



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 6
по дисциплине
«Функциональное и логическое
программирование»

Тема Условия

Студент Александров Э.И.

Группа ИУ7-53БВ

Преподаватель Строганов Ю.В.

Москва, 2024

Содержание

ВВЕДЕНИЕ	4
1 Аналитическая часть	5
2 Конструкторская часть	6
3 Технологическая часть	7
ЗАКЛЮЧЕНИЕ	9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10
Приложение А	11

ВВЕДЕНИЕ

Цель данной работы заключается в разработке функций для решения квадратного уравнения с использованием различных конструкций языка Lisp. Задачи включают реализацию алгоритмов, учитывающих как действительные, так и комплексные корни уравнения, а также использование различных подходов к программированию: логические операторы and и or, условные конструкции if и cond.

1 Аналитическая часть

Квадратное уравнение имеет вид $ax^2 + bx + c = 0$, где a , b и c — коэффициенты. Решение квадратного уравнения возможно с помощью дискриминанта $D = b^2 - 4ac$. В зависимости от значения дискриминанта можно выделить три случая:

- 1) Если $D > 0$, у уравнения два различных действительных корня.
- 2) Если $D = 0$, у уравнения один действительный корень.
- 3) Если $D < 0$, у уравнения два комплексных корня.

В данной работе реализованы три функции, каждая из которых использует разные конструкции языка Lisp для обработки этих случаев.

Вывод

В аналитической части была рассмотрена теория квадратных уравнений и методы их решения, что послужило основой для реализации функций.

2 Конструкторская часть

Алгоритмы для решения квадратного уравнения были реализованы с использованием различных подходов:

- 1) Функция `quadratic-solve-and-or` использует логические операторы `and` и `or` для проверки условий.
- 2) Функция `quadratic-solve-if` применяет условные конструкции `if` для обработки различных случаев.
- 3) Функция `quadratic-solve-cond` использует конструкцию `cond`, что позволяет более элегантно обрабатывать множественные условия.

Каждая из функций проверяет валидность коэффициентов и вычисляет корни уравнения в зависимости от значения дискриминанта.

Вывод

В конструкторской части были разработаны три функции, каждая из которых реализует алгоритм решения квадратного уравнения с использованием различных конструкций языка Lisp.

3 Технологическая часть

Реализация была выполнена на языке программирования Lisp. Для написания и тестирования кода использовалась среда разработки VS Code с установленным пакетом Common Lisp, а также компилятор Steal Bank Common Lisp. Код функций включает в себя расчёт дискриминанта, проверку валидности коэффициентов и вычисление корней уравнения.

Коды алгоритмов:

Листинг 3.1 — Метод and-or

```
(defun quadratic-solve-and-or (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (if (or (= a 0) (and (= b 0) (= c 0)))
        (error "Not valid coef: a, b, and c")
        (if (and (> discriminant 0) (not (= a 0)))
            (list (+ real-part (/ imaginary-part (* 2 a))) ;  $x_1 = \frac{-b + \sqrt{D}}{2a}$ 
                  (- real-part (/ imaginary-part (* 2 a))) ;  $x_2 = \frac{-b - \sqrt{D}}{2a}$ 
                  )
            (if (= discriminant 0)
                (list real-part) ;  $x = -b / (2a)$ 
                (list (complex real-part (/ imaginary-part 2)) ;;  $x_1 = real + i * imaginary$ 
                      (complex real-part (/ (- imaginary-part) 2)))))) ;;;  $x_2 = real - i * imaginary$ 
    )
  )
```

Листинг 3.2 — Метод if

```
(defun quadratic-solve-if (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (if (or (= a 0) (and (= b 0) (= c 0)))
        (error "Invalid coefficients: a, b, and c cannot all be zero.")
        (if (> discriminant 0)
            (list (+ real-part (/ imaginary-part (* 2 a))) ;  $x_1 = \frac{-b + \sqrt{D}}{2a}$ 
                  (- real-part (/ imaginary-part (* 2 a))) ;  $x_2 = \frac{-b - \sqrt{D}}{2a}$ 
                  )
            (if (= discriminant 0)
                (list real-part) ;  $x = -b / (2a)$ 
                (list (complex real-part (/ imaginary-part 2)) ;;  $x_1 = real + i * imaginary$ 
                      (complex real-part (/ (- imaginary-part) 2)))))) ;;;  $x_2 = real - i * imaginary$ 
    )
  )
```

```

(list real-part) ;  $x = -b / (2a)$ 
(list (complex real-part (/ imaginary-part 2)) ;  $x1 = real + i*$ 
      imaginary
      (complex real-part (/ (- imaginary-part) 2)))))) ;  $x2 = real$ 
      -  $i*imaginary$ 

```

Листинг 3.3 — Метод cond

```

(defun quadratic-solve-cond (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (cond
      ((or (= a 0) (and (= b 0) (= c 0)))
       (error "Invalid coefficients: a cannot be zero, and b and c cannot
              both be zero."))
      ((> discriminant 0)
       (list (+ real-part (/ imaginary-part 2))
             (- real-part (/ imaginary-part 2))))
      ((= discriminant 0)
       (list real-part))
      (t
       (list (complex real-part (/ imaginary-part 2))
             (complex real-part (/ (- imaginary-part) 2)))))))

```

Тестовые данные:

- 1) Для $a=1, b=-3, c=2$ (два действительных корня: 1 и 2)
- 2) Для $a=1, b=2, c=1$ (один действительный корень: -1)
- 3) Для $a=1, b=0, c=1$ (два комплексных корня: i и $-i$)

Все тесты пройдены успешно.

Вывод

В технологической части была реализована программа для решения квадратного уравнения на языке Lisp, протестированы различные случаи, и все тесты прошли успешно.

ЗАКЛЮЧЕНИЕ

Цель работы заключалась в разработке функций для решения квадратного уравнения с использованием различных конструкций языка Lisp. В результате были реализованы три функции, каждая из которых учитывает действительные и комплексные корни уравнения. Все задачи, поставленные в начале работы, были успешно выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Пол Грэм, ANSI Common Lisp - СПб.: Символ-Плюс, 2012. - 448 с.
2. Сайт [geeksforgeeks.org](https://www.geeksforgeeks.org/cond-construct-in-lisp/?ysclid=m2a5cqzzjd192302838) — [<https://www.geeksforgeeks.org/cond-construct-in-lisp/?ysclid=m2a5cqzzjd192302838>].
3. Сайт [docs.racket-lang.org](https://docs.racket-lang.org/reference/if.html) — [<https://docs.racket-lang.org/reference/if.html>]

Приложение А

Код программы:

Листинг 3.4 — Метод and-or

```
(defun quadratic-solve-and-or (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (if (or (= a 0) (and (= b 0) (= c 0)))
        (error "Not valid coef: a, b, and c")
        (if (and (> discriminant 0) (not (= a 0)))
            (list (+ real-part (/ imaginary-part (* 2 a))) ;  $x_1 = \frac{-b + \sqrt{D}}{2a}$ 
                  (- real-part (/ imaginary-part (* 2 a))) ;  $x_2 = \frac{-b - \sqrt{D}}{2a}$ 
                  / (2a))
            (if (= discriminant 0)
                (list real-part) ;  $x = -b / (2a)$ 
                (list (complex real-part (/ imaginary-part 2)) ;;  $x_1 = real + i*imaginary$ 
                      (complex real-part (/ (- imaginary-part) 2))))))) ;  $x_2 = real - i*imaginary$ 
  )
```

Листинг 3.5 — Метод if

```
(defun quadratic-solve-if (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (if (or (= a 0) (and (= b 0) (= c 0)))
        (error "Invalid coefficients: a, b, and c cannot all be zero.")
        (if (> discriminant 0)
            (list (+ real-part (/ imaginary-part (* 2 a))) ;  $x_1 = \frac{-b + \sqrt{D}}{2a}$ 
                  (- real-part (/ imaginary-part (* 2 a))) ;  $x_2 = \frac{-b - \sqrt{D}}{2a}$ 
                  / (2a))
            (if (= discriminant 0)
                (list real-part) ;  $x = -b / (2a)$ 
                (list (complex real-part (/ imaginary-part 2)) ;  $x_1 = real + i*imaginary$ 
                      (complex real-part (/ (- imaginary-part) 2)))))) ;  $x_2 = real - i*imaginary$ 
  )
```

Листинг 3.6 — Метод cond

```
(defun quadratic-solve-cond (a b c)
  (let* ((discriminant (+ (* b b) (* -4 a c)))
        (real-part (/ (- b) (* 2 a)))
        (imaginary-part (sqrt (abs discriminant))))
    (cond
      ((or (= a 0) (and (= b 0) (= c 0)))
       (error "Invalid coefficients: a cannot be zero, and b and c cannot
              both be zero."))
      ((> discriminant 0)
       (list (+ real-part (/ imaginary-part 2))
             (- real-part (/ imaginary-part 2))))
      (= discriminant 0)
      (list real-part))
      (t
       (list (complex real-part (/ imaginary-part 2))
             (complex real-part (/ (- imaginary-part) 2)))))))
```