

Assignment 1

Team number: 3

Team members

Name	Student Nr.	Email
Isaac Lewis	2687776	i.lewis@student.vu.nl
Martynas Vaznonis	2701013	m.vaznonis@student.vu.nl
Bekir Altun	2694301	b.e.altun@student.vu.nl
Matthijs Hulsebos	2681382	m.c.hulsebos@student.vu.nl

This document has a maximum length of 5 pages (excluding the contents above).

Introduction

Author(s): Bekir

Calculators are much used tools to make calculations on the fly. Online calculators are used by regular people just as much as they are used to make highly specialised calculations. Out of simplicity, many people use calculators in their web browser as they have a lot of freedom on the internet which allows them access to specialised calculators. For convenience this is of course not the best approach, as looking for the best calculator can be a tedious task, and websites may be hosting annoying ads which are best avoided. As a solution we strive to offer a modular calculator programmed in Java, because its popularity makes integration simple.

The calculator itself will show its contents in a command line interface (CLI) in real time, as the other option of a GUI would take away from the possible freedom the modularity has to offer. The real time interface will make following your steps simple, and should offer a frustration-free experience. To make calculations fault proof, a system will be implemented to check the calculations and answers internally, to offer reliability. As a basic feature a calculation history with the option to undo/redo your calculations will be available, as this will make rechecking calculations simple and offer additional reliability above what the internal system checks have to offer. Finally, on top of all, the calculator will support third-party plugins to make the calculator highly customizable in a simple way. If the calculator plugins aren't simple to develop, there would be no point in using it as making your own calculator would be better at that point. That is why it is important to put major focus on the modular aspect of the calculator.

As the calculator is highly specialised and many simple calculators already exist, the calculator will mostly be aimed at people with special interest. Engineers, programmers and creative content creators should be able to use the system with ease as they mostly have

experience coding anyway. At the same time, accountants or people working in a financial field should also be able to use the calculator with relative ease, as they also often need specialised calculations. In this way we try to offer a simple modular calculator to people that need specialised, hassle-free calculators.

Finally, as inspiration we refer to existing models. An example of this is found at <https://www.jq22.com/demo/jcalculator/>, which makes integration of the calculator extremely simple and yet is very effective at what it does. Unlike the above example, we will offer a CLI instead of a GUI and make the calculator more modular, making specialised usage more likely. The base calculator should thus be simple to implement into your own website or web browser, and adding plugins should be just as simple as implementing the calculator itself. The development of plugins is something left over to third-party developers, but programming in the domain of the calculator should be an intuitive task. Basic arithmetic should be available as a prebuilt plugin, both as an example and as something customizable to offer maximum freedom.

Features

Author(s): Martynas

Functional features

As a preamble to the table, you can discuss the main line of reasoning you applied for defining the functional features provided in the table.

In the **champion** field you put the name of the team member who is responsible for the modelling/implementation of the feature during the whole project (from assignment 1 to assignment 3).

ID	Short name	Description	Champion
F1	Plugins	The functionality of the calculator can be augmented with 3rd party extensions	Martynas
F2	CLI	The user can enter operands and operators from any of the modules he has and the interface will show the result dynamically	Isaac
F3	Arithmetic	The base version of the calculator comes prepackaged with arithmetic modules	Matthijs
F4	Undo	The user can undo their computations one by one	Bekir
F5	Store	Online plugin store where users can browse and download plugin	Isaac

Quality requirements

Author(s): Isaac

As a preamble to the table, you can discuss the main line of reasoning you followed for coming up with the quality requirements listed below.

ID	Short name	Quality attribute	Description
QR1	Commands sanity checks	Reliability	Command inputs should be verified to be valid
QR2	Extensible calculation	Maintainability	The calculation mechanism shall be easily extensible to allow for new calculation methods
QR3	Instantaneous results	Performance	There should not be noticeable delay for basic calculations
QR4	Correct results	Reliability	The calculations should not have any errors

Each quality requirement must be tagged with the corresponding quality attribute (see corresponding slides of the first lecture for knowing them).

Maximum number of pages for this section: 2

Java Libraries

Author(s): Matthijs

This is an initial overview of the libraries that will likely be incorporated in the project. The libraries and dependencies can change over time.

[PF4J](#)

PF4J enables the creation and management of third-party plugins for the calculator. The library essentially functions as a microframework which focuses on retaining a simple core. PF4J's lightweight, simple, and extensible properties were the primary reason why we have chosen to incorporate it in our tech stack.

[Log4j 2](#)

Logging is to be implemented with the Apache Log4j 2 library, the successor of the Log4j library. Logs can be used to keep a history of all computations by the user and the events in the application. This will both help with debugging and reverting user's computations. Log4j 2 was chosen as it is one of the most widely used libraries with a large number of features.

[Java ASCII Render](#)

The calculator will be fully based in a CLI without a GUI. This library helps graph computational results and display other simple visualisations in the console. Java ASCII Render was chosen as it supports plotting and tables, both useful graphics for the calculator.

Time logs

Team number	3			
Member	Activity	Week number	Hours	
All	Team Meeting	1	2	
Martynas	Feature requirements	1	1	
Isaac	Quality Requirements	1	1	
Matthijs H.	Explore Java libraries	1	1	
Bekir A.	Finishing Intro	1	2	
All	Team Meeting	2	1	
All	Team Meeting	2	1	
Matthijs H.	Select required libraries	2	1	
		TOTAL	10	