Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

**Кафедра «Компьютерные системы и программные технологии»**

**Отчёт по лабораторной работе**

**Дисциплина**: Низкоуровневое программирование

**Тема**: Симулятор машин Тьюринга 5-tuple

Выполнила

студентка гр.23531/1 <*подпись*> Буй Куок Дат

Руководитель

доцент, к.т.н. <*подпись*> К. А. Гагарский

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**ЗАДАНИЕ**

**НА ВЫПолнение курсовоГО ПРОЕКТА**

студентке группы 23531/1 Буй Куок Дат

***1. Тема проекта (работы):*** Симулятор машин Тьюринга 5-tuple

***2. Срок сдачи законченного проекта (работы)*** 21.07.19 г.

***3. Исходные данные к проекту (работе)***:

Описание машины, содержимое входной ленты задаются во входных текстовых файлах (разных), результат работы машины выводится в текстовый файл. Форматы файлов фиксируются в ТЗ.

Необходимо обеспечить возможность имитации работы машины до ее останова или в течение заданного числа шагов.

Необходимо обеспечить возможность отладки машины (вывод промежуточных конфигураций, возобновление работы машины).

Допускается введение разумных ограничений на алфавит машины, а также максимальное число соостояний (миллионоы). Размер ленты должен быть ограничен только доступным симулятору объемом памяти.

***4. Содержание пояснительной записки***: введение,описание работы, испытание работы, вывод, список источников

***Дата получения задания***:

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А.Гагарский

*(подпись)*

Задание приняла к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ Буй Куок Дат

*(подпись студента)*

# **ВВЕДЕНИЕ**

В данном курсовом проекте мною будет написана программа, являющаяся симулятором машины Тьюринга вида 5-tuple, что соответствует моему индивидуальному заданию по низкоуровневому программированию.

В ней будут реализованы основные опции машины Тьюринга вида 5-tuple, такие как загрузка описания машины, исходной ленты и команд из соответствующих входных файлов, печать ленты и команд на экран, пошаговое и мгновенное выполнение программы, содержащейся в файле с командами.

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc14592998)

[**1.** **Описание работы:** 4](#_Toc14592999)

[**а.** **Испольные классы:** 4](#_Toc14593000)

[**б.** **Испольные входные файлы :** 6](#_Toc14593001)

[**2.** **Испытание программы:** 7](#_Toc14593002)

[**3.** **Вывод :** 7](#_Toc14593003)

[**4.** **Список источник** 7](#_Toc14593004)

1. **Описание работы:**
2. **Испольные классы:**
3. **ТuringMachine :**

* Реализованы основные функции машины
* **Метод loadDeacription :** для загрузки алфавита и состояний из текстового файла

private void loadDescription(String inputName) throws Exception {  
  
 BufferedReader br = new BufferedReader(new FileReader(new File(inputName)));  
  
 String strAlphabet = br.readLine();  
 if (!strAlphabet.matches("(.,)\*.")) {  
 throw new Exception("format error");  
 }  
 String strStatus = br.readLine();  
 if (!strStatus.matches("(.+,)\*.+")) {  
 throw new Exception("format error");  
 }  
 listAlphabet = Arrays.*asList*(strAlphabet.split(","));  
 listStatus = Arrays.*asList*(strStatus.split(","));  
  
 br.close();  
}

**- Метод loadCommand:** для загрузки команд из текстового файла

private void loadCommand(String inputName) throws Exception {  
  
 BufferedReader br = new BufferedReader(new FileReader(new File(inputName)));  
  
 String current = br.readLine();  
 String[] string;  
 while (current != null) {  
 if (!current.matches(".+,.+,.+,.+,.+")) {  
 throw new Exception("format error");  
 }  
 string = current.split(",");  
 if (listStatus.indexOf(string[0]) == -1 && listStatus.indexOf(string[4]) == -1) {  
 throw new Exception("This status is not in the list");  
 }  
 if (listAlphabet.indexOf(string[1]) == -1 && listAlphabet.indexOf(string[2]) == -1) {  
 throw new Exception("This character is not in the list");  
 }  
 if (!string[3].equals("L") && !string[3].equals("R") && !string[3].equals("T")) {  
 throw new Exception("format error");  
 }  
  
 hashMap.put(new Pair<>(string[0], string[1]), string);  
 current = br.readLine();  
 }  
 br.close();  
}

**- Метод loadTape:** для загрузки входной лент и первого состояния из текстового файла

private void loadTape(String inputName) throws Exception {  
  
 BufferedReader br = new BufferedReader(new FileReader(new File(inputName)));  
  
 this.pointer = Integer.*parseInt*(br.readLine());  
 this.initialState = br.readLine();  
 String strTape = br.readLine();  
  
 if (listStatus.indexOf(initialState) == -1) {  
 throw new Exception("This status is not in the list");  
 }  
  
 for (int i = 0; i < strTape.length(); i++) {  
 tape.addLast(String.*valueOf*(strTape.charAt(i)));  
 }  
  
 if (pointer < 0) {  
 int count = -1;  
 while (count != pointer) {  
 tape.addFirst("\_");  
 count--;  
 }  
 tape.addFirst("\_");  
 } else if (pointer >= strTape.length()) {  
 int count = strTape.length();  
 while (count != pointer) {  
 tape.addLast("\_");  
 count++;  
 }  
 tape.addLast("\_");  
 }  
  
 br.close();  
}

**- Метод run: выполнение программы.**

public void run() throws IOException {  
 BufferedWriter bw = new BufferedWriter(new FileWriter(new File("process.txt")));  
 BufferedWriter bw2 = new BufferedWriter(new FileWriter(new File("result.txt")));  
  
 boolean stop = false;  
 Pair<String, String> currentPair = new Pair<>(initialState, tape.get(pointer));  
 while (!stop) {  
 String[] nextState = hashMap.get(currentPair);  
 bw.write("command : " + Arrays.*toString*(nextState));  
 bw.newLine();  
  
 if (nextState == null) {  
 stop = true;  
 System.*out*.println("khong du trang thai");  
 } else {  
 stop = (nextState[4].equals("STOP"));  
 tape.set(pointer, nextState[2]);  
 switch (nextState[3]) {  
 case "L": {  
 pointer--;  
 if (tape.get(pointer) == null) {  
 tape.addFirst("\_");  
 }  
 break;  
 }  
 case "R": {  
 pointer++;  
 if (tape.get(pointer) == null) {  
 tape.addLast("\_");  
 }  
 break;  
 }  
 }  
 }  
 bw.write(this.toString());  
 bw.newLine();  
 assert nextState != null;  
 currentPair = new Pair<>(nextState[4], tape.get(pointer));  
 }  
  
 bw2.write("RESULT : " + this.toString());  
 bw.close();  
 bw2.close();  
}

1. **Класс DoublyLinkedListlmpl : реализовано tape на основе этого класса.**
2. **Испольные входные файлы :**

* input.txt : этот файл содержит положение головки, первое состояние и входной лент
* alphabet.txt : этот файл содержит символ из алфавита и множество символов (пробел заменяется знаком подчеркивания «\_»)
* command.txt : этот файл содержит список команд программы

1. Выходные файлы :

* process.txt : этот файл содержит текущие ленты (лент получается после выполнения каждого комманды).
* resut.txt : этот файл содержит результат.

1. не сколько заметок:

* по умолчению вхоный лент начинается разложить из клетки с индексом 0.
* Направление перемещения: R (вправо), L (влево) или T (на месте)
* Состояние STOP — это конечное состояние: попав в него, автомат заканчивает работу.
* В входных текстовых файлах содержит следующий пример : a) Циклический сдвиг влево на заданное число разрядов (оба аргумента в двоичном коде). b) данны два числа (А и В) в двоичной системе счисления, разделенные знаком "-". Вычислить разность (головка находиться в последной клетки- значить 7)

1. **Испытание программы:**
2. Циклический сдвиг влево на заданное число разрядов (оба аргумента в двоичном коде):

- входный лент : 1011\_0000100

- правильный результат : 1000000

- результат программы:



* + **Верно**

1. данны два числа (А и В) в двоичной системе счисления, разделенные знаком "-". Вычислить разность (головка находиться в последной клетки- значить 7)

- входный лент : 1010-111

- правильный результат : 11

- результат программы:



* + - **Верно**

1. **Вывод :**

- В данном курсовом проекте мною была написана программа, реализующая симулятор машин Тьюринга типа 5-tuple.

- Она поддерживает базовый функционал, необходимый для корректной работы программы, а также полностью соответствует заданным мне в индивидуальном задании требованиям.

1. **Список источник**

**http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/euctm.pdf**