

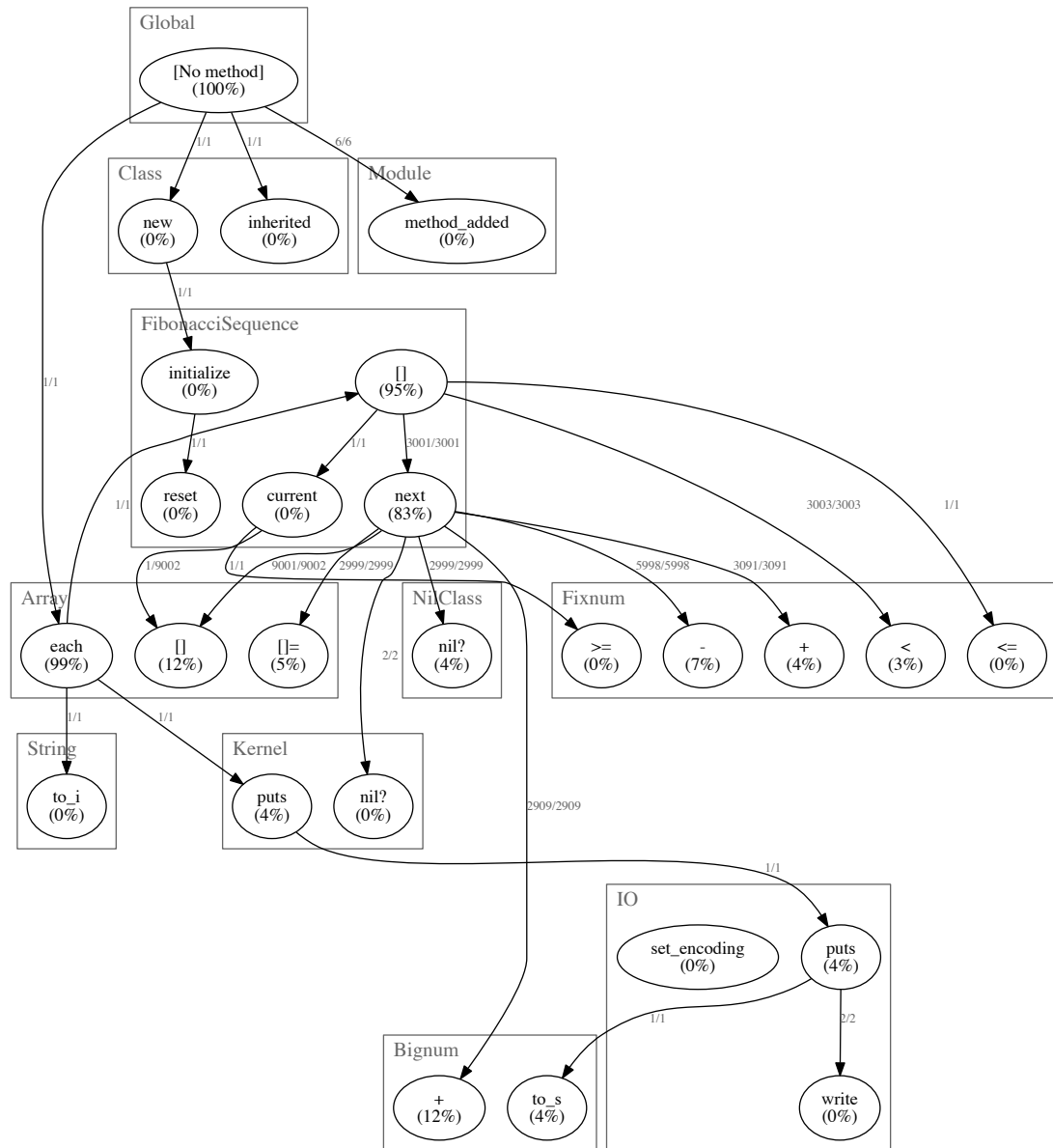
# *ZPRÁVA O PROFILOVÁNÍ*

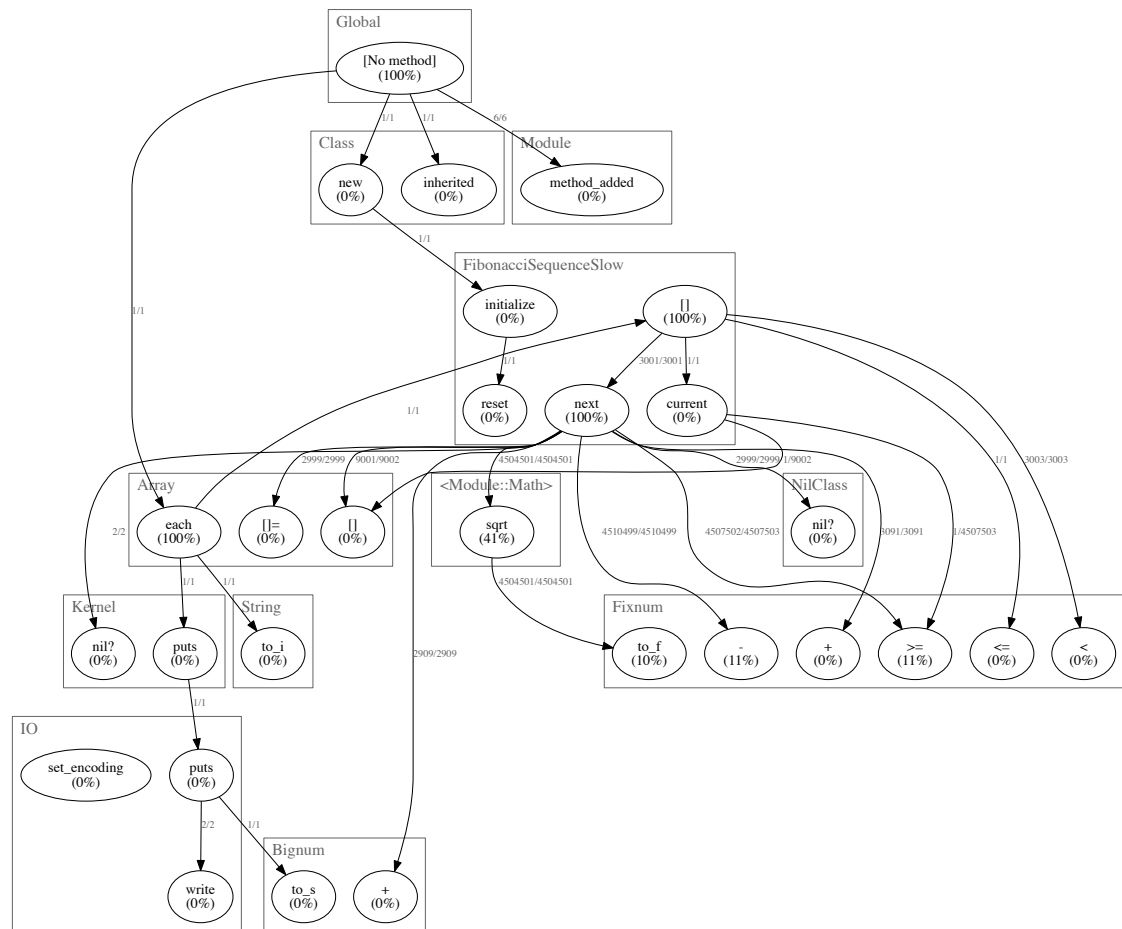
IVS 2016

*Štěpán Vích*

*4. březen, 2013*

# FibonacciSequence - Normální implementace





Z grafů výše je patrné, že metoda `Math.sqrt()` zabírá u zpomalené verze implementace nejvíce výpočetního času (okolo 40%) a je vhodné jí upravit, nebo v našem případě úplně odstranit, jelikož nemá v programu jakýkoliv význam. Čím větší bude parametr, který předáváme metodě [], tím větší bude časový rozdíl mezi rychlejší a pomalejší verzí implementace Fibonacciho posloupnosti. To je jednoduše vysvětlitelné tím, že čím je parametr větší, tím vícekrát se volá funkce `next()` a vícekrát se zopakuje metoda `Math.sqrt()`. Je dobré podotknout, že grafy zobrazující rozložení výpočetního času, byli generovány pomocí shell utilit `ruby-prof` a `dot`.

Štěpán Vích  
3. březen, 2016