

# 1. Om te beginnen

## 1.1 Installatie

Je wilt leren programmeren, je hebt de beschikking over een computer en je hebt Free Pascal gedownload. Nadat je het pakket gedownload en geïnstalleerd hebt zit je met een mooi icoontje op je bureaublad. Op de website komen alleen maar Engelstalige handleidingen te voorschijn. Hoe nu verder? Op dit punt pikken we met dit boek de draad op.

Als je het volledige programmapakket Free Pascal 2.0 wilt installeren, heb je een computer nodig met minimaal 120 megabyte aan vrije ruimte. Welk besturingssysteem je draait maakt weinig uit, Free Pascal is beschikbaar voor alle bekende en ook een aantal minder bekende besturingssystemen. In dit boek concentreren we ons op de twee meest gebruikte systemen, dat zijn Microsoft Windows en Linux. Maak je echter geen zorgen als je bijvoorbeeld een Apple Macintosh hebt, de details zullen wat verschillen maar je kunt dit boek gewoon gebruiken.

### 1.1.1 Linux

We zullen nu beschrijven hoe je Free Pascal installeert op een Linuxsysteem. Als je Windows gebruikt, blader dan even door naar het volgende kopje, daar beschrijven we hoe je Free Pascal installeert op een computer die Windows draait.

Allereerst raden we je eens aan te kijken of Free Pascal ingebouwd zit in jouw Linuxdistributie. Als dat zo is, kan je het installatieprogramma van je distributie starten, bij de software Free Pascal aanklikken, en je bent klaar. Zit Free Pascal nog niet in jouw distributie geen Free Pascal mee, dan zul je het moeten downloaden en kun je de nu volgende installatieprocedure volgen.

Log eerst in als root op je computer, zodat je alle rechten hebt om het installatieprogramma zijn werk te laten doen. Start vervolgens een terminal. Zoek daarvoor in het start-menu op het scherm maar de "Xterm" of "X-terminal". Je kunt ook op Ctrl+Alt+F1 drukken om op de console in te loggen en daar de terminal te gebruiken.

Controleert eerst of er genoeg ruimte vrij is op je systeem. Dit kun je controleren door een terminal te openen en achter de prompt het "df"-commando in te tikken. Free Pascal wordt geïnstalleerd in /usr. Als je intikt:

```
~> df /usr
```

... dan krijg je een overzicht van de vrije ruimte op de harde schijf waar /usr op staat. Als het aantal kilobytes dat vrij is groter is dan 120000, dan heb je genoeg ruimte vrij.

Download nu Free Pascal met je webbrowser vanaf de website, <http://www.freepascal.org>. Je zult een bestand downloaden dat een naam heeft als fpc-2.0.4.i386-linux.tar, de exacte naam kan licht verschillen afhankelijk van wat de meest recente versie is en of je een 32- of 64-bits uitgave downloadt. Download dit bestand en plaats het in de directory /tmp.

Ga nu weer terug naar je terminal, en verplaats jezelf naar de directory /tmp, door te typen:

```
~> cd /tmp
```

Maak een directory voor de installatiebestanden en ga daar in:

```
~> mkdir fpc-installatie
```

```
~> cd fpc-installatie
```

Pak de installatiebestanden uit:

```
~> tar xvf ../fpc-2.0.4.i386-linux.tar
```

Start vervolgens het installatieprogramma:

```
~> ./install.sh
```

De eerste vraag die gesteld wordt is de zogeheten «install prefix», oftewel de installatiedirectory. Je kunt die het beste laten staan op /usr/local door gewoon op enter te drukken. Vervolgens wordt gevraagd of de tekstmode-IDE geïnstalleerd dient te worden. Bevestig door een "y" te typen. Doe hetzelfde met de FCL, packages, documentatie en demo's.

Heb je niet voldoende ruimte, dan kan je overwegen de documentatie, packages en/of demo's weg te laten. Het weglaten van de IDE of FCL raden we af.



## Installatieprogramma Free Pascal 2.0.4 Linux

Nadat het installatieprogramma zijn werk heeft gedaan kom je terug op de prompt. Free Pascal is nu geïnstalleerd. Je het starten door in een terminal "fp" te typen. Een mooiere schermuitvoer krijg je op de console, als je dat wilt, druk op Ctrl+Alt+F1 om naar de console te schakelen, log daar in en typ "fp". Je schakelt weer terug naar het grafische scherm met Ctrl+Alt+F7.

### 1.1.2 Windows

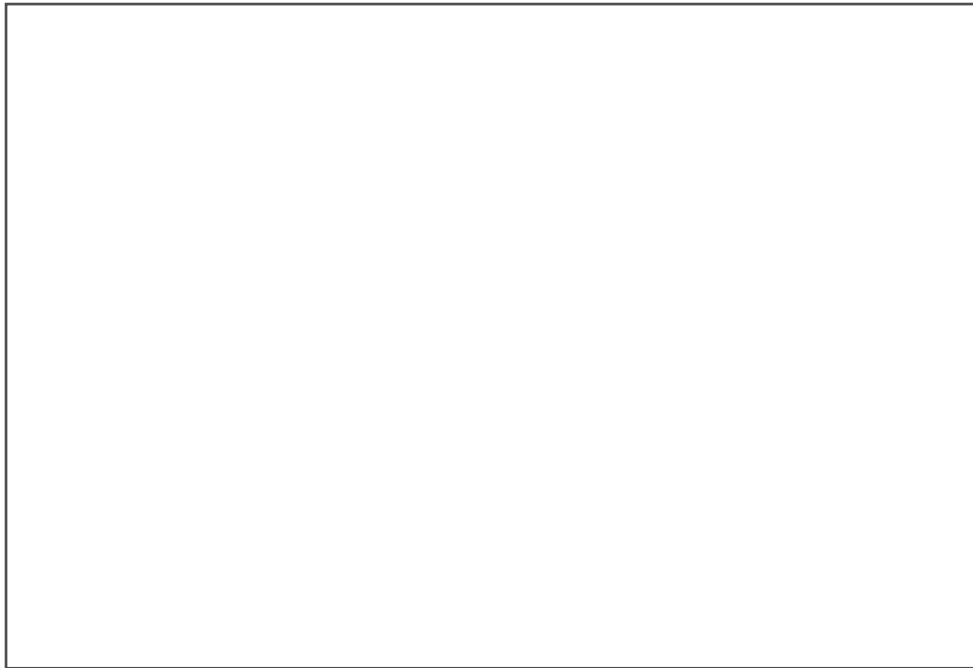
We zullen nu beschrijven hoe je Free Pascal 2.0.4 installeert op een Windowssysteem. De Windowsversie van Free Pascal maakt gebruik van een installatieprogramma. Download om te beginnen Free Pascal van de website, <http://www.freepascal.org>. Na het kiezen van de Windowsversie zal je een bestand downloaden dat een naam heeft als fpc-2.0.4.i386-win32.exe, dat kan afhankelijk van de versie licht verschillen.. Door dit bestand te openen start je het installatieprogramma.

Het installatieprogramma opent met een welkomstboodschap, die je kunt beantwoorden door op «Next» te klikken. Vervolgens vraagt het programma om de installatiedirectory. Je kunt de suggestie `c:\fpc\versienummer` overnemen, maar mocht er op je C:-schijf niet genoeg ruimte zijn, dan kun je dit aanpassen. Als je de directorynaam aanpast, let dan op dat je geen directorynaam kiest waar een spatie in voorkomt. De reden dat je dit niet moet doen is dat één van de hulpprogramma's die Free Pascal gebruikt, de GNU-linker, niet goed met spaties in bestands- en directorynamen overweg kan.

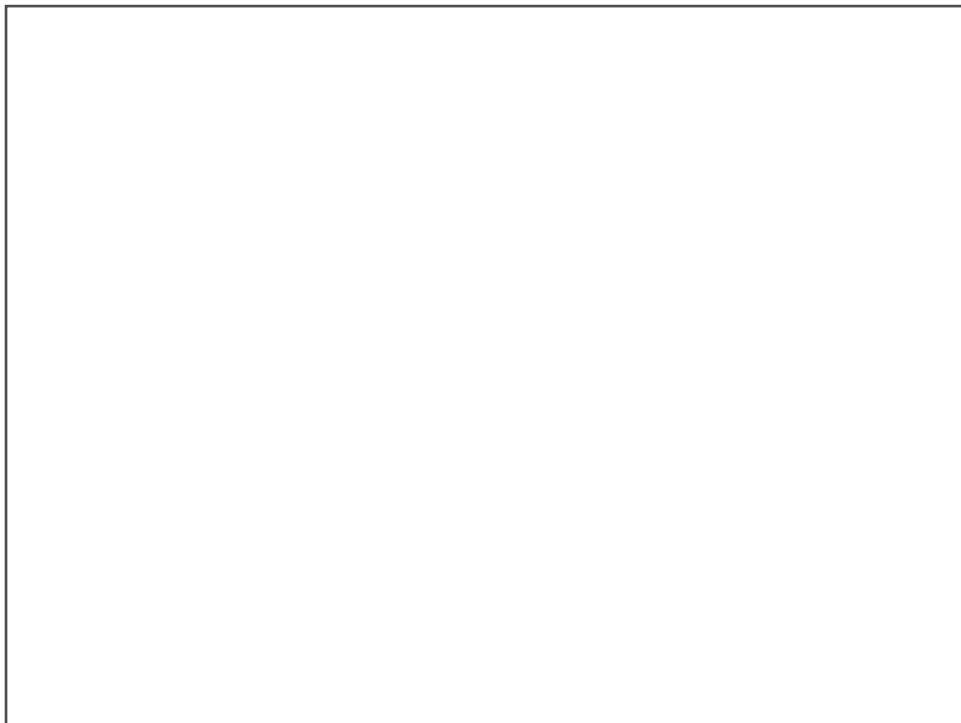


Klik hierna weer op «Next». Het installatieprogramma zal nu vragen wat voor installatie je wilt. De meeste mensen zullen willen kiezen voor een volledige installatie. Mocht je krap in de ruimte zitten, dan zou je kunnen overwegen een aangepaste installatie te doen. Zorg dat je in ieder geval de «Integrated Development Environment» installeert. Nadat je weer op «Next» hebt geklikt wordt een programmamap voor het startmenu voorgesteld. Het voorstel “Free Pascal” is eigenlijk in alle gevallen prima, klikt daarom weer op «Next». In het laatste scherm kun je kiezen welke bestandstypen geassocieerd worden met Free Pascal. Tenzij je nog een Pascalcompiler geïnstalleerd hebt kun je ze standaardinstellingen accepteren. Klik op «Next». Klik vervolgens op «Install» om de installatie te starten.

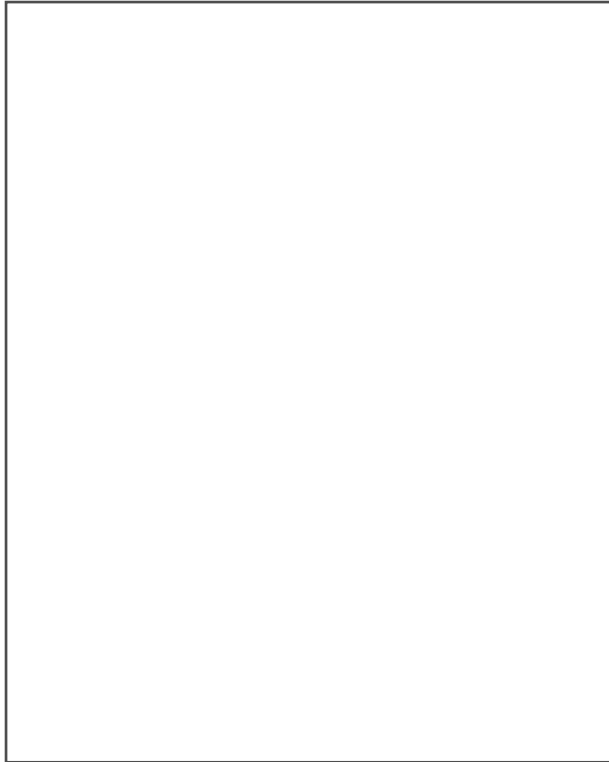
Nadat de installatie voltooid is, moeten we nog één ding doen, namelijk een werkmapije aanmaken waar we de programma's die we gaan schrijven neerzetten. Open daarom de Windowsverkenner en maak bijvoorbeeld op de C:-schijf een directory “programmas” aan.



Om het ons makkelijk te maken, zorgen we dat Free Pascal standaard in deze directory begint. Daartoe zoeken we het op in het startmenu, en kiezen met de rechtermuisknop de “Eigenschappen”.



In het scherm dat zich nu opent vullen we bij “Beginnen in” de locatie van de directory die we net gemaakt hebben in:



Free Pascal is nu klaar voor gebruik.

## 1.2 Wat is een programma?

Nu we de programmeertaal Free Pascal op de schijf geïnstalleerd hebben, is het natuurlijk de bedoeling dat we hiermee zelf programma's leren maken. Het moet dan natuurlijk wel duidelijk zijn wat nu eigenlijk een computerprogramma is.

Een computerprogramma is een verzameling van opeenvolgende instructies die door de centrale verwerkingseenheid van de computer kunnen worden uitgevoerd. Zo'n programma kan communiceren met de buitenwereld door gegevens te vragen vanaf het toetsenbord, of door deze gegevens in te lezen vanaf een informatiedrager zoals bijvoorbeeld de harde schijf. Ook kan een programma informatie toegezonden krijgen via een telefoon-, datalijn of internet. In zo'n geval wordt een modem gebruikt.

Het programma hoeft natuurlijk niet alleen te communiceren door middel van het vragen van gegevens; het programma moet ook informatie kunnen verstrekken. Deze informatie wordt afgebeeld op het scherm van de monitor, naar een printer gestuurd of door middel van een modem op het telefoon- of op een datanet gezet.

Een tekstverwerker is één van de meest bekende toepassingen op de computer. Als we naar zo'n programma kijken dan is die communicatie ook duidelijk te zien. De letters, die samen een tekst vormen, zijn de gegevens waar het programma om vraagt. Deze gegevens worden verstrekt door ze met behulp van een toetsenbord in te tikken. Als het programma eenmaal over deze gegevens beschikt kunnen we het opdracht geven om de tekst naar een printer te sturen. Resultaat is een op papier afgedrukte tekst.

We kunnen zo'n tekst ook bewaren. Als we dit willen, geven we aan de tekst een naam en het programma schrijft de tekst op de harde schijf of op een diskette. De gegevens die onder een bepaalde naam op de schijf zijn opgeborgen noemen we een bestand.

Als we de tekst weer nodig hebben vertellen we met behulp van het toetsenbord aan het programma dat we een tekst willen inlezen. Daarna geven we de naam van het bestand op. Het programma leest nu de eerder opgeborgene tekst weer in en beeldt die vervolgens af op het scherm.

Zoals ik al zei is een programma een verzameling voor de computer uitvoerbare instructies. Deze verzameling is meestal in gedeelten verdeeld. Ieder van die delen is gericht op het uitvoeren van een bepaalde taak.

In het voorbeeld van de tekstverwerker zal de keuze voor het afdrukken van de tekst met zich meebrengen dat het gedeelte van het programma uitgevoerd wordt, dat teksten naar de printer stuurt. De beweging tussen het gedeelte van het programma dat vraagt wat er gebeuren moet en het gaan naar het gedeelte dat teksten naar de printer brengt wordt een sprong genoemd. Binnen een programma kan dus gesprongen worden naar die delen van het programma die voor de uitvoering van de taak van het programma noodzakelijk zijn.

Naast het springen is herhaling binnen een programma erg belangrijk. Als we het voorbeeld van de tekstverwerker aanhouden dan zou je het inlezen van tekst vanuit een bestand onder kunnen brengen in een herhaling. Dit gaat ongeveer als volgt:

1. zoek het opgegeven bestand op de schijf
2. open het bestand en lees de eerste letter
3. lees de volgende letter
4. als het niet de laatste letter is ga dan terug naar regel 3.
5. ...

Zoals we zien wordt in regel vier telkens teruggekeerd naar regel drie als de laatste letter nog niet ingelezen is. Als het wel de laatste letter is dan wordt er doorgegaan met de instructie die op regel vijf staat.

Met regel vier is iets bijzonders aan de hand. In deze regel wordt namelijk een keuze gemaakt op basis van een vergelijking. Eerst wordt er gekeken of het de laatste letter van het bestand is. Als dit niet het geval is, dan wordt er teruggekeerd naar regel drie. Als het echter wel de laatste letter is dan wordt er niet teruggekeerd en wordt gewoon begonnen met de uitvoering van regel vijf. Het maken van dit soort vergelijkingen is in een programma buitengewoon belangrijk.

### **1.3 Computertaal**

Nu we ongeveer weten hoe de structuur van een computerprogramma is wordt het tijd dat we het gaan hebben over computertalen. Een computertaal is net als een gewone taal, een hoeveelheid woorden, die op een bepaalde manier gegroepeerd moeten worden. Hier gaat het om opdrachten, die je aan de computer kunt geven in voor mensen begrijpelijke taal.

Deze opdrachten worden opgeschreven in een door de taal voorgeschreven vorm. Zo'n geschreven programma wordt broncode genoemd. Ook de Engelse term "source code" wordt hiervoor veel gebruikt.

De broncode wordt aangeboden aan een programma dat in staat is de voor mensen leesbare instructies om te zetten naar voor de machine leesbare instructies. Zo'n programma noemen we een compiler. De compiler krijgt dus tekst aangeboden en voert een lijstbestand van instructies in machinecode uit. Dit laatste bestand noemen we de doelcode ofwel objectcode.

Tenslotte wordt de doelcode aangeboden aan een programma dat hiervan een uitvoerbaar programma in

machinecode maakt. Dit programma noemen we een linker.

Bij Free Pascal is de tekstverwerker om teksten in broncode te maken, de compiler om hiervan objectcode te maken en de linker om van deze objectcode een uitvoerbaar programma te maken ondergebracht in één programmeeromgeving. Deze programmeeromgeving wordt de IDE genoemd. Dit is een afkorting van Integrated Development Environment. In gewoon Nederlands betekent dat: tot één geheel gemaakte ontwikkelomgeving. Vanuit deze ontwikkelomgeving kunnen allerlei opdrachten aan de compiler worden meegegeven, kan een helpfunctie worden aangeroepen en kunnen fouten worden opgespoord.

Een programma wordt gemaakt om een bepaalde taak geautomatiseerd uit laten voeren. Dit uitgangspunt heeft tot consequentie dat het probleem waarvoor het programma uitkomst moet bieden glashelder geformuleerd moet zijn. Als het probleem niet helder is dan zal de programmeur voor de verkeerde oplossingen kiezen.

Als het probleem duidelijk is dan zal er een globale oplossing voor het probleem bedacht worden. Deze globale oplossing zal weer uiteengelegd worden in verschillende deeloplossingen, die op hun beurt weer uiteengelegd kunnen worden in deeloplossingen.

In de kleinste deeloplossingen zullen we in staat zijn het probleem stapsgewijs op te lossen. Laten we eens een gewoon huis-tuin-en-keukenprobleem nemen als hoe ga ik naar mijn werk?

Het probleem is:

Welke acties moet ik ondernemen als ik besloten heb met welk soort vervoer ik naar mijn werk ga.

De globale oplossing is:

Als ik geen abonnement voor de bus gekocht heb ga ik bij mooi weer op de fiets en anders ga ik met de auto.

De deeloplossingen die hieruit voortkomen zijn:

- Beschrijving acties voor openbaar vervoer
- Beschrijving acties voor gebruik fiets
- Beschrijving acties voor gebruik auto
- Het maken van een keuze op grond van de probleemstelling

### **Acties openbaar vervoer**

pak je busabonnement  
stop de krant in je zak  
wandelen naar de bushalte  
wacht daar op de bus die je nodig hebt  
stap in  
lees de krant tot je op de plaats van bestemming bent  
stap uit  
wandelen naar je werk

### **Acties gebruik fiets**

haal de fiets uit de schuur  
controleer de banden  
als ze te zacht zijn pomp ze dan op

rijdt langs de fietsroute naar je werk  
berg de fiets op in het fietsenhok

Dit voorbeeld kan natuurlijk nog behoorlijk uitgebreid worden je kunt de verschillende routes nog gaan beschrijven, je kunt beschrijven wat je doet als de bus staakt of als je fiets gestolen is enzovoort.

Waar het hier om gaat is dat een bepaalde manier van denken duidelijk wordt.

Het langs deze weg opzetten van een programma wordt ook wel gestructureerd programmeren genoemd.

Pascal dwingt ons tot een gestructureerde opzet van een programma. Het is zelfs zo dat Professor Niklaus Wirth de programmeertal Pascal speciaal ontwikkeld heeft om aan zijn studenten gestructureerd programmeren bij te brengen.

Een Pascal-programma bestaan uit een hoofdprogramma en procedures en functies.

In ons voorbeeld zou het maken van de keuze het hoofdprogramma zijn en de verschillende acties de procedure.

Als ik het voorbeeld zou gieten in de vorm van een Pascalprogramma dan zou dit er als volgt uitzien:

**PROGRAM VERVOER;**

**PROCEDURE OpenbaarVervoer;**

BEGIN

pak je busabonnement;  
stop de krant in je zak;  
wandelen naar de bushalte;  
wacht daar op de bus die je nodig hebt;  
stap in;  
lees de krant tot je op de plaats van bestemming bent;  
stap uit;  
wandelen naar je werk

END;

**PROCEDURE GebruikFiets**

BEGIN

haal de fiets uit de schuur;  
controleer de banden;  
als ze te zacht zijn pomp ze dan op;  
rijdt langs de fietsroute naar je werk;  
berg de fiets op in het fietsenhok

END;

**PROCEDURE GebruikAuto**

BEGIN

wandelen naar de parkeerplaats waar je auto staat;  
controleer brandstof en olie;  
stap in;  
start de auto;  
rijdt langs de autoroute naar je werk;  
zoek een parkeerplaats;



```
    stap uit;  
    wandel naar je werk  
END;
```

```
BEGIN  
  Als ik een busabonnement heb dan  
    OpenbaarVervoer  
  anders als het regent of als er vandaag regen verwacht wordt dan  
    GebruikAuto  
  anders  
    GebruikFiets  
END.
```

Het zal duidelijk zijn dat dit geen echt Pascalprogramma is, maar dat het voorbeeld over vervoer gegoten is in de vorm van een Pascalprogramma. De compiler zal dit natuurlijk gewoon weigeren.

Toch kun je uit het voorbeeld een aantal dingen leren. Ik had het over een hoofdprogramma en een aantal procedures. Ik heb van de keuzen het hoofdprogramma gemaakt en de verschillende acties zijn procedures geworden.

De onderste negen regels van het voorbeeld vormen samen het hoofdprogramma en start met BEGIN. Daarna komen de verschillende opdrachten. Het hoofdprogramma eindigt met END en een punt. Op deze manier begint en eindigt ieder Pascal programma.

Ook de procedures starten met BEGIN en eindigen met END alleen wordt hier de END niet met een punt afgesloten maar met een punt-komma. Ook iedere opdracht wordt

beëindigd met een punt-komma behalve als onmiddellijk achter de opdracht een END staat. Bijvoorbeeld de laatste regel van de Procedure GebruikAuto gaat vooraf aan een END en wordt dus niet afgesloten met een punt-komma. Als je op deze regel wel een punt-komma zet dan zal er geen foutmelding gegeven worden. De compiler accepteert het. Hetgeen er gebeurt is dat de compiler een opdracht voor de computer maakt waarin staat dat er niets moet gebeuren. Er wordt een lege opdrachtregel aangemaakt en dat is verspilling.

Bovenaan ieder programma staat PROGRAM, gevolgd door de naam die je aan het programma geeft en een punt-komma. De uitvoering van het programma begint op de eerste regel van het hoofdprogramma. Daar wordt onderzocht of er een busabonnement gekocht is. Indien dit bevestigd wordt, dan wordt er gesprongen naar de procedure OpenbaarVervoer en worden de acties van deze procedure uitgevoerd. Is er geen busabonnement EN het regent OF er wordt vandaag regen verwacht, dan wordt er gesprongen naar GebruikAuto. Als het voorgaande allemaal niet opgaat dan is het dus weer om te fietsen en wordt de procedure GebruikFiets uitgevoerd.

Onder PROGRAM kan eventueel gezet worden welke delen van de programmabibliotheek gebruikt zullen worden. Free Pascal levert een uitgebreide programmabibliotheek met het pakket mee. De zeer uitgebreide opdrachtenset van de taal is ondergebracht in een aantal onderdelen. Deze onderdelen worden units genoemd. Er is bijvoorbeeld een unit DOS. In deze unit zijn alle Free Pascal opdrachten opgeslagen om DOS opdrachten uit te voeren. Zo is er ook een unit CRT. Deze unit bevat de opdrachtenset om met het tekstscherf te werken. Die delen van de programmabibliotheek, die je nodig hebt bij het vervaardigen van je programma, die gebruik je. De delen die je niet nodig hebt gebruik je niet.

Het gebruik van de programmabibliotheek is zeer eenvoudig. We zetten op de tweede regel van het programma USES <naam unit> en we kunnen de procedures en functies van de unit gebruiken. Free Pascal

levert een zeer groot aantal units mee. Bij het pakket worden onder andere de volgende units meegeleverd:

Baseunix:	Een unit om toegang te krijgen tot systeemfuncties van Unixbesturingssystemen.
Classes:	Een unit met enkele basisklassen voor het classes-OOP-model.
Crt:	Procedures voor eenvoudige presentatie op het scherm en geluid.
Dos :	Een unit om toegang te krijgen tot functies van het besturingssysteem.
Graph:	Voor de grafische functies en procedures.
Keyboard:	Een unit voor geavanceerde toetsenbordinput.
Math:	Een unit met wiskundige functies.
Matrix:	Een unit voor het werken met 2-, 3- en 4-dimensionale matrixen.
Mouse:	Een unit om toegang te krijgen tot de muis.
Objects:	Een unit met enkele basisobjecten voor het object-OOP-model.
Printer:	Stuurt teksten naar de printer.
System:	Levert de procedures en functies voor de uitvoering van een programma
Sysutils:	Een unit om toegang te krijgen tot functies van het besturingssysteem, stringfuncties en datum- en tijdfuncties.
Ucomplex:	Een unit om te werken met complexe getallen.
Unix:	Een unit om toegang te krijgen tot niet-systeemfuncties van Unixbesturingssystemen.
Video:	Een unit voor directe toegang tot het beeldscherm.
Windows:	Een unit die de Win32-API bevat.

Het bovenstaande is een zeer beknopt overzicht. Zo worden er ook nog meer geleverd:

- Free Vision-units - Free Vision is een kloon van het bekende Turbo Vision, een programmeersysteem voor gevorderde programmeurs. De behandeling van Free/Turbo Vision valt buiten het kader van dit boek. Belangstellenden verwijs ik naar het boek "Turbo Vision" in de praktijk, dat van mijn hand verschenen is bij de uitgeverij Oets.
- FCL-units - De FCL is een objectgeoriënteerd systeem voor programmering van databases, gecomprimeerde bestanden, grafische bestanden, XML-bestanden en meer.
- OpenGL-units - Voor het maken van grafische 2D- en 3D applicaties en computerspellen.
- GTK-units - Voor het maken van grafische applicaties met de GTK- of GTK2-toolkit.
- Hash-units - Voor het berekenen hashwaarden volgens enkele veelgebruikte algoritmen.
- Pajpeg - Units voor het werken met JPEG-bestanden.
- Pajlib - Units voor het werken met zip- en gz-bestanden.
- Meer....

## Opgaven:

1. Werk op dezelfde manier als gedaan is met het pseudoprogramma Vervoer een pseudoprogramma uit waarbij je uitwerkt op welke manier je in de badkamer kiest voor het gebruik van de wastafel, de douche en het ligbad. Als uitgangspunt neem ik, dat ik mezelf eens per dag een uitgebreide wasbeurt geef. Voor zo'n uitgebreide wasbeurt gebruik ik of de douche of het ligbad. Als ik meer dan drie kwartier de tijd heb dan neem ik een ligbad anders neem ik de douche. Als ik mijn haar moet wassen gebruik ik altijd de douche. Bij tussentijdse reinigingsbeurten gebruik ik de wastafel.