Aufgabe 1

```java
public static int f1 ( int n) {
 int x = 0;
 for ( int i = 0; i < n ; i ++ )
    for ( int j = 0; j <= i; j ++ )
        for ( int k = 0; k < 5; k ++ )
            x = x + j;
 return x;
}
```

Die Schleife `for ( int i = 0; i < n ; i ++ )` bestimmt nur die Laufvariable i und ist daher nicht ausschlaggebend für die Laufzeit.

Die Schleife `for ( int j = 0; j <= i; j ++ )` läuft mit der laufvariable i durch. Nach der Anwendung der Gaußschen Summenformel erhalten wir also folgendes:
$1 + 2 + 3 + \ldots + n = n * (n + 1)/2$

Die Schleife `for ( int k = 0; k < 5; k ++ )` läuft je Iteration immer 5 mal durch.

Zusammen ergibt sich also die Formel: $(n * (n + 1)/2) * 5$.

Die Gleichung lässt sich vereinfachen zu: $(5/2) * n^2 + (5/2) * n$

Die Laufzeit von f1 ist also : $O(n^2)$

```java
public static int f2 ( int n) {
    if ( n <= 1 )
        return 2;
    else
        return f2 (n -1) * f2 (n -2) ;
}
```

Um die Laufzeit von f2 zu berechnen müssen wir die Aufrufe von `return f2 (n -1) * f2 (n -2) ;` bestimmen. Der Restliche Teil von f1 benötigt für jeden durchlauf O(1).

Jeder rekursive Aufruf hat wieder zwei weitere untergeordnete Aufrufe. Daher hat die Funktion eine Laufzeit von: $O(2^n)$

Aufgabe 2

a) $f(n) = n^3 + n$, zeige: $f(n) \in O(n^3)$

Zeige also $f(n) \leq c * n^3$ für alle $n \geq n0$.

Wir formulieren f(n) um zu: $f(n) = n^3 + n \leq n^3 + n^3$ für $n \geq 1$

Es ist also $f(n) \leq 2n^3$

Somit haben wir eine Konstante c = 2 gefunden. Für $n0 = 1$ gilt $f(n) \leq 2n^3$ für alle $n \geq n0$. Daher können wir sagen, dass $f(n) \in O(n^3)$.

b) $f(n) = log(n^2)$, zeige: $f(n) \in O(log(n))$

Zeige also $f(n) \leq c * log(n)$ für allen $n \geq n_0$

Nach Logarithmen Regeln können wir f(n) auch wie folgt darstellen $f(n) = 2 * log(n)$

Um also unsere Bedingung zu erfüllen können wir $c = 2$ wählen. So gilt dann für alle $n \geq 1$ unsere Behauptung $f(n) = 2 * log(n) \leq 2 * log(n)$ und $f(n) \in O(log(n))$ stimmt.

c) $f(n) = \sqrt{n}$, zeigen : $f(n) \in \Omega(log(n))$

Zeige also $f(n) \geq c * log(n)$für alle $n \geq n_0$

Wir wenden den Log auf beiden Seiten von f(n) an und erhalten so $log(f(n)) = 1/2 * log(n)$

Wir können nun zeigen, dass $log(f(n)) \in \Omega(log(n))$ Daraus gibt sich $log(f(n)) \geq (1/2) * log(n)$ für allen $n \geq 1$

Wir wählen $c = 1/2$ DAmit gilt: $log(f(n)) \geq (1/2) * log(n)$ für alle $n \geq 1$

Daraus ergibt sich unsere Behauptung als korrekt, da es ein c gibt für welches alle $n \geq n_0$ gilt.

d) $f(n) = 4n^4 + n^3$, zeige $f(n) \in \Theta(2n^4 + 15n^3)$

$c1 * g(n) \leq f(n) \leq c2 * g(n)$

$f(n) \leq c2 * g(n)$
$f(n) = 4 * n^4 + n^3 = n^3(4n + 1)$
$4n \leq 10n$ ==> $4n + 1 \leq 10n + 75$ ==> $n^3(4n + 1) \leq n^3(10n + 75)$
==> $4 * n^4 + n^3 \leq 5 * (2n^4 + 15)$ für alle $n \geq 1$
Daher gilt gilt $f(n) \in O(2 * n^4 + 15 * n^3) für c = 5 und n_0 = 1.$

$f(n) \geq c1 * g(n)$
$4 * n^4 + n^3 \geq c1 * (2 * n^4 + 15 * n^3)$
$2 * n^4 + 15 * n^3 \leq 17 * n^4$ ==> $c1 * (2 * n^4 + 15 * n^3)$ ≤ $c1 * (17 * n^4)$ für alle $n \geq 1$

Nun wählen wir $c1 = 1/17$ und $n_0 = 1$. Dann haben wir:
$1/17 * (2 * n^4 + 15 * n^3) \leq 1/17 * (17 * n^4) = n^4$

$4 * n^4 + n^3 \geq n^4$ ==> $f(n) \geq 1/17 * (2 * n^4 + 15 * n^3)$ für alle $n \geq 1$

Daher gilt $f(n) \in \Theta(2 * n^4 + 15 * n^3)$