

Array-luokka

Säiliöluokat

Olio-ohjelmointi

Lapin AMK / EM

Array-luokka

- System.Array-luokka sisältää monia taulukoiden käsittelyyn tarvittavia metodeja
- Array-luokan metodeja:
 - Binäärihaku: BinarySearch
 - Taulukon kopiointi: Copy
 - Taulukon tyhjentäminen/alkioiden alustus oletusarvoon: Clear
 - Ehdot täyttävän alkion haku: Find
 - Taulukon lajittelu: Sort

Säiliöluokat (Collections)

- Taulukoiden sijaan on usein tehokkaampaa ja kätevämpää käyttää ns. säiliöluokkia (collection classes)
- Säiliöluokat ovat valmiita toteutuksia dynaamisille tietorakenteille, kuten linkitetyille listalle, hajautustaulukolle, pinolle tai jonolle

Säiliöluokat (Collections)

- Säiliöluokkia on kahdenlaisia: ei-geneerisiä ja geneerisiä
- Ei-generiset säiliöt ovat nimiavaruudessa `System.Collections`
- Geneeriset säiliöluokat ovat nimiavaruudessa `System.Collections.Generic`

Ei-geneeriset säiliöluokat

- Ei-geneeriset säiliöön oliot tallennetaan Object-tyyppisinä viitteinä
- Koska ei-geneeriset säiliöt voivat sisältää vain Object-viitteitä, niihin ei voi sellaisenaan tallentaa struktuureja
 - Ks. struktuurien paketointi
- Kun ei-geneerisestä säiliöstä haetaan alkio, on (yleensä) tehtävä tyyppimuunnos takaisin alkuperäiseen tyyppiin:
 - Esim. `String nimi = (String) myStack;`

Geneeriset säiliöluokat

- Uusissa ohjelmissa on suositeltavaa käyttää vain geneerisiä säiliötä!
- Säiliöön tallennettavien alkioiden tyyppi määritellään säiliön luomisen yhteydessä
 - Esim. `List<Opiskelija> opiskelijaLista = new List<Opiskelija>();`
- Geneeriseen säiliöön voi tallentaa sellaisenaan myös struktuureja
 - Esim. `List<int> numeroLista = new List<int>();`

Esimerkkejä säiliöluokista

- Ei-geneerisiä säiliöluokkia
 - ArrayList
 - Stack (System.Collections.Stack)
 - Hashtable
- Geneerisiä säiliöluokkia
 - List<>
 - LinkedList<>
 - Stack<> (System.Collections.Generic.Stack)
 - Dictionary<> (Hashtablen geneerinen vastine)
 - SortedDictionary<>

List<T>-säiliöluokka

- Esimerkkejä List-luokan metodeista:
 - Yksittäisen alkion lisäys listan loppuun: **Add**
 - Toisen säiliön alkioden lisäys listan loppuun: **AddRange**
 - Binäärihaku: **BinarySearch**
 - Listan tyhjentäminen: **Clear**
 - Ehdot täyttävän alkion haku: **Find**
 - Alkion lisäys listan keskelle: **Insert**
 - Alkion poisto listasta: **Remove**
 - Listan alkioden palautus taulukkona: **ToArray**

Säiliöluokan valinta

- Käytä List-luokkaa tavallisen taulukon sijasta
- Käytä LinkedList-luokkaa List-luokan sijasta, jos poistoja ja lisäyksiä (keskelle listaa) on paljon, mutta hakuja vähän
- Käytä Dictionary-luokkaa key-value (avain-arvo) –parien tallentamiseen
 - Esim. opiskelijanumero (string) / Opiskelija:
Dictionary<String, Opiskelija> opiskelijaLista = new Dictionary<String, Opiskelija>();

Säiliöluokan valinta

- Käytä geneeristä Stack-luokkaa pino-rakenteen toteuttamiseen
- Käytä SortedDictionarya, kun haluat pitää arvot lajiteltuina Key-arvon mukaiseen järjestykseen
- Kaikilla säiliöluokilla on runsaasti valmiita metodeja esim. alkioden lisääkseen, poistoon, hakuun, lajitteluun