

# **Отчет по лабораторной работе №4 по курсу «Функциональное программирование»**

Студент группы 8О-307 Алексюнина Юлия, № по списку 1.

Контакты: juvyjuli@gmail.com

Работа выполнена: 17.07.2020

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 № 4)**

Определить обычную функцию line-length, вычисляющую длину отрезка - экземпляра класса line. Причем начало отрезка задаётся в полярных координатах (экземпляром polar), а конец - в декартовых (экземпляром cart).

```
(setq ln (make-instance 'line
```

```
  :start (make-instance 'polar ...)
```

```
  :end (make-instance 'cart ...)))
```

## **4. Оборудование студента**

Ноутбук HP, процессор Intel® Core™ i3-6006U CPU 2.00GHz 1.99GHz, память 4ГБ, 64-разрядная система.

## **5. Программное обеспечение**

ОС Windows 10 версия 1903, программа LispWorks Personal Edition 6.1.1

## **6. Идея, метод, алгоритм**

Задача состоит в следующем: имеются две точки, причем одна из них в полярных координатах, другая в декартовых., и нужно вычислить длину отрезка между ними.

Мы переводим ту, что в полярных, в декартовы, и затем по формуле длины отрезка, вычисляем ее.

## 7. Сценарий выполнения работы

## 8. Распечатка программы и её результаты

### Программа

```
(defclass cart ()
  ((x :initarg :x :reader cart-x)
   (y :initarg :y :reader cart-y)))

(defclass polar ()
  ((radius :initarg :radius :accessor radius)
   (angle :initarg :angle :accessor angle))

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

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

(defclass line()
  (
    (start :initarg :start :accessor line-start)
    (end :initarg :end :accessor line-end)
  )
)

(defmethod print-object((lin line) stream)
  (format stream "[LINE ~s ~s]"
          (line-start lin) (line-end lin)
  )
)

(defmethod cart-x ((p polar))
```

```

(* (radius p) (cos (angle p))))

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

(defun pow2 (x) (* x x))

(defmethod calc-dist ((c1 cart) (c2 cart))
  (sqrt2 (+ (pow2 (abs (- (cart-x c2) (cart-x c1))) ) (pow2 (abs
(- (cart-y c2) (cart-y c1)))))))
  )

(defun line-length(lin)
  (setq dot1 (to-cart (line-start lin)))
  (calc-dist dot1 (line-end lin))
)

```

## Результаты

```

CL-USER 16 : 4 > (setq lin (make-instance 'line
      :start (make-instance 'polar :radius 5 :angle (/ pi 2))
      :end (make-instance 'cart :x 0 :y 0))
)
[LINE [POLAR radius 5 angle 1.5707963267948966D0] [CART x 0 y 0]]

```

```

CL-USER 17 : 4 >
(line-length lin)
5.000023178253949D0

```

```

CL-USER 18 : 4 > (setq lin (make-instance 'line
      :start (make-instance 'polar :radius 10 :angle (/ pi 4))
      :end (make-instance 'cart :x 4 :y -7))
)

```

```
[LINE [POLAR radius 10 angle 0.7853981633974483D0] [CART x 4 y -  
7]]
```

```
CL-USER 19 : 4 >
```

```
(line-length lin)
```

```
14.402306143468408D0
```

## 9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание

## 10. Замечания автора по существу работы

Я познакомилась с классами в Common Lisp, задача не вызвала у меня особых затруднений.

## 11. Выводы

В данной лабораторной работе я познакомилась со классами, общими функциями и методами классов в языке Коммон Лисп и написала программу, которая высчитывает длину отрезка с заданными концами в полярных и декартовых координатах. Реализовано три класса – полярных и декартовых координат и класс линии, которая этими координатами задаётся. Программа работает правильно и прекрасно себя показала на составленных тестах.