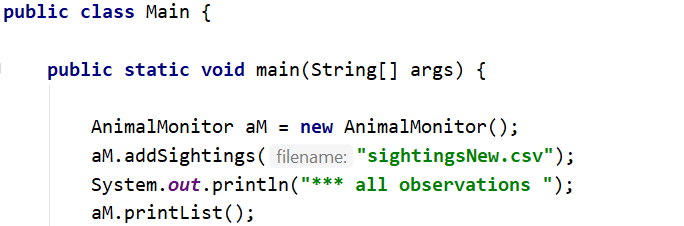
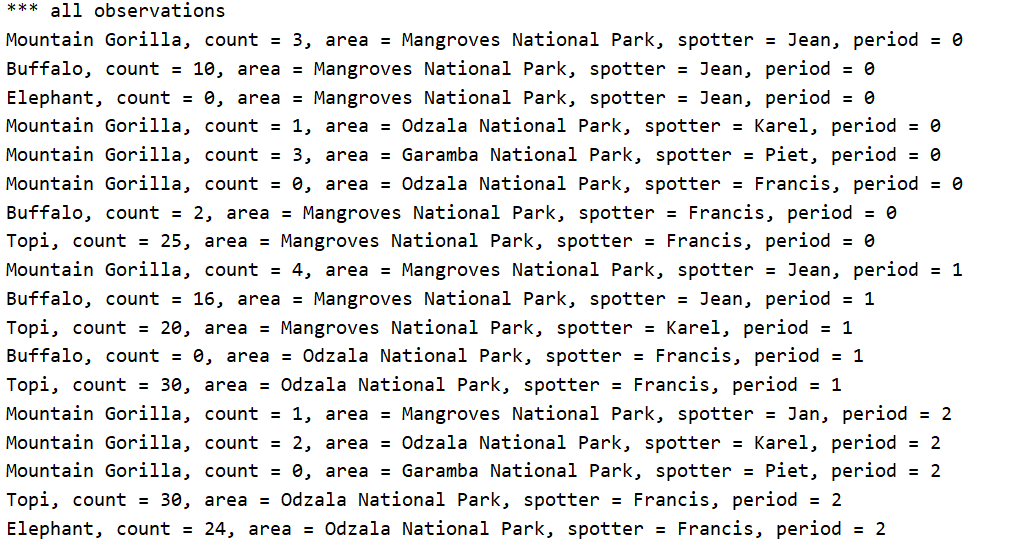
# Opgave 8 Java 8

Omdat we de file SightingsNew moesten inlezen moesten er een paar aanpassingen gebeuren aan het programma, omdat we voor de plaats nu een String inlezen i.p.v. een integer. Nadat we dit hadden aangepast gingen we kijken of we alles konden afdrukken.

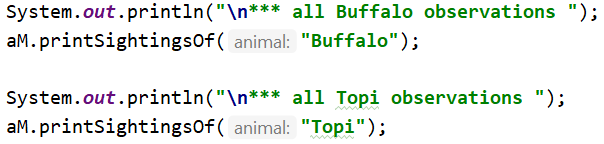


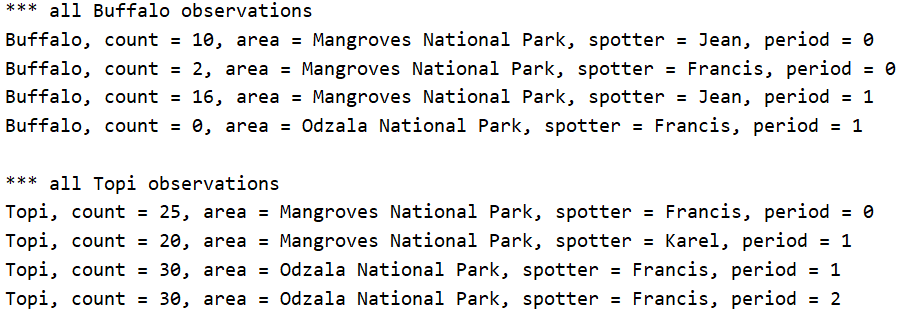
Dit geeft een hele hoop informatie terug



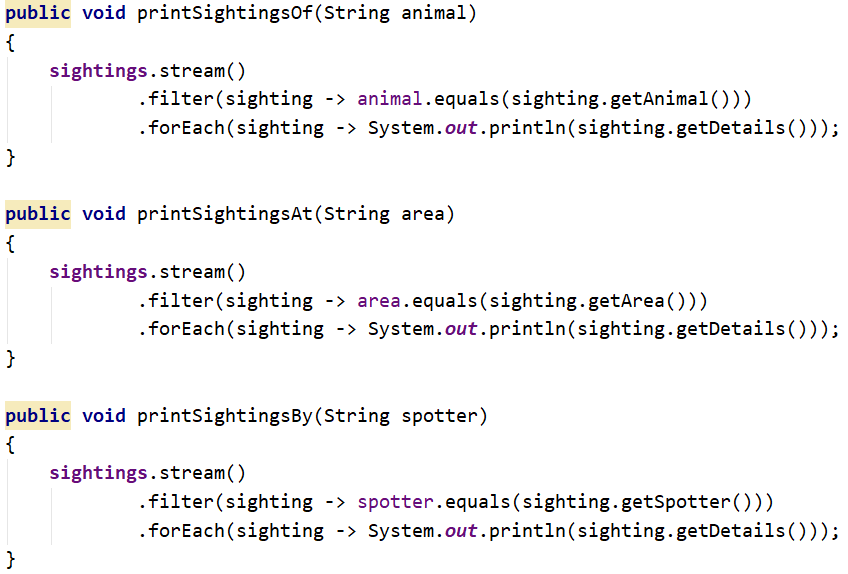
# Filtering

We hebben al een filter methode gekregen genaamd printSightingsOf, hiermee kunnen we een dier als parameter meegeven en krijgen we alle observaties van dat dier terug.



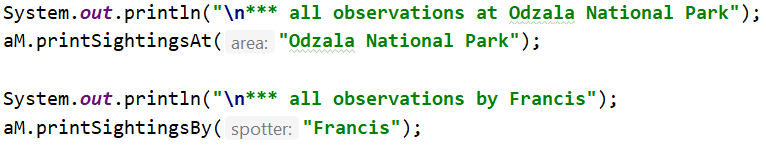


Op deze methode maken we ook verschillende variaties. Zo hebben we één waarop we filteren op de plaats waar ze zijn gespot (printSightingsAt) en een andere die filter op de persoon die het dier heeft gespot. De methodes zien er als volgt uit:

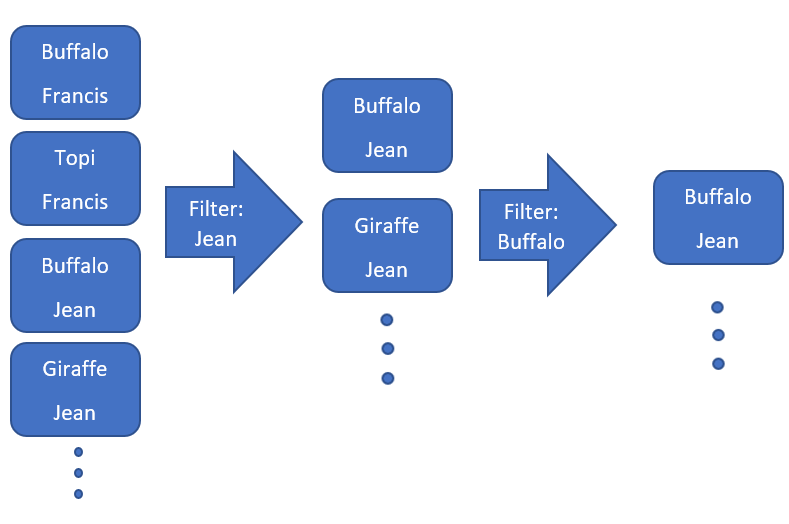


We zien dat de methode bestaat uit een filter die gaat kijken of de string van de meegegven parameter gelijk is als die van de gegevens in de stream. Na het filteren worden deze met een for each allemaal afgedrukt. De info die wordt afgedrukt krijgen we uit de method getDetails, die alle onfo achter elkaar zet.

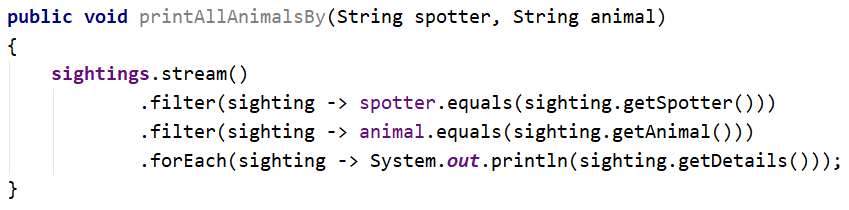
We testen deze methodes uit in de main.



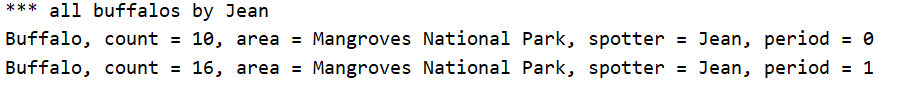
We kunnen ook de gefilterde stream nog eens filteren. In dit voorbeeld filteren we eerst op de spotter Jean en daarna op het dier Buffalo. Visueel ziet dit er zo uit.



De methode is niet veel verschillend t.o.v. de vorige, hier komt er enkel een extra filter achter de vorige.

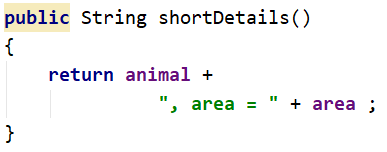


Als we een voorbeeld van hierboven uittesten krijgen we:

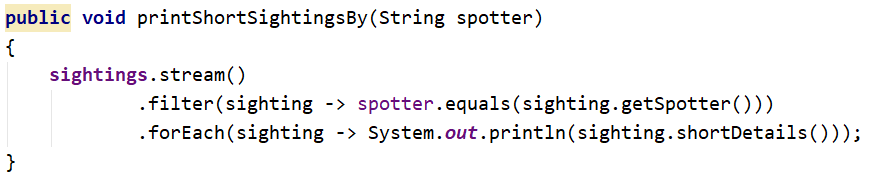


# Mapping

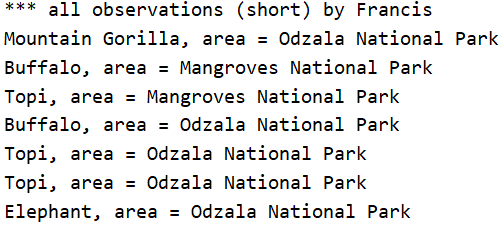
De opdracht hier is het maken van de methode shortDetails. Dit geeft in plaats van alle info enkel de plaats terug.

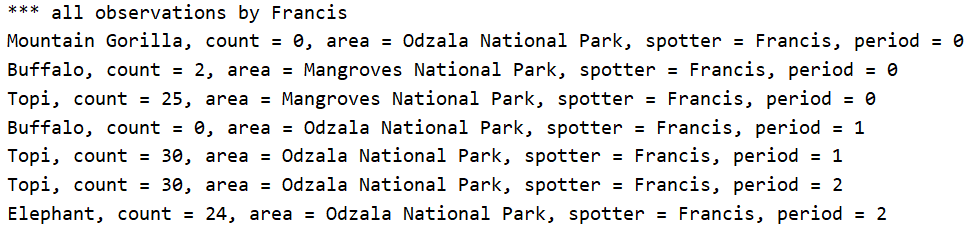


We kunnen nu de vorige methode printSightingBy aanpassen zodat het enkel de area terugkrijgen i.p.v. alle info.



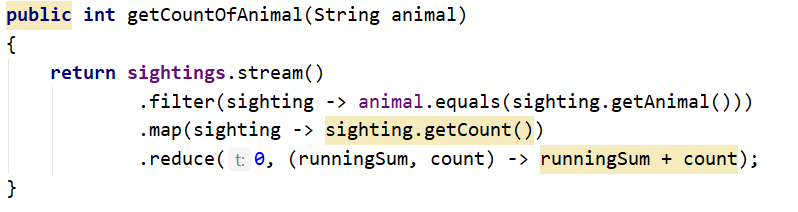
Hieronder kunnen we het verschil zien tussen de 2 methodes. Als we bijvoorbeeld enkel de area willen weten is dit wel overzichtelijker.





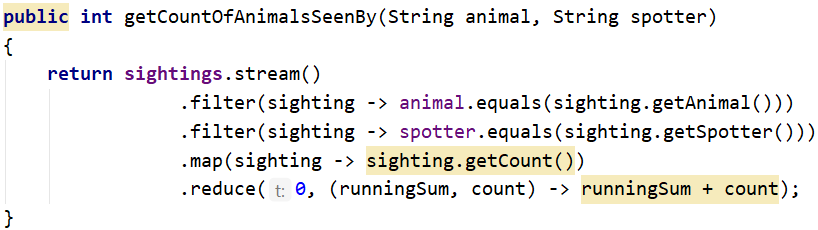
# Reducing

De eerste methode die we hier ontwerpen is om te tellen hoe vaak een bepaald dier is gespot.

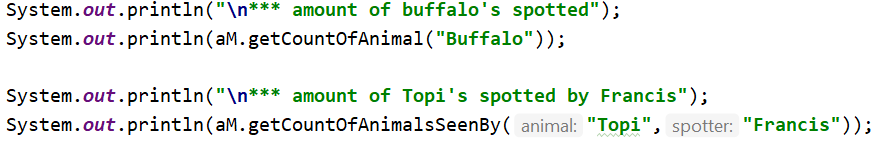


We tellen gewoon alle aantallen op, maar doen eerst een filter op dieren zodat we enkel het meegegeven dier tellen.

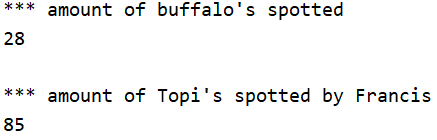
We kunnen ook zorgen dat een bepaald dier enkel wordt geteld als het gespot is door een bepaald persoon. Dit is zoals het vorige maar dan met een extra filter.



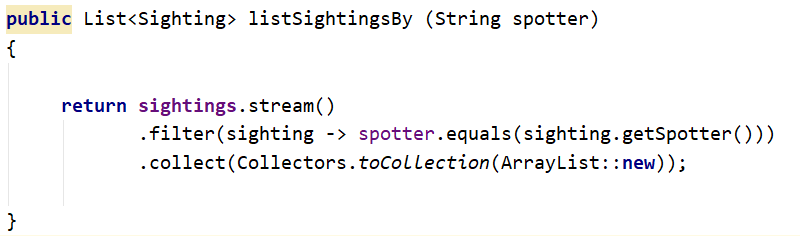
In de main ziet dit er zo uit:



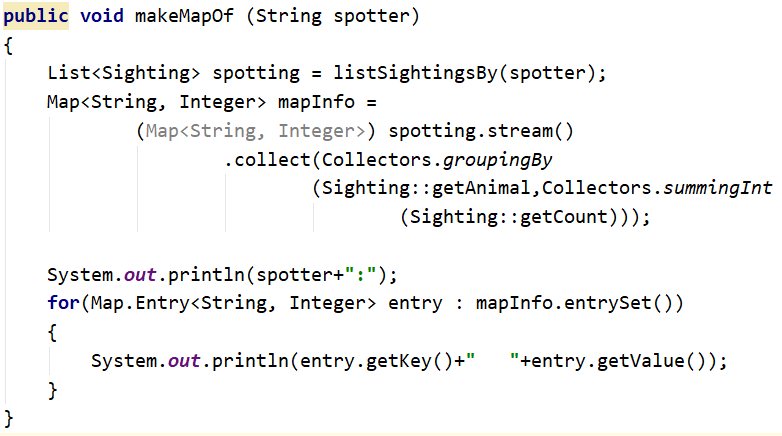
Met als resultaten:



De volgende opdracht was een ArrayList maken met de gegevens in van een bepaalde spotter.



De laaste opdracht was de gegevens van een bepaalde spotter in een Map zetten, de methode ziet er zo uit:



In het bovenste deel wordt er een map opgemaakt die het dier als Key opslaat en een aantal keren gespot als Value. De aantal worden ook opgeteld bij meerdere spottingen. In het tweede gedeelde wordt elke entry van die map afgedrukt.

In de main geeft dit een mooi overzicht:

