FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Cvičení 4

KIV/VSS

Patrik Harag

harag@students.zcu.cz

A18N0084P, nar. 10. května

9. prosince 2019

Tabulka 1: Průměrné doby vykonání metody toArray v milisekundách

Kolekce	Velikost		
	100 000	1 000 000	10 000 000
java.util.ArrayList	0.079 ± 0.001	1.039 ± 0.002	11.041 ± 1.005
java.util.LinkedList	0.585 ± 0.004	11.675 ± 0.358	121.389 ± 0.098
java.util.Vector	0.079 ± 0.001	1.042 ± 0.002	11.109 ± 1.105

Zadání

6. Porovnejte rychlost operace toArray nad kolekcemi ArrayList a LinkedList s dostatečným množstvím dat.

Konfigurace prostředí

Hardware

- Notebook Lenovo G510,
- Intel Core i3-4000M 2.40 GHz,
- 8 GB RAM,
- WD Green 3D NAND SSD 240GB 2.5”.

Platforma

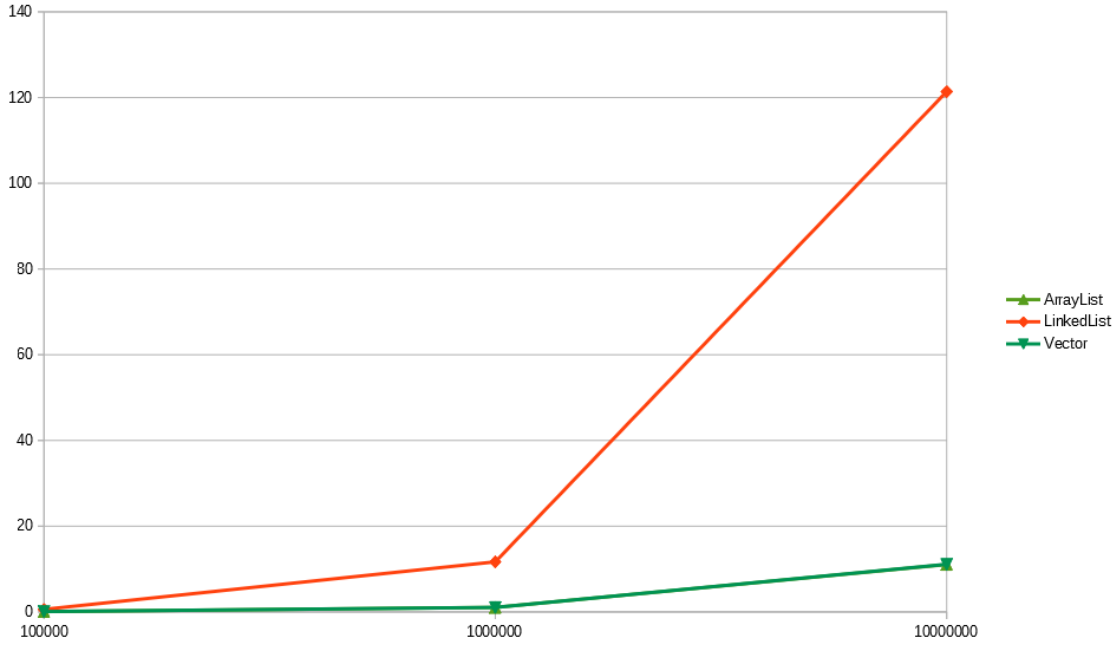
- Windows 10 Pro, 64 bit (10.0.18362 Build 18362)
- JDK 1.8.0_144
- Apache Maven 3.5.0

Způsob měření

Pro tvorbu benchmarku byl použit framework JMH. Konfigurace je kombinace velikosti kolekce (100 000, 1 000 000, 10 000 000) a typu kolekce (ArrayList, LinkedList, Vector). Kolekce se plní objekty typu Integer. Pro každou konfiguraci se provádí 5 warmup iterací a 10 měřených iterací. Iterace trvá 8 sekund. Celkem tedy benchmark trvá 18 minut.

Výsledky

Výsledky měření jsou zobrazeny v Tabulce 1. Graf na Obrázku 1 dále naměřené hodnoty vizualizuje.



Obrázek 1: Průměrné doby vykonání metody toArray v milisekundách

Závěr

Metoda toArray třídy java.util.ArrayList je rychlejší než metoda toArray třídy java.util.LinkedList, protože zkopírování bloku paměti je vždy rychlejší než iterace po prvcích a kopírování jedné hodnoty za druhou. Navíc byla změřena kolekce java.util.Vector, která dává, dle očekávání, podobné výsledky jako java.util.ArrayList. Téměř stejné jsou dokonce i odchylky.

I přesto, že měření byla provedena pouze pro tři různé velikosti kolekce, lze výsledků v Tabulce 1 pozorovat lineární závislost. Metoda toArray u kolekce s desetinásobným počtem prvků trvá vždy přibližně desetkrát tak dlouho.