HO
GENT

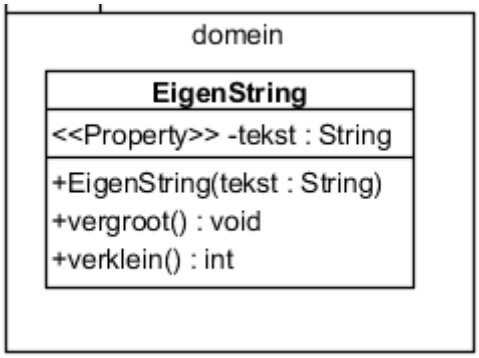
OOSDII

Strings en reguliere expressies

Table of Contents

1. Oefening Strings

- Maak een klasse EigenString.



- Aan een object van die klasse kan je vragen om te vergroten of te verkleinen.
  - Geef je de opdracht om te vergroten, dan wordt het String-object voorafgegaan door 3 spaties en gevolgd door 2 spaties.
  - Geef je de opdracht om te verkleinen, dan worden alle spaties uit het String-object verwijderd. Er wordt weergegeven hoeveel spaties er werden verwijderd

2. Oefening Namen

- Schrijf een applicatie die een willekeurige naam opzoekt in een gegeven lijst van namen.
  - Komt de naam voor in de lijst, dan volgt een gepaste vermelding en de alfabetische plaats van de ingegeven String in de lijst van namen.
  - Komt de naam niet voor in de lijst, dan verschijnt enkel een gepaste vermelding.
- Voorzie een domeinklasse Namen, die de data verwerkt. Daarnaast maak je een grafische user interface in Java FX waarin de opzoekingen kunnen gebeuren.
- In de persistentielaag bevindt zich een klasse NamenMapper, die een ongeordende lijst van namen teruggeeft, vb. "Roobrouck", "Roobroeck", "Bernard", "Barbier", "Brandt","Vandermeersch","Van Schoor","Vanenens",…
  Nog enkele opmerkingen:
  - Schrijf de Java-code op een efficiënte manier, zodanig dat bij wijziging van de meegegeven namen de code niet moet worden aangepast.
  - Wordt de te zoeken naam zonder hoofdletter ingegeven, dan vang je dit op in uw zoekopdracht.
- Oplossing versie1 :
  - De namen worden eerst alfabetisch gesorteerd op de volgende manier: we voorzien een tweedimensionale array waarbij de eerste rij alle namen voorstelt die beginnen met een A, de tweede rij alle namen die beginnen met een B, enz.
  - Per rij voorzien we de nodige elementen om het aantal namen bij te houden. Daarvoor gebruiken we een array-teller die bijhoudt hoeveel namen er precies beginnen met een A, met een B, enz. Je zorgt ervoor dat per rij de namen alfabetisch gesorteerd staan, zodanig

dat het latere opzoekwerk versneld wordt.

- Oplossing versie2 :
  - Gebruik methodes uit de klasse Arrays.

3. Oefening encryptie

- Schrijf een applicatie die een zin kan encrypteren en terug decrypteren.
- Het encrypteren verloopt als volgt:
  - De zin wordt opgesplitst in zijn woorden. Afhankelijk van de positie van het woord in de zin (in het voorbeeld: “We” = eerste woord, “gaan” = tweede, “deze” = derde, “zin” = vierde, enz.) worden de letters van het woord vervangen door nieuwe letters met als nieuwe unicode: de unicode van de ‘oude’ letter + positie van het woord in de zin (= parameter verschuiving).
  - Karakters die geen letters zijn, worden niet gewijzigd!
  - Let op: als de nieuwe unicode buiten de grenzen van de unicodes voor de letters valt, dan dient deze herberekend te worden, waarbij je terug bij de ‘a’ of ‘A’ begint. Als in het eerste woord een letter ‘z’ staat, dan wordt deze bijvoorbeeld ‘a’ en als in het vierde woord een letter ‘X’ staat, dan wordt deze een ‘B’.
    Voorbeeld:
    In “We” (eerste woord, dus verschuiving = 1): W wordt X en e wordt f
    In “gaan” (verschuiving = 2): g wordt i, a wordt c (tweemaal) en n wordt p
    In “deze” (verschuiving = 3): d wordt g, beide e’s worden h’s en z wordt c
    In “zin” (verschuiving = 4): z wordt d, i wordt m en n wordt r.
    Enz…..
- Bij het decrypteren gebeurt het omgekeerde: let hier dus ook op met unicodes die voor die van de letter ‘a’ of ‘A’ zouden vallen.
- Als de gebruiker op één van beide knoppen klikt, gebruik je de methode “crypteer” in de gui-klasse om dit event op te vangen. Deze methode krijgt drie parameters mee: het invoervak, het uitvoervak en een boolean die aangeeft of het om encryptie (true) of decryptie (false) gaat.