Fibonacciho posloupnost

Způsob implementace

Úkol byl počítat Fibonacciho posloupnost třemi způsoby:

S použitím rekurze a pamatováním mezivýsledků:

Pro zapamatování mezivýsledků bylo využito datové struktury pole. Do tohoto pole se na pozici indexu ukládá mezivýsledek daného čísla. V metodě se nejprve porovnává, zda zadané číslo není 0 nebo 1. V těchto dvou případech se na 0 index pole vloží hodnota 0 respektive na pozici v poli s indexem 1 vloží číslo 1. Dále se porovnává zadané číslo, pro které chceme vypočítat konečné číslo, zda je v poli. Jestliže v poli není vypočítá se, k vypočtu se používá stejný postup rekurzivně. Tyto vypočtené hodnoty se uloží do pole na index čísla pro které se tento výpočet posloupnosti počítá.

P[n]	Hodnota
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55

Programová složitost

Počítání pomocí rekurze: 2+3+6*(N-2) +6*(N-3)

Počítání bez použití rekurze: 6+8*N

Testování

1. Testování posloupnosti pro nulu – hraniční bod

```
Zadej kolikátý prvek chceš vypočítat: 0

Provedené výpočty:
1. Rekurzivně
0
2. Nerekurzivně
0
3. Rekurzivně s pamatováním prvků
0
Pro informaci uvádíme výpis pole
0, BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Testování posloupnosti pro jedničku – hraniční bod

```
run:
Zadej kolikátý prvek chceš vypočítat: 1

Provedené výpočty:
1. Rekurzivně
1
2. Nerekurzivně
1
3. Rekurzivně s pamatováním prvků
1
Pro informaci uvádíme výpis pole
0, 1, BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Testování pro dvojku – první bod, který se netestuje

```
run:
Zadej kolikátý prvek chceš vypočítat: 2

Provedené výpočty:
1. Rekurzivně
1
2. Nerekurzivně
1
3. Rekurzivně s pamatováním prvků
1
Pro informaci uvádíme výpis pole
0, 1, 1, BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Testování pro trojku – první součet, kde není nula

```
run:
Zadej kolikátý prvek chceš vypočítat: 3

Provedené výpočty:
1. Rekurzivně
2
2. Nerekurzivně
2
3. Rekurzivně s pamatováním prvků
2
Pro informaci uvádíme výpis pole
0, 1, 1, 2, BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Testování pro desítku – namátkově vybrané číslo

```
run:
Zadej kolikátý prvek chceš vypočítat: 10

Provedené výpočty:
1. Rekurzivně
55
2. Nerekurzivně
55
3. Rekurzivně s pamatováním prvků
55
Pro informaci uvádíme výpis pole
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, BUILD SUCCESSFUL (total time: 0 seconds)
```

6. Testování pro dvacítku – dvojnásobek vybraného čísla

```
run:
Zadej kolikátý prvek chceš vypočítat: 20

Provedené výpočty:
1. Rekurzivně
6765
2. Nerekurzivně
6765
3. Rekurzivně s pamatováním prvků
6765
Pro informaci uvádíme výpis pole
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, BUILD SUCCESSFUL (total time: 0 seconds)
```

7. Testování výpočtu větších hodnot již nemá příliš velký smysl