

Übungen zum Bioinformatik-Tutorium

Blatt 4

Termin: Dienstag, 13.11.2014, 11 Uhr

1. Schleifen

Erstelle eine Klasse Summe in deinem Tutorium-Projekt.

- (a) Implementiere in der Main-Methode eine for-Schleife, die alle Zahlen von 1 bis 42 aufaddiert.

```
int sum = 0;
for (int i = 1; i <= 42; i++)
{
    sum += i;
}
System.out.println(sum);
```

- (b) Wandle den Code aus der Main-Methode in eine statische Methode um, die dir die Summe aller Zahlen von 1 bis n ausgibt.

Der Methodenkopf soll so aussehen:

```
public static int getSumFrom1ToN(int n)
```

```
public static int getSumFrom1ToN(int n){
    int sum = 0;
    for (int i = 1; i <= n; i++)
    {
        sum += i;
    }
    return sum;
}
```

- (c) Teste deine Methode für mehrere Ausgaben. Erst mal für 42, um zu testen, ob deine beiden Ergebnisse übereinstimmen.

- (d) Teste deine Methode auch für große Zahlen. Zum Beispiel für 10.000.000. Ist das Ergebnis immer noch korrekt? Wenn nein, fällt dir ein Weg ein, diesen Fehler zu beheben?

Ein Integer hat nur 2^{32} Stellen in Binärcode. Der maximale positive Wert, den er annehmen kann, ist $2^{31} - 1$. Addiert man auf diesen maximalen Wert nochmal 1, läuft der Integer über und man erhält den kleinsten möglichen Wert, 2^{31} . Benötigst du größere Zahlen, könntest du mit Longs arbeiten... Aber auch ein Long hat seine Grenzen.

2. Codon-Zerlegung

Schreibe ein Programm, das eine Sequenz in Codons (dreier Triple) aufteilt und ausgibt.

Für die Eingabe CCCAGATTTGCG soll zum Beispiel die Ausgabe CCC-AGA-TTT-GCG erzeugt werden (was sich in "Prolin-Arginin-Phenylalanin-Alanin" übersetzen lässt, da immer jeweils 3 Nucleotide, also 1 'Codon', eine Aminosäure kodieren).

Tipp: Gehe durch die Sequenz nicht Zeichen für Zeichen, sondern mache 3er Sprünge und gib immer 3 Zeichen auf einmal aus.

```
public class Zerlegen {
    public static void main(String [] args)
    {
        if ( args.length < 1)
            throw new RuntimeException("Not_enough_parameter");

        String sequence = args[0];

        for (int i=0;i<sequence.length - 2;i = i + 3) {
            String codon = sequence.substring(i, i + 3);
            System.out.print(codon + "_");
        }
    }
}
```

3. Suche in einer DNA-Sequenz (optional)

Schreibe ein Programm, dem man 2 Parameter übergeben kann: Eine lange DNA-Sequenz, und eine kurze, die in der langen Sequenz gesucht werden soll. Die Stelle, an der sie gefunden wurde, soll

ausgegeben werden.

(Vorsicht: Der Computer beginnt beim zählen bei Null! Die Stelle 0 ist also eigentlich die Stelle 1, und die Stelle 5 ist eigentlich die Stelle 6. Vor der Ausgabe sollte die Stelle deshalb um eins hochgezählt werden!)

```
public class Suche {
    public static void main(String [] args)
    {
        if ( args.length < 2)
            throw new RuntimeException("Not_enough_parameter");

        String sequence = args[0];
        String code = args[1];

        for (int i=0;i<sequence.length;i++) {
            String codon = sequence.substring(i, i + code.length);
            if (codon.equals(code)) {
                int stelle = i + 1;
                System.out.println("Gefunden_an_Stelle_" + stelle);
                break;
            }
        }
    }
}
```