Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

Отчет по практике №2

«Анализ задачи. Абстракция программ и данных.

Синтаксис языка программирования»

Выполнил

Студент гр.728-2

_____Геворгян Д.Р.

_____

Принял

Преподаватель КИБЭВС

_____ Кальнеус М. А.

_____

Томск 2019

1 Введение

Цель работы: Знакомство с основными элементами языка программирования, расширенной формой записи Бэкуса-Наура для записи синтаксиса языка программирования.

2 Ход работы

2.1 Описание темы

Коммивояжер желает посетить ряд городов и вернуться в исходный город, минимизируя суммарную длину переездов. Эта задача в математической форме формулируется как задача нахождения во взвешенном графе гамильтонова цикла минимальной длины и называется задачей коммивояжера. Алгоритм ближайшего соседа - один из простейших эвристических алгоритмов решения задачи Коммивояжера. Относится к категории «жадных» алгоритмов. Формулируется следующим образом: пункты обхода плана последовательно включаются в маршрут, причем каждый очередной включаемый пункт должен быть ближайшим к последнему выбранному пункту среди всех остальных, еще не включенных в состав маршрута.

Необходимо предоставить пользователь интерфейс для указания вершин графа, после чего задача Коммивояжера должна быть решена и решение выведено на экран, результат необходимо сохранить в виде изображения в файл.

2.2 Декомпозиция программы

Рисунок 2.1 - Декомпозиция программы “Алгоритм ближайшего соседа в задаче Коммивояжера”

2.2.1 Модуль 1

Наименование: Ввод количества городов

Назначение: Принятие на вход целочисленной переменной, обозначающей количество городов, через которые будет проходить Коммивояжер

Входные данные: Символ, который вводит пользователь

Выходные данные: Переменная типа int

Возможные ошибки: Будет введено отрицательное или дробное число, буква

Модуль 2

Наименование: Ввод координат каждого города

Назначение: Указание пользователем координат каждого города

Входные данные: Строка символов типа string, которые вводит пользователь

Выходные данные: массив с переменными типа double, хранящий координаты

Возможные ошибки: Будет введена буква

Модуль 3

Наименование: Вычисление расстояний

Назначение: Вычисление расстояний между точками по их координатам при помощи формулы векторной алгебры

Входные данные: Элементы массива, хранящего координаты

Выходные данные: Массив с переменными типа double, хранящий величины расстояний

Возможные ошибки: При верно введенных исходных данных ошибки не предвидятся

Модуль 4

Наименование: Поиск

Назначение: Поиск города, расстояние до которого минимально относительно других

Входные данные: элементы массива, хранящего расстояния

Выходные данные: Номер найденного города

Возможные ошибки: Не предвидятся

Модуль 5

Наименование: Переход

Назначение: Переход с одного города на другой путём смены сравниваемого элемента массива расстояния

Входные данные: Номер найденного города

Выходные данные: Элемент массива расстояния

Возможные ошибки: Переход на город, в котором мы уже были

Модуль 6

Наименование: Вывод

Назначение: Вывод решения на экран

Входные данные: Информация о том, что Коммивояжёр вернулся из последнего города в первый

Выходные данные: Изображение решенной задачи

Возможные ошибки: Неверный путь изображения

2.2 Синтаксис языка программирования в РБНФ:

identifier ::= alphabetic character, { alphabetic character | digit } .

number ::= ["-"], digit, { digit } .

string ::= ““ , { all characters - "" }, "" .

assignment :: = identifier , "$" , (number | identifier | string | math_operation).

alphabetic character ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z".

digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .

operator ::= ‘+’|’-’|’*’|’/’|‘%’.

math_operation ::= [“(“], (digit | identifier), operator, (digit | identifier), [“)”], {operator, math_operation}.

bool ::= “<” | “>” | “==” | “>=” | “<=” | “!=”

and_or_operator ::= “and” | “or”

bool_operation ::= [“(“], (digit | identifier), bool, (digit | identifier), [“)”], { and_or_operator, bool_operation}.

if_operator ::= “if”, bool_operation, white_space, “{“, white_space, {math_operation | if_operation | assignment | while_operation}, white_space, “}”, white_space, [“else”, white_space, “{“, white_space, {math_operation | if_operation | assignment | while_operation}, white_space, “}”].

while_operator ::= “while”, bool_operation, white_space, “{“, white_space, {math_operation | if_operation | assignment| while_operation}, white_space, “}”, white_space.

Input ::= “Input”, “(”, identifier, “)”.

Output ::= “Output”, “(“, (number | string | identifier), “)”.

white_space ::= ? white space characters ? .

all_characters ::= ? all visible characters ? .

2.3 Пример программы с использованием синтаксиса разработанного языка.

```
Input(a)
Input(b)
if (a>b)
{
	Output(“a>b”)
}
else
{
	Output(“a<=b”)
}
c$(a+b)
a$(c/b)
while (c!=0) or (c>a)
{
c$(c-1)
}
else
{
c$1000
}
```

3 Заключение

Проведен анализ задачи программы «Алгоритм ближайшего соседа в задаче Коммивояжёра», разработан синтаксис языка программирования и записан в форме РБНФ, приведен пример программы, записанной на разработанном языке.