Java Advanced – Week 4: Lambda expressies

GitHub: https://github.com/PXLJavaAdvanced1819/week4-lambda-expressies

Opdracht 1

De klasse **NumberMachine** bevat een lijst van integers die verwerkt moeten worden. De klasse heeft ook een methode *processNumbers*, die de lijst van integers zal gaan inspecteren en afhankelijk hiervan een selectie van getallen zal samenzetten in een String waarde (met telkens een streepje er tussen: bv. *1-2-3*), die daarna als return waarde wordt gegeven. De manier waarop de integers verwerkt worden, wordt bepaald door een object dat de interface NumberFilter implementeert.

Schrijf de functionele interface NumberFilter met als enige methode ***boolean check(int number)***.

Maak een klasse ***NumberSelector*** met als member variabele een NumberMachine object. Dit object wordt aan de constructor meegegeven en daar ook in de member variabele opgeslagen. Definieer in deze klasse de methode ***showEvenNumbers().*** Zorg dat deze methode de methode *processNumbers* van het NumberMachine object oproept en de NumberFilter op die manier implementeert zodat enkel even getallen afgeprint zullen worden. Doe dit door gebruik te maken van een anonieme geneste klasse.

Test nadien je code met de EvenNumbersTest.

Maak nu een tweede methode: ***showNumbersAbove()***, eveneens in de NumberSelector klasse. Deze methode krijgt een integer als parameter mee en zal opnieuw de *processNumbers* methode gebruiken, maar deze keer om enkel getallen uit te printen die boven het meegegeven getal liggen. Definieer de instantie van NumberFilter met behulp van een **lambda expressie**.

Test nadien je code met de NumbersAboveTest.

Maak tenslotte een derde methode ***printHexNumbers()***. Deze methode moet de integers printen in hexadecimale notatie, die steeds bestaat uit 4 karakters. (bv. 46 => 2E) Hier kan je geen gebruik maken maken van de NumberFilter, omdat deze een boolean moet returnen. Maak voor deze oefening een methode ***convertNumbers*** in NumberMachine, die een standaard functionele interface gebruikt die geschikt is voor deze methode. De methode geeft de hexadecimale waarden in een String als returnwaarde terug. De waarden zijn opnieuw gescheiden door een streepje -. Gebruik tenslotte de methode *convertNumbers* in combinatie met een lambda expressie om het gewenste resultaat te bekomen.

Test je code opnieuw met de HexNotationTest.

Opdracht 2

De klasse **VideoGame** is reeds gegeven, kijk kort even welke data een object van deze klasse kan opslaan.

Creëer nu een **GameCollection** klasse, met een ArrayList van VideoGames als member variabelen. Voorzie een *addGame* methode die een VideoGame object als argument krijgt en dit object gaat toevoegen aan de collectie.

Voeg nu een methode ***selectGames*** toe aan de GameCollection klasse. Deze methode krijgt als argument een object dat één of andere filter implementeert. Zoek zelf in je cursus of op internet welke standaard functionele interface je hier best voor kan gebruiken. De methode *selectGames* zal door de lijst met games in de collectie lopen en alle VideoGames die voldoen aan de eisen van de meegegeven filter, toevoegen aan een nieuwe ArrayList van VideoGames. Deze ArrayList is het resultaat en wordt teruggegeven als return waarde door deze methode.

Maak tenslotte een klasse **GameBrowser**. Deze klasse heeft als member variabele een GameCollection object. Deze variabele wordt gezet in de constructor (meegegeven als argument).

Maak een methode ***showGamesForSearch*** in GameBrowser. Deze methode krijgt een String *search* als parameter en gaat daarmee de games uit de collectie filteren die deze String in hun naam bevatten. Dit doe je door een gepaste implementatie van het filter object te voorzien en deze mee te geven aan de *selectGames* methode van GameCollection. Doe dit in deze methode met behulp van een **anonieme geneste klasse**.
Geef nadien de resulterende ArrayList van VideoGames terug als return waarde.

Je kan de methode controleren met de GamesSearchTest.

Schrijf een methode ***showFreeGames*** in deze klasse. Roep daarvoor opnieuw de methode *selectGames* uit GameCollection aan en voorzie een gepaste implementatie van de filter, maar deze keer aan de hand van een **lambda expressie**. Geef nadien de resulterende ArrayList van VideoGames terug als return waarde.

Controleren kan je met de FreeGamesTest.

Voeg nadien een methode toe met de naam ***showGamesInGenre***. Deze heeft een String *genre* als parameter en gaat daarmee alle games in de collectie selecteren die dat genre bevatten. Verder werkt de methode analoog aan de methode *showFreeGames*. (werk opnieuw met een lambda expressie)

Check of alles naar behoren werkt met de GamesGenreTest.