



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats in 2027 en wordt dan georganiseerd door HAN University of Applied Sciences. Zodra daarover meer informatie beschikbaar is, is deze hier te vinden.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden_nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.



Programma-ontwikkeling in inleidend programmeeronderwijs

H. Koppelman, N.M. van Diepen
Faculteit der Informatica, Universiteit Twente

Inleiding

De cursus Inleiding in het Programmeren is bedoeld voor technische studenten aan de Universiteit Twente (studielast: honderd uur). De globale leerdoelen zijn: het leren van een programmeertaal en het systematisch leren ontwikkelen van programma's. Deze cursus is onlangs vernieuwd, met als doelstelling het tweede leerdoel nadrukkelijker aan bod te laten komen. In deze presentatie komen een aantal uitgangspunten aan de orde waarop die vernieuwing is gebaseerd.

Kern

Vier fasen

Er worden vier fasen onderscheiden om tot een programma te komen: de analysefase, de ontwerpfase, de implementatiefase en de testfase. Deze fasen worden expliciet gemaakt en afgebakend. Elke fase levert een produkt op dat duidelijk omschreven is. In de analysefase moeten de studenten als probleemoplossers bijvoorbeeld vanuit een beschrijving van een probleem en in interactie met de probleemsteller een probleemdefinitie geven.

Specificaties en implementaties

Centraal in de cursus staat het onderscheid tussen de specificatie en de implementatie van een procedure of functie. De specificatie beschrijft het effect, de implementatie de wijze waarop dat effect wordt bereikt. De specificatie bevat alle informatie die nodig is voor de gebruiker, en ook alle informatie die nodig is voor de implementator. Een specificatie beschrijft welk effect het gebruik van een procedure of functie heeft, mits aan gegeven voorwaarden is voldaan. Een gebruiker hoeft dus niet verder 'omlaag' te kijken dan de specificatie, een implementator niet verder 'omhoog'.

Het gebruik van bouwstenen

Programmeren, en met name het ontwerpen, is een proces waarbij gebruik wordt gemaakt van al klaarstaande bouwstenen. Tot deze bouwstenen behoren procedures en functies, die in diverse modules ter beschikking staan. Voor de gebruikers zijn alleen de specificaties van belang, niet de implementaties. Tot de bouwstenen worden ook gerekend vaste manieren om klassen van problemen aan te pakken. De gedachte is dat niet elk (deel)probleem weer een nieuw probleem is, waarvoor een nieuw algoritme moet worden bedacht. Een deelprobleem kan vaak beschouwd worden als een bijzonder geval van een bekend probleem, waarvan de structuur van de oplossing al vastligt. Zo kunnen veel lussen beschouwd worden als bijzondere gevallen van een beperkt aantal vaste schema's. Zulke schema's worden in de cursus aangereikt. Voorbeelden zijn: een schema voor lineair zoeken en schema's voor de verwerking van invoer.

Expliciete kwaliteitscriteria

Studenten moeten niet alleen programma's kunnen schrijven, maar ook de kwaliteit ervan kunnen beoordelen op een aantal criteria. Daarbij gaat het in deze cursus om de criteria correctheid, leesbaarheid en begrijpelijkheid, modulariteit en gestructureerdheid, en efficiëntie. Deze criteria worden omschreven, en er worden positieve en negatieve voorbeelden genoemd. Verschillende algoritmes voor hetzelfde probleem worden vergeleken op de genoemde criteria.

Ondersteuning bij de ontwerpfase

De complexiteit van oplossingen kan worden beheerst als deelproblemen onafhankelijk van elkaar en van het probleem waarvan ze deel zijn kunnen worden opgelost. Die onafhankelijkheid wordt in de cursus nagestreefd door elke procedure en functie die wordt geïdentificeerd als deelprobleem, te specificeren: de specificatie moet alle informatie bevatten om de betreffende procedure of functie te kunnen implementeren. Het splitsen van problemen in deelproblemen in het kader van de top-downmethode wordt ondersteund door criteria te

noemen waaraan splitsingen moeten voldoen.

Een voorbeeld van zo'n criterium is: verschillende abstractieniveaus in de probleemstelling moeten in het algoritme gescheiden zijn. Als bijvoorbeeld wordt gevraagd te bepalen of een palindroom aanwezig is in een gegeven rij woorden, is er sprake van twee abstractieniveaus: het niveau van een rij elementen (waarvan de interne structuur niet van belang is) die moet worden doorzocht, en het niveau van een woord dat op een eigenschap moet worden getoetst. Ook bij het bottom-up ontwerpen is abstractie van belang: de gebruiker van bouwstenen is geïnteresseerd in specificaties, niet in implementaties.

Onderwijskundige uitgangspunten

Programmeren leer je door het te doen: zelfwerkzaamheid (gekoppeld aan intensieve begeleiding) staat centraal. Verder wordt gebruik gemaakt van de completion strategy: er worden programma's aangeboden die studenten moeten begrijpen en aanvullen. Voordelen hiervan zijn de voorbeeldfunctie van goed ontworpen en goed gedocumenteerde programma's, en de mogelijkheid met complexere en dus interessantere problemen te kunnen werken.

Slot

Met de vernieuwde cursus is tot dusver op beperkte schaal geëxperimenteerd. Zowel docenten als studenten zijn overwegend tevreden.