Loend List

* Sailitab elementide lisamisjarjestuse
* Loendi keskele lisamine/kustutamine

• Realisatsioonid:

– ArrayList

• muutuva pikkusega massiiv

• kiire juhupöördumine (indeksi abil)

• elemendi keskele lisamine või keskelt kustutamine on aeglane

– LinkedList

• üksteisele viitavate objektide kogum

• mõistliku kiirusega jadapoordumine

• aeglane juhupöördumine

• elemendi keskele lisamine/kustutamine on kiire

Hulk Set

• Hoiab unikaalseid elemente

– Tüüpoperatsioon: otsing, kas objekt q kuulub hulka: contains(q)

– Tuvastatakse equals() ja hashCode() meetodi abil

• Realisatsioonid

– HashSet

• Kiire lisamine\* ja pöördumine / otsing: *O(1)*

• Andmestruktuuriks paisktabel (*hashtable*)

• Eeldab hashCode() realiseerimist

– TreeSet

• Andmestruktuuriks puu

• Moistliku kiirusega lisamine ja otsing: *O( log(N) )*

• Elemendid on puus automaatselt sorteeritud

– LinkedHashSet

• Poordumiskiirus samavaarne HashSet'iga

• Säilitab elementide lisamise järjekorra

Sõnastik Map

• Ingl.k. *associative array, dictionary, map*

• Hoiab võti-väärtus paare

– Võtmed on unikaalsed (Set)

– Väärtus võib korduda

– Lisamisjärjekorda ei säilita

• Realisatsioonid

– HashMap

• Võtmed hoitakse paisktabelis

– TreeMap

• Võtmed hoitakse puustruktuuris

• Kasutamine

– Lisamine: put(Object key, Object value)

– Eemaldamine: remove(Object key)

– Vaartuse kusimine: get(Object key)

– Otsing: containsKey(Object o) , containsValue

Object.equals()

Objekte vordlev equals() meetod peab vastama järgmistele tingimustele:

– Refleksiivsus – objekt on iseendaga alati võrdne

x: x.equals(x)

– Sümmeetrilisus – ei ole vahet, kas ma võrdlen x ja y-it või y ja x-i

x, y: x.equals(y)y.equals(x)

– Transitiivsus – kui esimene objekt on vordne teisega ja teine kolmandaga, peab ka

esimene objekt vorduma kolmandaga

x, y, z: x.equals(y) ∧y.equals(z)⇔x.equals(z)

– Kooskolalisus – equals() meetodi kaivitamine ei tohi muuta võrreldavaid objekte

x, y: sama tulemus korduva x.equals(y) korral

– Mittetühisus – ukski objekt ei vordu null-iga

x ≠ null: x.equals(null) == false

hashcode()

• HashMap / HashSet korral kohustuslik!

– Korrektne too ja joudlus soltub hashCode() realisatsioonist

– hashCode() paigutab paisktabelis objektid erinevatesse kimpudesse (*bucket*)

• Realiseerimine:

– Igal valjakutsel peaks tagastuma sama vaartus

– Erineva sisuga objektil soovitavalt erinev vaartus (annab

uhtlasema jaotuse)

– Kui on realiseeritud equals(), tuleb realiseerida ka hashCode()

• Kasutada samu andmevalju, mis equals() realisatsioonis

equals() ja hashCode() peavad olema kooskolas:

– x,y: x.equals(y) x.hashCode() == y.hashCode()

– Realiseerides ainult equals(), onnestub hulka lisada sisult duplikaadid ning sonastik ei toota ootusparaselt