

# Kasviplanktonin laskentaohjelma

Ajatuksenamme on toteuttaa kasviplanktonin laskentaohjelma mikroskooppikäyttöön. Ohjelmasta on aikaisempia versioita muun muassa C++, ActiveX, Delphi ja JavaScript/HTML -ympäristöistä. Käytetyt työvälineet ovat osittain vanhentuneita tai vaikeasti ylläpidettävissä. Pyrimme tässä löytämään joitakin vaihtoehtoja näiden tilalle.

Tarvitsemamme komponentit ovat helppokäyttöinen tietojen syöttöikkuna, mikroskooppikuvan yhtäaikainen näyttö, tietokanta tietojen tallentamiseen sekä mahdollisuus kuvien samanaikaiseen numeeriseen analysointiin.

Vaatumuksenamme on myös ohjelman toimivuus eri käyttöjärjestelmissä (itselläni Linux, biologeilla pääasiassa Windows-koneet).

## JavaScript ja HTML

Ohjelman prototyyppiversiossa työkaluina ovat olleet JavaScript ja HTML. Ohjelman kehitys on jäänyt puolitiehen, ja syykin tähän on varsin selvästi luettavissa lähdekoodista: JavaScriptin kasaantuva monimutkaisuus on tehnyt ongelmasta jotakuinkin mahdottoman ratkaista. Tietojen esittäminen sujuvasti JavaScriptillä vaatisi palvelinkielen tuen ja suunnittelun tulisi lähteä tietotyyppien suunnasta.

Pidämme JavaScriptin kuitenkin mukana yhtenä mahdollisuutena, mikäli käyttöliittymäkirjastojen asentaminen osoittautuu ongelmaksi. Tällöin toteutustavaksi tulee Haskell-kielinen verkkopalvelin, joka tietokannan avulla luo laskentaan käytettävän verkkosivun toimintoihin.

## Käyttöliittymäkirjastot

Linux- ja Windows-käyttöjärjestelmissä toimivina käyttöliittymäkirjastoina tulevat kysymykseen lähinnä Gtk, Wx ja Qt. Gtk-käyttöliittymäkirjasto on hankalasti siirtymävaiheessa (Gtk2/Gtk3), mutta muutoin vakiintunut varsinkin Linuxissa. Windowsiin helpoiten siirrettävä lienee Wx. Kuitenkin kaikki vaihtoehdot ovat kokeiltavissa eikä valinta välttämättä rajoitu näihin kolmeen.

## Haskell

Haskell on nykyaikainen käännettävä (ja tarvittaessa myös tulkattava) funktionaalinen ohjelmointikieli. Haskell-kieli on toiminut opetuskielenä funktio-ohjelmoinnin kurssilla, ja siten oletuskieli ohjelman toteutusta ajatellen.

Haskell-kielen pienen käyttäjäkunnan vuoksi graafiset käyttöliittymäkirjastot muodostavat jonkinasteisen ongelman.

Windowsissa suositeltava asennusvaihtoehto lienee *Haskell-platform*. Haskell-platform sisältää muun muassa GHC-kääntäjän (Glasgow Haskell Compiler) ja sen vuorovaikutteisen tulkin GHCI. Kääntäjäympäristö vie kiintolevytilaa noin 700 MB.

Asennuksen jälkeen hyvä kokeilukohde on Haskell-kielen vuorovaikutteinen tulkki `ghci`. Linux-ympäristössä tulkki käynnistyy komentotulkissa komennolla `ghci`. Seuraavassa on käynnistetty vuorovaikutteinen tulkki `ghci` ja käytetty tulkkia yksinkertaisena laskimena:

```
$ ghci
GHCi, version 7.8.4: http://www.haskell.org/ghc/  :? for help
> 64 * 12
768
```

Asennettaessa GTK-käyttöliittymäkirjasto kirjaston tuominen vuorovaikutteiseen tulkkiin sujuu ilman virheilmoituksia:

```
> import Graphics.UI.Gtk
>
```

Haskell-tulkki sulkeutuu komennolla :q

```
> :q
Leaving GHCi.
bash ~ $
```

## Python

Haskell-kielen vaihtoehtona on tulkattava ohjelmointikieli Python. Python on suuren käyttäjäkuntansa vuoksi hyvin tuettu ohjelmointikieli. Kaikki tunnetuimmat käyttöliittymäkirjastot (Gtk2, Gtk3, Wx, Qt) toimivat varsin hyvin Python-kielellä niin Windowsissa kuin Linuxissa. Python-kielen oma oletuskäyttöliittymäkirjasto on Tk. Verrattuna Haskell-kieleen Python-kieli ei ole aivan yhtä ilmaisuvoimainen, eikä kaikissa tapauksissa tulkattavana kielenä suorituskyvyltään samaa luokkaa (tosin poikkeuksiakin on).

Python-kielen vuorovaikutteinen tulkki toimii samaan tapaan kuin Haskell-kielessä. GTK-käyttöliittymäkirjastojen onnistuneesta asentamisesta myös Python-kielessä kertoo kirjastojen tuominen tulkkiin ilman virheilmoituksia:

```
$ python
Python 2.7.11 (default, Sep 29 2016, 13:33:00)
Type "help", "copyright", "credits" or "license" for more information.
>>> import pygtk
>>> pygtk.require('2.0')
>>> import gtk
>>>
```

Python-kieli on pitkään ollut siirtymässä versiosta 2 versioon 3, mutta siirtyminen ei ole sujunut täysin kitkattomasti. Tämän vuoksi esimerkiksi Linuxeissa oletuksena asentuva Python on versiota 2.

Python-tulkki sulkeutuu painamalla CTRL-D tai kirjoittamalla `quit()` komentokehotteeseen.

```
>>> quit()
bash ~ $
```

## Tietokanta

Tehtävään sovelias tietokantaohjelmisto on sulautettu Sqlite-tietokanta. Sqlite tallentaa tietokannan tavallisena tiedostona kiintolevyllä eikä siksi vaadi erillistä tietokantamoottoria tai käyttäjänhallintaa. (Käyttäjillä, joilla on oikeus tiedostoon, on oikeus tietokantaan.) Tietokantaan voimme sisällyttää käyttöliittymän, oletusasetukset, käännökset, kuvatiedostojen sijainnin ja laskennan tulokset. Sqlite-tietokanta toimii kaikissa tässä esitellyissä toteutusvaihtoehdoissa.

Haskell- ja Python-kielen tapaan Sqlite on mahdollista käynnistää myös komentoriviltä:

```
$ sqlite3 biomass.sqlite
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> .tables
cities          databaseFields  permutations    settings
sqlite> select * from databaseFields;
field|label
sampleID|Sample ID:
date|Date and time:
place|Location:
site|Site:
latitude|Latitude (N):
longitude|Longitude (E):
```

Sqlite-tietokannan tuloste on ihanteellinen myös dokumentoinnin kannalta taulukon tulostuessa asiallisen näköisenä jotakuinkin sellaisenaan:

field	label
sampleID	Sample ID:
date	Date and time:
place	Location:
site	Site:
latitude	Latitude (N):
longitude	Longitude (E):

Sqlite-tulkki sulkeutuu komennolla .q

```
sqlite> .q
bash ~ $
```

## Yhteenveto

Käsittelimme tässä seuraavia mahdollisuuksia ohjelmointikielen ja käyttöliittymäkirjaston valinnaksi:

- Haskell, Python, JavaScript/HTML
- Gtk, Qt, Tk, Wx
- Sqlite

Jatko riippuu pitkälle siitä, miten ohjelmointiympäristöjen asennus sujuu eri käyttöjärjestelmissä.