|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт искусственного интеллекта | | |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры | | |

|  |  |
| --- | --- |
| **КУРСОВОЙ ПРОЕКТ**  **по дисциплине «Функциональное программирование»** | |
| на тему «Функциональный калькулятор на функциональном языке программирования Lisp» | |
| Студент группы            КММО-11-24 | *О. А. Конилец* |
| Руководитель | Г. А. Милонов |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2025 г. | *(подпись студента)* |
|  |  |  |
| «Отчет утвержден.  Допущен к защите» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2025 г. | *(подпись руководителя)* |

Москва 2025

**Оглавение**

[Техническое задание 3](#_Toc199457577)

[Аннотация 3](#_Toc199457578)

[Требования к функциональным характеристикам 3](#_Toc199457579)

[Требования к надёжности программы 4](#_Toc199457580)

[Требования к составу и параметрам технических средств 4](#_Toc199457581)

[Порядок контроля и приёмки 5](#_Toc199457582)

[Структура программного обеспечения 5](#_Toc199457583)

[Алгоритм работы программы 5](#_Toc199457584)

[Используемые методы 5](#_Toc199457585)

[Структура программы с описанием функций составных частей и связи между ними 6](#_Toc199457586)

[Структуры данных 6](#_Toc199457587)

[Входные данные 6](#_Toc199457588)

[Выходные данные 6](#_Toc199457589)

[Методика и результаты тестирования 6](#_Toc199457590)

[Приложения 9](#_Toc199457591)

[Исходный код 10](#_Toc199457592)

Техническое задание

Аннотация

Наименование программы – «Функциональный калькулятор на функциональном языке программирования Common Lisp».

Назначение программы – вычисление математических выражений с возможностью определения новых функций и последующим их использованием.

Краткая характеристика программы – данное программное средство предоставляет пользователю возможность вычислять математические выражения с использованием встроенных операций (+, -, \*, /, ^, %, | |, !), встроенных функций (lg, ln, log, sin, cos, tg, arcsin, arccos, arctg, d/dx) встроенных констант (e, π), а так возможностью определять новые математические функции, которые также можно будет использовать в вычисляемых математических выражениях. Область применения – научные исследования с проведением сложных математических расчётов, расчёт бизнес показателей (рентабельности проектов, оценка доходности, расчёт затрат на производство) и другие области деятельности, где необходимы вычисление выражений с применением настраиваемых функций.

Используемые при разработки инструменты – онлайн компилятор функционального языка Common Lisp (https://onecompiler.com/commonlisp).

Основание для разработки программы – учебный план по специальности 01.04.02 «Прикладная математика и информатика» для дисциплины «Функциональное программирование».

Требования к функциональным характеристикам

Требования к функциональности (подробное описание сценария использования):

1. Программное обеспечение должно предоставлять возможность корректного выполнения математических операций: сложение «+», вычитание и отрицательное значение числа «-», умножение «\*», деление «/», возведение в степень «^», остаток от деления «%», абсолютное значение «| |», факториал «!», а также скобки «()».
2. Программное обеспечение должно предоставлять возможность корректного вычисления математических функций: десятичный логарифм «lg», натуральный логарифм «ln», логарифм x по основанию y «x log y», синус «sin», косинус «cos», тангенс «tg», обратный синус «arcsin», обратный косинус «arccos», обратный тангенс «arctg», взятие производной «d/dx».
3. Программное обеспечение должно предоставлять возможность использования математических констант: число Эйлера (е), число Пи (π).
4. Программное обеспечение должно предоставлять возможность определение новых математических функций одной переменной x в формате «название\_функции = вычисляемое выражение функции». Название функций не должно совпадать с названиями встроенных функций.
5. Математические операции должны обрабатываться в общепринятом порядке.
6. В случае ошибки программное обеспечение должно выводить сообщение об ошибки и указать на неё.

Требования по организации входных и выходных данных, их передача в программу или из неё:

* Входные данные: вычисляемое математическое выражение с использование встроенных операций, встроенных функций, встроенных констант и определённых ранее функций; определяемая функция в формате «название = вычисляемое выражение» с возможностью использовать переменную х. Пробелы игнорируются.
* Выходные данные: в случае вычисляемого математического выражения – правильный ответ, в случае определения новой функции – выводится входная функция, означающая успешное определение новой функции, в случае ошибки – вывод ошибки с указанием места ошибки.
* Передача в программу и из неё происходит с помощью консоли (stdin/stdout).

Временные характеристики выполнения программы: нормальное выполнение вычисления выражений и определения новых функций не должно превышать 1 секунду за каждые 100 операций в вычисляемом выражении или новой определяемой функции. При этом новые определённые функции считаются не одной операцией, а числом встроенных операций, используемых при определении функции.

Требования к надёжности программы

В случае некорректной входной информации программа будет выводить сообщение о ошибке с указанием места ошибки.

Требования к составу и параметрам технических средств

Для запуска программы необходим стабильный доступ в интернет к сайту <https://onecompiler.com/commonlisp>, куда нужно вставить исходный код программы, ввести входную информацию и запустить программу.

Порядок контроля и приёмки

При приёмке программы должны быть проведены тесты. Тесты должны демонстрировать соответствие программы каждому функциональному требованию. Корректность работы и правильность результатов теста будет проверяться с помощью ожидаемых результатов.

Структура программного обеспечения

Алгоритм работы программы

Используемые методы

Для данного программного обеспечения была разработана следующая грамматика:

E := T1 E'

E' := +T1 E' | -T1 E' | eps

T1 := T2 T1'

T1' := \*T2 T1' | /T2 T1' | ^T2 T1' | %T2 T1' | eps

T2 := T21 T3 T22

T21 := -T21 | eps

T22 := !T22 | eps

T3 := N | (E) | lg(E) | ln(E) | (E)log(E) |

sin(E) | cos(E) | tg(E) | arcsin(E) | arccos(E) |

arctg(E) | d/dx(E) | fn(E) | |E|

N := num | pi | e | x

В соответствии с данной грамматикой присутствуют методы E, E', T1, T1', T2, T21, T22, T3, N.

Остальные методы:

* functionalCalculate – главный метод, запускающий функциональный калькулятор.
* set1, set2, set3, set4 – вспомогательные методы, вставляющие элемент в список на 1, 2, 3 и 4 позицию соответсвующе.
* NNegDegreeSF1, NNegDegreeSF2, NChange – вспомогательные методы, помогающие парсить число.
* Factorial – вычисляет факториал числа.
* errorIndex – вспомогательный метод, помогающий вывести сообщение об ошибке.
* digitChar, charWOlen, subseqWOlen – вспомогательные методы для digit-char-p, char и subseq соответственно. Предотвращают ошибки.
* derivative – функция взятия производной.
* checkFunction – проверяет наличие пользовательских функций на данной позиции.
* executeFunction – исполняет пользовательскую функцию.

Структура программы с описанием функций составных частей и связи между ними

Структуры данных

Входные данные

Представляют собой данные, вводимые во время исполнения программы (пробелы игнорируются):

* Вычислимые математические выражения, аргументы функций всегда в скобках, пример: 1+ 2 \* 3, 12%10, ln(e), c(e).
* Определение математических выражений в формате «название\_функции=вычислимое выражение».

Выходные данные

Представляют собой данные, выводимые во время исполнения программы:

* Результат вычисления математических выражений, например: 1, 42, 3x^2, -sin(x).
* Вывод определения функции, означающий успешное определение пользовательской функции, например: a=x, b=sin(x), c=(x^3+lnx).
* Вывод ошибок с указанием места ошибки, например:

Ошибка – деление на 0:

1/0

^

Методика и результаты тестирования

Для запуска тестов необходим стабильный доступ в интернет к сайту <https://onecompiler.com/commonlisp>, куда нужно вставить исходный код программы, ввести входную тестовую информацию, запустить программу и сравнить с ожидаемым результатом.

| Проверяемые требования | Входные данные | Выходные данные | Ожидаемый результат | Соответствие ожидаемым результатам |
| --- | --- | --- | --- | --- |
| Математические операции сложения, умножения, скобки корректный порядок операций, игнорирование пробелов, обработка ошибок | 1+ 2 \* 3  4 \* 2 + 42  (1+2)\*3  4\*(2+42)  42\*0  42\*  42\*+ | 7  50  9  176  0  Ошибка(42\*)  Ошибка(\*+) | 7  50  9  176  0  Ошибка(42\*)  Ошибка(\*+) | Да |
| Математические операции вычитания и деления, скобки корректный порядок операций, обработка ошибок | 3–15/5  45/5-12  (3-15)/5  45/(5-12)  0/13  0/0 | 0  -3  -2.4  -6.428571  0  Ошибка(0) | 0  -3  -2.4  -6.428571  0  Ошибка(0) | Да |
| Математическая операция возведения в степень, отрицательное значение числа | 2^10  2^-3  3^0 | 1024  0.125  1 | 1024  0.125  1 | Да |
| Математическая операция остаток от деления, отрицательное значение числа, обработка ошибок | 12%10  -12%10  10%-10  10%0 | 2  8  0  Ошибка(0) | 2  8  0  Ошибка(0) | Да |
| Математическая операция абсолютного значения | |1|  |0|  |-1| | 1  0  1 | 1  0  1 | Да |
| Математическая операция факториала, обработка ошибок | 5!  0!  1!  -1!  0.25!  -0.25! | 120  1  1  Ошибка(-1)  Ошибка(0.25)  Ошибка(-0.25) | 120  1  1  Ошибка(-1)  Ошибка(0.25)  Ошибка(-0.25) | Да |
| Математическая функция десятичного логарифма, обработка ошибок | lg(1)  lg(10)  lg(0)  lg(-1) | 0  1  Ошибка(0)  Ошибка(-1) | 0  1  Ошибка(0)  Ошибка(-1) | Да |
| Математическая функция натурального логарифма, число Эйлера, обработка ошибок | ln(1)  ln(e)  ln(0)  ln(-1) | 0  1  Ошибка(0)  Ошибка(-1) | 0  1  Ошибка(0)  Ошибка(-1) | Да |
| Математическая функция логарифм x по основанию y, обработка ошибок | 1 log 42  10 log 10  100 log 10  10 log 1  -10 log 10  10 log -10 | 0  1  2  Ошибка(1)  Ошибка(-10)  Ошибка(-10) | 0  1  2  Ошибка(1)  Ошибка(-10)  Ошибка(-10) | Да |
| Математическая функция синуса, число Пи | sin(0)  sin(1)  sin(π/2)  sin(-π/2) | 0  0,8414709848  1  -1 | 0  0,8414709848  1  -1 | Да |
| Математическая функция косинуса, число Пи | cos(0)  cos(1)  cos(π/2)  cos(-π) | 1  0,54030230586  0  -1 | 1  0,54030230586  0  -1 | Да |
| Математическая функция тангенса, число Пи, обработка ошибок | tg(0)  tg(π/4)  tg(-π/2)  tg(3π/2) | 0  1  Ошибка(-π/2)  -1 | 0  1  Ошибка(-π/2)  -1 | Да |
| Математическая функция обратного синуса, обработка ошибок | arcsin(0)  arcsin(1)  arcsin(-1)  arcsin(2) | 0  π/2  -π/2  Ошибка(2) | 0  π/2  -π/2  Ошибка(2) | Да |
| Математическая функция обратного косинуса, обработка ошибок | arccos(0)  arccos(1)  arccos(-1)  arccos(2) | π/2  0  π  Ошибка(2) | π/2  0  π  Ошибка(2) | Да |
| Математическая функция обратного тангенс | arctg(0)  arctg(1)  arctg(-1) | 0  π/4  -π/4 | 0  π/4  -π/4 | Да |
| Взятие производной, определение пользовательских функций, вычисление пользовательских функций, определение числа Эйлера | a=x  b=sin(x)  c=(x^3+lnx)  d/dx(a)  d/dx(b)  d/dx(c)  a(e)  b(e)  c(e) | a=x  b=sin(x)  c=(x^3+lnx)  1  cos(x)  3\*x^2+1/x  e  0,047425172189  21,08553692318 | a=x  b=sin(x)  c=(x^3+lnx)  1  cos(x)  3\*x^2+1/x  e  0,047425172189  21,08553692318 | Да |

Приложения

Исходный код