

Oefening 5: MVC

1. Model

Maak een klasse *BoodschappenLijst* waarin je een lijst van *Boodschappen* kan bijhouden. Gebruik hiervoor een *ArrayList*. Een boodschap wordt beschreven in een aparte klasse *Boodschap* (aantal, eenheid en omschrijving volstaan als attributen). De klasse *BoodschappenLijst* moet volgende methodes aanbieden:

- Een constructor, setter, getter en *toString*
- Een methode *voegToe* waarmee je een nieuwe boodschap toevoegt aan de lijst met boodschappen;
- Een methode *volgendeBoodschap* waarmee je de lijst van boodschappen circulair overloopt, dwz., als er bijvoorbeeld drie boodschappen in de lijst zitten, dan krijg je de eerste keer dat deze methode wordt opgeroepen de eerste terug, de tweede keer de tweede, de derde keer de derde boodschap, de vierde keer opnieuw de eerste, enzovoort. Om deze methode te implementeren, kan je de klasse bijvoorbeeld een bijkomend attribuut *huidigeIndex* geven.

Als je in dit programma voor 1 item een negatief getal of 0 geeft, gaan we dit verbieden door het opnemen van beveiligingsmaatregelen in de klasse *BoodschappenLijst*: zorg ervoor dat je geen boodschappen kan maken / toevoegen met een ongepast aantal door een uitzondering van het type *IllegalArgumentException* gegooit te gooien.

Uitbreidingen :

- Verander je klasse *BoodschappenLijst* zodanig dat de methode *toString* gebruik maakt van een *Iterator* over je *ArrayList*.
- Maak je eigen *TeWeinigUitzondering* als subklasse van *Exception* en gebruik deze in de plaats van *IllegalArgumentException*

2. View

Implementeer initieel een GUI die er (ongeveer) als volgt uit ziet:



Deze GUI heeft de volgende componenten:

- Drie tekstveldjes waarmee de gebruiker een boodschap kan ingeven;
- Een knop "Voeg toe" en een knop "Volgende";
- Een label waarmee een boodschap kan worden getoond aan de gebruiker (initieel "Welkom")

3. Controller

De view uit de vorige sectie moet gekoppeld worden aan het model dat je in de eerste sectie ontwikkeld hebt. Concreet is het de bedoeling dat deze GUI gebruikt kan worden voor het beheren en bekijken van een BoodschappenLijst:

- Als de gebruiker op de knop “Voeg toe” klikt, dan wordt de boodschap die hij via de tekstveldjes heeft ingevuld, toegevoegd aan de BoodschappenLijst.
- Met de knop “Volgende” kan de gebruiker circulair navigeren doorheen de BoodschappenLijst.

Maak een klasse Controller die een BoodschappenLijst en een GUI aanmaakt en op de gepaste manier aan elkaar koppelt – hierbij moet je je implementatie van de GUI wat aanpassen, maar die van BoodschappenLijst blijft normaalgezien onveranderd.

Deze klasse bevat ook de main-methode en start dus uiteindelijk de volledige applicatie.

De gegooide uitzondering moet in de GUI of Controller worden opgevangen, zodat er een gepaste foutendialoog op het scherm getoond kan worden (zie klasse JOptionPane):

Uitbreidingen :

- Vervang het tekstveld voor de eenheid door een aantal radiobuttons (vb. stuks, liter, kg,...)
- Maak aparte view klassen voor het input gedeelte en het output gedeelte (voor 1 boodschap). Beide zijn afgeleide klassen van de klasse JPanel. Voeg deze subpanels uiteindelijk toe aan je window (JFrame) en controleer of het geheel correct samenwerkt. (zie ook figuur)
- Maak naast de view voor één boodschap ook een view-gedeelte voor de volledige boodschappenlijst door gebruik te maken van een JScrollPane met een JTextArea. Maak ook hier een aparte klasse.

Herschrijf nu de applicatie zodat al de deel-views samen worden getoond worden en op het juiste moment worden aangepast. Het geheel kan er dan als volgt uitzien :

