|  |  |  |
| --- | --- | --- |
| **ACTIES OP HET SCHERM** | **VOICE-OVER** | **DUUR** |
|  | Error handling is simple, right? Your application writes a message to the screen and exits. Essentially, yes, but if you want that error message to make sense to the user of your application, it requires a lot more work, and especially a switch of perspective. As a programmer, you have to view your application from the point of view of a user, as if it were a black box. Error messages that don’t make sense are quite frustrating. |  |
| 1. Show random\_mod\_01.f90 | 1. Show allocatables |  |
| 1. Show main\_01.f90 | 1. Show error handling for command line arguments |  |
| 1. Compile and run | 1. Discuss runtime error |  |
| 1. Show random\_mod\_02.f90 | 1. Discuss stat argument |  |
| 1. Compile and run | 1. Discuss error message |  |
| 1. Show random\_mod\_03.f90 | 1. Discuss optional ierr |  |
| 1. Show main\_02.f90 | 1. Discuss handling error in main context |  |
| 1. Show main\_03.f90 | 1. Discuss negative arguments |  |
| 1. Compile and run |  |  |
|  | Errors can be handled at various levels in your application. Handling them at the level that makes most sense to the user of your application will certainly help those users, but it requires some work and imagination on your part. |  |
| **TOTALE DUUR** | | *Maak je screencast niet langer dan ca. 6 minuten.* |