МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

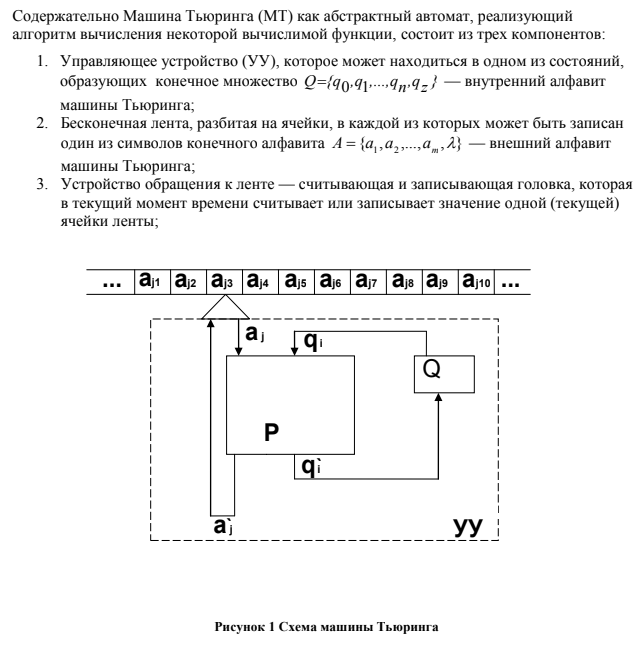
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ассистент | |  |  | | | |  | | Д. А. Кочин |
| должность, уч. степень, звание | |  | подпись, дата | | | |  | | инициалы, фамилия |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 | | | | | | | | | | |
| Изучение принципов функционирования машины Тьюринга | | | | | | | | | | |
| по дисциплине: ТЕОРИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ | | | | | | | | | | |
|  | | | | | | | | | | |
| РАБОТУ ВЫПОЛНИЛА | | | | | | | | | | |
| СТУДЕНТКА ГР. | 4931 | | |  | 21.10.2022 |  | | Е.Ю. Ильченко | | |
|  |  | | |  | подпись, дата |  | | инициалы, фамилия | | |
|  |  | | |  |  |  | |  | | |

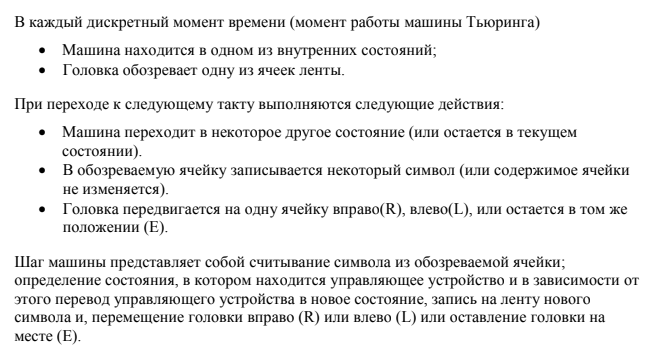
Санкт-Петербург 2022

1. **Цель работы:**

Изучить принципы функционирования машины Тьюринга.

1. **Основные сведения из теории:**





1. **Постановка задачи:**

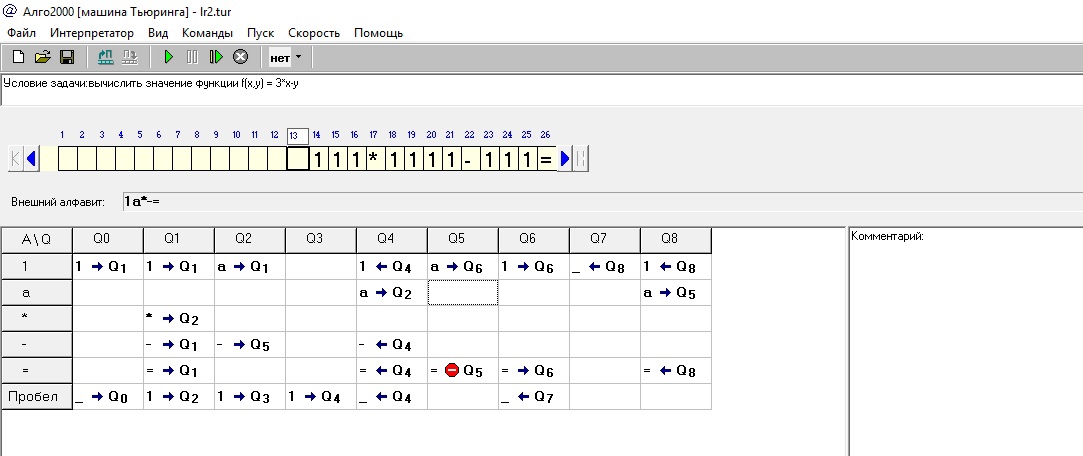
|  |  |
| --- | --- |
| **Вариант 39** | 3x - y |

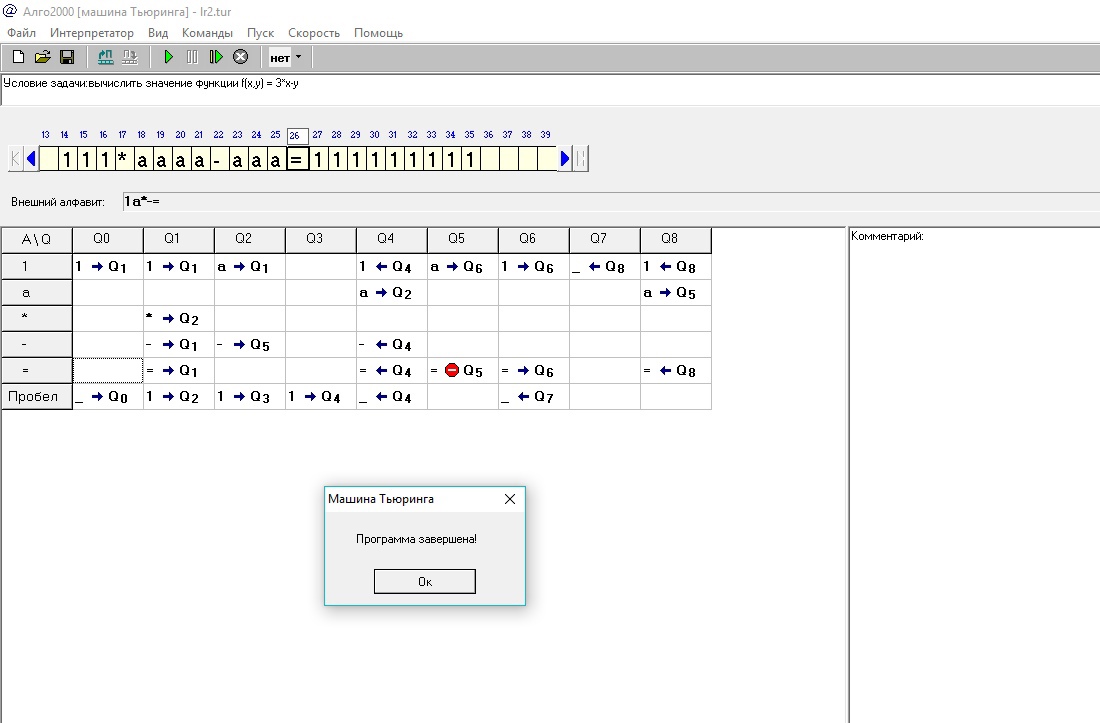
1. Необходимо написать программу для машины Тьюринга, реализующую вычисление арифметической функции согласно выданному варианту задания. Должна быть составлена совокупность команд P. Для выполнения данного задания следует использовать приложение Algo2000.

Аргументы задаются набором ”1”. Пример 2\*3, будет выглядеть следующим образом 11\*111.

Работа машины Тьюринга должна начинаться со стандартной начальной конфигурации и заканчиваться стандартной конечной конфигурацией.

1. Во второй части лабораторной работы требуется создать программу на языке высокого уровня имитирующую работу машины Тьюринга.
2. **Совокупность команд для машины Тьюринга**

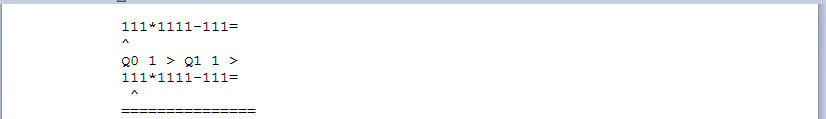


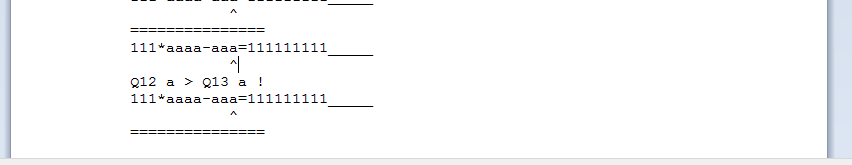


1. **Листинг программы:**

import java.io.\*;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Vector;  
  
import static java.sql.DriverManager.*println*;  
  
public class Main {  
 static class mTuringa {  
 private String[] alphabet;  
 private HashMap<String, String> rules = new HashMap<String, String>();  
 private String input = ""; *// Лента* private int counter = 0; *// Положение головки* private String instance = "Q0"; *// Состояние* private boolean exit = false; *// Флаг выхода* private String output = ""; *// Вывод* public void start() throws IOException {  
 while (!exit) {  
  
 if (input.length() - 1 == counter) input += "\_";  
  
 output += input + "\n";  
 for (int i = 0; i < counter; i++)  
 output += " ";  
 output += "^\n";  
 output += instance + " "+ input.charAt(counter) + " > " + rules.get(instance+input.charAt(counter))+"\n";  
 if (rules.get(instance + input.charAt(counter)) == null)  
 makeIteration("error");  
 else  
 makeIteration(rules.get(instance + input.charAt(counter)));  
  
 output += input + "\n";  
 for (int i = 0; i < counter; i++)  
 output += " ";  
 output += "^\n";  
 output += "===============\n";  
 }  
 BufferedWriter bfwrite = new BufferedWriter(new FileWriter("output.txt"));  
 bfwrite.write(output);  
 bfwrite.flush();  
 bfwrite.close();  
 }  
  
 public void loadData(String alphabet, String input) {  
 this.input = input;  
 this.alphabet = alphabet.split("\\s+");  
 }  
  
 public void makeConfig(Vector<String> rulesList) {  
 *// Для каждого правила заносим его в карту* for (String rule : rulesList) {  
 String[] rule1 = rule.split("\\s+");  
 rules.put(rule1[0] + rule1[1], rule1[3] + " " + rule1[4]+" "+rule1[5]);  
 }  
 }  
  
 private void makeIteration(String expr) {  
 if (expr == "error") {  
 *println*("Incorrect instance or char in input");  
 exit = true;  
 return;  
 }  
  
  
 String[] rule = expr.split("\\s+");  
 instance = rule[0];  
 input = input.substring(0, counter) + rule[1] + input.substring(counter + 1, input.length());  
 switch (rule[2]) {  
 case (">"):  
 counter++;  
 break;  
 case ("<"):  
 counter--;  
 break;  
 case ("!"):  
 exit = true;  
 }  
 }  
 }  
  
 public static void main(String[] args) throws IOException {  
 mTuringa turing = new mTuringa();  
 turing.loadData(new BufferedReader(new FileReader("alphabet.txt")).readLine(),  
 new BufferedReader(new FileReader("input.txt")).readLine());  
 *// Считываем правила* Vector<String> rules = new Vector<String>();  
 String line;  
 try {  
  
 BufferedReader bufferreader = new BufferedReader(new FileReader("rules.txt"));  
  
  
 while ((line = bufferreader.readLine()) != null) {  
 rules.add(line);  
 }  
  
 } catch (FileNotFoundException ex) {  
 ex.printStackTrace();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 turing.makeConfig(rules);  
 turing.start();  
 }  
}

1. **Пример результата выполнения:**





1. **Вывод:**

В ходе выполнения данной лабораторной работы были изучены принципы функционирования машины Тьюринга.