Übungsaufgabe 5.1

Aufgabe 1

Jo, ist implementiert!

Aufgabe 2

Die Laufzeiten sind identisch, weil unsere erst Liste ebenfalls rekursiv an bestimmten Positionen einfügt. Somit kommen sie bei einer Implementationsgröße von 100 auf 5050 Zeiteinheiten.

Aufgabe 3

Übungsaufgabe 5.2

Aufgabe 1

Ergebnis für 0 ist 1

Ergebnis für 1 ist 1

Ergebnis für 2 ist 1

Ergebnis für 3 ist 6

Ergebnis für 4 ist 11

Ergebnis für 5 ist 26

Ergebnis für 6 ist 66

Ergebnis für 7 ist 151

Ergebnis für 8 ist 361

Ergebnis für 9 ist 861

**for**(**int** i=0;i<=9;i++){

System.*out*.println("Ergebnis für " + i + " ist " + Methods.*fancyFunction*(i));

}

Aufgabe 2

Jo, ist implementiert!

Aufgabe 3

Jo, ist 27.

**for**(**int** i=0;i<=Integer.*MAX\_VALUE*;i++){

**int** result = Methods.*fancyFunction*(i);

System.*out*.println("Ergebnis für " + i + " ist " + result);

**if**(result < 0){

**break**;

}

}

Aufgabe 4

**public** **static** List<Integer> *array* = **new** ArrayList<Integer>();

**public** **static** **int** fancyFunction(**int** n) {

*array*.set(n, *array*.get(n)+1);

**int** accu = 0;

**if** (n < 3) {

accu = 1;

} **else** {

accu = *fancyFunction*(n-1) + 2\**fancyFunction*(n-2) + 3\**fancyFunction*(n-3);

}

**return** accu;

}

Für n = 19

**int** result = Methods.*fancyFunction*(n-1);

**int** n = 20;

**for** (**int** i = 0; i < n; i++) {

Methods.*array*.add(i, 0);

}

[10609, 16377, 19513, 10609, 5768, 3136, 1705, 927, 504, 274, 149, 81, 44, 24, 13, 7, 4, 2, 1, 1]

Aufgabe 5 & 6

**public** **static** **int** fancyFunctionOpt(**int** n) {

*fancyArray* = **new** ArrayList<Integer>();

**for**(**int** i = 0; i < n+1; i++) {

*fancyArray*.add(-1);

}

**return** *fancyHelper*(n);

}

**private** **static** **int** fancyHelper(**int** n) {

**if**(*fancyArray*.get(n) != -1) {

**return** *fancyArray*.get(n);

} **else** **if** (n < 3) {

**return** 1;

} **else** {

*fancyArray*.set(n, *fancyHelper*(n-1) + 2\**fancyHelper*(n-2) + 3\**fancyHelper*(n-3));

**return** *fancyArray*.get(n);

}

}