

Übungen zum Propädeutikum Programmierung in der Bioinformatik

Blatt 2

Termin: Dienstag, 22. Oktober 2019

Übung 1 *Basic Loop*

Benutze erst eine **For-Loop** und dann eine **While-Loop** um den String "Bioinformatik" jeweils fünf mal zu printen.

Lösung 1

```
1 public static void main(String[] args) {
2
3     for(int i = 0; i<=4; i++){
4         System.out.println("Bioinformatik");
5     }
6
7     int x = 0;
8     while(x<=4){
9         System.out.println("Bioinformatik");
10        x = x + 1;
11    }
12
13 }
```

Übung 2 *Basic Array*

1. Erstelle ein Array *words* , das die strings "DNA", "RNA", "Amino Acid" und "Protein" enthält.
2. Printe dir die Länge des Arrays *words* aus.
3. Printe dir das 2. Element dieses Arrays aus. (Hint: index = 1)
4. Ändere mittels geeigneter Anweisungen das Element 'DNA' in "deoxyribonucleic acid" und das Element "RNA" in "ribonucleic acid" um. Gebe dir diese beiden Elemente einzeln aus.

Lösung 2

```
1 public static void main(String[] args) {
2     //Aufgabe 2.1 - init array
3     String[] words = {"DNA", "RNA", "Amino Acid", "Protein"};
4
5     //Aufgabe 2.2 - Länge ausgeben
6     System.out.println("Länge des Arrays: " + words.length);
7
8     //Aufgabe 2.3 - 2. Element ausgeben
9     System.out.println(words[1]);
10
11    // Aufgabe 2.4 - Ändern und Ausgeben
12    words[0] = "deoxyribonucleic acid";
13    words[1] = "ribonucleic acid";
14    System.out.println(words[0]);
15    System.out.println(words[1]);
16
17 }
```

Übung 3 *Looping through an Array*

1. Benutze eine Loop deiner Wahl um alle Elemente des Arrays *words* aus der Aufgabe 2.1 zu printen.
2. Verändere die Loop so, dass nur die ersten 2 Elemente des Arrays geprinted werden.
3. Erstelle eine weitere Loop, welches alle Elemente printed, aber das Element 'Amino Acid' auslässt.

Lösung 3

```
1 public static void main(String[] args) {
2
3     //Aufgabe 3.1 - init array & loop
4     String[] words = {"DNA", "RNA", "Amino Acid", "Protein"};
5
6     for(int i=0; i<words.length ; i++){
7         System.out.println(words[i]);
8     }
9
10    //Aufgabe 3.2 - Nur die ersten 2 Elemente
11    for(int i=0; i<words.length ; i++){
12        if(i==2){
13            break;
14        }
15        System.out.println(words[i]);
16    }
17
18    // Aufgabe 3.3 - continue bei "Amino Acid"
19    for(int i=0; i<words.length ; i++){
20        if(words[i].equals("Amino Acid")){
21            continue;
22        }
23        System.out.println(words[i]);
24    }
25
26 }
```

Übung 4 *Odd Numbers*

Benutze eine geeignete Loop um die Summe aller **ungeraden** Zahlen zwischen zwei nach Größe geordneten, positiven Integer-Werten zu berechnen. Findest du eventuell eine Lösung, in welcher weniger Schleifendurchläufe benötigt werden, als im naiven Ansatz?

Given: Zwei positive Integer-Werte: a=144 und b=655

Return: Die Summe aller ungeraden Zahlen zwischen a und b (eingeschlossen).

Sample Dataset:

```
a=100 b=200
```

Sample Output:

```
7500
```

Lösung 4

Summe aller ungeraden Zahlen zwischen 144 und 655: **102400**

```
1 public static void main(String[] args) {
2
3     // naiver Ansatz
4     int a = 144;
5     int b = 655;
6     int sum = 0;
7
8     while(a <= b){
9         if(a%2 != 0){
10            sum = sum + a;
11        }
12        a++;
13    }
14
15    // weniger Schleifendurchläufe und weniger Modulo-Berechnungen
16    int a = 144;
17    int b = 655;
18    if(a % 2 == 0){
19        a++;
20    }
21
22    int sum = 0;
23    while(a <= b){
24        sum = sum + a;
25        a = a + 2;
26    }
27 }
28
```