Overview

1. Simple calculus calculating language for mathematical function evaluation, derivation and integration.


3. Dynamic and Strong typed.
Motivation

Why Calcul\textsuperscript{2}? 

1. Calculating Calculus
   - With powerful built-in math functions

2. Math-like Coding
   - Easy to understand at first glance

3. Studying Tool
   - For teachers and students on calculus

http://www.ppthi-hoo.com
Function Declaration: \( f(x) = x^2 \);

Function Evaluation: \( f(3) \);

Function Derivation: \( f'(x) \);

Function Integral: \( f(x)(1,3) \);

Multivariate Functions:

Declaration: \( g(x,y) = x + y \cdot 2 \);

Function Evaluation: \( g(1,2) \);

Function Derivation: \( g'(x) \).
```c
main()
{
    f(x) = 2 * x;
    g(x) = sin(x);
    h(x) = f + g;
    h;
    h'(x);
}
```

**output**

\[ 2 \times x + \sin(x), \quad 2 + \cos(x) \]
input.cul

Scanner?

Token

Parser?

Ast

Semantic?

Failed

Codegen

output.cpp

g++ compiler

a.exe

calcul2.h
Implementation

Development Tools/Environment

- Eclipse
- Vim
- GitHub
- Linux/Mac/Windows
- Ocamlyacc/Ocamllex
- g++ Compiler
- Makefile
```cpp
int main()
{
    double printer;

    vector<string> f_var;
    f_var.push_back("x");
    vector<double> f_begin, f_end, f_now;
    FTree f(f_var);

    f.AddNode(new FNode(T_OP,0,PLUS));
    f.AddNode(new FNode(T_OP,0,POWER));
    f.AddNode(new FNode(T_VAR,0,0));
    f.AddNode(new FNode(T_VAL,3.));
    f.AddNode(new FNode(T_OP,0,TIMES));
    f.AddNode(new FNode(T_VAL,3.));
    f.AddNode(new FNode(T_VAR,0,0));

    f_now.clear();
    f.Derive("x") -> Print();
    cout << "\n";

    return 0;
}
```

Lesson Learnt

- Project Lives on Teamwork
- Plan Limited by Time
- Problems Lessen by Testing
- Practical Lively Things