

Мета роботи: практичне ознайомлення з засобами функціонального програмування на прикладі мови Lisp.

Завдання роботи: скласти та реалізувати в інтерпретаторі Lisp функціональні вирази, відповідно до вимог пунктів завдання.

Теоретичні відомості

Lisp – одна з найпоширеніших мов функціонального програмування. Найпростішими (примітивними) об'єктами даних у Ліспі є *атоми* та *списки*. (Назва мови є скороченням від слів «обробка списків» (LISt Processing)). Атоми поділяються на символи та числа. Список в Lisp прийнято позначати круглими дужками, а його елементи розділяти пробілами, наприклад: (q w e r (t) y).

Функціональна програма складається з сукупності визначених функцій, які, в свою чергу, можуть викликати інші функції. Обчислення починається з виклику деякої функції. Виклик функції в Lisp записується у вигляді списку: (name arg1 arg2 ...), де name – ім'я функції; arg1, arg2, ... – її аргументи. Функція викликається з набором аргументів і повертає єдине значення. Якщо ж необхідно задати список, в якому обчислення функції виконувати не потрібно, то перед списком ставлять ключове слово QUOTE або символ одиночної лапки. Повторні обчислення у функціональній програмі здійснюються за допомогою рекурсії, оскільки оператори передачі керування у функціональних мовах відсутні.

Мова Lisp має п'ять *примітивних функцій*:

1. (CAR list) - голова списку.
2. (CDR list) - хвіст списку.
3. (CONS object list) - об'єднання (конкатенація) об'єкта зі списком.
4. (EQL atom1 atom2) - порівняння двох атомів.

5. (ATOM *object*) - перевірка, чи є object атомом.

Функції CAR та CDR називаються *селекторними функціями*, оскільки вони дають можливість обирати або знищувати частину об'єкта. Використовуються також функції, які є комбінаціями (не більше 5) функцій CAR та CDR.

Наприклад: \$ (CADDR '(q w e r t y))

Функція конструктора CONS використовується для додавання об'єкту до заданого списку. Об'єкт, який додається, стає головою списку. Наприклад:

\$ (CONS apple '(q w))
(apple q w)

Функція порівняння EQ порівнює значення двох своїх аргументів, які обов'язково повинні бути атомами-ідентифікаторами, та повертає значення істини (T) або хибності (F, NIL).

\$ (EQL 'qw 'qw)
T

Також, в багатьох реалізаціях Ліспу визначені інші функції порівняння: EQL, аргументами якої можуть бути не лише атоми-ідентифікатори, а й атоми-числа; і EQUAL, яка порівнює два довільних S-вирази.

Предикат ATOM повертає T, якщо об'єкт є атомом, і F в іншому випадку. Порожній список NIL вважається атомом.

\$ (ATOM '(q))	\$ (ATOM 3)
F	T

Особливістю мови Лісп є те, що хибне значення ("брехня") подається як порожній список, що записується як () або nil, а будь-який інший вираз вважається носієм значення "істина" (часто в цій ролі виступає атом T).

Функції призначення застосовуються для надання значень програмним змінним. Найбільш використовуваною є функція SET,

яка виконує заміну значення вказаного символу вказаним об'єктом. При виконанні операції заміни необхідно розрізнити символ та значення. При старті інтерпретатора Lisp значенням кожного символу є він сам; наприклад, значенням символу DOG буде DOG. Присвоїмо символу DOG значення CAT: (SET 'DOG 'CAT). Тепер введемо команду: (SET DOG 'HEN). При цьому значення HEN ми присвоюємо не символу DOG, а значенню символу DOG, тобто символу CAT, а значення символу DOG залишилося без зміни.

Поряд з примітивними функціями в Lisp можуть бути використані функції, визначені користувачем. Визначення функції в Ліспі має наступний вигляд:

```
(DEFUN name (arg1 arg2 ...)  
  task1  
  task2  
  . . . . . )
```

де name – ім'я функції;

arg1, arg2, ... – аргументи (параметри);

task1, task2, ... – послідовність задач.

Порядок виконання роботи

1. Ознайомтесь з теоретичними відомостями.

2. Подібно до наведених прикладів, складіть власні варіанти списків, по одному до кожного завдання А) та Б).

А) Список містить два підсписки. Загальна кількість елементів 7-8.

Приклад: '(1 (2 3 4) (5 6 7 8))

Б) Голова та хвіст списку є списками, що містять підсписки.

Приклад: '(((1) 2 3) (4 (5) (6) 7))

3. Запустіть інтерпретатор мови Lisp. Для кожного з списків, складених в п.2, напишіть вирази для отримання кожного з елементів.

4. Виконайте в інтерпретаторі Lisp наступні команди. Зафіксуйте та поясніть отримані результати виконання.

А) (CONS NIL NIL)

Б) (EQL (CDR '(q)) NIL)

В) (CONS (CAR '((q w))) (CDR '((q (w e)))))

Г) (ATOM (CDR '(q NIL)))

Д) (EQL NIL 'NIL)

5. За наведеними вихідними даними побудуйте вираз для отримання заданого результату.

Приклад.

Дано: (q (w e)). Необхідно отримати: (w e)

Розв'язання:

1) (CDR '(q (w e))) = ((w e))

2) (CAR '((w e))) = (w e)

Відповідь: вираз має вигляд: CAR (CDR '(q (w e)))

А) Дано: (A, B, C), (X, Y, Z). Отримати: (A, Y, Z)

Б) Дано: ((q (w (e) r) t) y). Отримати: ((q) w (e) r)

В) Дано: ((one 1) (two 2 3) (three 4 5 6)). Отримати: 5

Г,Д) Придумайте ще два власних варіанти до завдання 3 та розв'яжіть їх.

6. Придумайте та реалізуйте по 2 приклади на використання кожної з наступних функцій мови Lisp: ATOM, NULL, NTH, LAST, CONS, +, -, ADD1, SUB1, MOD, REM, LIST, COND.

7. Підготуйте звіт про виконану роботу. Дайте відповіді на контрольні запитання.

Контрольні запитання

1. Як визначаються списки в мові Lisp?
2. Які функції Lisp називаються примітивними функціями?
3. Як записати функцію у Lisp-програмі?
4. Поясніть дію символу одиночної лапки, що стоїть перед списком у Lisp-програмі.
5. Поясніть сутність операції заміни значення символу в мові Lisp.
6. Напишіть функції Lisp, які дозволять отримати кожен з п'яти елементів заданого списку.