

HO GENT

OOSDII

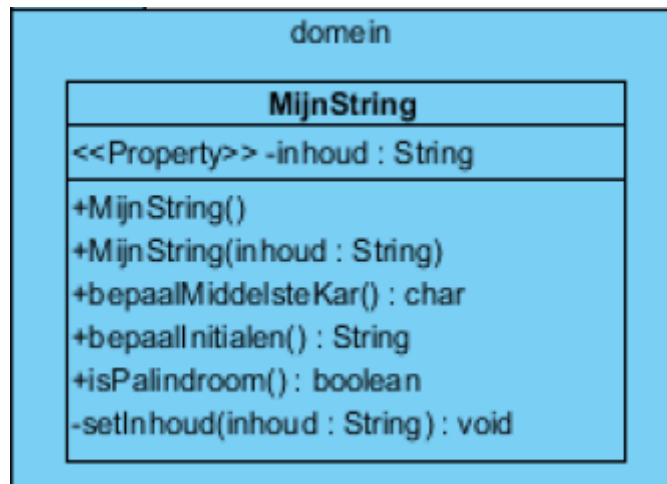
Strings en reguliere expressies

Table of Contents

1. Oefening MijnString	1
1.1. bepaalMiddelsteKar	1
1.2. isPalindroom	1
2. Oefening MijnString(2)	1
2.1. keerOm	1
2.2. isPalindroom	1
2.3. Werk onderstaande gui uit	1
3. Oefening wartaal	2
3.1. maakWartaal	2
4. Oefening validatie	4
5. Oefening splitsen van teksten	4
6. Oefening wartaal(2)	4
6.1. maakWartaal	5
6.2. splitsOp	5
7. Oefening PatternMatcher	6
7.1. PatternMatcher1	6
7.2. PatterMatcher2	6
8. Oefening EigenString	6
8.1. Maak een klasse EigenString	6
8.2. Console-applicatie	7
8.3. FX-applicatie	8
9. Oefening Namen	9
10. Oefening registratie	10

1. Oefening MijnString

Werk de domeinklasse MijnString uit (zie chamilo voor het startproject met testklasse)



1.1. bepaalMiddelsteKar

Geeft het middelste karakter van inhoud terug.

1.2. isPalindroom

Bepaalt of inhoud een palindroom is.

2. Oefening MijnString(2)

Voeg de volgende methode toe in de klasse MijnString

2.1. keerOm

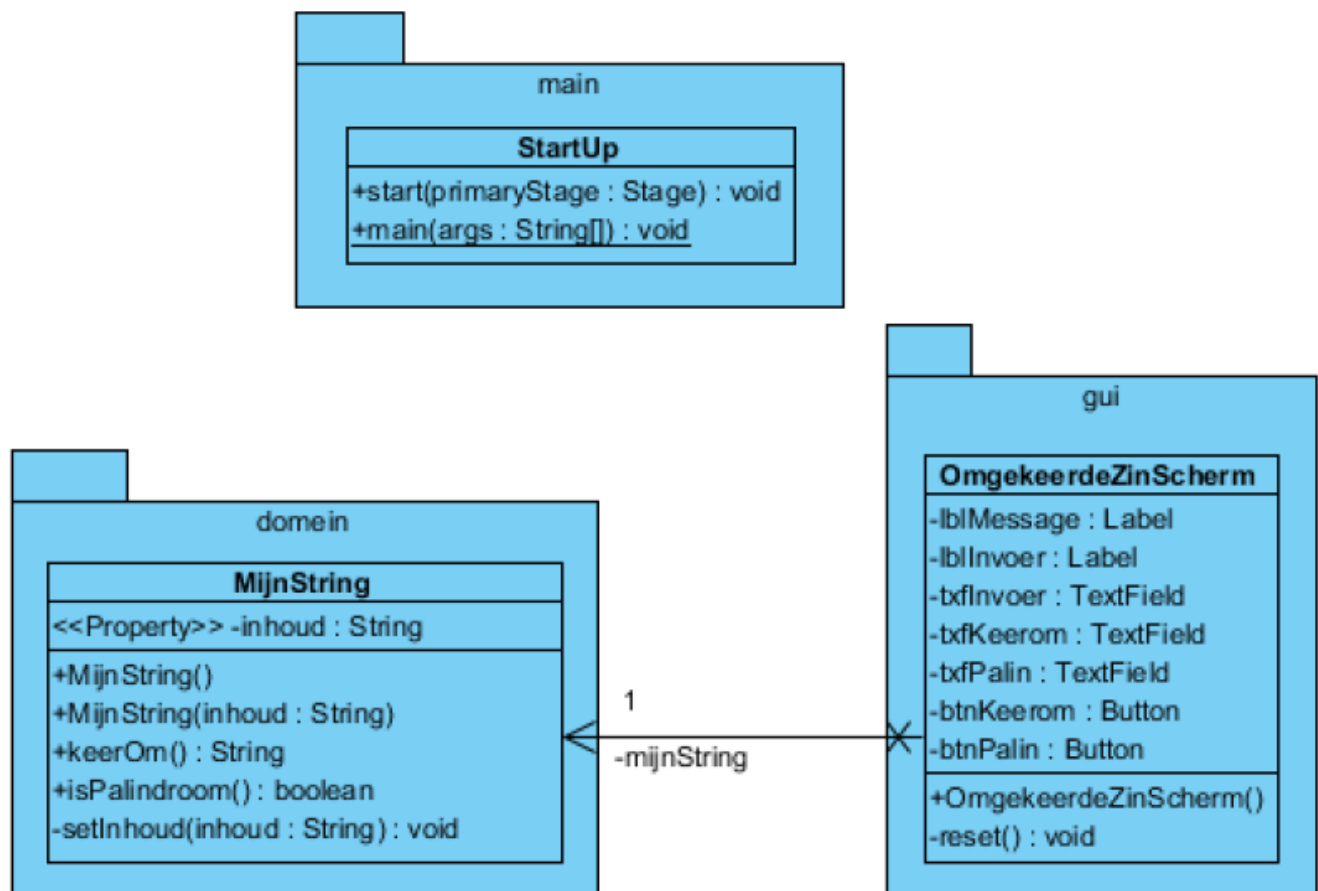
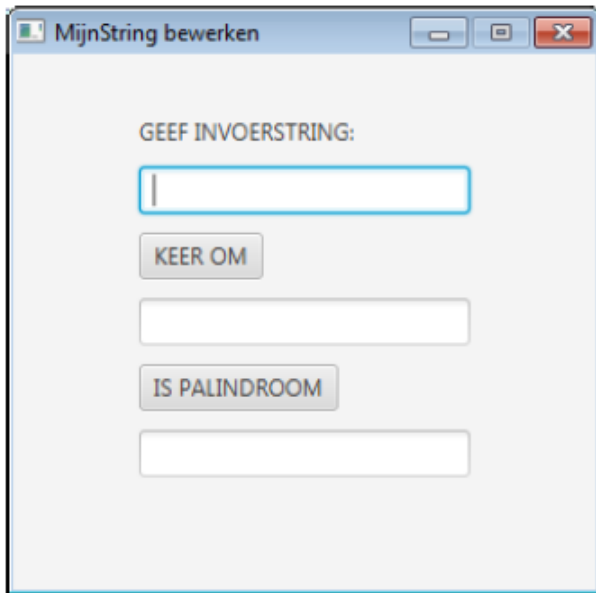
Geeft de inhoud in omgekeerde volgorde terug.

2.2. isPalindroom

Herschrijf de implementatie; gebruik nu een StringBuilder-methode.

2.3. Werk onderstaande gui uit

en gebruik MijnString.



3. Oefening wartaal

Breid de klasse `MijnString` uit met de methode

3.1. maakWartaal

die 2 karakters (oud en nieuw) als parameter heeft en die in het attribuut elk voorkomen van het eerste karakter vervangt door het tweede karakter.

Het resultaat wordt door de methode teruggegeven, met andere woorden het attribuut zelf wordt

niet gewijzigd!

Vb. "lepel". maakWartaal('e','a') geeft lapal

"LEPEL".maakWartaal('e','a') geeft "LAPAL"

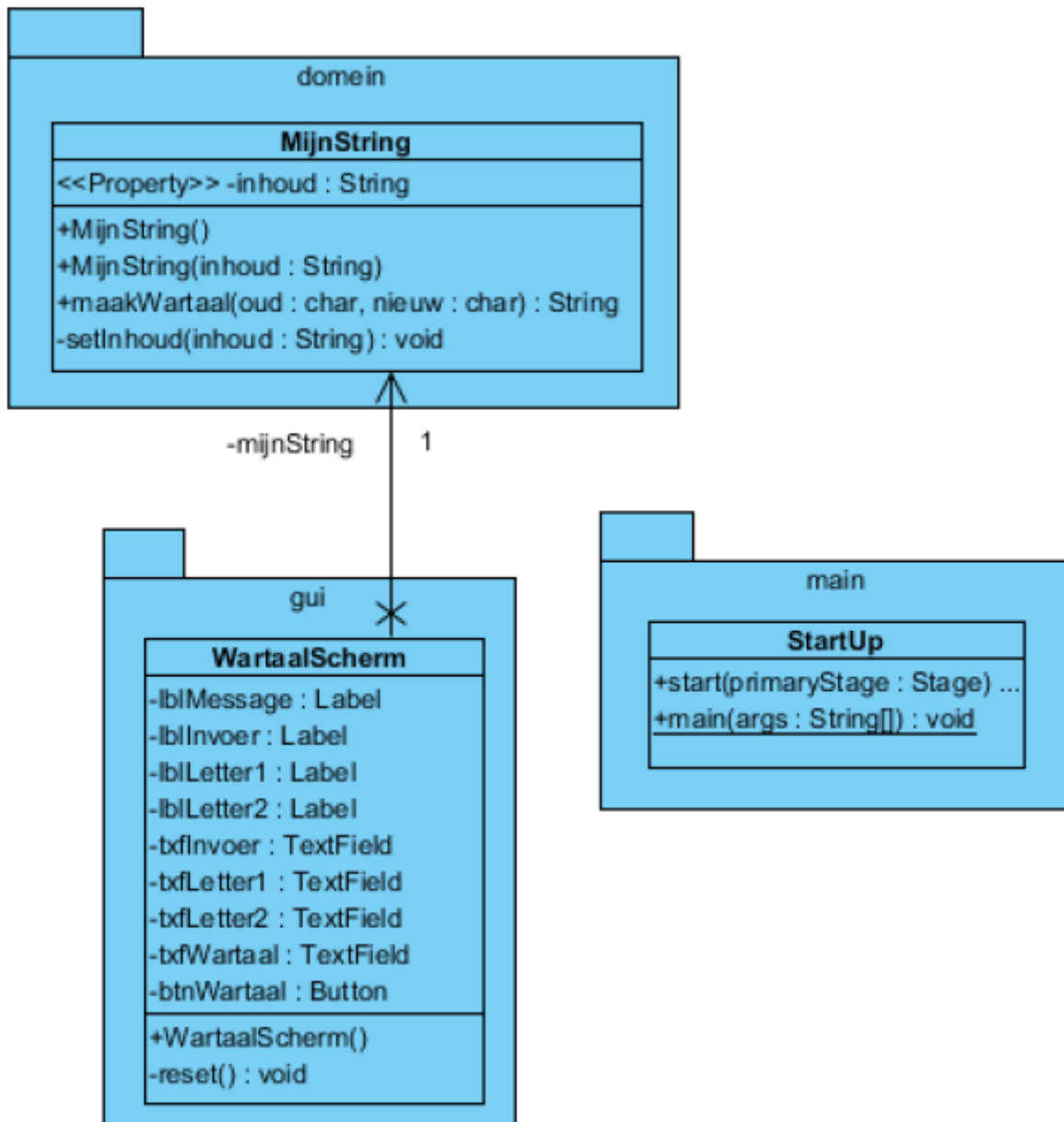
The image shows two side-by-side screenshots of a Java Swing window titled "MijnString bewerken".

Left Screenshot:

- Label: "GEEF INVOERSTRING:"
- Text input field: Empty.
- Label: "1ste LETTER:"
- Text input field: Empty.
- Label: "2de LETTER:"
- Text input field: Empty.
- Button: "MAAK WARTAAL" (disabled).
- Text output field: Empty.

Right Screenshot:

- Label: "GEEF INVOERSTRING:"
- Text input field: "een eenvoudige test".
- Label: "1ste LETTER:"
- Text input field: "e".
- Label: "2de LETTER:"
- Text input field: "a".
- Button: "MAAK WARTAAL" (active).
- Text output field: "aan aanvoudiga tast".



4. Oefening validatie

Valideer een e-mail, huisnummer, postcode en naam.

Gebruik `Validate.java` (Chamilo), specificaties validaties in opgave.

5. Oefening splitsen van teksten

Gebruik `StartUpSplitKleineOef.java` (Chamilo), specificaties in opgave.

6. Oefening wartaal(2)

Gebruik opnieuw de domeinklasse `MijnString` en

6.1. maakWartaal

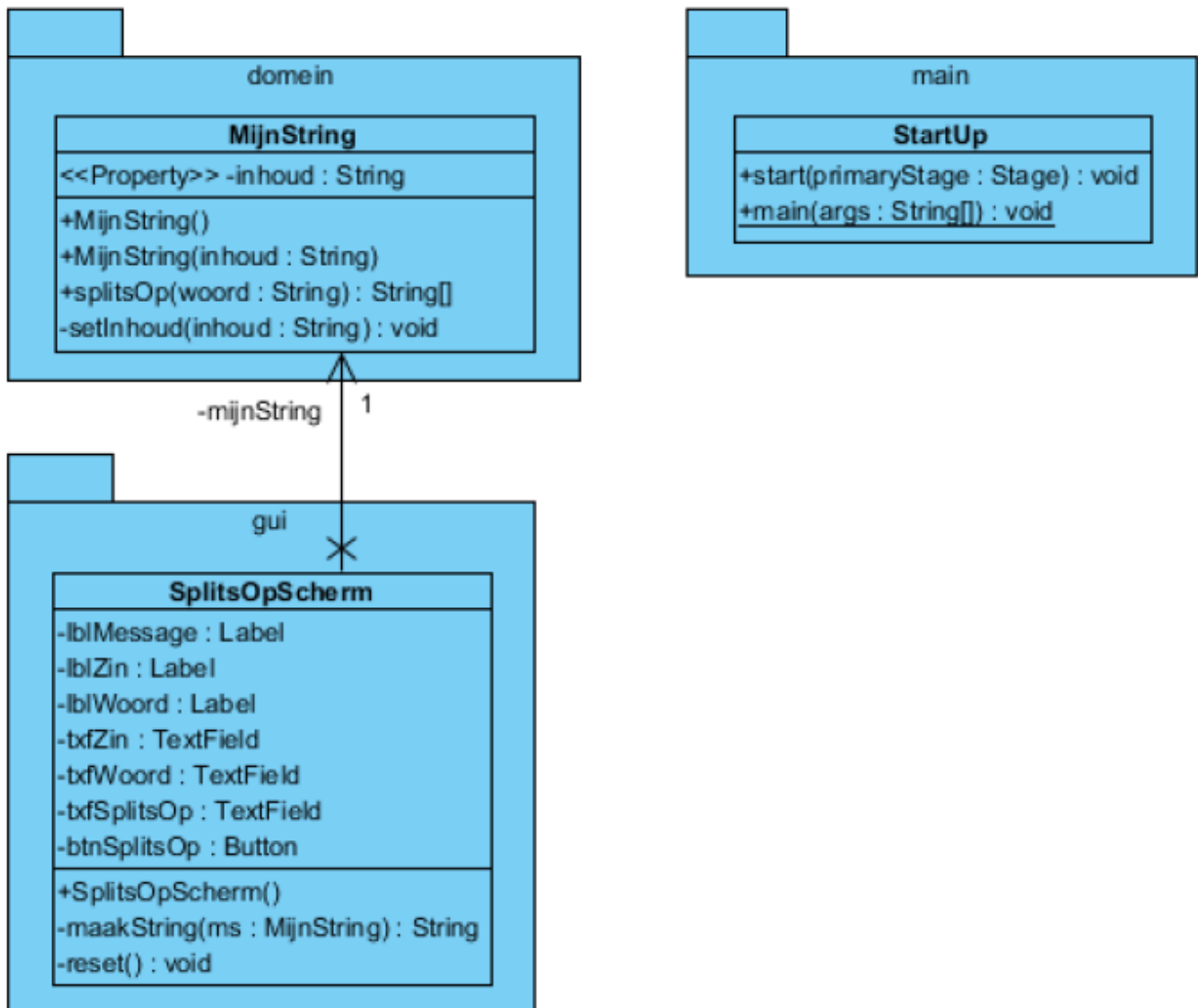
Herschrijf de implementatie; gebruik nu een reguliere expressie

6.2. splitsOp

Deze methode splitst inhoud op volgens het meegegeven woord(=scheidingsteken).



The screenshot shows a Java Swing window titled "MijnString bewerken". Inside the window, there are two text input fields. The first field is labeled "GEEF EEN ZIN:" and contains the text "Oefening dit is een voorbeeld dit is een :". The second field is labeled "GEEF EEN WOORD:" and contains the text "dit". Below these fields is a button labeled "OPSPLITSEN". At the bottom of the window, there is a text area displaying the result of the split operation: "[Oefening, is een voorbeeld, is een zin]".



7. Oefening PatternMatcher

7.1. PatternMatcher1

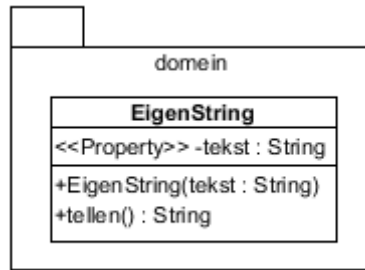
specificaties in opgave

7.2. PatterMatcher2

specificaties in opgave

8. Oefening EigenString

8.1. Maak een klasse EigenString.



Schrijf een applicatie waarbij de gebruiker een zin opgeeft. Nadat hij op Enter gedrukt heeft, verschijnt het resultaat van volgende tellingen:

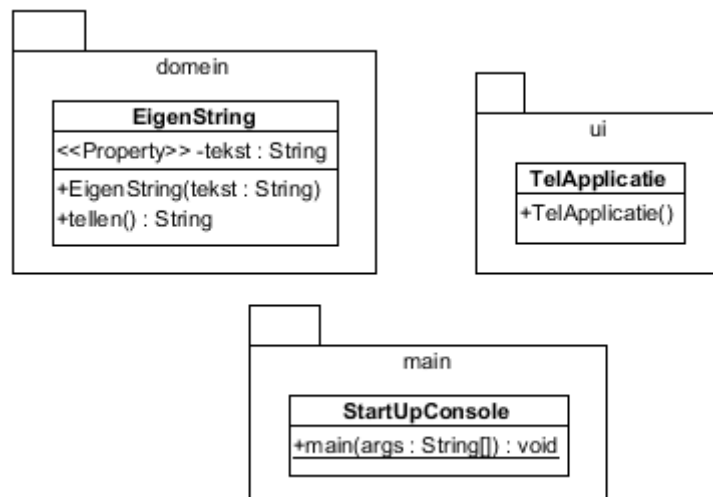
1. Het aantal klinkers
2. Het aantal medeklinkers
3. Het aantal cijfers
4. Het aantal andere karakters (alle tekens niet behorende tot a, b en c)
5. Het aantal hoofdletters (zowel klinkers als medeklinkers)
6. Het aantal kleine letters (idem als e)

Maak gebruik van de domeinklasse EigenString; werk een methode ‘telKaraktersVolgensSoort’ uit, die een String teruggeeft met de gevraagde informatie.

Werk uit met een console-applicatie en een FX-applicatie.

8.2. Console-applicatie

UML:



Uitvoer:

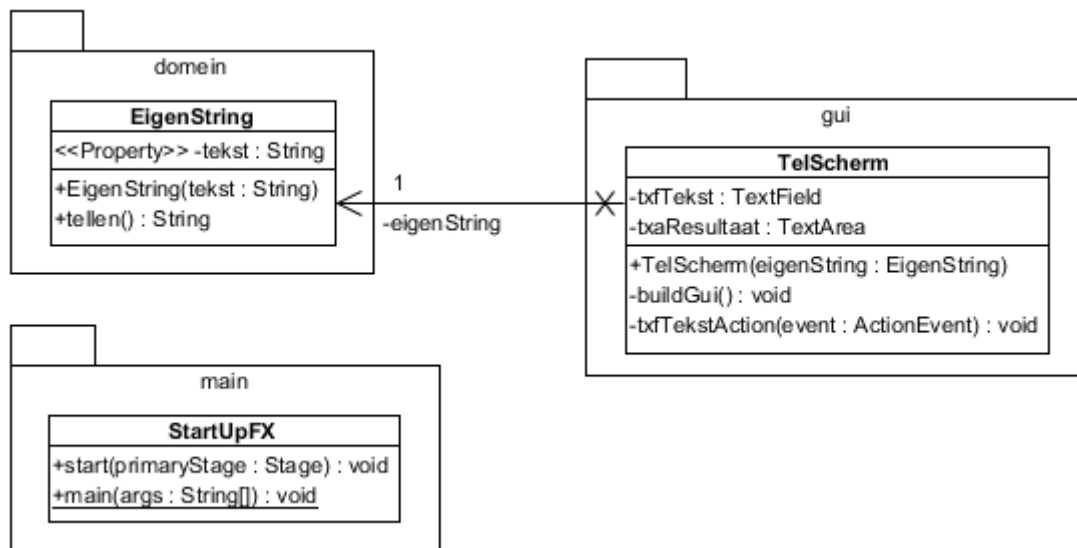
```
run-single:
Geef een tekst: In mei leggen alle vogels 1 ei.
In de zin " In mei leggen alle vogels 1 ei. "
```

Soort karakter	Aantal
Klinker	11
Medeklinker	12
Cijfer	1
Andere	7
Kleine letter	22
Hoofdletter	1

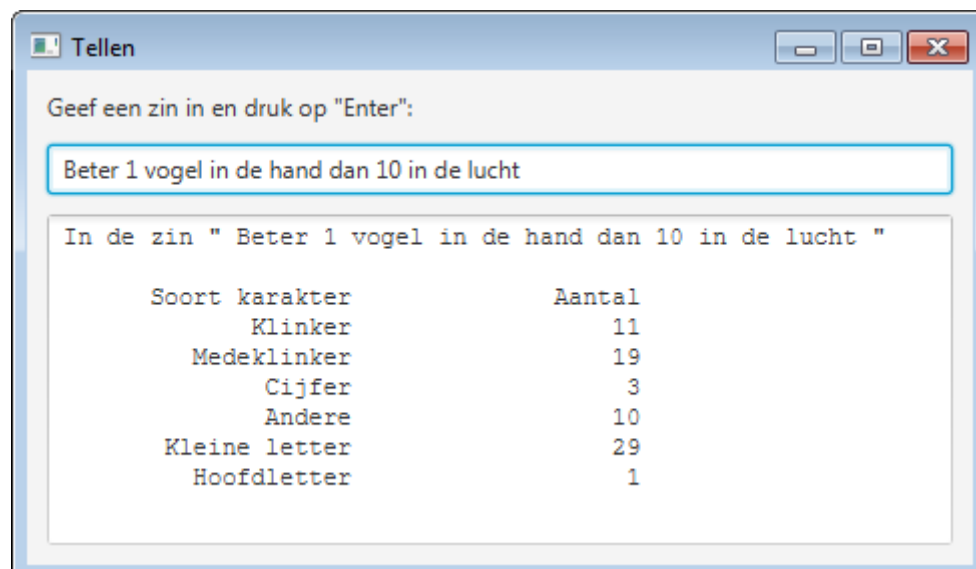
BUILD SUCCESSFUL (total time: 16 seconds)

8.3. FX-applicatie

UML:



Uitvoer:

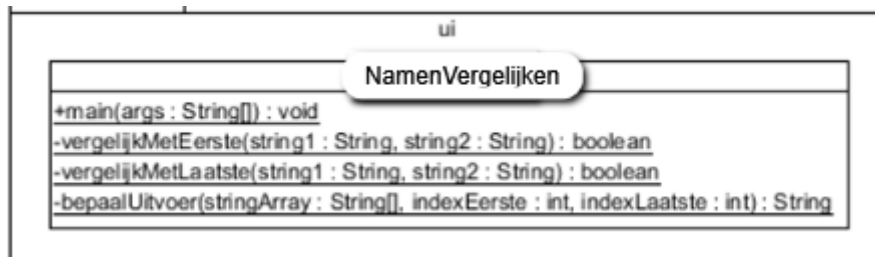


9. Oefening Namen

Schrijf een applicatie (main-methode en 3 hulpmethodes) die aan de gebruiker vraagt om maximaal 10 verschillende namen in te geven. (Om te stoppen met de invoer wordt de lege string ingegeven.)

Na de ingave worden de ingevoerde namen onder elkaar afgedrukt, met als vermelding EERSTE bij de naam die alfabetisch het eerst komt en LAATSTE bij de naam die alfabetisch het laatst komt.

UML:



- De methoden **vergelijkMetEerste** en **vergelijkMetLaatste** vergelijken respectievelijk de nieuwe naam met de eerste op dat moment/de laatste op dat moment; we geven true terug als de nieuwe naam de nieuwe eerste of de nieuwe laatste wordt.
- **bepaalUitvoer** bouwt de lijst met namen en de toevoegingen op.

Uitvoer:

```
Output - OefH14_Oef2 (run-single)

run-single:
Geef een naam in: Jan
Geef een naam in: Maarten
Geef een naam in: Stijn
Geef een naam in: Brecht
Geef een naam in: Kaat
Geef een naam in: Pieter
Geef een naam in: Gerrit
Geef een naam in: Pascale
Geef een naam in: Paul
Geef een naam in: Julie
Jan
Maarten
Stijn    LAATSTE
Brecht   EERSTE
Kaat
Pieter
Gerrit
Pascale
Paul
Julie

BUILD SUCCESSFUL (total time: 1 minute 12 seconds)
```

10. Oefening registratie

Maak een registratie-venster en controleer aan de hand van reguliere expressies of de ingevoerde gebruikersnaam en het wachtwoord voldoen aan de volgende regels:

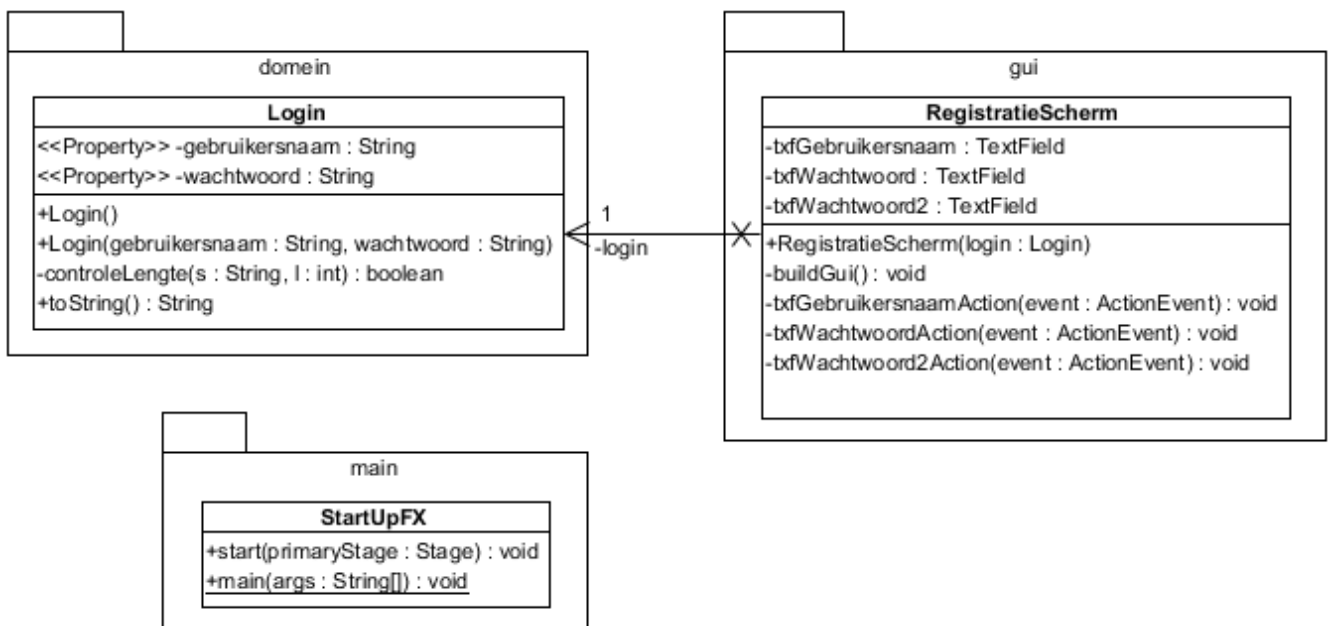
- Gebruikersnaam: 2 woorden, eerste woord moet starten met een hoofdletter en minstens 4 karakters bevatten, het tweede woord moet eindigen met een cijfer en minstens 8 karakters bevatten
- Wachtwoord: 1 woord, minstens 6 karakters waarvan minstens 2 cijfers en de rest allemaal kleine letters

In een derde invoervak moet het wachtwoord bevestigd worden. Controleer of dit hetzelfde is als het eerste ingevoerde wachtwoord. Indien in orde, dan verschijnt er een boodschap: ingevoerde gegevens worden aanvaard.

Na de enterhit in een invoervak wordt de invoer onmiddellijk gecontroleerd. Indien correct komt de cursor automatisch in het volgende invoervak te staan. Indien foutief wordt de foutieve inhoud geselecteerd en krijg je een volgende kans om iets in te voeren.

Maak een domeinklasse Login en controleer de data in de setters volgens bovenstaande regels.

UML:



Uitvoer:

Registratie

Gebruikersnaam:

Wachtwoord:

Bevestig wachtwoord:

Registratie

Gebruikersnaam:

Wachtwoord:

Bevestig wachtwoord:

Fout

Het wachtwoord moet minstens 2 cijfers bevatten en kleine kars

OK

Registratie

Gebruikersnaam:

Wachtwoord:

Bevestig wachtwoord:

Fout

Gelieve het wachtwoord correct te bevestigen

OK

