

# Übungen zum Propädeutikum Programmierung in der Bioinformatik

## Blatt 6

Termin: Dienstag, 26. November 2019

### Übung 1 Erweiterung von *DnaSequence*

Falls du bisher noch nicht Übung 1 vom Übungsblatt der letzten Woche gemacht hast, sollte diese erst fertiggestellt werden. Alternativ steht eine fertige *DnaSequence*-Klasse auf der Propädeutikumswebsite zum Download bereit. Versichere dich aber, dass du verstehst was die Klasse macht bevor du losprogrammierst.

1. Setze den Sichtbarkeitsmodifikator von allen Attributen in *DnaSequence* auf `private`. Schreibe dann zu allen Attributen der Klasse jeweils einen Getter und einen Setter.
2. Schreibe danach eine statische Variante von `isValidSequence()` mit der Signatur `isValidSequence(String seq)`. Überlege dir warum es sinnvoll sein kann, gerade diese Methode statisch zu implementieren.
3. Schreibe einen *zusätzlichen* Konstruktor `DnaSequence(String sequence)`. Dieser soll, nachdem keine `id` übergeben wird, der Sequenz einfach eine fortlaufende Zahl als `id` zuweisen.

### Übung 2 Rosalind-Recycling: Transcribing DNA into RNA

Auf Blatt 3 war verlangt die Rosalind-Aufgabe **Transcribing DNA into RNA** zu bearbeiten. Erweitere *DnaSequence* um eine Methode `getRnaSequence()` welche dir die Sequenz eines *DnaSequence*-Objekt in RNA übersetzt und als String zurückgibt.

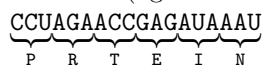
### Übung 3 Rosalind: Protein-Klasse

In dieser Aufgabe sollst du eine weitere Klasse *Protein* schreiben, die eine Aminosäuresequenz modelliert.

Nach dem **Zentralen Dogma der Molekularbiologie** wird DNA in RNA übersetzt, und die RNA in ein Protein. Auf vereinfachte Weise wurde bereits in der vorherigen Aufgabe die **Transkription** (DNA→RNA) durch die Methode `getRnaSequence()` modelliert. In der *Protein*-Klasse soll nun die **Translation** (RNA→Protein) modelliert werden.

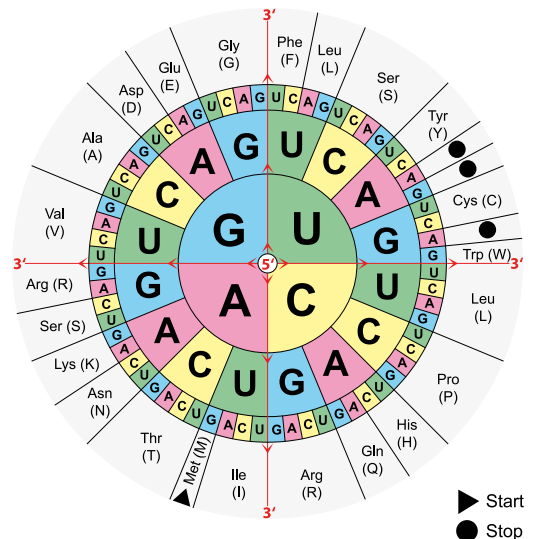
1. Die Klasse soll (wie *DnaSequence*) die Attribute `String id`, `String sequence`, sowie Getter und Setter besitzen.
2. Die Methode `rnaToProtein(String sequence)` soll implementiert werden. Diese nimmt als Input einen RNA-String an und übersetzt diesen anhand des **genetischen Codes**<sup>1</sup> in eine Aminosäuresequenz. Diese Sequenz soll dann zurückgegeben werden.

Eine Beispielsequenz würde (vgl. Abbildung) so übersetzt:



3. Nun sollen *zwei* Konstruktoren implementiert werden:

- (a) `Protein(String rnaseq)`
- (b) `Protein(DnaSequence dnaseq)`



*Codon-Sonne* an der von innen nach außen für jedes Basentriplett (*Codon*) die zugehörige Aminosäure abgelesen werden kann

Ob dein Algorithmus korrekt funktioniert kannst du wieder auf **Rosalind: Translating RNA into Protein** überprüfen.

<sup>1</sup>Im Rosalind-Testdatensatz wird der Einfachheit halber davon ausgegangen, dass jeder RNA-String mit einem Start-Codon beginnt und einem Stop-Codon endet. Es muss also nicht nach dem Beginn gesucht werden.