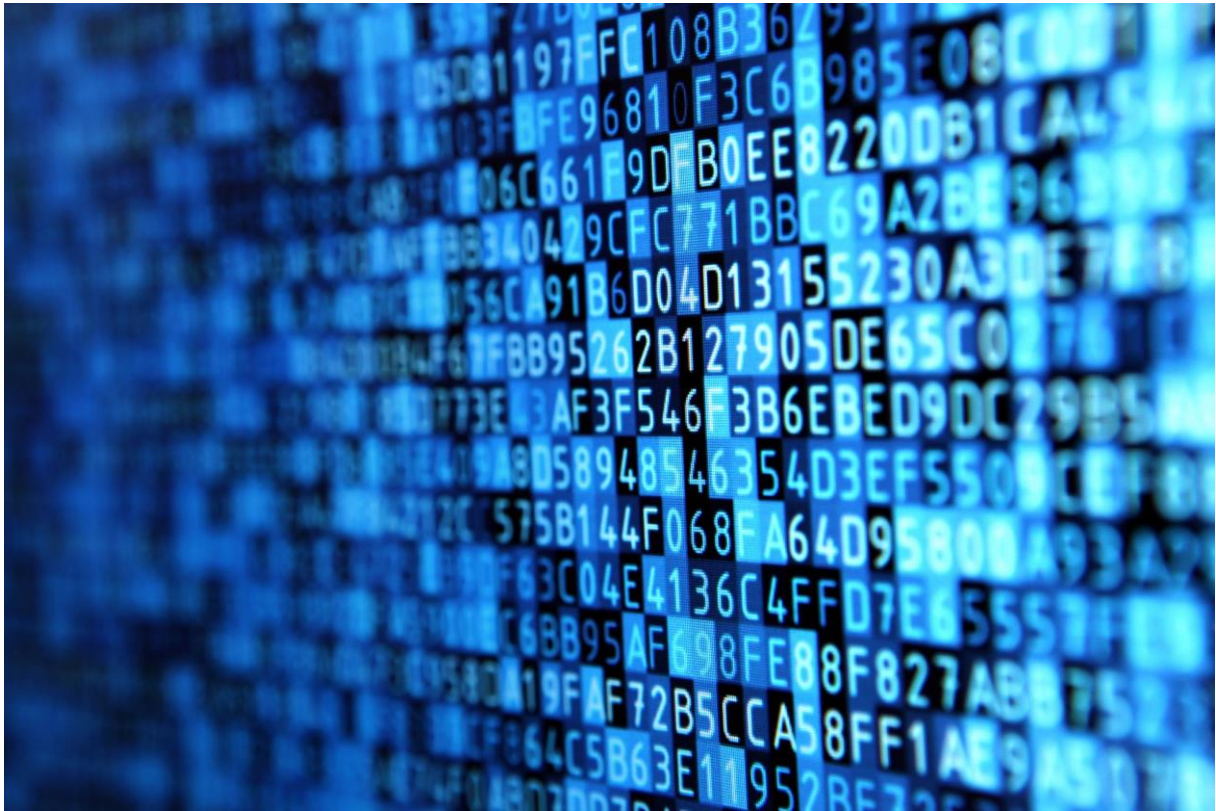


TESTDOCUMENT MALLAKA LANGUAGE



Tobias van den Hoogen

Studentnummer: 1742361

Datum: 13-11-2022

Versie: 1.0

INHOUD

Inleiding.....	3
Tests	3
Unit tests	3
Integration Tests	3
System Tests.....	4
Overige kwalitatieve criteria	4
Turing compleetheid	4
Opvangen van excepties.	5
Verwijzingen.....	6

INLEIDING

Dit document dient als testdocument van de Interpreter van de Mallaka Language waarin alle verschillende softwaretests toegelicht worden. De reden voor deze toelichtingen zijn om aan te tonen dat de code over voldoende kwaliteit beschikt.

TESTS

De Interpreter wordt op drie verschillende manieren getest. Namelijk met Unit tests, Integration Tests en System Tests. De reden voor het kiezen van deze tests is om de Functional Suitability van de software aan te tonen volgens de ISO 25000 standaard. (ISO 25000, sd). Hierdoor weet de gebruiker dat de Mallaka Interpreter functioneel werkt. Daarnaast kan hij ook zelf deze tests uitvoeren.

UNIT TESTS

Onder Unit tests wordt er verstaan dat de individuele modules apart worden getest (Software Testing Help, 2022). De reden waarom er voor Unit Testing is gekozen is om aan te tonen dat elke individuele module binnen de Interpreter werkt. Bij de Mallaka Language zijn er sprake van drie modules. Deze zijn Lexer, Parser en Interpreter. Bij elk van deze drie onderdelen worden de volgende functionaliteiten getest:

- Declareren van variabelen.
- Optellen van getallen.
- Aftrekken van getallen.
- Toepassen van de wortel.
- Delen van getallen.
- Toepassen van haakjes.
- If Statements.
- For loops.
- Until Loops.
- Functie declaratie.
- Functie Toepassing.

Hierbij wordt er per module het volgende getest:

- **Lexer:** Controleren of bij de tekstinvoer (de normale invoer) de juiste tokenlijst uitkomt.
- **Parser:** Controleren of bij de tokenlijst de juiste Parse Tree uitkomt.
- **Interpreter:** Controleren of bij de Parse Tree de juiste resultaat uitkomt.

INTEGRATION TESTS

Met Integration tests wordt er gecontroleerd of het combineren van de verschillende modules de gewenste functionaliteit aanbiedt. Hiermee wordt aangetoond dat het combineren van de verschillende individuele modules de gewenste uitvoer geeft. Bij de Mallaka Language is er sprake van twee combinaties, namelijk de Lexer -> Parser en de Parser -> Interpreter. Bij elk van deze twee onderdelen worden de volgende functionaliteiten getest:

- Declareren en samenvoegen van twee lijsten.
- Boolean operaties (AND, OR, NOT)
- Functie declareren en toepassen.
- Delen en machtsverheffing.

Hierbij wordt er per onderdeel het volgende getest:

- **Lexer -> Parser:** Controleren of bij de tekstinput de juiste Parse Tree uitkomt.
- **Parser -> Interpreter:** Controleren of bij de Token lijst de juiste resultaat uitkomt.

SYSTEM TESTS

Bij System Tests wordt het hele systeem getest. Oftewel, alle modules die aan elkaar verbonden zijn. Hierdoor kan worden aangetoond dat het gehele project functioneel is. Hierdoor kan het project dan ook (mits extra gestelde requirements) gedeployed worden. Bij de Mallaka Language geldt het volgende systeem: Lexer -> Parser -> Interpreter. Bij dit systeem wordt er de volgende functionaliteiten getest:

- Het berekenen van de gemiddelde waarde van een lijst.
- Het simuleren van een 4-input NOR Functie.

Hierbij wordt het volgende getest:

- **Lexer -> Parser -> Interpreter:** Controleren of bij de tekstinput de juiste resultaat geeft.

OVERIGE KWALITATIEVE CRITEREA

Alhoewel de testen al enige vormen van kwaliteit aantonen aan de code van de Interpreter. Zijn er ook andere onderdelen binnen de Mallaka Language die de kwaliteit van de code aantonen. Dit zijn de volgende:

- Turing compleetheid.
- Opvangen van excepties.

In de komende secties worden deze twee onderdelen verder toegelicht.

TURING COMPLEETHEID

Volgens (Binance Academy, sd) is een programmataal Turing compleet als hij de taken van een Turing machine kan uitvoeren. Bij de Mallaka Language is daar van sprake. Hierbij voldoet de Mallaka Language aan het volgende:

Requirement	Functionaliteiten binnen de Mallaka Language
De mogelijkheid om data te lezen en op te slaan.	Variabelen.
De mogelijkheid om data elkaar te laten verwijzen. (De head/write head simuleren)	Lijsten.
De mogelijkheid om een Finite State Machine te simuleren.	Functies en If-statements.
De mogelijkheid om een halt-status te simuleren.	For loops en Until statements.

OPVANGEN VAN EXCEPTIES.

Om ervoor te zorgen dat de gebruiker duidelijke meldingen krijgt als er iets foutgaat binnen de Mallaka Language worden er excepties opgevangen. Dit ondersteunt de Maintainability van de ISO 25000 standaard (ISO 25000, sd). Bij de Mallaka Language worden de volgende excepties opgevangen.

Excepties	Toelichting	Voorbeeld
InvalidCharError	Een van de karakters binnen de invoer is ongeldig en komt niet voor binnen de Mallaka Language.	#jklfdjalkdjf
InvalidSyntaxError	De syntax is ongeldig.	carrot +;
DivisionByZeroError	Er wordt door nul gedeeld.	a = 8 / 0;
NotDefinedError	Een opgeroepen variabele is niet gedefinieerd.	7 + a;
ConditionError	De gegeven statement is geen conditie.	whatif(a = 10)-><-;
InvalidParameterName	De parameter naam is ongeldig.	process a {7};
InvalidNumberOfArguments	Bij het oproepen van een functie worden er meer of minder parameters gegeven dan gedefinieerd zijn.	process a {b}->(b);<-; run a {1, 3, 5};
InvalidOperation	De operatie tussen twee datatypen is ongeldig.	[1,2,3] ^ 10.0;
OutOfBoundsIndex	Er wordt buiten een lijst geïndexeerd.	[1,2,3] \$ 5;
UnexpectedEndOfFile	Het lezen van de bestand is onverwacht geëindigd.	a = 10
InvalidInput	De invoer is ongeldig.	include "";
FileNotFound	Het gegeven bestand is niet gevonden.	include "filedoesnotexist.ma"
DeveloperException	Deze error wordt alleen gegeven als er met de code wordt aangepast waardoor de functionaliteit van de code in het geding komt.	

VERWIJZINGEN

Software Testing Help. (2022, 10 25). *The Differences Between Unit Testing, Integration Testing and Functional Testing*. Opgehaald van softwaretestinghelp.com: <https://www.softwaretestinghelp.com/the-difference-between-unit-integration-and-functional-testing/>