Предикат — это функция, которая отвечает на вопрос, является ли некоторое утверждение истинным или ложным для данного аргумента.

**1. Функция all**

lisp

Копировать код

(defun all (p x)

(cond

((null x) t) ; Если список пустой, возвращаем true

((funcall p (car x)) ; Проверяем первый элемент списка

(all p (cdr x))) ; Рекурсивно проверяем оставшиеся элементы

(t nil))) ; Если предикат не выполнен, возвращаем nil

Это рекурсивная функция, которая проверяет, удовлетворяют ли все элементы списка предикату p. Рассмотрим её шаги:

* **Параметры функции:**
  + p — это предикат, т.е. функция, которая возвращает либо T, либо NIL для каждого элемента.
  + x — это список элементов, который нужно проверить.
* **Логика работы:**
  + (null x): Если список пустой, то по определению можно считать, что все элементы удовлетворяют предикату, и возвращается T. Это стандартное поведение для рекурсивных функций на списках: пустой список обычно считается удовлетворяющим любому предикату.
  + (funcall p (car x)): Функция car возвращает первый элемент списка. Мы применяем к этому элементу предикат p с помощью функции funcall. Если результат этого вызова — T, то переходим к следующему элементу с помощью рекурсивного вызова (all p (cdr x)). Функция cdr возвращает "хвост" списка (все элементы, кроме первого).
  + (t nil): Если хотя бы для одного элемента предикат не выполняется, то возвращаем NIL. Это срабатывает, когда предикат на текущем элементе списка возвращает NIL, то есть когда хотя бы для одного элемента условие не выполнено.

Пример:

lisp

Копировать код

(all 'even-p '(2 4 6)) ; Вернет T, так как все элементы четные

(all 'even-p '(2 3 6)) ; Вернет NIL, так как 3 — нечетное число

**2. Функция even-p**

lisp

Копировать код

(defun even-p (n)

"Предикат для проверки, является ли число N четным.

Возвращает T, если N четное, иначе NIL."

(evenp n)) ; Используем встроенную функцию evenp

Эта функция — предикат для проверки, является ли число четным. Она использует встроенную функцию evenp, которая возвращает T, если число чётное, и NIL, если оно нечётное.

**3. Функция positive-p**

lisp

Копировать код

(defun positive-p (n)

"Предикат для проверки, является ли число N положительным.

Возвращает T, если N положительное, иначе NIL."

(> n 0))

Эта функция проверяет, является ли число положительным. Она возвращает T, если число больше нуля, и NIL в противном случае.

**4. Примеры использования**

Теперь давайте рассмотрим примеры, приведенные в коде:

lisp

Копировать код

(format t "Все элементы четные? ~A~%" (all 'even-p '(2 4 6))) ; Вернет T

(format t "Все элементы четные? ~A~%" (all 'even-p '(2 3 6))) ; Вернет NIL

(format t "Все элементы четные? ~A~%" (all 'even-p '())) ; Вернет T (пустой список)

(format t "Все элементы положительные? ~A~%" (all 'positive-p '(1 2 3))) ; Вернет T

(format t "Все элементы положительные? ~A~%" (all 'positive-p '(-1 2 3))) ; Вернет NIL

* **Первый пример:**

lisp

Копировать код

(all 'even-p '(2 4 6))

Здесь мы проверяем, все ли элементы списка (2 4 6) чётные. Так как все элементы — четные, результат будет T.

* **Второй пример:**

lisp

Копировать код

(all 'even-p '(2 3 6))

В этом случае элемент 3 — нечетное число, поэтому результат будет NIL.

* **Третий пример:**

lisp

Копировать код

(all 'even-p '())

Пустой список по определению удовлетворяет любому предикату, включая even-p, так что результат будет T.

* **Четвертый пример:**

lisp

Копировать код

(all 'positive-p '(1 2 3))

Все числа в списке положительные, поэтому результат будет T.

* **Пятый пример:**

lisp

Копировать код

(all 'positive-p '(-1 2 3))

В этом случае элемент -1 — отрицательное число, и результат будет NIL.

**Итог**

Этот код реализует универсальную функцию all, которая проверяет, удовлетворяют ли все элементы списка заданному предикату. В качестве примеров использованы предикаты для проверки четности и положительности чисел.