheidenheid aan oplossingsmethodes en kritisch kiezen welk algoritme ze gaan inzetten voor de oplossing van een probleem.

Veel concrete wetenschappelijke problemen vinden een oplossing in de wiskunde. Sommige problemen in de wiskunde kunnen exact worden opgelost met behulp van een algoritme. Zulke algoritmen worden directe methoden genoemd. Toch zijn voor de meeste problemen geen directe methoden beschikbaar of kiest men er niet voor. Er wordt dan een numerieke methode uitgewerkt, een algoritme dat de wiskundige oplossing zo goed mogelijk benadert. Een benadering leidt tot fouten in je oplossing. Op voorhand moet je inschatten of die fouten een invloed hebben op de oplossing van het probleem of niet.

Bij het inzetten van numerieke methodes is het belangrijk om te weten dat het hier meestal niet over exacte oplossingen gaat en dat er bijgevolg fouten optreden zoals afrondingsfouten of problemen omwille van eindigheid, random keuzes ... Die fouten zijn, afhankelijk van de concrete toepassing, verwaarloosbaar of zo groot dat er voor een andere numerieke methode moet worden gekozen.

Efficiënt software ontwikkelen om gegevens uit databanken te verwerken, beheren en presenteren

Binnen Informaticawetenschappen neemt software de belangrijkste plaats in en die plaats wordt steeds groter. De belangrijkste ontwikkelingen van de laatste tijd zijn softwarematig. Denk aan AI, cloudtoepassingen, IoB (Internet of Behaviors - software voor het verzamelen en verwerken van gegevens via internet) ... Het is dan ook essentieel dat de leerlingen een goede basis softwareontwikkeling krijgen.

Dat ze systematisch software leren ontwikkelen en de verschillende stappen doorlopen, vertrekkend van een eigen modulair ontwerp.

Voor het implementeren wordt een objectgeoriënteerde programmeertaal gebruikt. Dit biedt het voordeel dat functioneel bij elkaar horende elementen worden gegroepeerd in een klasse, module, functie met een duidelijk afgetekende taak. Het vereenvoudigt uitbreiding en herbruikbaarheid. Documentatie speelt hierin ook een grote rol. Een goed gedocumenteerd ontwerp en programma kan naderhand eenvoudiger worden uitgebreid en herwerkt. De leerlingen dienen ook voldoende aandacht te besteden aan de laatste fase, de testfase, een belangrijke fase in het softwareontwikkelingsproces. Het is essentieel om te testen of alles goed op elkaar aansluit en werkt zoals het moet werken.

Aangezien de studierichting gesitueerd is in het domein Economie & organisatie, trekken we volledig de kaart van data. De software die we ontwikkelen, gebruikt databanken om er gegevens uit te halen of aan te passen en ze nadien te presenteren.

In deze studierichting worden softwaretoepassingen ontwikkeld die aansluiten bij de (bedrijfs)economische wereld.

Databanken ontwerpen, beheren en bevragen

Alles in de hedendaagse digitale maatschappij draait om data. Iedereen probeert om zoveel mogelijk data te verzamelen om daar wijzer van te worden, hun acties, strategieën daardoor te laten bepalen. Die data moet natuurlijk ergens op een gestructureerde manier worden bewaard. De vraag naar mensen die in staat zijn om goed gebouwde relationele databanken te ontwerpen, te beheren en te bevragen is groot.

Dit bevragen kan gebeuren op verschillende manieren: een gegeven opvragen, een (gesorteerde) lijst opvragen al dan niet gefilterd, aggregatiefuncties gebruiken ... Doordat we werken in een relationele databank kunnen we gegevens uit meerdere tabellen combineren.

Een databank is natuurlijk niet statisch. Gegevens wijzigen, er komen gegevens bij, er vallen gegevens weg. Het up-to-date houden van een databank is cruciaal voor data-analyse.

Voor het bevragen en beheren van een databank wordt SQL ingezet.

3.3 Opbouw

Het leerplan bestaat uit volgende onderdelen:

- Relationale databanken maken, bevragen en beheren
 - Structuur en ontwerp van een relationele databank
 - Aanpassen en bevragen van een relationele databank
- Algoritmiek
- Efficiënt software ontwikkelen

3.4 Leerlijnen

3.4.1 Samenhang met de tweede graad

In de studierichting Bedrijfswetenschappen van de tweede graad krijgen de leerlingen een basis programmeren aangereikt.



3.4.2 Samenhang in de derde graad

Softwareontwikkeling komt ook voor in Informatica- en communicatiewetenschappen. De leerplandoelen worden daar gerealiseerd in een STEM-context. In deze studierichting worden ze uitgewerkt binnen de bedrijfseconomische context.

Het onderdeel algoritmiek is bestemd voor een aantal domeinoverschrijdende studierichtingen en Informatica- en communicatiewetenschappen. Het wordt in relatie met softwareontwikkeling aangebracht in deze richting.

3.5 Aandachtspunten

- De informaticakennis van de leerlingen in deze studierichting kan heel verscheiden zijn. Dit is afhankelijk van de richting die ze hebben gevolgd in de tweede graad en van de kennis die ze hebben verworven buiten de school door zelfstudie of door deel te nemen aan initiatieven zoals CoderDojo, CodeFever, STEM-workshops ... Er zal voldoende differentiatie moeten worden ingebouwd om met die verschillen in voorkennis om te gaan. Zorg voor uitdagende opdrachten voor leerlingen die al een goede (basis)kennis verworven hebben. Je kan hen ook inzetten bij het leerproces van de andere leerlingen.
- Het is belangrijk dat leerlingen inzicht krijgen in de opbouw van databanken en databanken leren ontwerpen, bevragen en analyseren. De leerlingen hebben geen voorkennis over die materie. Het is niet nodig om met grote, ingewikkelde databanken te werken die tientallen tabellen bevatten. Ontwerp relatief eenvoudige databanken met een duidelijke structuur uit een bedrijfsomgeving.
- Het is de bedoeling van de onderdelen “Algoritmiek” en “Efficiënt software ontwikkelen” om een goede basis te leggen waarop kan worden verder gebouwd in het hoger onderwijs. Het is dan ook aangewezen te werken met haalbare projecten die de leerlingen zelf kunnen uitwerken. Het is niet de intentie van het leerplan dat leerlingen, bij het uitwerken van hun oefeningen, op internet de nodige code zoeken en die implementeren in hun oplossing. Om de leerlingen in het hoger onderwijs slaagkansen te geven bij algoritmiek en softwareontwikkeling moeten zij de basisoefeningen zelf uitwerken. Uiteraard mag bestaande code wel worden gebruikt als voorbeeld om uit te leren. Voor het uitwerken van de leerplandoelen “Algoritmiek” en “Efficiënt software ontwikkelen” vertrek je steeds van een concrete probleemstelling.
- Het is niet aangewezen om leerplandoelen afzonderlijk, in de volgorde van het leerplan, af te werken. In elke toepassing, wordt normaal gezien aan één of meerdere leerplandoelen gewerkt. Je kan de leerplandoelen van algoritmiek ook verwerken bij het uitwerken van de leerplandoelen softwareontwikkeling. Je kan bv. het sorteren van gegevens in een module aan bod laten komen.
- Het leerplandoel over numerieke methodes is gelinkt aan het vak Wiskunde. Wiskundige concepten aangeleerd tijdens de lessen Wiskunde worden via numerieke methodes omgezet in programmacode.

3.6 Leerplanpagina

Wil je als gebruiker van dit leerplan op de hoogte blijven van inspirerend materiaal, achtergrond, professionalisering en lerarennetwerken, surf dan naar de [leerplanpagina](#).



4 Leerplandoelen

4.1 Relationale databanken maken, bevragen en beheren

Minimumdoelen, specifieke minimumdoelen of doelen die leiden naar BK

4.1.1 Structuur en ontwerp van een relationele databank

LPD 1 De leerlingen lichten de structuur en de werking van relationele databanken toe.

Duiding: Het begrippenkader van een relationele databank: tabel, rij, kolom, veld, record, primaire en referentiële sleutel, index, relaties, referentiële integriteit en cascading update/delete.

Een goed gestructureerde databank voor een applicatie bevat volgende kenmerken: onderhoudbaar, integriteit, consistentie en redundantie.

Wenk: Je kan het belang van goed gestructureerde databanken aantonen via voorbeelden van databanken met een slechte structuur (redundantie, inconsistentie ...).

LPD 2 De leerlingen implementeren een relationele databank op basis van een eigen ontwerp.

★ Datamodel: ontwikkeling, normalisering, vertaling naar relationele databank

Duiding: De normalisatieregels van Codd worden gebruikt tot en met de derde normaalvorm.

Wenk: Het is aan te raden dat je voldoende aandacht besteedt aan het ontwikkelen van een datamodel en het maken van een eigen ontwerp door de leerlingen. Voor het ontwikkelen van een datamodel kan je een grafische voorstelling zoals ERD gebruiken.

Wenk: Het is niet nodig elk ontwerp van een databank te implementeren. Voor de implementatie gebruik je een serverdatabase zoals MySQL, SQL Server, MariaDB, Aurora, PostgreSQL ... ofwel in een eigen installatie (on-premise) of in een cloud systeem

Wenk: Omdat een duidelijke en volledige probleemstelling tot een te lange opgave zou leiden, is het aan te raden dat de leraar als 'klant/gebruiker' fungeert zodat hij eventuele vragen van begrenzing in de les kan beantwoorden.

LPD K1 De leerlingen breiden aan de hand van een concrete probleemstelling de structuur van een relationele databank uit.

Duiding: Bij het aanpassen van een relationele databank de gevolgen van de aanpassing correct inschatten.



4.1.2 Aanpassen en bevragen van een relationele databank

Minimumdoelen, specifieke minimumdoelen of doelen die leiden naar BK

LPD 3 De leerlingen gebruiken een gestructureerde querytaal voor de bevraging van een relationele databank.

- ★ Selecteren, groeperen, sorteren, berekenen en filteren

Wenk: Je laat de leerlingen bij de bevraging van een relationele databank eerst werken met een enkele tabel en naderhand met meerdere gerelateerde tabellen. Het groeperen wordt ingezet in functie van het toepassen van aggregatiefuncties om totalen, aantallen, gemiddelden ... te berekenen.

Wenk: Je kan het belang toelichten van SQL als standaard voor het bevragen van een relationele databank in een client-server omgeving.

Wenk: Je kan de leerlingen subqueries leren maken.

LPD 4 De leerlingen gebruiken een gestructureerde querytaal voor het wijzigen, toevoegen en verwijderen van gegevens in een relationele databank.

Wenk: Je kan gebruik maken van stored procedures om SQL-instructies te hergebruiken met paramaters.

4.2 Algoritmiek

Minimumdoelen, specifieke minimumdoelen of doelen die leiden naar BK

LPD 5 De leerlingen passen een gestructureerde programmeertaal toe bij het ontwerpen van een oplossing.

- ★ Gebruik van softwarebibliotheken

2de graad: II-Inf-d LPD 2, 3: gestructureerde programmeertaal, softwarebibliotheken

Wenk: Er wordt een tekstuele programmeertaal gehanteerd. Gebruik de afspraken inherent aan de gebruikte programmeertaal. Math is in veel programmeertalen een softwarebibliotheek voor wiskundige functies.

Wenk: Je kan de leerlingen naast variabelen ook het gebruik van constanten aanleren. Wijs hen op het belang van de gegevenstypes. Je kan delen van de code in functies laten uitwerken en zo decompositie toepassen. Je kan de leerlingen leren om een logische en zinvolle naamgeving voor variabelen, constanten en functies te gebruiken.

LPD 6 De leerlingen ontwerpen en implementeren algoritmen met datastructuren voor concrete problemen.

Duiding: De keuze van het algoritme en de datastructuur zijn verbonden met elkaar.

- Wenk: Het is niet nodig om alle datastructuren aan bod te laten komen. Je kan vertrekken van ééndimensionale- en meerdimensionale arrays en afhankelijk van de aangebrachte concrete problemen andere datastructuren aan bod laten komen.
- Wenk: Je kan de leerlingen een concreet probleem op verschillende manieren en met verschillende datastructuren laten oplossen en dan analyseren wat de beste combinatie is voor bv. het sorteren van gegevens.
- Wenk: Je kan verschillende soorten sorteeralgoritmes aanreiken en door afzonderlijke groepen leerlingen laten beoordelen, uitwerken en vergelijken met elkaar op gebied van snelheid, geheugengebruik, invloed van hoeveelheid gegevens ... Voor datastructuren kan je dezelfde methodieken hanteren.
- Wenk: Het is niet nodig om elk algoritme ook effectief te programmeren. In het geval er meerdere algoritmes zijn om een probleem op te lossen, kan een voorafgaande analyse al bepalend zijn voor de uiteindelijke keuze die wordt omgezet in een programma.

LPD 7 De leerlingen passen algoritmische technieken toe bij het programmeren van zelf ontworpen oplossingen.

- Wenk: Mogelijke algoritmische technieken zijn: brute-force (Engels voor “brute kracht”), dynamisch programmeren, gretig (greedy), recursie, verdeel-en-heers. Het is niet nodig om alle algoritmische technieken aan bod te laten komen. Je kiest afhankelijk van het concrete probleem voor een geschikte algoritmische techniek.
- Wenk: Je kan de leerlingen een concreet probleem met verschillende algoritmische technieken laten oplossen en dan analyseren wat de beste methode is, wat het verschil is in gebruik op gebied van geheugengebruik, doorlooptijd.
- Wenk: Voorbeelden van eenvoudige algoritmen die zich lenen om met of zonder recursie op te lossen, zijn het bepalen van de grootste gemene deler van 2 gehele getallen, het berekenen van $n!$ (faculteit), het opstellen van de rij van Fibonacci ...

LPD 8 De leerlingen ontwerpen oplossingen die externe gegevens in- en uitvoeren.

- Wenk: Bij invoer van en uitvoer naar externe gegevensbronnen handelt het over in- en uitvoer van gegevens die in bestanden staan. Dat kan een CSV -, tekst-, JSON-, XML-bestand, een databank of een rekenblad zijn. Het kan ook handelen over invoer van of uitvoer naar een website of de cloud.
- Wenk: Het is niet nodig om elk bestandstype aan bod te laten komen. Je kiest een invoer en uitvoer in overeenstemming met het probleem dat je wil oplossen.

LPD 9 De leerlingen lossen concrete wiskundige problemen op met behulp van numerieke methodes.

- Wenk: Het betreft wiskundige problemen vanuit verschillende toepassingsgebieden zoals economie, wetenschappen, wiskunde ...
Voorbeelden:
- nulwaarden, extrema, afgeleiden, bepaalde integralen van functies,



- matrixvermenigvuldiging, oplossen van stelsels eerstegraadsvergelijkingen.

Bij voorkeur komt een numerieke methode aan bod juist nadat de leerstof binnen de wiskunde is behandeld.

Wenk: Het is niet de bedoeling dat de leerlingen complexe numerieke algoritmes zelf ontwerpen. Dit kan wel voor eenvoudige algoritmes. Bij complexere algoritmes wordt het algoritme klassikaal besproken of individueel of in groep bestudeerd. Het is belangrijk dat de leerlingen begrijpen hoe het algoritme werkt en wat de eventuele beperkingen ervan zijn zoals afrondingsfouten, eindigheid ...

Wenk: Het is niet de bedoeling dat de leerlingen die algoritmes allemaal zelf omzetten in programmacode. Veel numerieke methodes zijn al uitgewerkt in de softwarebibliotheken en vind je daar terug in de vorm van functies, procedures ... Leer de leerlingen gebruik maken van voorgeprogrammeerde numerieke algoritmes in de softwarebibliotheek en de beschikbare documentatie te raadplegen.

Wenk: Je kan een concreet probleem vanuit de lessen Economie of Wiskunde gebruiken. Het is aan te bevelen om daarvoor samen te werken met de collega's van de andere vakken.

4.3 Efficiënt software ontwikkelen

Minimumdoelen, specifieke minimumdoelen of doelen die leiden naar BK

LPD 10 De leerlingen maken een ontwerp voor een softwaremodule en passen de principes van objectgeoriënteerd programmeren toe.

Wenk: Je leert de leerlingen een klassendiagram te maken met daarin de attributen en methoden van de klasse, met inbegrip van interfaces.

Wenk: Je kan m.b.v. een grafisch model het class diagram voor het ontwerp van een softwaremodule voorstellen. Je kan dit met de hand tekenen of een softwarepakket hiervoor gebruiken.

LPD 11 De leerlingen maken een modulair ontwerp met een 3-lagenarchitectuur voor een softwaretoepassing en passen de basisprincipes voor een goed ontwerp toe.

Wenk: De basisprincipes voor een goed modulair ontwerp zijn: lage koppeling (low coupling), sterke cohesie (high cohesion), enkelvoudige verantwoordelijkheid (single responsibility).

Wenk: Het is zinvol om soms te vertrekken vanuit een werkend programma en daar functionaliteit of een laag (layer) aan toe te voegen of de werking aan te passen. Er wordt een nieuw ontwerp gemaakt of een bestaand ontwerp aangepast.

Wenk: Het is niet nodig om elk ontwerp om te zetten in een werkend programma.

LPD 12 De leerlingen implementeren objectgeoriënteerde softwaremodules op een manier die herbruikbaarheid ervan bevordert en samenwerking vereenvoudigt.

- Wenk: Met herbruikbaarheid wordt bedoeld dat modules op die manier worden ontwikkeld dat ze vlot kunnen worden gebruikt in of aangepast aan andere contexten.
Leer de leerlingen een schikking te hanteren bij het maken van programma's.
Het gebruik van een consequente naamgeving vereenvoudigt de herbruikbaarheid van en samenwerking van softwaremodules.
Het toevoegen van commentaar bevordert de leesbaarheid van het programma voor jezelf en voor anderen.
- Wenk: Leer de leerlingen om met versiebeheer te werken. Je kan gebruik maken van een open source dienst die git voor versiebeheer aanbiedt bv. GitHub Classroom, AWS CodeCommit, Azure DevOps ...
- Wenk: Je kan een bestaande implementatie aanpassen op basis van een verbeterd of aangepast ontwerp.
- Wenk: Laat leerlingen code van andere leerlingen nalezen of voortbouwen op een ontwerp of implementatie van een medeleerling. Je kan leerlingen hun implementatie laten vergelijken met die van medeleerlingen.

LPD 13 De leerlingen houden bij het implementeren van softwaremodules rekening met gebruiksvriendelijkheid voor de eindgebruiker.

- Wenk: Gebruiksvriendelijkheid houdt o.a. in:
- validatie,
 - foutafhandeling,
 - exception handling.
- Wenk: Je kan de leerlingen duidelijk maken dat het belangrijk is om voldoende informatie en instructies op de gebruikersinterface (user interface) aan te brengen.
- Wenk: Bij het ontwerpen van een gebruikersinterface kan je vaak al gebruiksvriendelijkheid inbouwen. Zorg voor een gebruikersinterface die zoveel mogelijk fouten tracht te voorkomen bv. keuzerondjes i.p.v. aankruisvakjes, validatie bij invoer, invoermaskers ...

LPD 14 De leerlingen voeren CRUD (create, read, update en delete) operaties uit op een relationele databank in de data laag vanuit de domeinlogica laag.

- Duiding: Dit doel is gelinkt aan de leerplandoelen 3 en 4. Dezelfde query-instructies zullen hier terug komen.
- Wenk: De gevaren toelichten van het simultaan programmatorisch aanpassen van dezelfde gegevens, m.a.w. concurrency problemen.

LPD 15 De leerlingen ontwikkelen als presentatielaag een responsive website met hedendaagse webtechnologie.

- Wenk: De nodige aandacht wordt besteed aan het loskoppelen van inhoud en opmaak.
- Wenk: Je kan een oefening online zetten om de leerlingen te laten zien wat responsive betekent.



Wenk: Je kan <https://www.w3schools.com/> aanreiken aan de leerlingen als hulpbron voor de HTML- en CSS-code. De leerlingen kunnen met HTML de structuur van een website opzetten en met CSS stijlelementen toevoegen aan een website.

LPD 16 De leerlingen ontwikkelen server-side scripts om een databank vanuit een website aan te passen en te bevragen.

Duiding: Dit doel is gelinkt aan de leerplandoelen 3 en 4. Dezelfde query-instructies zullen hier terug komen.

LPD K2 De leerlingen ontwikkelen en implementeren client-side-scripts voor een websitetoeepassing.

LPD 17 De leerlingen hanteren aanwezige hulpbronnen bij het implementeren van softwaremodules en schrijven documentatie bij eigen ontworpen softwaremodules.

Wenk: Leer de leerlingen de mogelijke hulpbronnen van de ontwikkelomgeving gericht en kritisch gebruiken: helpfunctie, programmadocumentatie (van externe modules) en softwareraamwerken, automatisch aanvullen ... De basis moeten de leerlingen zonder externe hulp kunnen implementeren.

Wenk: Leer de leerlingen de mogelijke hulpbronnen gericht en kritisch gebruiken: helpfunctie, fora, internet ...

LPD 18 De leerlingen testen eigen implementaties om eventuele fouten te identificeren en verwijderen.

Wenk: De soorten fouten zijn: logische fouten, syntaxfouten en runtimefouten.

Wenk: Het is belangrijk dat leerlingen leren om een programma voldoende en gericht te testen. Ook alle mogelijke uitzonderingen worden meegenomen in die test. Als de leerlingen naderhand wijzigingen uitvoeren aan het programma testen ze het programma opnieuw.
Laat de leerlingen testgegevens opstellen voor ze starten met het uitwerken van hun programma.

Extra: Je kan een unit test voor een klasse maken.

LPD 19 De leerlingen debuggen eigen implementaties om de bronnen van eventuele fouten op te sporen.

Wenk: Je kan de debugger op meerdere manieren inzetten voor het opsporen van de bronnen van fouten: stap voor stap doorlopen, gegevens onderzoeken, voorwaarden controleren afhankelijk van de waarde van variabelen, breekpunten (breakpoints) plaatsen ...

5 Lexicon

Het lexicon bevat een verduidelijking bij begrippen die in het leerplan worden gebruikt. Die verduidelijking gebeurt enkel ten behoeve van de leraar.

3-lagen architectuur

De 3-lagen architectuur bestaat uit de volgende drie lagen: data laag (persistentie-laag), domein-logica (business-logica), presentatielaag (User Interface).

Datastructuren

De volgende datastructuren kunnen worden geïmplementeerd: één dimensionale- en meer dimensionale arrays, lijst (list), stapel (stack), wachtrij (queue), hashtable (hashtable), boom en graaf.

De meeste datastructuren zijn varianten van arrays. De manier waarop de array wordt benaderd of geïmplementeerd, bepaalt welke datastructuur het juist is. Hierbij wordt de datastructuur ofwel aangesproken via een index of via een sleutel.

Gestructureerde programmeertaal

Een gestructureerde programmeertaal maakt gebruik van controlestructuren, blokstructuren, functies ... De controlestructuren zijn: opeenvolging (sequentie), keuze (selectie), herhaling (iteratie).

Principes objectoriëntatie

De principes van objectoriëntatie zijn: abstractie, inkapseling (encapsulation), overerving (inheritance) en polymorfisme (polymorphism).

Responsive website

Een responsive website past zich aan naargelang het scherm (laptop, tablet, smartphone) waarop het wordt getoond.

6 Basisuitrusting

Basisuitrusting verwijst naar de infrastructuur en het (didactisch) materiaal die beschikbaar moeten zijn voor de realisatie van de leerplandoelen.

Om de leerplandoelen te realiseren dient de school minimaal de hierna beschreven infrastructuur en materiële en didactische uitrusting ter beschikking te stellen die beantwoordt aan de reglementaire eisen op het vlak van veiligheid, gezondheid, hygiëne, ergonomie en milieu. We adviseren de school om de grootte van de klasgroep en de beschikbare infrastructuur en uitrusting op elkaar af te stemmen.

6.1 Infrastructuur

Een leslokaal

- dat qua grootte, akoestiek en inrichting geschikt is om communicatieve werkvormen te organiseren;
- met een (draagbare) computer waarop de nodige software en audiovisueel materiaal kwaliteitsvol werkt en die met internet verbonden is;
- met de mogelijkheid om (bewegend beeld) kwaliteitsvol te projecteren;
- met de mogelijkheid om geluid kwaliteitsvol weer te geven;
- met de mogelijkheid om draadloos internet te raadplegen met een aanvaardbare snelheid.

Toegang tot (mobile) devices voor leerlingen.



6.2 Materiaal en gereedschappen waarover elke leerling moet beschikken

Om de leerplandoelen te realiseren beschikt elke leerling minimaal over onderstaand materiaal. De school bespreekt in de schoolraad wie (de school of de leerling) voor dat materiaal zorgt. De school houdt daarbij uitdrukkelijk rekening met gelijke kansen voor alle leerlingen.

- pc of laptop met administratorrechten;
- de nodige actuele software, services, online platformen of servers om de leerplandoelen te realiseren.

7 Glossarium

In het glossarium vind je synoniemen voor en een toelichting bij een aantal handelingswerkwoorden die je terugvindt in leerplandoelen en (specifieke) minimumdoelen van verschillende graden.

Handelingswerkwoord	Synoniem	Toelichting
Analyseren		Verbanden zoeken tussen gegeven data en een (eigen) besluit trekken
Beargumenteren	Verklaren	Motiveren, uitleggen waarom
Beoordelen	Evalueren	Een gemotiveerd waardeoordeel geven
Berekenen	Berekeningen uitvoeren	
Berekeningen uitvoeren	Berekenen	
Beschrijven	Toelichten, uitleggen	
Betekenis geven aan	Interpreteren	
Een (...) cyclus doorlopen	Een (...) proces doorlopen	Via verschillende fasen tot een (deel)resultaat komen of een doel bereiken
Een (...) proces doorlopen	Een (...) cyclus doorlopen	Via verschillende fasen tot een (deel)resultaat komen of een doel bereiken
Evalueren	Beoordelen	
Gebruiken	Hanteren, inzetten, toepassen	
Hanteren	Gebruiken, inzetten, toepassen	
Identificeren		Benoemen; aangeven met woorden, beelden ...
Illustreeren		Beschrijven (toelichten, uitleggen) aan de hand van voorbeelden
In dialoog gaan over	In interactie gaan over	
In interactie gaan over	In dialoog gaan over	
Interpreteren	Betekenis geven aan	
Inzetten	Gebruiken, hanteren, toepassen	
Kritisch omgaan met	Kritisch gebruiken	
Kwantificeren		Beredeneren door gebruik te maken van verbanden, formules, vergelijkingen ...

Onderzoeken	Onderzoek voeren	Verbanden zoeken tussen zelf verzamelde data en een (eigen) besluit trekken
Onderzoek voeren	Onderzoeken	Verbanden zoeken tussen zelf verzamelde data en een (eigen) besluit trekken
Reflecteren over		Kritisch nadenken over en argumenten afwegen zoals in een dialoog, een gedachtewisseling, een paper
Testen	Toetsen	
Toelichten	Beschrijven, uitleggen	
Toepassen	Gebruiken, hanteren, inzetten	
Toetsen	Testen	
Uitleggen	Beschrijven, toelichten	
Verklaren	Beargumenteren	Motiveren, uitleggen waarom

8 Concordantie

8.1 Concordantietabel

De concordantietabel geeft duidelijk aan welke leerplandoelen specifieke minimumdoelen (SMD) realiseren.

Leerplandoel	Specifieke minimumdoelen
1	SMD 07.05.01
2	SMD 07.05.02
3	SMD 07.05.03
4	SMD 07.05.03
5	SMD 07.01.01
6	SMD 07.01.01
7	SMD 07.01.01
8	SMD 07.01.01
9	SMD 07.01.01
10	SMD 07.03.01; SMD 07.03.02
11	SMD 07.03.01
12	SMD 07.03.02; SMD 07.03.03



13	SMD 07.03.03
14	SMD 07.03.03
15	SMD 07.03.03
16	SMD 07.03.03
17	SMD 07.03.03
18	SMD 07.03.04
19	SMD 07.03.04

8.2 Specifieke minimumdoelen

07.01.01 De leerlingen programmeren zelf ontworpen oplossingen voor concrete problemen.

Onderliggende (kennis)elementen:

- Algoritmen en datastructuren
- Algoritmische technieken
- Numerieke methodes
- Gebruik van softwarebibliotheken
- Gestructureerde programmeertaal
- Invoer van en uitvoer naar externe gegevensbronnen

Memorie van toelichting:

Algoritmische technieken zoals recursie, brute-force, gulzig (greedy), verdeel-en-heers en dynamisch programmeren.

Datastructuren zoals lijst, stapel (stack), wachtrij (queue), boom, graaf, hashtabel.

Externe gegevensbronnen zoals bestanden, databank, website, cloud.

07.03.01 De leerlingen maken een modulair ontwerp voor een softwaretoepassing.

07.03.02 De leerlingen passen principes toe van objectgeoriënteerd programmeren in een eigen ontwerp.

Memorie van toelichting:

Principes van objectgeoriënteerd programmeren zoals abstractie, overerving, polymorfisme, inkapseling.

07.03.03 De leerlingen implementeren softwaremodules op een manier die herbruikbaarheid ervan bevordert en samenwerking vereenvoudigt.

Memorie van toelichting:

Met herbruikbaarheid wordt bedoeld dat modules op die manier worden ontwikkeld dat ze vlot kunnen worden gebruikt in of aangepast aan andere contexten.

07.03.04 De leerlingen testen en debuggen eigen implementaties om eventuele fouten te identificeren en op te lossen.

Memorie van toelichting:

Strategieën zoals:

- tussenresultaten afdrukken, gegevens onderzoeken, voorwaarden controleren afhankelijk van variabelen;
- stap voor stap doorlopen, breakpoint plaatsen;
- testgegevens opstellen en unit testen.

- 07.05.01 De leerlingen lichten de structuur en de werking van relationele databanken toe.
- 07.05.02 De leerlingen implementeren een relationele databank op basis van een eigen ontwerp.
- Memorie van toelichting:
Vaardigheden zoals diagrammen gebruiken, een datamodel maken, normalisatieregels toepassen.
- 07.05.03 De leerlingen gebruiken een gestructureerde querytaal voor de bevraging en de aanpassing van gegevens in een relationele databank.
- Memorie van toelichting:
Bevraging zoals selecteren, groeperen, sorteren, berekenen, filteren.
Aanpassing zoals wijzigen, verwijderen, toevoegen.
Combineren zoals door gebruik te maken van join, subquery.



Inhoud

1	Inleiding.....	3
1.1	Het leerplanconcept: vijf uitgangspunten	3
1.2	De vormingscirkel – de opdracht van secundair onderwijs	3
1.3	Ruimte voor leraren(teams) en scholen	4
1.4	Differentiatie	5
1.5	Opbouw van leerplannen.....	6
2	Situering	7
2.1	Samenhang in de derde graad	7
2.1.1	Samenhang met het leerplan Informatica- en communicatiewetenschappen binnen de finaliteit...7	
2.1.2	Samenhang over de finaliteiten heen.....	7
2.2	Plaats in de lessentabel.....	7
3	Pedagogisch-didactische duiding	7
3.1	Informaticawetenschappen” en het vormingsconcept	7
3.2	Krachtlijnen	8
3.3	Opbouw.....	9
3.4	Leerlijnen.....	9
3.4.1	Samenhang met de tweede graad	9
3.4.2	Samenhang in de derde graad	10
3.5	Aandachtspunten	10
3.6	Leerplanpagina.....	10
4	Leerplandoelen	11
4.1	Relationele databanken maken, bevragen en beheren.....	11
4.1.1	Structuur en ontwerp van een relationele databank	11
4.1.2	Aanpassen en bevragen van een relationele databank.....	12
4.2	Algoritmie.....	12
4.3	Efficiënt software ontwikkelen	14
5	Lexicon	16
6	Basisuitrusting	17
6.1	Infrastructuur	17
6.2	Materiaal en gereedschappen waarover elke leerling moet beschikken	18
7	Glossarium.....	18

8	Concordantie	19
8.1	Concordantietabel.....	19
8.2	Specifieke minimumdoelen.....	20