Academiejaar: 2013-2014
Opleiding: Bachelor in de Toegepaste Informatica
Fase: 1
Examinator(en): M. Lens, M. Vandemaele, F. Vogels, M. Veracx
OPO: MBI08A - Beginselen van objectgericht programmeren
OLA: MBI08a - Beginselen van objectgericht programmeren
Activiteit: Examen Januari
Toegelaten hulpmiddelen: Geen
Datum: Donderdag 9 januari 2014
Beginuur: 8h 30
Toegestane tijdsduur: 3h

KHLEUVEN
GEZONDHEIDSZORG EN TECHNOLOGIE
ASSOCIATIE KATHOLIEKE UNIVERSITEIT LEUVEN

Student: …………………………………………… Nummer: ……………… Reeks: ……………………

PROBLEEMSTELLING

Voor professor Weetal moet je een programma voor een examen met meerkeuzevragen maken.

Prof. Weetal maakt in de main methode verscheidene meerkeuzevragen. Een meerkeuzevraag bestaat uit :

- de vraag zelf, bvb. "Vind je Java leuk";
- enkele te kiezen antwoorden, bvb.
- "Ja", "Nee" of "Kweenie"
- Het juiste antwoord, bvb. “Ja” (Dat dit het juiste antwoord is, wordt niet getoond aan de student.)

Prof. Weetal mag het aantal vragen, en ook per vraag het aantal mogelijke antwoorden willekeurig laten variëren.

Elke student krijgt bij uitvoering alle vragen (en antwoorden) in een andere volgorde. Om dit te implementeren, moet je gebruik maken van de klasse **Math** of **Random** (zie verder).

Als een student voor alle vragen een keuze gemaakt heeft, krijgt hij zijn totale score en dan zijn score met een correctie voor het raden (zie verder).

Voor communicatie met de gebruiker **moet** gebruik gemaakt worden van de klasse **GUI** waarvan je de Javadoc in bijlage vindt.

Wees consequent:

- Let op klasse-, methode- en variabele namen: gebruik zo omschrijvend mogelijke namen en volg de Java conventies
- Let op je parameter, instantie variabele en return types
- Maak bij initialisatie het verschil tussen String, int, char en double
- Herbruik code als je die al geschreven hebt

Hints:

- Als je een vraag niet kan beantwoorden, kan je altijd verder gaan naar de volgende vraag.
- De elementen van een rij door mekaar schudden, kan als volgt :
    - genereer willekeurig ("at random") 2 indexen en wissel de elementen op die indexen om;

    - doe dit een aantal keer (bvb. de lengte van de rij gedeeld door 2).
- Een willekeurig getal tussen tussen 0 (inclusief) en p (exclusief), krijg je door
    - (int)(Math.random() * p)
    - of Random r = new Random(); r.nextInt(p);
- De main methode schrijf je als
    - public static void main(String[] args)

DE TESTGEGEVENS

Wanneer het programma uitgevoerd wordt, krijgt de student (dankzij **GUI**) onderstaande schermen.

Deze test bevat 5 vragen.
Selecteer telkens het meest juiste antwoord.

Na klikken op "OK", komt er bv.:

Vraag 1: Wat is geen Java keyword?

Na klikken op één van de 4 mogelijkheden, is het antwoord 0 of 1 of 2 of 3 al naargelang de 1°, 2°, 3°, 4° mogelijkheid gekozen wordt.

Enz...
Als de student op alle vragen een antwoord gegeven heeft, krijgt hij nog informatie over zijn score:

Je scoorde: 3.0/5
met giscorrectie is dit 1.5/5

DE KLASSEN

Voorzie minstens 2 klasses waarbij het mogelijk is om de gegevens, zoals hierboven geïllustreerd, bij te houden.

Bekijk wat er gevraagd wordt in de vraag Main methode, en begin eventueel eerst met je klassendiagram te tekenen in vraag Klassendiagram.

- Voorzie de nodige constructoren, waarbij je de minimale gegevens kunt meegeven als parameters.

Voorzie een eerste klasse met de mogelijkheid om…

- per vraagstelling, een rij van mogelijke antwoorden en welk element van die rij het juiste antwoord is, bij te houden.
  - Als het eerste antwoord het juiste is, is het juiste antwoord dus 0
  - Er moeten minstens 2, en mogen maximaal 10 mogelijke antwoorden zijn.
- de instantievariabelen mogen nooit null zijn
- alle mogelijke antwoorden op een vraag willekeurig door elkaar te verplaatsen, waarbij het juiste antwoord correct onthouden blijft
- per vraag de score op basis van het antwoord te berekenen
  - waarbij er de mogelijkheid is om dit al dan niet met giscorrectie te doen
    - als juist +1
    - als fout + 0 of met giscorrectie -0.75

Voorzie een tweede klasse met de mogelijkheid om…
- alle mogelijke vragen bij te houden in een rij
  - maar geen 2 vragen met dezelfde vraagstelling te hebben
  - het aantal vragen moet meer dan 1 zijn, maar minder dan 10
- op basis van deze rij alle mogelijke vragen willekeurig door elkaar te verplaatsen
- alle vragen achtereenvolgens te stellen aan de gebruiker (gebruik makend van **GUI**) en diens antwoorden te onthouden
- en de totale score (met of zonder giscorrectie) kan berekend worden

Academiejaar: 2013-2014
Opleiding: Bachelor in de Toegepaste Informatica: **BOP**
Fase: 1
Student: ………………………………………………… Nummer: ………………… Reeks: ………………………

DE MAIN METHODE

Vul hieronder de main-methode op de stippellijnen aan. Gebruik de klasse **GUI** en jouw klassen. Het resultaat is zoals hoger onder "TESTGEGEVENS".

```
public class WeetAlMain {
   public static void main(String[] args) {
   // Definieer een examen met 5 mogelijke vragen
   ....................................................................................................
   ....................................................................................................
   // Voeg de eerste vraag toe aan het examen
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   // Voeg de tweede vraag toe aan het examen
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   // Wijzig de vraagvolgorde en bij elke vraag de antwoordvolgorde willekeurig
   ....................................................................................................
   ....................................................................................................
   // Zeg tegen de gebruiker dat het examen start
   ....................................................................................................
   // Neem het examen af en verzamel de antwoorden
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   // Bereken de score met en zonder giscorrectie
   ....................................................................................................
   ....................................................................................................
  // Geef de student zijn score
   ....................................................................................................
   } }
```

HET KLASSENDIAGRAM

Vul aan, op basis van de probleemomschrijving en je main methode
- welke klassen,
- instantievariabelen/velden
- en methodes

je nodig hebt.

GUI

WeetAlMain

+main(String args[])

Class GUI

```
public class GUI
extends java.lang.Object
```

Gegeven:
Een klasse die de grafische user interface (= de interactie met de gebruiker) geïmplementeerd heeft.
Hiervan maak je gebruik i.p.v. System.out.println of JOptionPane.

Constructor Summary

Constructors

Constructor and Description
`GUI``()`

Method Summary

Methods

Modifier and Type	Method and Description
`void`	**`geefInfo`**`(java.lang.String titel, java.lang.String tekst)` Deze methode geeft een tekstkader aan de gebruiker met een titel en een tekst: bv
`int`	**`stelVraagEnKrijgAntwoord`**`(int vraagnummer, java.lang.String vraag, java.lang.String[] options)` Deze methode stelt een vraag aan de gebruiker, en geeft de mogelijke antwoordopties .

Constructor Detail

GUI

```
public GUI()
```

Method Detail

stelVraagEnKrijgAntwoord

```
public int stelVraagEnKrijgAntwoord(int vraagnummer,
                                    java.lang.String vraag,
                                    java.lang.String[] options)
```

Deze methode stelt een vraag aan de gebruiker, en geeft de mogelijke antwoordopties .
de gebruiker moet hieruit 1 optie kiezen, en het volgnummer van zijn keuze wordt teruggeven.

Bijvoorbeeld:

Parameters:

`vraagnummer` - = 1
vraag = "Is Java leuk?"
options = { "Neen, te saai", "Ja, uitdagend", "Neen, ik snap er niets van" };

Geeft de gebruiker volgende kader:

```
________________________________________________________________
|
| Vraag 1: Is Java leuk?
|
| [Neen, te saai] [Ja, uitdagend] [Neen, ik snap er niets van]
|_______________________________________________________________
```

Returns:

Als de gebruiker op "Ja, uitdagend" klikt, geeft de functie int 1 terug

geefInfo

```
public void geefInfo(java.lang.String titel,
                     java.lang.String tekst)
```

Deze methode geeft een tekstkader aan de gebruiker met een titel en een tekst:
bv

Parameters:

`titel` = "Je resultaten met giscorrectie"
`tekst` = "Je scoorde 3/4\nBravo!"

geeft de gebruiker volgende kader:

```
________________________________________________________________
| Je resultaten met giscorrectie
|
| Je scoorde 3/4
| Bravo!
|_______________________________________________________________
```