**Vysoká škola polytechnická Jihlava**

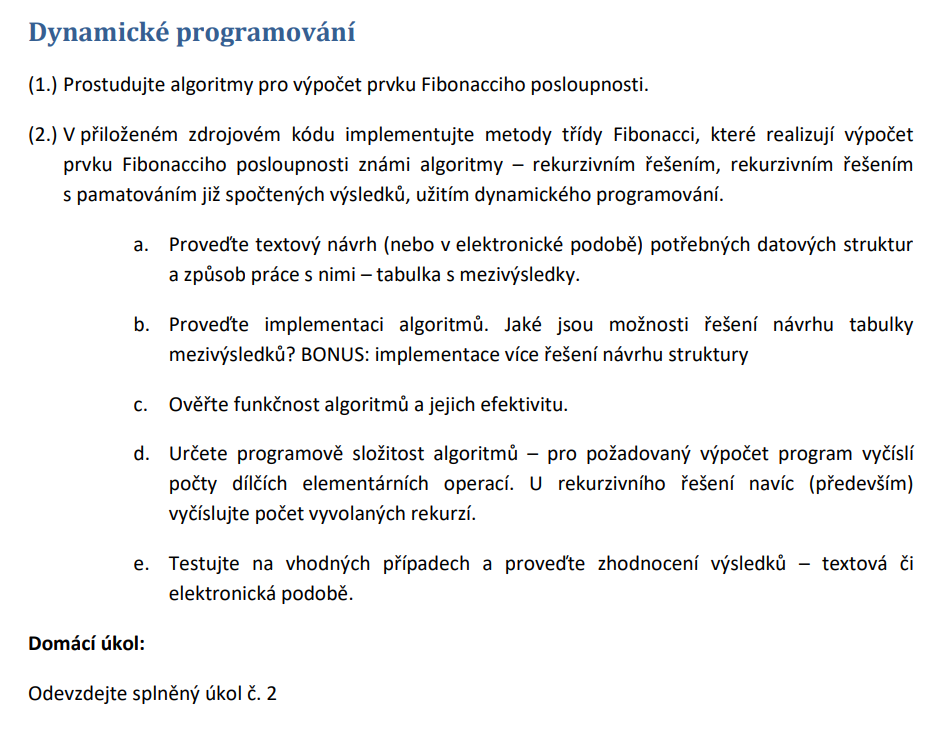
Katedra technických studií

**Pokročilé programovací techniky  
Třetí sada úkolů**

Autoři práce: Dominik Bula, Tomáš Cink, Jáchym Hruška, Vít Štěpánek, Ondřej Vaňkát

Jihlava 2019

# Zadání



# Řešení

## Bod a)

Viz přiložené soubory „bod\_a.pdf“ a „bod\_a2.pdf“.

## Bod b)

Pro implementaci viz přiložený program.

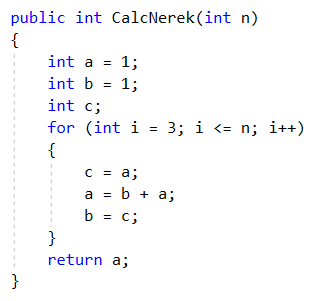
Možnosti řešení implementace tabulky jsou např.:

* Pole proměnných typu INT
* .NET Arraylist
* .NET List (typu INT)

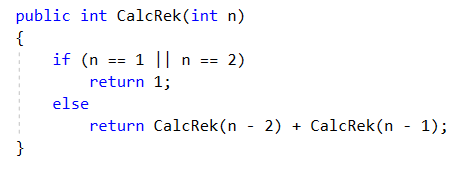
## Bod c), d) a e)

Z definice:

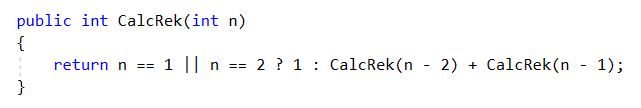
* nerekurzivní algoritmy mají delší zápis, ale menší složitost
* rekurzivní algoritmy mohou mít naopak velmi krátký zápis, ale velkou složitost
* některé úkoly jde naprogramovat pouze pomocí rekurzivního algoritmu



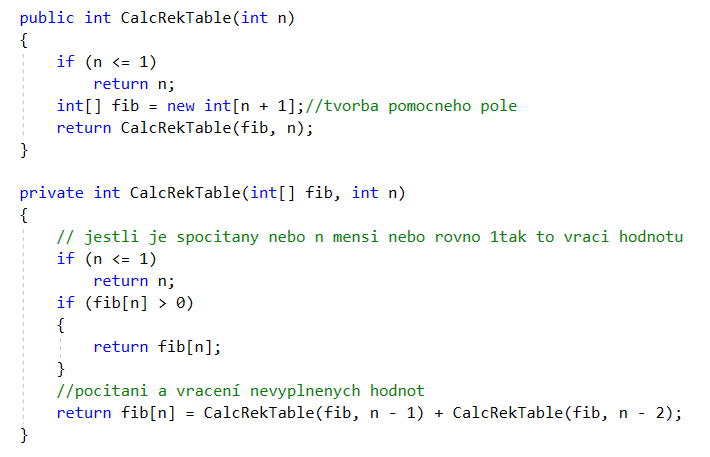
Obrázek 1: Nerekurzivní algoritmus



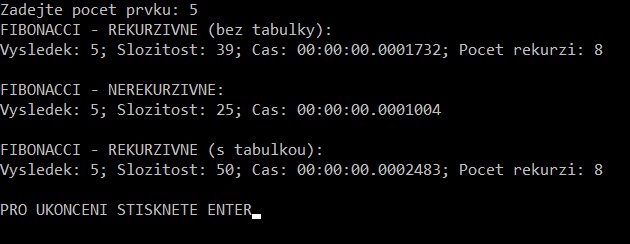
Obrázek 2: Rekurzivní algoritmus (bez tabulky)



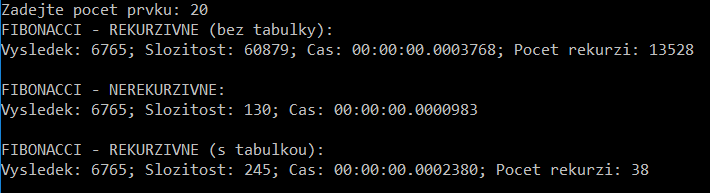
Obrázek 3: Rekurzivní algoritmus (bez tabulky) - ternární operátor



Obrázek 4: Rekurzivní algoritmus (s tabulkou)



Obrázek : Výsledky (pro číslo 5)



Obrázek : Výsledky (pro číslo 20)

Z tabulek výsledků a z obrázků můžeme potvrdit, že nerekurzivní algoritmy mají opravdu delší kód, ale menší složitost. Nerekurzivní algoritmy mají zase kratší zápis (zejména s použitím ternárního operátoru), ale větší složitost.

Co se týče našeho rekurzivního algoritmu s tabulkou, tak pro číslo 5 má horší složitost a tím pádem i časovou náročnost než algoritmus rekurzivní bez tabulky. Avšak pro číslo 20 dostáváme již značně lepší výsledky, a to díky dříve spočteným datům.