

# Opdrachten methoden 2

---

## Casus 1 – Reversi

Schrijf een programma waarbij de gebruiker zijn naam kan invullen. Vervolgens toont het programma de naam van de gebruiker achterstevoren. Dus als iemand Jan Klaassen invult dan is de uitkomst nessaalK naJ.

Eisen:

1. De berekening (omdraaien van een stuk tekst) moet in een methode gebeuren.
2. Het algoritme moet met een lus werken.

Ontwerpidee: gebruik een string variabele voor de naam (input) en een tweede string-variabele voor de uitkomst. De tweede variabele bouw je karakter voor karakter op met een lus (while of for).

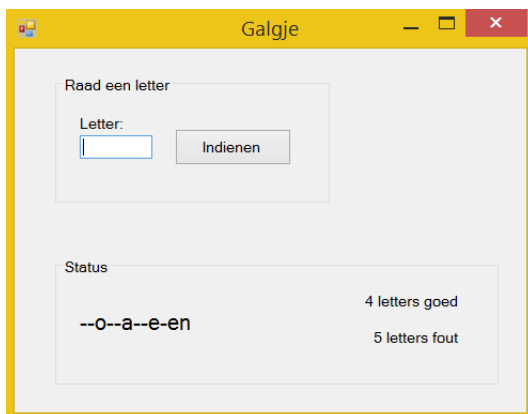
## Casus 2 – Faculteit

Schrijf een programma dat van een willekeurig getal (in te voeren door gebruiker) de faculteit uitrekent.

Eisen:

1. De berekening (uitrekenen faculteit van een geheel getal) moet in een methode gebeuren.
2. Het algoritme moet met een lus werken.

## Casus 3 – Galgje



Programmeer het spel Galgje waarbij de gebruiker een vooraf ingesteld woord moet raden.

De gebruiker mag 10 keer een letter raden. Heeft hij daarna het woord goed dan heeft hij gewonnen, anders wint de computer.

Ontwerpideeën: houdt in een (string-) variabele het te raden woord bij en houdt in een andere variabele bij wat de gebruiker allemaal heeft geraden.

Eventuele uitbreidingen:

### Feature 1:

Maak een PictureBox waarin je middels een aantal plaatjes van een galgje en poppetje het aantal fouten laat zien. Gebruik bijv. de volgende tekening <http://defransejuf.nl/wp->

content/uploads/2016/10/Jeu-du-pendu.png (Links to an external site.)Links to an external site. en knip de verschillende toestanden uit en noem die galgjeX.png waarbij X een nummer is van 1 tm 8.

Laad dynamisch elke keer het volgende plaatje in als er een fout wordt gemaakt. Kijk eens naar de methode PictureBox.Load().

#### Feature 2:

Laat zien welke letters geweest zijn. Je kunt bijv. alle letters laten zien in een zwarte kleur. Letters die goed geraadde zijn maak je groen, letters die fout geraadde zijn maak je rood.

#### Feature 3:

Uitbreiding op feature 2: maak de (zwarte dus niet gekozen) letters clickable. Als je er op clickt wordt dezelfde event handler als onder de knop Indienen aangeroepen.

#### Feature 4:

Zorg dat als je een letter intypt deze automatisch wordt "ingediend", dus dat je niet ook nog eens op de knop moet clicken. Maak tevens na het indienen van een letter de TextBox leeg en zorg dat deze altijd de focus meteen weer heeft zodat je meteen een nieuwe letter kan intypen.

#### Feature 5:

Zorg ervoor dat je alleen letters kan intypen in de TextBox. Accepteer tevens hoofdletters. Er zijn meerdere manieren om dit op te lossen. Kijk bijv. eens naar deze StackOverflow bron waar een vergelijkbaar probleem wordt besproken of gebruik een variant op de TextBox nl. MaskedTextBox.

Type je geen letter in, bijv. een cijfer, dan speelt er een errorgeluidje af.

#### Feature 6:

Onthoud de score(s) van een bepaalde speler en schrijf deze weg naar een highscore bestand. Laat hiervoor de speler zich eerst identificeren. Maak een highscore overzicht scherm waarin alle namen met de gemiddelde aantal benodigde pogingen worden getoond.

#### Feature 7:

Uitbreiding op 6: Maak een login mogelijkheid waar je middels een wachtwoord jezelf moet identificeren zodat niet andere spelers met jouw highscore aan de haal gaan. Sla dit wachtwoord gehashed op in hetzelfde bestand waar ook de namen en highscores staan. Gebruik bijv. MD5 hiervoor. Elke regel in het bestand bestaat uit "<username> <md5 wachtwoord> <highscore>".

Omdat MD5 niet veilig is, probeer een ander hashing algoritme te vinden.

## Casus 4 - Boter-kaas-eieren tegen de computer

Je hebt al eerder een algoritme gemaakt voor boter-kaas-en-eieren.



Dat algoritme kun je programmeren om zo een spel te maken waarbij iemand tegen de computer boter-kaas-en-eieren kan spelen.

Denk van te voren na over hoe je dit gaat programmeren: welke variabelen heb je nodig en wanneer ga je die variabelen een waarde geven?