

Cvičenie 2 – LinkedList a Profiler

Cieľ:

Cieľom cvičenia je implementovať jednostranne zreťazený zoznam, vytvoriť iterátor pre túto štruktúru a otestovať implementáciu v testovacej aplikácii. Dôraz sa pritom opäť kladie na efektívnu implementáciu štruktúry. Na záver porovnáme implementáciu zoznamu pomocou poľa a pomocou zreťazenia.

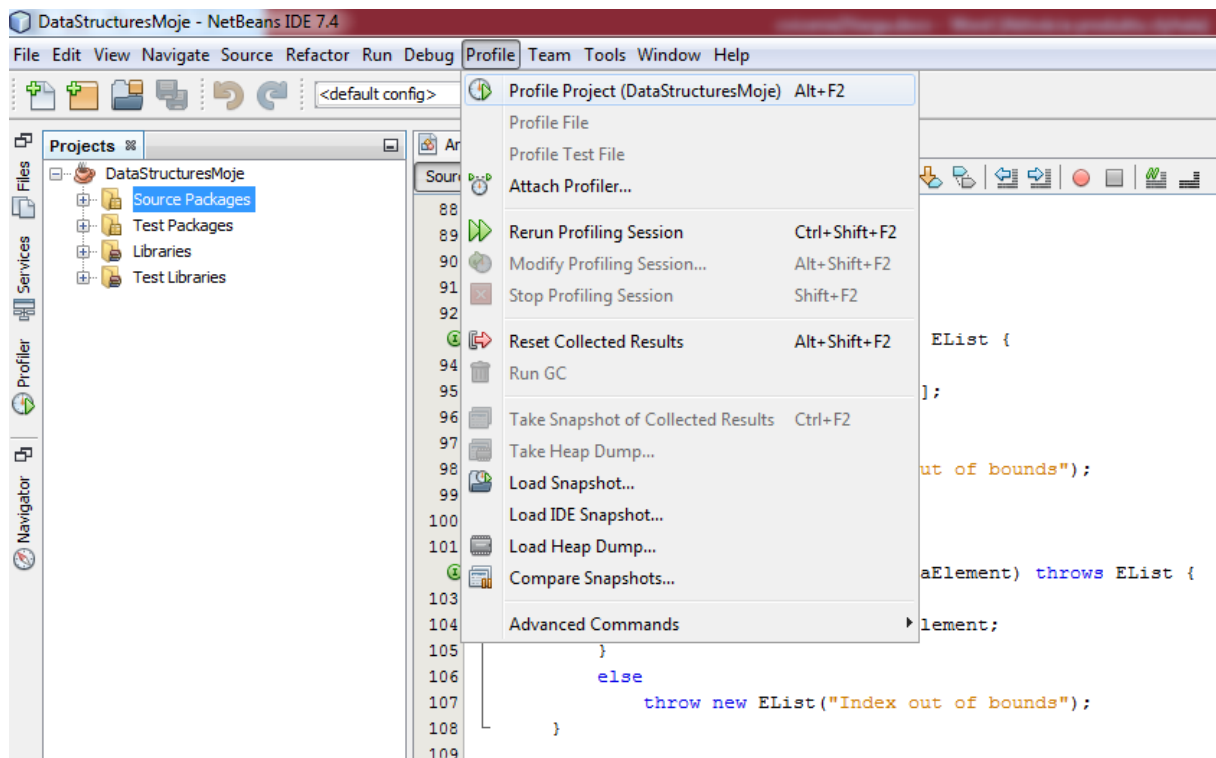
Postup:

1. Do balíčka `datastructures.lists` vytvorte balíček `datastructures.lists.LinkedList` a pridajte doň dve triedy:
 - a. `LinkedList`, ktorá implementuje rozhranie `IList<E>` (poznáte z predchádzajúceho cvičenia).
 - b. `LinkedListIterator`, ktorá implementuje rozhranie `Iterator<E>` (tiež poznáte z predchádzajúceho cvičenia).
2. Implementujte rozhrania v týchto triedach tak, aby bol `LinkedList` jednostranne zreťazený zoznam. (Možno budete potrebovať ďalšie triedy, tiež ich pridajte do tohto balíčka. Dôraz kladte na použitie správnych modifikátorov prístupu – `private`, `protected`, `public`, `package`.)
3. Do triedy `FrameMain` pridajte konečný atribút typu `LinkedList`, ktorý ste implementovali na tomto cvičení. Ako generický parameter zvolte triedu `Person` z balíčku `data` (prípadne vlastnú triedu dátovú, ak ste sa tak po minulom cvičení rozhodli).
4. V konštruktore triedy `FrameMain` vytvorte inštanciu Vášho `LinkedList`-u.
5. Zaregistrujte Vašu štruktúru do ovládacieho panelu pomocou volania `panelLinkedList.init(aLinkedList,aEditorPerson)`. Ak ste pracovali správne môžete štruktúru ovládať z užívateľského rozhrania.
6. Porovnajte výkon Vášho `ArrayList`-u a `LinkedList`-u. Zistite, na ktoré typy úloh sa viac hodí `ArrayList`, a na ktoré `LinkedList`.

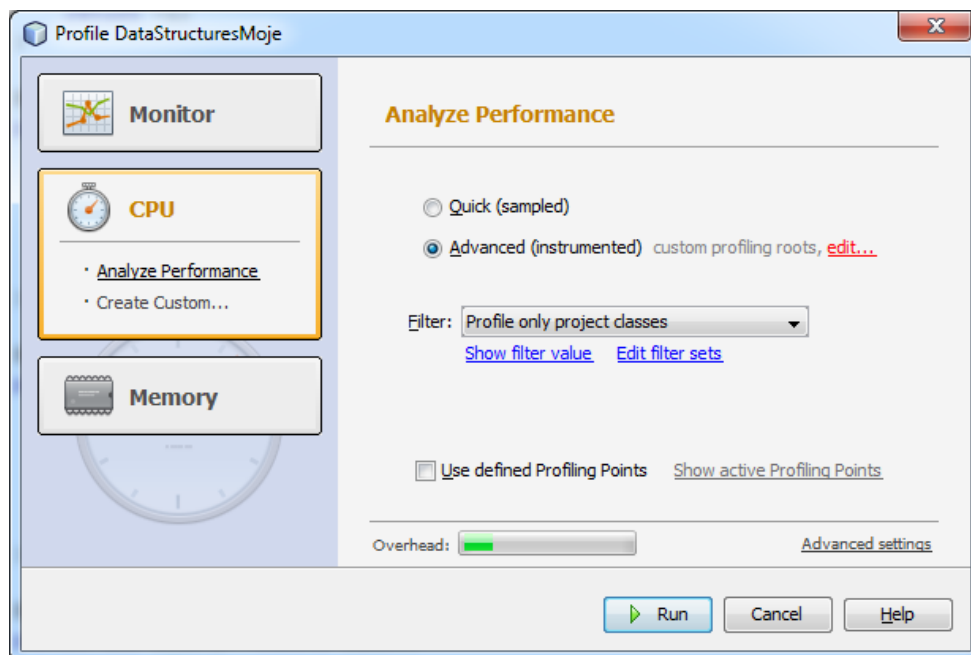
Čo sa dnes môže hodiť:

Testovacia aplikácia Vám dáva orientačnú informáciu o dĺžke trvania jednotlivých operácií. Pre podrobné merania (a porovnanie) sa však používajú tzv. *profiler-y*. Profiler si v Netbeans vytvoríte veľmi jednoducho:

- 1) V menu Profile si vyberiete Profile Project:



- 2) Vyberte si monitor CPU, Filter nastavte na triedy tohto projektu, nastavte Advanced mód a kliknite na [edit...](#)



- 4) Zavriete dialóg cez OK a na predchádzajúcom dialógu (krok 2) stlačte run (spustíte aplikáciu).
- 5) V ľavej časti uvidíte ovládacie prvky pre profiler (preklikajte si všetky). V Live Results sa budú postupne ukazovať časy pre jednotlivé funkcie. Spustíte Komplexný test v aplikácii pre aspoň 100 000 operácií a v pravej časti okna uvidíte výsledky. Takto ľahko porovnáte výkon jednotlivých operácií pre jednotlivé štruktúry a zodpoviete úlohu 6 dnešného cvičenia.



Čo môžete spraviť doma:

Okrem jednostranne zreťazených zoznamov existujú obojstranne zreťazené zoznamy (prvky majú referencie aj na predchodcu) a cyklické zoznamy (posledný prvok ukazuje na prvý a prvý na posledný). Pri týchto implementáciách sa niektoré operácie podstatne zjednodušia. Implementujte štruktúry *DoubleLinkedList* a *CyclicalList* (každé vo svojom balíčku, budú podporovať generický interface *IList* a svoj iterátor). Zaregistrujte ich do rozhrania (rovnako ako *ArrayList* alebo *LinkedList*). Porovnajte výkon týchto štruktúr so štruktúrami, ktoré ste implementovali v škole. Pri správnom využití dedičnosti štruktúr je to veľmi jednoduché. Opäť si položte a zodpovedzte otázku, ktorá štruktúra sa na čo hodí.