

In deze vijfde week werk je voor het laatst aan ons xonix-spel. Je implementeert geluid (sound) en maakt het programma helemaal af.

Ter voorbereiding op het project maak je nog twee kleine opgaven om te oefenen met streams, maps en filters.

### Opdracht 1.

In het xonix-spel pas je geluidssamples toe, waarmee het spel leuker wordt om te spelen. Gebruik de voorbeelden uit het college hiervoor. Maak daarbij gebruik van een enum-klasse. Als in jouw programma alle relevante gebeurtenissen door een 'command' worden afgehandeld, dan hoef je de geluiden alleen in de controller te gebruiken. Mocht dat bij jouw implementatie niet het geval zijn, probeer het programma dan zo te wijzigen, dat je alle geluiden oproept in de controller.

### Opdracht 2.

Schrijf een klein programma in Java, waarin je streams gebruikt. Lees een bestand met data in, bewerk die en schrijf het resultaat weer weg in een bestand.

Voor deze opdracht is een file met data voor je beschikbaar (*data.txt*). Het gaat om de getallen in dit bestand. Lees de getallen in via een *stream* met

```
java.nio.file.Files.lines (java.nio.file.Paths.get (filename))
```

Hierachter kun je *filter*, *map*, etc. gebruiken.

De eerste regels bevatten commentaar. Die beginnen met een '#'. Denk erom dat regels ook leeg kunnen zijn. De commentaarregels en de lege regels verwijder je met een eerste *filter*.

Zet daarna de strings per regel om in een *double*. Dit doe je door een *map* met gebruikmaking van

```
java.lang.Double.parseDouble (string)
```

Selecteer vervolgens alle getallen die groter zijn dan -2,0 en kleiner zijn dan 3,0. Schrijf deze getallen weg in een nieuw bestand (*out.txt*). Gebruik daarvoor *forEach* en *pw::println*, waarbij *pw* de *PrintWriter* is.

Maak bij *filters* en *maps* gebruik van reguliere expressies.

### Opdracht 3.

Schrijf ook voor deze opdracht een klein Java-programma, waarin je gebruik maakt van *streams*. Ook voor deze opdracht is een bestand beschikbaar, in dit geval met telefoonnummers (*phone.txt*). Deze telefoonnummers moeten in het goede *format* worden gezet.

1. Alle telefoonnummers zijn Nederlandse nummers.
2. Sommige beginnen met een +, dat is een algemeen teken voor het internationale toegangsnummer, in Nederland is dat 00.
3. Andere beginnen met 00. Dat is dat internationale toegangsnummer.
4. Na een + of een 00 volgt het landnummer. Dat moet altijd 31 zijn, omdat in het bestand alleen telefoonnummers voor Nederland zijn opgenomen.
5. Na het landnummer 31 moeten nog precies negen cijfers volgen.
6. Alle andere nummers moeten met een enkele 0 beginnen en in totaal tien cijfers bevatten.
7. De telefoonnummers kunnen ook nog streepjes (-) en spaties bevatten. Die moeten alle verwijderd worden.

Schrijf de telefoonnummers weg in een nieuw bestand als Nederlandse telefoonnummers. Elk nummer begint dus met een enkele *0* en bevat in totaal tien cijfers. Als een telefoonnummer niet in het correcte *format* kan worden geschreven, verwijder die dan. Maak ook bij deze opdracht bij *filters* en *maps* van reguliere expressies.

**Tot slot**

De beide bestanden *data.txt* en *phone.txt* staan op Blackboard onder Opdrachten. Dit zijn de laatste weekopdrachten. Hopelijk heb je de opdrachten met plezier uitgewerkt en heb je er veel van geleerd.

De deadline voor de opdrachten van deze vijfde en laatste week is vrijdag 23 december 2016. Je kunt je oplossing laten goedkeuren door de practicumdocent en uploaden via BlackBoard. De weggeschreven bestanden hoeven niet te worden ingeleverd.