FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Semestrální práce z KIV/VSP

Benchmarkování

David Košek
A14N0132P
26. 5. 1991
kosek@students.zcu.cz

1. Zadání

Porovnejte rychlost vyčištění (clear) kolekce ArrayList a LinkedList. Zajistěte aby kolekce byla dostatečně naplněná.

2. Popis testovacího prostředí

Popis HW, na kterém probíhalo testování.

Typ zařízení	Notebook Lenovo ThinkPad T520
Procesor	Intel® Core™ i7-2670QM CPU @2.20GHz
Operační paměť	8,00 GB (Použitelné: 7,89GB)
Disk	Samsung SSD 850 EVO 250G

Popis SW, na kterém probíhalo testování.

Verze systému Windows
- Windows 7 Professional
- Server Pack 1
- 64bitový operační systém

Programové a spouštěcí prostředí
- Eclipse IDE for Java Developers
- Version: Luna Service Release 2 (4.4.2)
- Build id: 20150219-0600
- JDK 1.7.0_79

JVM bylo spouštěno s parametry pro výpis garbage kolekce:

-Xms6g -verbose:gc -XX:+PrintGCDateStamps

Příkaz udává minimálně 6GB paměti a výpis do konzole.

3. Měření

Do každé z testovacích kolekcí jsem vložil 100 000 unikátních testovacích objektů. Test byl spuštěn vždy 1000.

Postup měření

1) Naplnění kolekcí testovacími daty
2) Zaznamenání startovacího času (pomocí metody currentTimeMillis)
3) Provolání clear metody nad danou kolekcí
4) Zaznamenání času po dokončení operace clear
5) Uložení naměřených hodnot pro pozdější statistiky
6) Bod 2 až 5 je proveden pro každou kolekci

Tabulka naměřených hodnot

Kolekce	Průměrný čas metody clear v ms	Rozptyl	Odchylka
ArrayList	0.094	0.895	0.946
LinkedList	1.099	10.478	3.236
Vector	0.027	0.208	0.456
HashMap	0.204	1.995	1.412

Použité vzorce pro jednotlivé výpočty

Střední hodnota (průměrný čas)

$$\frac{(\sum_{k=0}^{n} měření_k)}{počet\ měření}$$

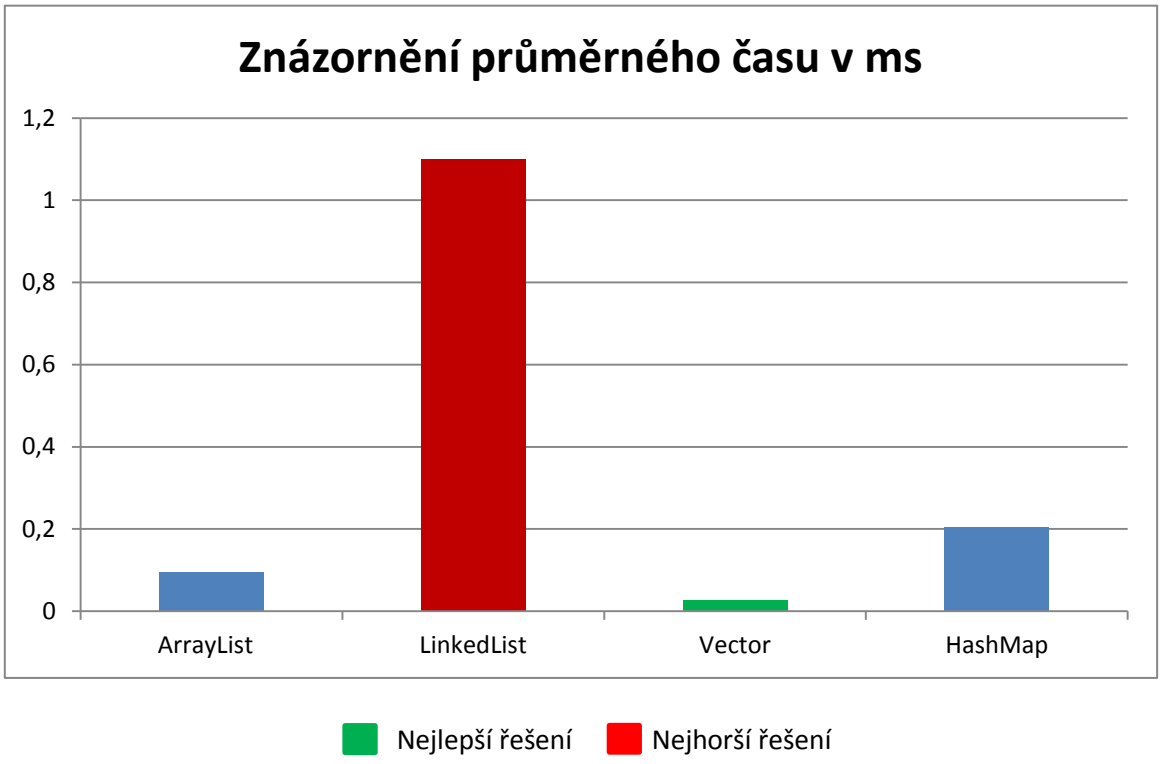
Rozptyl

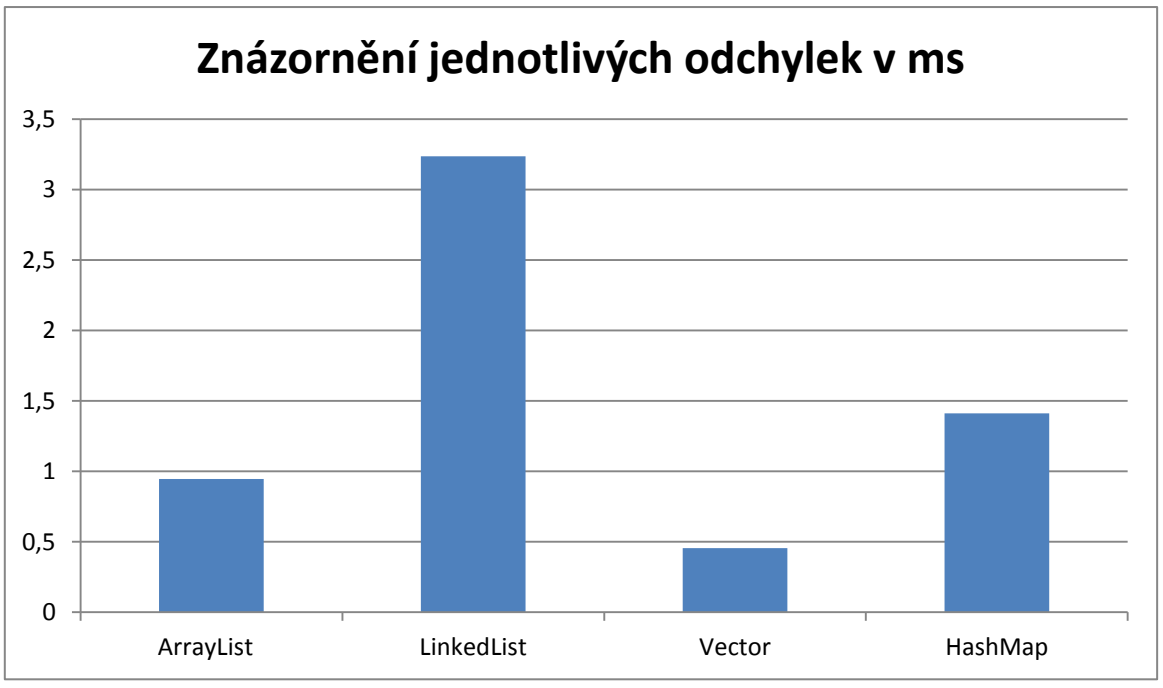
$$\frac{(\sum_{k=0}^{n} měření_k^2)}{počet\ měření} - (střední\ hodnota)^2$$

Odchylka

$$\sqrt{rozptylu}$$

4. Zhodnocení výsledků





- Čím menší odchylka, tím přesnější bylo každé měření.

5. Závěr

Pro měření času jsem chtěl využít z Java třídy System metodu nanoTime, protože je přesnější než známější metoda currentTimeMillis, ale její velká režie by nám test zpomalovala tudíž pro výsledné měření jsem použil druhou zmiňovanou metodu. Jako nejlepší z testovaných kolekcí bych prohlásil vector a poté arrayList, nejhůře dopadly výsledky LinkedListu. Při měření jsem nepřidával záměrně žádnou zátěž. Garbage kolekce se projevuje pro více měření. V našem případě pro 1000 měření a 100000 prvků se projeví pětkrát.