

## Java Oefeningen – Reeks 2 (extra)

### Vraag 5

1. Als uitbreiding van SocialNet gaan we de mogelijkheid voorzien om comments te maken. Maak hiervoor een **Comment**-klasse. Comments hebben een bericht, de gebruikersnaam (**String**) van diegene die de comment gepost heeft, het tijdstip dat het verstuurd werd (**int**) en een List van **Comments** die als replies dienen.
2. Voorzie de volgende methodes in **Comment**:
  - a. Een constructor die het bericht, de gebruikersnaam en het tijdstip (**int**) initialiseert. Het tijdstip wordt in minuten beschreven (bv. 672 → 11u12).
  - b. **reply** met als input een **Comment** die wordt toegevoegd aan de lijst van replies.
3. Maak een methode die een comment en al zijn replies correct uitprint, zoals op het voorbeeld hieronder weergegeven (Tip: recursie komt hier goed van pas). Je kan tabs uitprinten met de String "\t". (Bv: "A\t\tB" print "A" uit, gevolgd door twee tabs, en dan "B").

```
Tom zei om 9u50:  
De meeting gaat morgen door om 14u.
```

```
Jan zei om 10u11:  
OK
```

```
Sander zei om 10u15:  
Ik kan er pas om 14u30 zijn.
```

```
Tom zei om 10u33:  
Geen probleem.
```

```
Katrien zei om 11u21:  
In orde, tot dan.
```

4. Voeg een **Comment** als attribuut toe aan de klasse **Group**. Test de klasse uit in main().

## Vraag 6

Maak de methode **telsequentie**, met als invoer een List<Integer> **lijst** en een int **n**, die als returnwaarde een List<Integer> teruggeeft. Gegeven een lijst van (strict positieve) getallen, zal deze opeenvolgende getallen “tellen”, en met deze getallen een nieuwe lijst vormen.

Bijvoorbeeld:

De lijst: [1, 1, 2, 1, 3, 3, 3, 1, 3]

wordt beschreven door: 2x 1, 1x 2, 1x 1, 3x 3, 1x 1, 1x 3. De nieuwe lijst wordt dan:

[2, 1, 1, 2, 1, 1, 3, 3, 1, 1, 1, 3]

Herhaal dit **n** keer en geef het dan als outputwaarde teurg. Vb: als **lijst** = [1] en **n** = 6, krijg je:

[1]

[1, 1]

[2, 1]

[1, 2, 1, 1]

[1, 1, 1, 2, 2, 1]

[3, 1, 2, 2, 1, 1] → uitvoer

## Vraag 7

Morsecode is een communicatietechniek die symbolen encodeert als een opeenvolging van korte en lange signalen (bv. bieptonen, lichtsignalen, enz.). Een kort signaal wordt voorgesteld door een punt, een lang signaal als een streepje. Elk teken heeft zijn unieke voorstelling, bv: 'A' als .- , 'X' als -.-, '?' als ..-..

Het zinnetje "Hello world" wordt in Morse: .... . .-.. .-.. --- .-- --- .- .-.. -..

Punten en streepjes kunnen op hun beurt ook in Morse-code worden voorgesteld:

. als -.      - als ...-

Dit betekent dat we een Morse-bericht meermaals opnieuw kunnen encoderen als een Morse-bericht:

. → -. → ..- → -.-.....-.-.- → .....-.....-.....-.....-.....-.....- → enz.

Het vraagstuk dat je moet oplossen is het volgende: **als we een bericht in morsecode een gegeven aantal keer na elkaar coderen in morsecode, hoe lang is dan het uiteindelijke bericht in morsecode?** Je mag ervan uitgaan dat het invoerbericht zelf enkel uit streepjes en punten bestaat.

Implementeer de methode "**repeatedmorse**". Als invoer heb je begin-**String str** en **int n** stelt het aantal keer dat het bericht herhaaldelijk na elkaar geëncodeerd moet worden. Als uitvoer geef je een **int** terug die de lengte van het finale bericht teruggeeft.

Tip: de volgende methodes van **String** komen van pas:

`char charAt(int n)`: Geeft het karakter terug op de n<sup>de</sup> plaats.

`int length()`: Geeft de lengte terug van de string