

## Toteutusdokumentti

### Ohjelman yleisrakenne

Ohjelma koostuu java-koodista sekä sanasto-tiedostosta, joka annetaan xml-muotoisena. Ohjelma suoritetaan ajamalla crossword.crosswordmaker-pakkauksessa oleva Main-metodi. Käyttöliittymän avulla käyttäjä voi valita generoitavan krypton koon sekä aloitussanan. Ohjelman koodi on jaettu kolmeen pakkaukseen:

1. crossword.crosswordmaker  
Pakkauksessa on Main-metodin lisäksi käyttöliittymä.
2. crossword.lexicon  
Xml-muotoisen sanasto-tiedoston käsittely tapahtuu täällä.
3. crossword.logic  
Kaikki krypton generointiin tarvittava koodi löytyy tästä paketista.

Yksikkötestit ovat vastaavissa pakkauksissa.

### Saavutetut aika- ja tilavaativuudet

Ohjelman ydin on metodi, joka etsii jokaiseen krypton sanapaikkaan sanan. Tämä on toteutettu rekursion avulla niin, että jokaiseen sanapaikkaan kokeillaan jokaista siihen sopivaa sanaa (varatut kirjaimet huomioiden) kunnes päädytään lopputulokseen, jossa krypton jokainen sanapaikka on täytetty. Pseudokoodina:

boardOfWords ja positions ovat globaaleja muuttujia

```
layWords(index)
  if index == positions.size
    return boardOfWords
  position = positions[index]
  fittingWords = findWords(position)
  for each word in fittingWords
    boardOfWords.drawWord(word)
    solution = layWords(index + 1)
    if (solution != null) return solution
  return null
```

Koska kryptolauta on globaalimuuttuja, muistissa on kerrallaan ainoastaan yksi kryptolauta. Jos merkitään, että  $l$  = kryptolaudan leveys ja  $k$  = laudan korkeus, niin tilavaativuus on  $O(l \cdot k)$ .

Aikavaativuus on teoriassa eksponentiaalinen: jos jokaiseen sanapaikkaan pystyisi sijoittamaan  $n$  määrän sanoja ja kryptoon sijoitettavien sanojen määrää merkittäisiin  $s$ :llä, aikavaativuus olisi  $n^s$ , mutta käytännössä  $n$  pienenee hyvin nopeasti laudalle jo asetettujen sanojen rajoittaessa seuraavaksi lisättävien mahdollisten sanojen määrää.