**BANK VERRICHTINGEN**

We schrijven een programma dat al je verwerkingen van je bankrekening bijhoudt

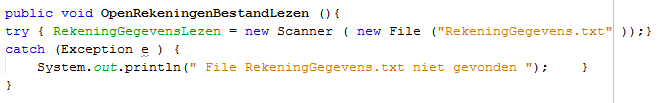
1. Werken met Files

Om gegevens van files te lezen : SCANNER

Om gegevens naar files te schrijven : FILEWRITER

Belangrijke Opmerking : om Fatal Errors te vermijden indien file niet bestaat of corrupt is, gebruiken we EXCEPTIONS

Voorbeeld bij lezen



1. Gebruik van Array

We gebruiken ArrayList met dynamisch aantal elementen( dus onbeperkt uitbreidbaar )



Opmerking : indien je een Array wil met statische aantal elementen ( vb 3 elementen)



1. Zoek en Sorteer Algoritmes

Zoeken naar

Hoogste / Laagste waarde

Aantal keren dat een waarde voorkomt

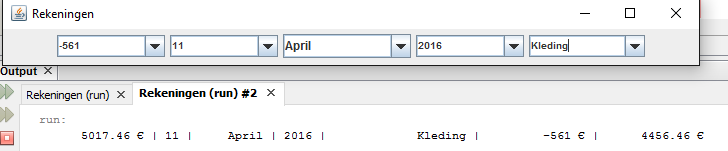
Sorteer

Van Laag naar Hoog

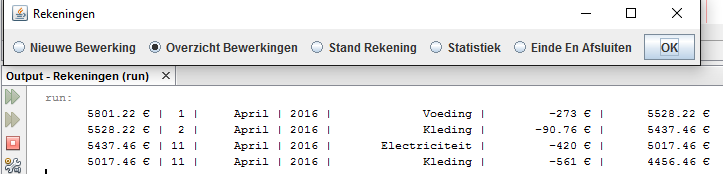
**OVERZICHT REKENINGEN**



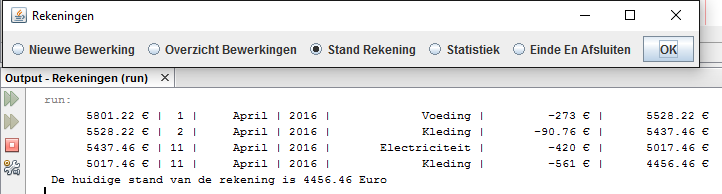
**NIEUWE BEWERKING**



**OVERZICHT BEWERKINGEN**



**STAND REKENING**



1. **RadioButtons en Comboxen**

**RadioButtons voor OVERZICHT REKENINGEN**



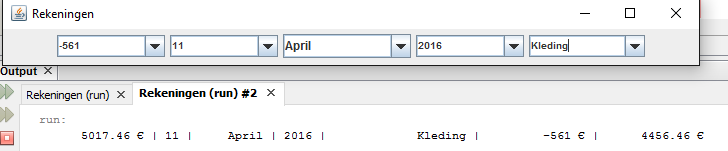
Reeds beschikbaar : NIEUWE BEWERKING

TOE TE VOEGEN : Overzicht Bewerkingen , Stand Rekening , Statistiek

**RadioButtons voor STATISTIEK**

Nog aan te maken

**Comboxen voor NIEUWE BEWERKING**



Reeds aanwezig : Bedrag en Beschrijving

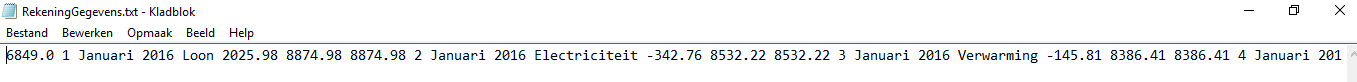
Nog bij te voegen : Datum

**Comboxen voor STATISTIEK**

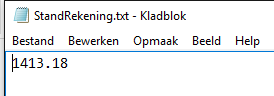
Nog aan te maken

1. **Input / Output naar file**

File **RekeningGegevens.txt** voor Lijst van Bewerkingen ( gesorteerd van eerste datum naar laatste datum)



File **StandRekening.txt** voor Huidige Stand Rekeningen ( bevat 1 getal , wordt telkens overgeschreven )



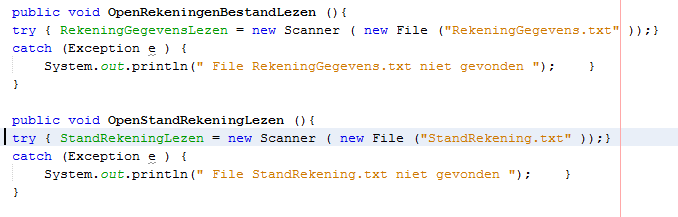
Om te lezen 🡺 **SCANNER**

Om te Schrijven 🡺 **FILEWRITER**

GEVOLG : je moet elke file 2 maal openen ( voor SCANNER en voor FILEWRITER)

* 1. **OPENEN OM TE LEZEN**

**LEZEN = SCANNER**



* 1. **OPENEN OM TE SCHRIJVEN**

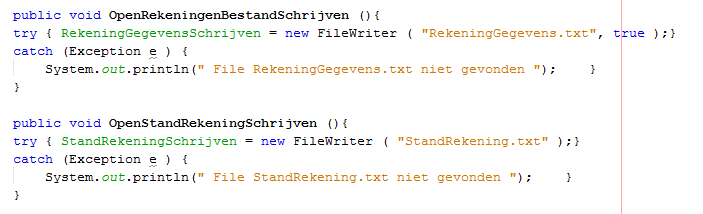
**SCHRIJVEN = FILEWRITER**

Een file wordt normaal altijd overschreven ( dus bestaande gegevens worden verwijderd en overschreven ).

Voorbeeld de file StandRekening

Als je niet wilt overschrijven , maar je wilt toevoegen ( append ) , moet je bij het openen de waarde ‘true’ meegeven.

Voorbeeld de file RekeningGegevens



* 1. **SCHRIJVEN NAAR FILE**

Gegevens in de file worden gescheiden door een delimiter , die vastgelegd is op “ “ ( blanco)

Indien je een andere delimiter wilt ( vb een “,” of een “\* “) , gebruik



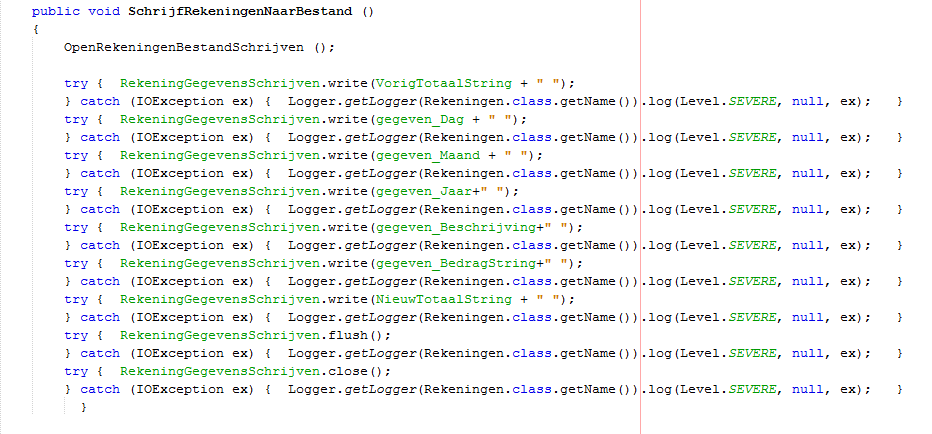
Vergeet NIET om deze delimiter mee weg te schrijven in je file , door + “ “ te gebruiken

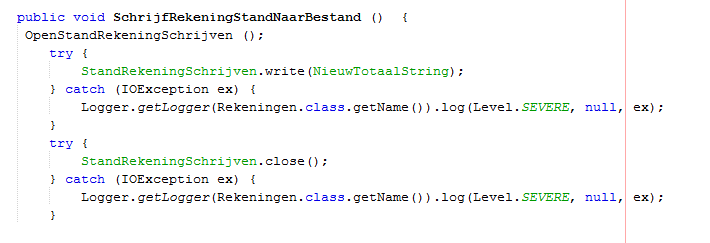
In deze file schrijven we alle gegevens weg in een rij , zonder nextLine te gebruiken

Hou daar rekening mee tijdens het lezen

Let op voor de try-catch

Op het einde wordt de file gesloten ( close)





* 1. **LEZEN VAN FILE**

**LEZEN = SCANNER**

Gegevens in de file worden gescheiden door een delimiter , die vastgelegd is op “ “ ( blanco)

Indien je een andere delimiter wilt ( vb een “,” of een “\* “) , gebruik



Om te springen naar het volgende element in je file , gebruik hasNext () en next()

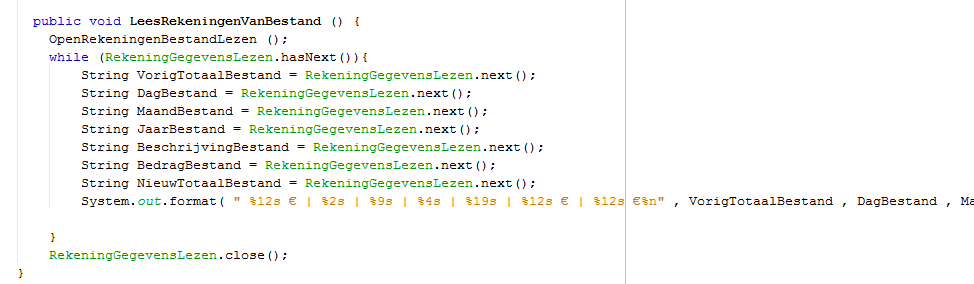
Om de gelezen gegevens te tonen op het console scherm , gebruik

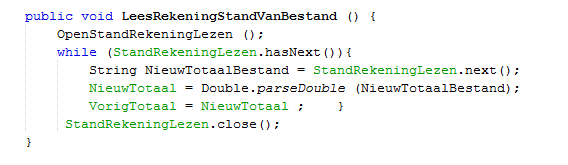
**System.out.format( " %12s € | %2s | %9s | %4s | %19s | %12s € | %12s €%n" , VorigTotaalBestand , DagBestand , MaandBestand , JaarBestand , BeschrijvingBestand , BedragBestand , NieuwTotaalBestand);**

**%12s : Eerste parameter ( VorigTotaalBestand ) te tonen als STRING in een veld van 12 tekens**

**%2s : Tweede parameter ( DagBestand ) te tonen als STRING in een veld van 2 tekens**

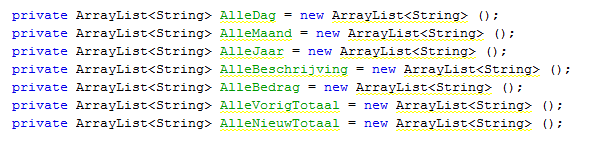
**etc**



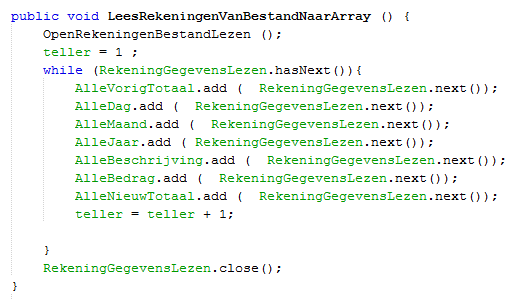


1. **Arraylist**

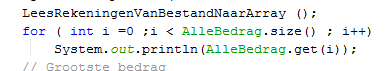
**Definitie**



**Invullen van ArrayList vanuit file**



**Lezen van gegevens van Arraylist**



1. **Statistiek : Sorteren , maximum etc**

**TO BE ADDED**