

# ITBC BI2 Opsporen van genetische mutaties & Python II

## Afvinkopdracht 5

---

### Afvinkopdracht 5: Object Oriëntatie

#### Oefenopdrachten

Maak de opdrachten

#### **Chapter 10 Classes and Object-Oriented Programming**

1. Pet Class (p494)
2. Car Class (p495)
8. Trivia Game (p497)

#### **Chapter 11 Inheritance**

3. Person and Customer Classes (p523)

### Afvinkopdracht

Deel 1

Image  
Not  
Available

DNA
-sequence: String
+getLength(): int +getTranscript(): String +setDNA(String)

- Schrijf een Class DNA. Deze class heeft als attribuut sequentie en als methodes setDNA(), getDNA(), getTranscript() en getLength().
  - sequentie is een string
  - setDNA() is een methode die een string accepteert en niets retourneert
  - getDNA() is een methode die niets accepteert en een String (de sequentie) retourneert
  - getTranscript() is een methode die niets accepteert en het transcript (RNA) retourneert
  - getLength() is een methode die niets accepteert en de lengte retourneert
- Bouw in de methode setDNA in dat de creatie van een DNA instance altijd alleen maar kan met DNA nucleotiden (ATGC) of N (van onbekend)
- Maak een constructor \_\_init\_\_ die bij de creatie van een DNA instantie meteen de sequentie accepteert van de te creëren DNA sequentie.
- Voeg als extra een methode toe die het GC percentage retourneert

## Deel 2

- Schrijf een applicatie die bijvoorbeeld het bestand [ftp://ftp.ensembl.org/pub/release-91/fasta/felis\\_catus/cdna/Felis\\_catus.Felis\\_catus\\_8.0.cdna.all.fa.gz](ftp://ftp.ensembl.org/pub/release-91/fasta/felis_catus/cdna/Felis_catus.Felis_catus_8.0.cdna.all.fa.gz) kan lezen en van iedere fasta entry een DNA sequentie object maakt.
- Alle DNA objecten worden in een lijst gezet
- Itereer over de lijst van DNA objecten en bepaal wat de sequentie is met het hoogste GC percentage. Print van deze sequentie het GC percentage, transcript en lengte.

## Bonus deel

Als je OO ontwerp goed is en het is goed geïmplementeerd dan kun je het eenvoudig uitbreiden met extra functionaliteit.

- Pas je programma zodanig aan dat het ook nog de naam van het gen toont. Dit is slechts een kleine aanpassing aan je code maar vergt wel dat je object oriëntatie goed begrijpt en goed hebt toegepast. Kortom: een half uur denken en 5 minuten programmeren.

2. Schrijf een super class voor DNA en noem deze sequentie. Neem alle functionaliteit die in deze super class kan hierin op.
3. Ontwikkel ook een class RNA, met als aanvullende methode `getTranslation()`

## Super bonus

De grote kracht van object orientatie is dat het dicht op de werkelijkheid staat en dat objecten zelf iets kunnen. Zo kun je uiteindelijk opdrachten geven aan objecten om iets te doen.

1. Breid je applicatie uit zodanig dat de DNA objecten zichzelf in de lijst kunnen sorteren op lengte. Stel dat je lijst `lijstDNA` heet zou je met de opdracht `lijstDNA.sort()` de opdracht geven alle DNA objecten in de lijst zichzelf te laten sorteren op lengte.