

Отчет по лабораторной работе № 5 по курсу «Функциональное программирование»

Студент группы 8О-307 МАИ *Бирюков Виктор*, №2 по списку
Контакты: vikvladbir@mail.ru
Работа выполнена: 21.05.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

Обобщённые функции, методы и классы объектов.

2. Цель работы

Научиться определять простейшие классы, порождать экземпляры классов, считывать и изменять значения слотов, научиться определять обобщённые функции и методы.

3. Задание (вариант №5.43)

Определите обычную функцию `der-polynom` с одним параметром — многочленом, т.е. экземпляром класса `polynom`.

Функция должна вычислять производную $P'(x)$.

4. Оборудование студента

Процессор AMD Ryzen 7 3700U @ 2.3GHz, память: 20Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Windows 10, LispWorks Personal Edition 7.1.2.

6. Идея, метод, алгоритм

Известно, что производная степени $\frac{dx^n}{dx} = nx^{n-1}$, $n \neq 0$, в то время как $\frac{dx^0}{dx} = \frac{d1}{dx} = 0$. Таким образом дифференцирование многочлена происходит по следующей схеме:

- терм вида ax^n , $n > 0$ становится термом $an \cdot x^{n-1}$
- терм, соответствующий свободному члену, удаляется.

Эту операцию над списком термов осуществляет функция `der-terms`. Функция `der-polynom` создает новый многочлен с тем же символом и измененными термами.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
; 5.43
```

```
(defun make-term (&key order coeff)
  (list order coeff))

(defun order (term) (first term))
(defun coeff (term) (second term))

(defclass polynom ()
  ((var-symbol :initarg :var :reader var)
   (term-list :initarg :terms :reader terms)))

(defgeneric zerop1 (arg)
  (:method ((n number))
    (zerop n)))

(defgeneric minusp1 (arg)
  (:method ((n number))
    (minusp n)))

(defmethod print-object ((p polynom) stream)
  (format stream МЧ"[ (~s)
~:{~:[~:[+~;-~]~d~[~2*~;~s~*~::~~s^~d~]~::~~}"
  (var p)
  (mapcar (lambda (term)
            (list (zerop1 (coeff term))
                  (minusp1 (coeff term))
                  (if (minusp1 (coeff term))
                      (abs (coeff term))
                      (coeff term))
                  (order term)
                  (var p)
                  (order term))))
```

```

      (terms p))))
(defun der-terms (terms)
  (loop for x in terms
    when (> (order x) 0)
    collect (make-term
              :order (1- (order x))
              :coeff (* (coeff x) (order x)))))
(defun der-polynom (p)
  (make-instance 'polynom
    :var (var p)
    :terms (der-terms (terms p))))

```

8.2. Результаты работы

```

CL-USER 1 > (setq p (make-instance 'polynom
  :var 'x
  :terms (list (make-term :order 2 :coeff 5)
                (make-term :order 1 :coeff 3.3)
                (make-term :order 0 :coeff -7))))
[МЧ (X) +5X^2+3.3X-7]

```

```

CL-USER 2 > (der-polynom p)
[МЧ (X) +10X+3.3]

```

```

CL-USER 3 > (der-polynom (der-polynom p))
[МЧ (X) +10]

```

```

CL-USER 4 > (der-polynom (der-polynom (der-polynom p)))
[МЧ (X) ]

```

```

CL-USER 5 > (setq p (make-instance 'polynom
  :var 'x
  :terms (list (make-term :order 100 :coeff 1))))
[МЧ (X) +1X^100]

```

```

CL-USER 6 > (der-polynom p)
[МЧ (X) +100X^99]

```

```

CL-USER 7 > (der-polynom (der-polynom p))
[МЧ (X) +9900X^98]

```

```
CL-USER 8 > (der-polynom (der-polynom (der-polynom p)))  
[MЧ (X) +970200X^97]
```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

10. Замечания автора по существу работы

В данной работе я решил использовать loop вместо явной рекурсии и это оказалось несколько проще. Сложность алгоритма — $O(n)$, где n — количество термов.

11. Выводы

В ходе выполнения лабораторной работы я познакомился с созданием простейших классов, а также с реализацией функций для их обработки.