

Viikkoraportti 5

Alkuperäisessä suunnitelmassa kaavailin toteuttavani sekä Huffman-koodaukseen että Lempel-Ziv – koodaukseen perustuvat tiivistysalgoritmit. Aikaisemmilla viikoilla olen kuitenkin keskittynyt nimenomaan Huffman-koodauksen toteuttamiseen. Viikolla 5 olen toteuttanut Lempel-Ziv –pakkauksen toteuttavan luokan LempelZivKoodaus sekä sen tarvitseman apuluokan Buffered12bitStream. Lisäksi olen toteuttanut Tarkastaja-luokan, jonka tehtävän on tarkastaa, että alkuperäinen tiedosto ja pakkauksen ja purkamisen tuloksena luotu tiedosto ovat identtiset.

Olen myös muokannut pääohjelmaa siten, että ohjelma kysyy nyt käyttäjältä ensin pakattavan tiedoston nimeä ja tämän jälkeen käytettävää pakkausalgoritmiä. Tämän jälkeen ohjelma pakkaa käyttäjän ilmoittaman tiedoston valitulla algoritmilla tiedostoon pakattu.dat ja purkaa tuon tiedoston sitten purettu.dat –tiedostoon. Pakkaamiseen ja purkamiseen kulunut aika mitataan, samoin alkuperäisen ja pakatun tiedoston pituus ja niistä laskettu pakkaussuhde. Tämän jälkeen vielä suoritetaan Tarkastaja-luokan metodi, joka kertoo, ovatko alkuperäinen tiedosto ja purettu.dat –tiedosto identtiset. Tarkastuksen tulos on 0, mikäli tiedostot ovat identtiset, ja jokin muu luku, mikäli näin ei ole.

Lisäksi olen testannut pakkausalgoritmejä erilaisilla testiaineistoilla, kuten tekstitiedostoilla, kuvatiedostoilla ja äänitiedostoilla.

Lempel-Ziv –toteutus on vienyt paljon aikaa tällä viikolla, ja JUnit-testaaminen on jäänyt sen takia jälkeen. Osin tämä johtuu myös siitä, että nykymuotoiset metodit ovat hankalasti testattavia, koska monet keskeiset metodit ottavat syötteenä tiedostojen käsittelyyn käytettävän olion, joten myös testaaminen vaatisi erityisesti testaamista varten luotujen syötetiedostojen käyttöä. Tämä puute on tarkoitus korjata refaktoroimalla koodia siten, että testaus on mahdollista ilman tiedostojen käyttöä.