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

Студент группы 8О-307 МАИ *Вельтман Лина*, №7 по списку
Контакты: kluuo@mail.ru
Работа выполнена: 14.05.2020

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 №23)

Определить обобщённую функцию и методы `on-single-line3-p` - предикат, принимающий в качестве аргументов три точки (радиус-вектора) и необязательный параметр `tolerance` (допуск), возвращающий `T`, три указанные точки лежат на одной прямой (вычислять с допустимым отклонением `tolerance`). Точки могут быть заданы как декартовыми координатами (экземплярами `cart`), так и полярными (экземплярами `polar`).

```
(defgeneric on-single-line3-p (v1 v2 v2 &optional tolerance))
(defmethod on-single-line3-p ((v1 cart) (v2 cart) (v3 cart)
  &optional (tolerance 0.001))
  ...)
```

4. Оборудование ПЭВМ студента

Ноутбук MacBook Pro (13-inch, 2017), процессор 2.3GHz Intel Core i5, память: 8Gb, разрядность системы: 64.

5. Программное обеспечение ЭВМ студента

macOS Catalina 10.15.4, реализация языка SBCL 1.4.16, текстовый редактор Sublime Text 3.

6. Идея, метод, алгоритм

Воспользуемся уравнением прямой, проходящей через 2 точки. Если 3 точки лежат на одной прямой, то для них выполняется равенство:

$$\frac{x_3 - x_1}{x_2 - x_1} = \frac{y_3 - y_1}{y_2 - y_1}.$$

Чтобы избежать погрешностей при делении, заменим частные на произведения:

$$(x_2 - x_1) * (y_3 - y_1) = (x_3 - x_1) * (y_2 - y_1).$$

Функция `on-single-line3-p` проверяет выполнение вышеприведенного равенства для всех элементов списка с помощью функции `on-single-line3`. Функция умеет работать как с декартовыми координатами, так и с полярными.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun square (x) (* x x))

(defclass cart ()                    ; имя класса и надклассы
  ((x :initarg :x :reader cart-x)    ; дескриптор слота x
   (y :initarg :y :reader cart-y))) ; дескриптор слота y


(defmethod print-object ((c cart) stream)
    (format stream "[CART x:~d y:~d]"
        (cart-x c) (cart-y c)
    )
)


(defclass polar ()
  ((radius :initarg :radius :accessor radius)    ; длина >=0
```

```
   (angle  :initarg :angle  :accessor angle)))  ; угол (-pi;pi]


(defmethod print-object ((p polar) stream)
  (format stream "[POLAR radius ~d angle ~d]"
          (radius p) (angle p)))


(defmethod radius ((c cart))
  (sqrt (+ (square (cart-x c))
           (square (cart-y c)))))


(defmethod angle ((c cart))
  (atan (cart-y c) (cart-x c))) ; atan2 в Си


(defmethod cart-x ((p polar))
  (* (radius p) (cos (angle p))))


(defmethod cart-y ((p polar))
  (* (radius p) (sin (angle p))))


(defun on-single-line3 (data)
  (<= (abs (- (* (- (cart-x (second data)) (cart-x (first data)))
          (- (cart-y (third data)) (cart-y (first data))))
          (* (- (cart-x (third data)) (cart-x (first data)))
          (- (cart-y (second data)) (cart-y (first data)))))
      ) (fourth data)
  )
)


(defgeneric on-single-line3-p (v1 v2 v3 &optional tolerance))

(defmethod on-single-line3-p ((v1 cart) (v2 cart) (v3 cart)
   &optional (tolerance 0.001))
  (on-single-line3 (list v1 v2 v3 tolerance)))
```

```
(defmethod on-single-line3-p ((v1 polar) (v2 polar) (v3 polar)
   &optional (tolerance 0.001))
  (on-single-line3 (list v1 v2 v3 tolerance)))
```

8.2. Результаты работы

```
MacBook-Pro-Lina:lab5 linuxoid$ sbcl
This is SBCL 1.4.16, an implementation of ANSI Common Lisp.
More information about SBCL is available at
   <http://www.sbcl.org/>.

SBCL is free software, provided as is, with absolutely no
   warranty.
It is mostly in the public domain; some portions are provided
   under
BSD-style licenses.  See the CREDITS and COPYING files in the
distribution for more information.
* (compile-file "./lab5_23.lisp")
; compiling file
   "/Users/linuxoid/Desktop/VUZICH/FP/lab5/lab5_23.lisp" (written
   19 MAY 2020 02:37:15 PM):
; compiling (DEFUN SQUARE ...)
; compiling (DEFCLASS CART ...)
; compiling (DEFMETHOD PRINT-OBJECT ...)
; compiling (DEFCLASS POLAR ...)
; compiling (DEFMETHOD PRINT-OBJECT ...)
; compiling (DEFMETHOD RADIUS ...)
; compiling (DEFMETHOD ANGLE ...)
; compiling (DEFMETHOD CART-X ...)
; compiling (DEFMETHOD CART-Y ...)
; compiling (DEFUN ON-SINGLE-LINE3 ...)
; compiling (DEFGENERIC ON-SINGLE-LINE3-P ...)
; compiling (DEFMETHOD ON-SINGLE-LINE3-P ...)
; compiling (DEFMETHOD ON-SINGLE-LINE3-P ...)

; wrote /Users/linuxoid/Desktop/VUZICH/FP/lab5/./lab5_23.fasl
; compilation finished in 0:00:00.028
#P"/Users/linuxoid/Desktop/VUZICH/FP/lab5/lab5_23.fasl"
NIL
NIL
* (load "lab5_23.lisp")
T
* (on-single-line3-p (make-instance 'cart :x 5 :y 6)
```

```
          (make-instance 'cart :x 7 :y 8)
          (make-instance 'cart :x 9 :y 4))
NIL
* (on-single-line3-p (make-instance 'cart :x 1 :y 2)
          (make-instance 'cart :x 2 :y 4)
          (make-instance 'cart :x 0 :y 1))
NIL
* (on-single-line3-p (make-instance 'cart :x 4 :y 8.00232)
          (make-instance 'cart :x 5 :y 10.0)
          (make-instance 'cart :x 5.00034 :y 10.0001))
T
* (on-single-line3-p (make-instance 'polar :radius 1 :angle (/ pi
   2))
                     (make-instance 'polar :radius 2 :angle (/ pi
   2))
                     (make-instance 'polar :radius 3 :angle (/ pi
   2)))
T
* (on-single-line3-p (make-instance 'polar :radius 3 :angle 1)
                     (make-instance 'polar :radius 5.01 :angle
   0.3)
                     (make-instance 'polar :radius 8 :angle 2.02))
NIL
* (on-single-line3-p (make-instance 'polar :radius 1 :angle 1)
                     (make-instance 'polar :radius 3 :angle 1)
                     (make-instance 'polar :radius 4 :angle
   1.000001))
T
```

9. Дневник отладки

Дата	Событие	Действие по исправлению
19.05.2020	Некорректная работа с полярными координатами	Добавлена функция по обработке полярных координат

10. Замечания, выводы

Благодаря данной лабораторной работы я познакомилась с классами и обобщенными функциями в языке Common Lisp. Простой и понятный синтаксис для создания классов является преимуществом. Также хотелось бы упомянуть, что обощенные функции позволяют не создавать несколько похожих функций для разных классов, а просто написать реализации одной функции, которая будет применима для нескольких классов, что очень удобно.