

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

Студент группы 8О-306 МАИ *Наумов Дмитрий*, №15 по списку

Контакты: dandachok@gmail.com

Работа выполнена: 27.04.2020

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 №41)

Определить обычную функцию `on-single-line-p` - предикат,

- принимающий в качестве аргумента список точек (радиус-векторов),
- возвращающий `T`, если все 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`).

```
(defvar *tolerance* 0.001)
```

```
(defun on-single-line-p (vertices)  
  ...)
```

## 4. Оборудование студента

Процессор Intel Core i3-2100 4 @ 2.2GHz, память: 8Gb, разрядность системы: 64.

## 5. Программное обеспечение

OS Ubuntu 18.04.4 LTS, clisp 2.49.60

## 6. Идея, метод, алгоритм

Программа состоит из следующих функций:

- `cart-x` вычисляет  $x$ -координату из полярных
- `cart-y` вычисляет  $y$ -координату из полярных
- `to-cart` преобразует координаты в декартовое представление:
  - если передается объект `cart`, то возвращает его
  - если передается объект `polar`, то вычисляет  $x, y$  и создает объект класса `cart`
- `on-line-p` проверяет, находится ли точка на прямой в каноническом представлении с параметрами  $x_1, y_1, x_2, y_2$

Функция просто проверяет выполнение следующего условия:

$$\frac{x - x_1}{x_1 - x_2} - \frac{y - y_1}{y_1 - y_2} \approx 0$$

- `on-single-line-p` проверяет все точки списка на принадлежность прямой  
В качестве параметров прямой берутся координаты первых двух точек.

## 7. Распечатка программы и её результаты

### 7.1. Исходный код

```
(defvar *tolerance* 0.001)
(defclass cart ()
  ((x :initarg :x :reader x)
   (y :initarg :y :reader y)))
(defclass polar ()
  ((radius :initarg :radius :accessor radius)
   (angle :initarg :angle :accessor angle)))
(defmethod cart-x ((p polar))
  (* (radius p) (cos (angle p))))

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

(defgeneric to-cart (arg)
  (:method ((c cart))
    c)
  (:method ((p polar))
    (make-instance 'cart :x (cart-x p) :y (cart-y p))))

(defmethod on-line-p ((c cart) x1 y1 x2 y2)
  (let ((a1 (- x1 x2)) (a2 (- y1 y2)))
    (cond ((< (abs a1) *tolerance*)
            (< (abs (- (x c) x1)) *tolerance*))
          ((< (abs a2) *tolerance*)
            (< (abs (- (y c) y1)) *tolerance*))
          (T (< (abs (- (/ (- (x c) x1) a1)
                        (/ (- (y c) y1) a2)))
              *tolerance*)))))

(defun on-single-line-p (vertices)
  (let ((x1 (x (to-cart (first vertices))))
        (y1 (y (to-cart (first vertices))))
        (x2 (x (to-cart (second vertices))))
        (y2 (y (to-cart (second vertices)))))
    (loop for point in vertices
      do (if (not (on-line-p (to-cart point) x1 y1 x2 y2))
              (return-from on-single-line-p NIL)))
    T))
```

## 7.2. Результаты работы

```
dandachok@dpc:~/Documents/Study/sem6/FP/lab5$ clisp
[1]> (load "solution.lisp")
;; Loading file solution.lisp ...
;; Loaded file solution.lisp
#P"/home/dandachok/Documents/Study/sem6/FP/lab4/solution.lisp"
[38]> (setq c1 (make-instance 'cart :x 1 :y 1))
#<CART #x55E960AC1C68>
[38]> (setq c2 (make-instance 'cart :x 2 :y 2))
#<CART #x55E960ACF4F8>
[38]> (setq p1 (make-instance 'polar :radius 5 :angle (/ pi 4)))
#<POLAR #x55E960ADCE08>
[38]> (setq p2 (make-instance 'polar :radius 5 :angle (/ pi 6)))
#<POLAR #x55E960AEA718>
[38]> (setq l1 (list c1 c2 p1))
(#<CART #x55E960AC1C68> #<CART #x55E960ACF4F8> #<POLAR
#x55E960ADCE08>)
[38]> (setq l2 (list c1 c2 p2))
(#<CART #x55E960AC1C68> #<CART #x55E960ACF4F8> #<POLAR
#x55E960AEA718>)
[38]> (on-single-line-p l1)
T
[38]> (on-single-line-p l2)
NIL
```

## 8. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
19.05.20	Неправильное объявление функции	Заменить макрос defun на defmethod	

## 9. Замечания автора по существу работы

## 10. Выводы

Сложность программы по времени и по памяти линейная.