

**Московский Авиационный Институт
(Национальный Исследовательский Университет)**

Факультет: “Информационные технологии и прикладная математика”
Кафедра: 806 “Вычислительная математика и программирование”

**Отчет по лабораторной работе №1
по курсу «Функциональное программирование»**

Студент: Полей-Добронравова Амелия Вадимовна

Группа: М8О-307Б, № по списку 16.

Преподаватель: Иванов Д. А., доц. каф. 806

Дата: 21.04.2021

Итоговая оценка:

Подпись преподавателя:

Москва, 2021

1. Тема работы

Примитивные функции и особые операторы Коммон Лисп.

2. Цель работы

Научиться вводить S-выражения в Лисп-систему, определять переменные и функции, работать с условными операторами, работать с числами, используя схему линейной рекурсии.

3. Задание (вариант № 1.38)

Запрограммируйте на языке Коммон Лисп функционал `compose`, который реализует композицию.

4. Оборудование студента

MacBook Pro (15-inch, Mid 2012), процессор 2,3 GHz Intel Core i7, память 8 ГБ.

5. Программное обеспечение

macOS Mojave 10.14.6

6. Идея, метод, алгоритм

По определению, композиция двух одноместных функций f и g есть функция: $x \rightarrow f(g(x))$.

Для написания программы необходимо использовать функционал **`funcall`**, означающий косвенный вызов функции и примитив **`function`**, возвращающий ссылку на глобальную функцию.

“Заменим” (вернем ссылку с помощью **`function`**) нашу функцию `compose` лямбда-конструкцией - безымянной функцией, принимающую аргументом X , который передадут при вызове `compose` из другого места программы.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

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

```
(defun cube (x) (* x x x))
```

```
(defun compose (F G)

(function (lambda (X) (funcall F (funcall G X)))) )


(funcall (compose #'square #'1+) 2)

(funcall (compose #'cube #'1-) 2)
```

Результаты

```
CL-USER 4 >
(funcall (compose #'square #'1+) 2)
9

CL-USER 5 > (funcall (compose #'cube #'1-) 2)
1

CL-USER 6 > (funcall (compose #'cube #'1-) 5)
64
```

9. Дневник отладки

№ Дата, время Событие Действие по исправлению Примечание

10. Замечания автора по существу работы

Данная задача учит определять переменные и функции, передавать функции как аргументы, что необходимо в данной парадигме программирования.

Сама реализация композиции двух функций бесполезна, потому что на практике её будут записывать через вложенные скобки, но как тренировка в составлении лямбда-выражений и ознакомление с синтаксисом - хорошее задание.

11. Выводы

Функции в лиспе являются настоящими сущностными первого класса. Функциональные объекты могут динамически создаваться, передаваться в качестве параметров или возвращаться в качестве результата. Таким образом, поддерживаются функции высшего порядка, то есть такие, аргументы и возвращаемые значения которых сами могут быть функциями.

Анонимные функции (лямбда-выражениями) могут использоваться вместо имени передаваемой функции. Они особенно полезны, когда мы хотим создать функцию для однократного использования, не засоряя программу

лишним именем. В общем случае их можно использовать для создания лексических замыканий.

Язык Common Lisp показался мне лаконичным, и скорее всего, при его качественном освоении, можно будет писать программы, пригодные для него, гораздо быстрее, чем на других языках.