The trebuchet

Henrik Skov Midtiby
Department of Physics and Chemistry,
University of Southern Denmark
henrik@midtiby.dk

1. Introduction

Java, C++ and Python are on the top ten of the most used programming languages. There are hundreds of languages, but not minding the hard competition, new languages are still created with the thought of doing better. An example of this is the relatively new programming language Julia, which has been developed with the idea to combine the best features of other languages. The purpose of this project is to find out how well Julia perform compared to some of the standard languages as of 2016. One of those languages is Python which Julia draws a lot of inspiration from, including syntax and the dynamic type system. The second language that Julia will be compared with is Java, these two do share similarities, but most of the similarities are behind the scene mechanics, such as garbage collection and compiler optimizations. The developers of Julia claim that Julia is as fast as C. Therefore we choose C++ as the last language to compare Julia with.

2. Julia

Julia is an object-oriented programming language, which has been under development since 2009. First released in 2012 and the newest stable version of Julia is version 0.4.5. The language has been created because the developers wanted a language with all the features they like from other languages. The developers wanted the language to be open source, which means that everybody can read and modify the language. One of the ideas was to make the language as simple, readable and easy to learn as possible. The language is made for high performance and scientific computations while still supporting general purpose programming.

3. Project Euler problem 116

```
function solve(tileSize, blockSize) #m=color block
  size n = black box size
   if tileSize > blockSize
      return 0
   end
   solutions = 0

   for i = tileSize : blockSize
      solutions += 1
      solutions += solve(tileSize, blockSize-i)
   end

   return solutions
end

function calc(size)
   result = solve(2, size) #Red tiles
   result += solve(3, size) #Green tiles
   result += solve(4, size) #Blue tiles

   return result
end

size = parse(Int32, ARGS[1])

calc(size)
```

Figur 1: *hej*

The only file you have to modify is the file poster.tex.
If you compile it with pdflatex, you will be able to include bitmap figures (PNG, BMP and JPG) and PDF files by using the class graphicx (\useclassgraphicx at the beginning of the document and e.g. \includegraphics[width=10.5cm]./figure.png in the document body). For this option, it is not possible to include EPS or PS figures.
If you compile poster.tex with latex, you will be able to include only EPS and PS figures using the same class graphix. (The template poster.tex already contains an image, see how it is included).

$$\tan(x) = \frac{\sin(x)}{\cos(x)} \tag{1}$$
$$\sin(x) = \int_0^\infty \cos(a)da \tag{2}$$

We suggest you use pdflatex to compile your poster, as we are 100% sure that it produces nice looking posters that can be easily printed on plotters.

4. How to compile the poster

MAKE sure you have both `a0size.sty` and `sciposter.cls` in yout tex path or in the [2] current directory, then run `pdflatex` on this file. Voilà.

Litteratur

[1] An Author *A reference*. A paper.

[2] An Other Author *A reference*. Some paper.