HOGESCHOOL

Java Fundamentals - Labo 11

Richtlijnen

- Onderstaande oefeningen worden in de Netbeans IDE uitgevoerd. Denk aan de debug mogelijkheden!

Objecten en methoden - arrays (1 en 2 dim)

1. **WoordSpel** Schrijf een programma dat een gebruiker moet laten raden naar een woord. Het te raden woord wordt willekeurig gekozen uit een array die vooraf gevuld werd met mogelijke woorden. De gebruiker geeft ofwel een letter in ofwel het volledige woord wanneer hij denkt het woord te kennen. Het programmaverloop ziet er als volgt uit :

```
Geef je letter of het volledige woord : t
beurt 1: t__t__
Geef je letter of het volledige woord :  n
beurt 2: t__t_n
Geef je letter of het volledige woord : s
beurt 3: t_st_n
Geef je letter of het volledige woord : testen
beurt 4: Proficiat !
Je vond het woord in 4 trials
```

   Voorzie een constante om het maximaal aantal beurten vast te leggen. Zolang het woord niet geraden werd en het maximum aantal beurten niet overschreden werd mag de speler verder raden. Maak een klasse `WoordSpel` waarin je de volgende datavelden bijhoudt:

   - een constante met het maximum aantal beurten
   - een array opgevuld met 10 verschillende woorden
   - het huidige woord dat geraden moet worden

- een booleaanse array die bijhoudt welke letter uit het huidige woord al geraden werd.
- het aantal gokbeurten

Voorzie tevens volgende methoden :

- een constructormethode om je datavelden te initialiseren.
- een `isGeraden` methode die test of alle booleans van de booleaanse array op true staan. Deze methode neemt geen parameters en geeft zelf ook een booleaanse waarde terug.
- een `gok` methode die test of de meegegeven letter in het woord voorkomt. Indien ja, dan moet de booleaanse array aangepast worden. De methode geeft een String terug die alle tot dan toe correct geraden letters weergeeft. De andere letters worden verstopt achter een underscore karakter.
- een `gok` methode die test of het meegegeven woord het te raden woord is. Deze versie van de gok methode is een **overload versie** van de vorige. Deze methode neemt nu een string als parameter ipv een karakter. Als output geeft deze methode opnieuw een String terug. Wanneer de gok correct was, wordt het te raden woord volledig weergegeven en de booleaanse array wordt aangepast. Indien de gok niet juist was, wordt opnieuw de string met alle tot dan toe geraden karakters weergegeven.

2. **Matrix** Schrijf een klasse om matrices te definiëren. Als dataveld voorzie je een tweedimensionale array van integer getallen. Schrijf een niet-default constructormethode om matrices te kunnen initialiseren. Schrijf tevens een print-methode om je matrix netjes uit te printen.
   Schrijf vervolgens een gebruikersprogramma `MatrixDemo`(aparte klasse met main methode) dat een tweedimensionale reële array inleest van het toetsenbord. De dimensies van de array worden eerst aan de gebruiker gevraagd vooraleer de getallen worden ingelezen. Vervolgens maak je met deze data een object aan van type `Matrix` en je gebruikt de print-methode om de matrix weer uit te printen.

3. **rijSommen, kolomSommen** Breid je klasse `Matrix` uit met de methoden `rijSommen` en `kolomSommen` . Deze methoden zullen respectievelijk de som van elke rij en de som van elke kolom in de matrix berekenen. Geef telkens een 1 dim array terug als resultaat met hierin de resultaten van de sommen. Breid ook je gebruikersprogramma `MatrixDemo` uit door de sommen te berekenen en uit te printen naar het scherm.

4. **aftoppen** Breid je klasse `Matrix` uit met de methode `aftoppen`. Deze methode loopt over alle elementen van de matrix en topt de absolute waarde van elk getal in de array af op n (n is een positief geheel getal en geef je mee als formele parameter van de methode `aftoppen`). Aftoppen doe je als volgt : als een getal groter is dan n, wordt de waarde op n afgerond; is de waarde kleiner dan -n, dan wordt de waarde op -n gezet.

   Maak een nieuw `Matrix` object aan in je klasse `MatrixDemo`. Definieer de elementen van de array op de volgende manier:

   ```
   (int)(Math.random() * 1000.0 - 500.0)
   ```

   Top vervolgens deze matrix af op 100 en print het resultaat naar het scherm.

5. **lokaleMaxima** Voeg nu de methode `lokaleMaxima` toe aan je `Matrix` klasse. Deze methode zoekt alle locale maxima van een matrix-object. Een getal is een lokaal maximum als de waarden van alle acht de omringende getallen kleiner zijn. Een getal aan de rand van de matrix kan dus geen lokaal maximum zijn. Laat je methode een nieuw matrix object teruggeven, waarin alleen de waarden 0 of 1 voorkomen, de lokale maxima worden aangeduid met de waarde 1.

   Maak een opnieuw `Matrix` object (met dimensies groter dan 3) aan in je klasse `MatrixDemo`. Definieer de elementen van de array op de volgende manier:

   ```
   (int)(Math.random() * 100.0 - 50.0)
   ```

   Geef een samenvatting van alle lokale maxima van dit matrix-object.