



## Stichting NIOC en de NIOC kennisbank

Stichting NIOC ([www.nioc.nl](http://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](http://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 op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

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

R. Smedinga [kennisbank@nioc.nl](mailto:kennisbank@nioc.nl).

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.



## Abstracte datatypen in Modula-2

K.G. van den Berg, N.M. van Diepen  
Faculteit Informatica, Universiteit Twente, Enschede

### Inleiding

Abstracte datatypen (ADT's) vormen een steeds belangrijker onderdeel van de leerstof in inleidende cursussen programmeren. In studieboeken wordt een grote verscheidenheid aangetroffen in de behandeling van ADT's. In deze bijdrage wordt een didactiek gegeven van ADT's in de imperatieve programmeertaal Modula-2. Deze benadering wordt geplaatst naast de aanpak van ADT's in de functionele programmeertaal Miranda (Miranda is een handelsmerk van Research Software Limited). De volgende stappen worden onderscheiden: het model, de signatuur, de typedefinitie, de algebraïsche en/of operationele specificatie, en de implementatie. Speciale aandacht krijgen: data hiding, sharing van datastructuren en het gebruik van pseudo-generische elementtypen.

### Abstracte datatypen in het programmeeronderwijs

Het eerstejaars informatica programmeeronderwijs aan de Universiteit Twente bestaat uit drie vakken: functioneel programmeren (in Miranda) in het eerste trimester, gevolgd door imperatief programmeren (in Modula-2) en in het derde trimester het vak programmeertechnieken (in beide programmeertalen). In het tweede trimester wordt nadruk wordt gelegd op data-abstractie en modulaire opbouw van programma's in Modula-2. Hierbij wordt uitgegaan van kennis van een functionele programmeertaal. In deze bijdrage wordt ingegaan op de aanpak van de behandeling van abstracte datatypen in het tweede trimester.

### De aanpak van abstracte datatypen

Er blijkt in studieboeken voor imperatief programmeren een grote verscheidenheid te zijn in de behandeling van ADT's. Er is een spectrum van mogelijkheden aan te geven dat loopt van een zuivere functionele aanpak naar een object georiënteerde aanpak. Vaak wordt slechts één mogelijkheid gepresenteerd en de keuze niet verantwoord. In het vak imperatief programmeren is een aanpak ontwikkeld waarin de verschillende mogelijkheden in een kader geplaatst kunnen worden en voor- en nadelen kunnen worden geëxpliciteerd.

De eerste twee stappen in de behandeling van ADT's zijn in principe onafhankelijk van de programmeertaal:

1. Het model: een informele beschrijving van het datatype en de operaties op dit type.
2. De signatuur: het vastleggen van het domein en codomein van de operaties. Een onderscheid wordt gemaakt tussen basisoperaties en afgeleide operaties. In een signatuurgraaf kan de relatie tussen typen en operaties worden weergegeven.

De volgende drie stappen zijn afhankelijk van de constructies die in een programmeertaal aanwezig zijn, zoals het functie(procedure)mechanisme, het parametermechanisme (var en value), de soorten variabelen (statische en dynamische variabelen).

3. De typedefinitie: de signatuur wordt geconcretiseerd in de syntax van de programmeertaal.
4. De specificatie: het vastleggen van de betekenis van de operaties. Hierbij worden twee soorten specificaties onderscheiden:
  - een algebraïsche specificatie: de betekenis van de operaties worden vastgelegd door middel van axioma's
  - een operationele specificatie: de betekenis van de operaties worden vastgelegd door middel van pre- en postcondities in termen van een beschrijvingsmodel.
5. De implementatie: het kiezen van een interne datastructuur (een statische of dynamische datastructuur), het bepalen van de representatie-invariant en de uitwerking van de algoritmen voor elke operatie.

### **Een voorbeeld abstract datatype**

De aanpak met de vijf stappen wordt geïllustreerd met de stack ADT. Hoewel dit voorbeeld helemaal uitgebeend lijkt is het door zijn bekendheid heel goed bruikbaar om de vaak impliciete aannamen duidelijk te maken.

Achtereenvolgens worden behandeld:

- het model en de signatuur voor de stack ADT
- de definitie, de algebraïsche specificatie en een implementatie voor de stack ADT met een polymorf elementtype in Miranda met een voorbeeldtoepassing
- de definitiemogelijkheden voor de stack ADT in Modula-2: zuiver functioneel, functioneel met zij-effect, met var-parameters, met een globale stack, met een stackobject
- de algebraïsche specificatie en de operationele specificatie waarin een Miranda datastructuur als beschrijvingsmodel wordt gebruikt
- het gebruik van een pseudo-generisch elementtypen; het wordt hiermee bovendien mogelijk eenvoudig een samenstelling te maken van abstracte datatypen
- de data hiding in Modula-2 door middel van het gebruik pointervariabelen met mogelijke zij-effecten van sharing, het gebruik van het assignment en de test op gelijkheid
- statische en dynamische implementaties met pointers.

### **Slot**

In deze aanpak van abstracte datatypen in het programmeeronderwijs wordt een aantal stappen onderscheiden. Hiermee kan een aantal ontwerpbeslissingen worden geëxpliciteerd. De aanpak stimuleert de studenten tot het gebruik van data-abstractie in modulair opgezette programma's. Hergebruik van standaard modules voor ADT's wordt bereikt door de invoering van pseudo-generische elementtypen.