МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

Донецкий Национальный Технический Университет

КАФЕДРА ПРОГРАМНОЙ ИНЖЕНЕРИИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Теория алгоритмов и формальных языков»

на тему: «Машина Тьюринга»

|  |  |
| --- | --- |
| Руководитель:  кафедры ПИ | Выполнил:  студент гр. ПИ-18б  Моргунов А. Г. |

Донецк – 2019

Реферат

Пояснительная записка к курсовому проекту содержит: 43 страниц, 10 рисунков, 22 таблиц, 3 источника, 3 приложения.

Объект исследования – рекурсивные функции, машины Тьюринга, нормальные алгоритмы Маркова.

Цель – определить алгоритм одноленточной и многоленточной распознающей машины Тьюринга, выполняющей поиск слова в алфавите L = {anb2nc4n ⎢n>=1} и программно реализовать его. Построить график временной сложности.

Результаты работы – Объектно-ориентированная система имитирующая работу многоленточной и одноленточной распознающей машины Тьюринга с возможностью вывода на экран графика временной сложности. Файл с листингом программы.

МАШИНА ТЬЮРИНГА, АЛФАВИТ, ГРАФИК ВРЕМЕННОЙ СЛОЖНОСТИ, ЛЕНТА, ЯЗЫК, РАСПОЗНАВАНИЕ, ПРОТОКОЛ, ПРИНАДЛЕЖНОСТЬ

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc26140149)

[1. ОПИСАНИЕ АНАЛИТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА В ВИДЕ ЭЛЕМЕНТАРНЫХ МАШИН ТЬЮРИНГА И КОМПОЗИЦИИ МТ 5](#_Toc26140150)

[1.1 Машина Тьюринга 5](#_Toc26140151)

[1.2 Композиция машины Тьюринга 8](#_Toc26140152)

[2. РАЗРАБОТКА АНАЛИТИЧЕСКОЙ И ПРОГРАММНОЙ МОДЕЛИ АЛГОРИТМА ДЛЯ РАСПОЗНАЮЩЕЙ МАШИНЫ ТЬЮРИНГА 10](#_Toc26140153)

[2.1 Формальное определение машины распознающей Тьюринга 10](#_Toc26140154)

[2.1.1 Многоленточная МТ 10](#_Toc26140155)

[2.1.2 Одноленточная МТ 12](#_Toc26140156)

[2.2 Протоколы работы МТ 13](#_Toc26140157)

[2.2.1 Протоколы многоленточной МТ 13](#_Toc26140158)

[2.2.2 Протоколы одноленточной МТ 15](#_Toc26140159)

[2.3 Программная модель МТ 18](#_Toc26140160)

[2.3.1 Одноленточная МТ 18](#_Toc26140161)

[2.3.2 Многоленточная МТ 18](#_Toc26140162)

[2.4 Протоколы работы машины Тьюринга, построенные программно 19](#_Toc26140163)

[2.4.1 Листинг многоленточной МТ 19](#_Toc26140164)

[2.4.2 Листинг одноленточной МТ 21](#_Toc26140165)

[2.5 Расчет временной сложности 24](#_Toc26140166)

[3. РАЗРАБОТКА АНАЛИТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА С ИСПОЛЬЗОВАНИЕМ НОРМАЛЬНЫХ АЛГОРИТМОВ МАРКОВА 25](#_Toc26140167)

[4. ОПИСАНИЕ ФОРМАЛЬНОЙ МОДЕЛИ АЛГОРИТМА НА ОСНОВЕ РЕКУРСИВНЫХ ФУНКЦИЙ 26](#_Toc26140168)

[ВЫВОДЫ 27](#_Toc26140169)

[ПЕРЕЧЕНЬ ССЫЛОК 28](#_Toc26140170)

[ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 29](#_Toc26140171)

[ПРИЛОЖЕНИЕ Б. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 32](#_Toc26140172)

[ПРИЛОЖЕНИЕ В. КОД ПРОГРАММЫ 33](#_Toc26140173)

# ВВЕДЕНИЕ

Теория алгоритмов изучает алгоритмы, их особенности и разнообразные формальные модели их представления. Наиболее простыми и эффективными являются 3 формы: машина Тьюринга, нормальные алгоритмы Маркова, рекурсивные функции. Считается, что с их помощью можно представить любой алгоритм, и они с ним справятся за конечное количество шагов.

В современном мире эта проблема актуальна, потому что с распространением технологий возрастает потребность в реализации более сложных и оптимизации уже существующих алгоритмов.

Целью разработки является создание программной модели машины Тьюринга, распознающий язык L = {anb2nc4n ⎢n ≥ 1}, расчет и экспериментальная проверка расчета временной сложности МТ.

Применение – разработанная машина Тьюринга способна распознавать слова определенного вида над исходным алфавитом, что может быть использовано для проверки корректности результатов других расчетов или алгоритмов.

# ОПИСАНИЕ АНАЛИТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА В ВИДЕ ЭЛЕМЕНТАРНЫХ МАШИН ТЬЮРИНГА И КОМПОЗИЦИИ МТ

# 1.1 Машина Тьюринга

Маши́на Тью́ринга (МТ) — абстрактный исполнитель (абстрактная вычислительная машина). Была предложена Аланом Тьюрингом в 1936 году для формализации понятия алгоритма. Машина Тьюринга является расширением конечного автомата и, согласно тезису Чёрча — Тьюринга, способна имитировать всех исполнителей (с помощью задания правил перехода), каким-либо образом реализующих процесс пошагового вычисления, в котором каждый шаг вычисления достаточно элементарен[1].

Рассмотрим реализацию алгоритма, конструирующим в алфавите A={0,1} слова вида α=110102103…10n1, где n - произвольное натуральное число.

Рассмотрим функциональную таблицу многоленточной машины Тьюринга, выполняющий данную задачу(таблица 1.1).

Таблица 1.1 – Функциональная таблица

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | = | x | y | λ |
| q0 | q01R | - | - | - | - | q1=R |
| q1 | - | - | - | - | - | q21R |
| q2 | - | - | - | - | - | q31L |
| q3 | q31L | q30L | q4=L | - | q3yL | - |
| q4 | q5xR | - | - | q4xL | - | q11λR |
| q5 | - | - | q6 =R | q5xR | - | - |
| q6 | q61R | q60R | - | - | q6yR | q70L |
| q7 | q81L | q70L | - | - | - | - |
| q8 | q91R | q10yR | - | - | q8yL | - |
| q9 | q91R | q90R | - | - | q9yR | q31L |
| q10 | q101R | q100R | - | - | q10yR | q7 0L |
| q11 | q111R | q110R | q11λR | q11λR | q110R | q12λL |
| q12 | q12 1L | q12 0R | - | - | - | qzλR |

Система команд(таблица 1.2)

Таблица 1.2 – Система команд

|  |  |  |
| --- | --- | --- |
| q01→ q01R  q0 λ→ q1=R  q1 λ→ q21R  q2 λ→ q31L  q31→ q31L  q30→ q30L  q3=→ q4=L  q3 y→ q3yL  q41→ q5xR  q4 x→ q4xL  q4 λ→ q11λR  q5=→ q6 =R  q5 x→ q5xR | q61→ q61R  q60→ q60R  q6 y→ q6yR  q6 λ→ q70L  q71→ q81L  q70→ q70L  q81→ q91R  q80→ q10yR  q8 y→ q8yL  q91→ q91R  q90→ q90R  q9 y→ q9yR  q9 λ→ q31L | q101→ q101R  q100→ q100R  q10 y→ q10yR  q10 λ→ q7 0L  q111→ q111R  q110→ q110R  q11=→ q11λR  q11 x→ q11λR  q11 y→ q110R  q11 λ→ q12λL  q121→ q12 1L  q120→ q12 0R  q12 λ→ qzλR |

Диаграмма состояний(рис.1.2)

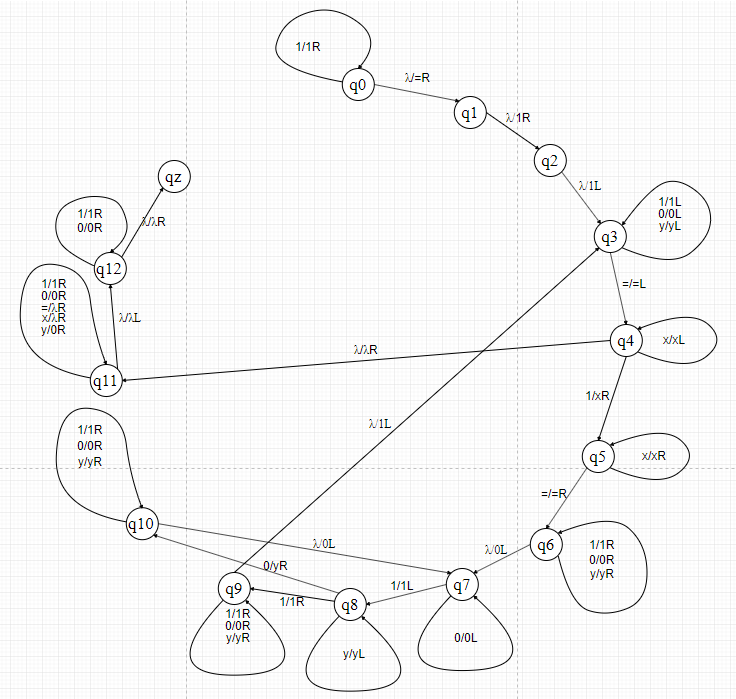


Рисунок 1.1 – Диаграмма состояний

Контрольные примеры(таблица 1.3 – таблица 1.4)

Таблица 1.3 – Пример №1

|  |  |  |
| --- | --- | --- |
| q01  1q0  1=q1  1=1q2  1=q311  1q3=11  q41=11  q4Y1=11  xq5=11  x=q611  x=1q61  x=11q6 | x=1q710  x=q8110  x=1q910  x=11q90  x=110q9  x=11q301  x=1q3101  x=q31101  xq3=1101  q4x=1101  q4Yx=1101  q11x=1101 | q11=1101  q111101  1q11101  11q1101  110q111  1101q11  110q121  11q1201  1q12101  q121101  q12Y1101  qz1101 |

Таблица 1.4 – Пример №2

|  |  |  |
| --- | --- | --- |
| q011  1q01  11q0  11=q1  11=1q2  11=1q31  11=q311  11q3=11  1q41=11  1xq5=11  1x=q611  1x=1q61  1x=11q6  1x=1q710  1x=q8110  1x=1q910  1x=11q90  1x=110q9  1x=11q301  1x=1q3101  1x=q31101  1xq3=1101  1q4x=1101  q41x=1101  xq5x=1101  xxq5=1101 | xx=q61101  xx=1q6101  xx=11q601  xx=110q61  xx=1101q6  xx=110q710  xx=11q8010  xx=11yq1010  xx=11y1q100  xx=11y10q10  xx=11y1q700  xx=11yq7100  xx=11q8y100  xx=1q81y100  xx=q911y100  xx=1q91y100  xx=11q9y100  xx=11yq9100  xx=11y1q900  xx=11y10q90  xx=11y100q9  xx=11y10q301  xx=11y1q3001  xx=11yq31001  xx=11q3y1001  xx=1q31y1001 | xx=q311y1001  xxq3=11y1001  xq4x=11y1001  q4xx=11y1001  q4Yxx=11y1001  q11xx=11y1001  q11x=11y1001  q11=11y1001  q1111y1001  1q111y1001  11q11y1001  110q111001  1101q11001  11010q1101  110100q111  1101001q11  110100q121  11010q1201  1101q12001  110q121001  11q1201001  1q12101001  q121101001  q12Y1101001  qz1101001 |

# 1.2 Композиция машины Тьюринга

Рассмотрим реализацию композиции машины Тьюринга для реализации алгоритма поиска количества простых чисел ≤ x

Для построения композиции сначала необходимо построить блок-схему алгоритма.

Изобразим блок-схему для основного алгоритма(рис 1.2)

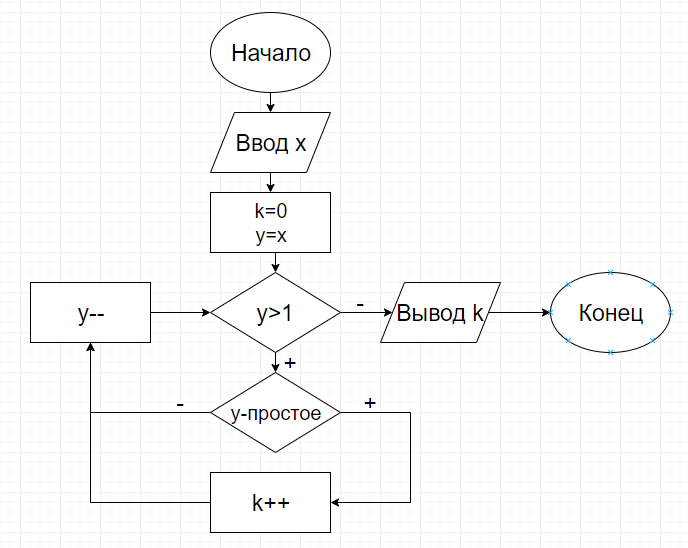


Рисунок 1.2 – Блок-схема основной МТ

Для реализации этого алгоритма требуется использование МТ, которая определяет простое ли число. Изобразим блок-схему для МТ, определяющую простоту числа(рис 1.3).

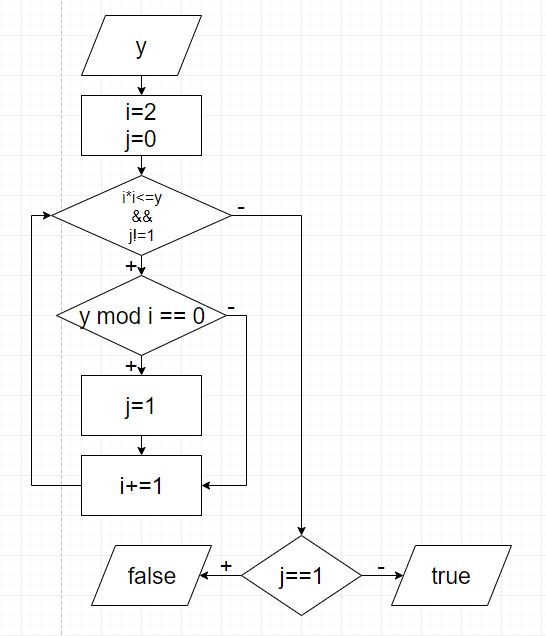


Рисунок 1.3 – Блок-схема МТ проверяющей простоту числа y

С помощью построенных блок-схем реализуем композиции(рис 1.4-1.5)

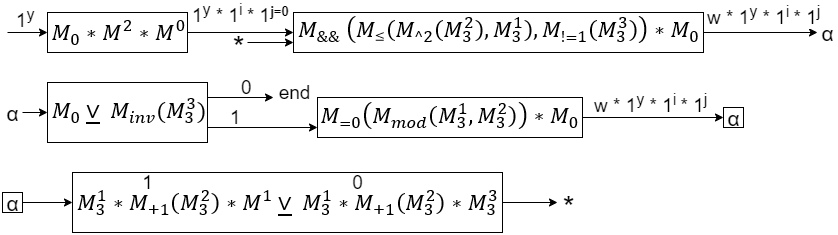


Рисунок 1.4 – Композиция МТ проверяющей число на простоту

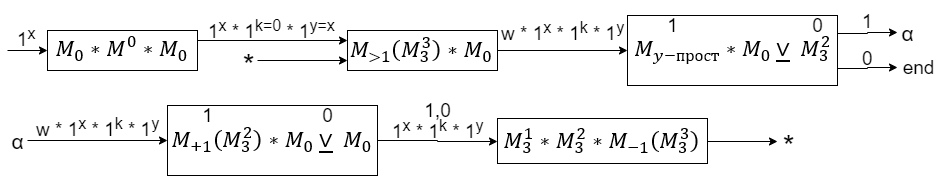


Рисунок 1.5 – Композиция основной МТ

Рассмотрим МТ, использованные в композиции:

М0 – копирование входного слова

Мn – установка константы n

М±1 - ±1

Minv – увеличение на 1

Mmod – остаток от деления

M!=1 – операция !=1

M>1 – операция >1

M&& - логическое «и»

M^2 – квадрат числа

– выбор аргумента

# РАЗРАБОТКА АНАЛИТИЧЕСКОЙ И ПРОГРАММНОЙ МОДЕЛИ АЛГОРИТМА ДЛЯ РАСПОЗНАЮЩЕЙ МАШИНЫ ТЬЮРИНГА

Рассмотрим реализацию машины Тьюринга для языка L = {anb2nc4n ⎢n>=1} , способ реализации в программе, построим график временной сложности.

# 2.1 Формальное определение машины распознающей Тьюринга

# 2.1.1 Многоленточная МТ

На первую ленту подается входное слово. Вторая используется как вспомогательная для счета.

Суть машины в том, чтобы сопоставить количество символов ‘a’,’b’,’c’ в отношении 1:2:4 соответственно в порядке a,b,c. Если слово подходит в конце ставим «+» , иначе «-». Для работы данного алгоритма была составлена система команд на двух лентах(таблица 2.1).

Таблица 2.1 – Система команд для многоленточной МТ

|  |  |  |
| --- | --- | --- |
| q0(a, λ) →(0;a, a; R, R)  q0(b, λ) →(1;b, λ; E, L)  q0(c, λ) →(8;c, λ; R, E)  q0(λ, λ) →(8; λ, λ; E, E)  q1(a, λ) →(8;a, λ; R, E)  q1(b, λ) →(8;b, λ; R, E)  q1(c, λ) →(4;c, λ; E, R)  q1(λ, λ) →(8; λ, λ; E, E)  q1(a, a) →(8;a, a; R, E)  q1(b, a) →(2;b, a; R, E)  q1(c, a) →(8;c, a; R, E)  q1(λ, a) →(8; λ, a; E, E)  q2(a, λ) →(8;a, λ; R, L)  q2(b, λ) →(8;b, λ; R, L)  q2(c, λ) →(8;c, λ; R, L)  q2(λ, λ) →(8; λ, λ; E, L) | q3(λ, λ) →(8; λ, λ; E, E)  q3(a, a) →(8;a, a; R, E)  q3(b, a) →(8;b, a; R, E)  q3(c, a) →(8;c, a; R, E)  q3(λ, a) →(8; λ, a; E, E)  q4(a, λ) →(8;a, λ; R, E)  q4(b, λ) →(8;b, λ; R, E)  q4(c, λ) →(8;c, λ; R, E)  q4(λ, λ) →(z;+, λ; E, L)  q4(a, a) →(8;a, a; R, E)  q4(b, a) →(8;b, a; R, E)  q4(c, a) →(5;c, a; R, E)  q4(λ, a) →(8; λ, a; E, E)  q5(a, λ) →(8;a, λ; R, E)  q5(b, λ) →(8;b, λ; R, E)  q5(c, λ) →(8;c, λ; R, E) | q6(c, λ) →(8;c, λ; R, E)  q6(λ, λ) →(8; λ, λ; E, E)  q6(a, a) →(8;a, a; R, E)  q6(b, a) →(8;b, a; R, E)  q6(c, a) →(7;c, a; R, E)  q6(λ, a) →(8; λ, a; E, E)  q7(a, λ) →(8;a, λ; R, E)  q7(b, λ) →(8;b, λ; R, E)  q7(c, λ) →(8;c, λ; R, E)  q7(λ, λ) →(8; λ, λ; E, E)  q7(a, a) →(8;a, a; R, E)  q7(b, a) →(8;b, a; R, E)  q7(c, a) →(4;c, a; R, R)  q7(λ, a) →(8; λ, a; E, E)  q8(a, λ) →(8;a, λ; R, E)  q8(b, λ) →(8;b, λ; R, E) |

Продолжение таблицы 2.1

|  |  |  |
| --- | --- | --- |
| q2(a, a) →(8;a, a; R, L)  q2(b, a) →(1;b, a; R, E)  q2(c, a) →(8;c, a; R, L)  q2(λ, a) →(8; λ, a; E, L)  q3(a, λ) →(8;a, λ; R, E)  q3(b, λ) →(8;b, λ; R, E)  q3(c, λ) →(8;c, λ; R, E) | q5(λ, λ) →(8; λ, λ; E, E)  q5(a, a) →(8;a, a; R, E)  q5(b, a) →(8;b, a; R, E)  q5(c, a) →(6;c, a; R, E)  q5(λ, a) →(8; λ, a; E, E)  q6(a, λ) →(8;a, λ; R, E)  q6(b, λ) →(8;b, λ; R, E) | q8(c, λ) →(8;c, λ; R, E)  q8(λ, λ) →(z;-, λ; E, E)  q8(a, a) →(8;a, a; R, E)  q8(b, a) →(8;b, a; R, E)  q8(c, a) →(8;c, a; R, E)  q8(λ, a) →(z;-, a; E, E) |

Подробное описание состояний:

q0 – переписывает символы «a» на вторую ленту. Если встречает «b» на главной ленте то переходит к состоянию q1, а по вспомогательной делает шаг влево. Иначе переходит в состояние q8 .

q1-продолжает проверять исходное слово, по вспомогательной ленте стоим на месте, если встречается «b» и на вспомогательной «a», то переходит к q2. Если по вспомогательной встречает лямбду, а по основной «с» то переходит к q4. Иначе переходит в q8.

q2- продолжает проверять исходное слово, если встречает «b» на основной ленте то по вспомогательной передвигаемся влево и возвращаемся в q1, таким образом зацикливая процесс проверки количества символов «b». Иначе переходит в q8.

q4- продолжает проверять исходное слово, по вспомогательной ленте стоим на месте, если встречается «c» и на вспомогательной «a», то переходит к q5. Если на основной и вспомогательной лентах λ, то ставит «+». Иначе переходит в состояние q8 .

q5- продолжает проверять исходное слово, по вспомогательной ленте стоим на месте, если встречается «c» и на вспомогательной «a», то переходит к q6. Иначе переходит в состояние q8 .

q6- продолжает проверять исходное слово, по вспомогательной ленте стоим на месте, если встречается «c» и на вспомогательной «a», то переходит к q7. Иначе переходит в состояние q8 .

q7- продолжает проверять исходное слово, если встречает «с» на основной ленте то по вспомогательной передвигаемся вправо и возвращаемся в q4, таким образом зацикливая процесс проверки количества символов «с». Иначе переходит в q8.

q8- переходит к правому концу слова и ставит «-» .

# 2.1.2 Одноленточная МТ

На первую ленту подается входное слово, на ней же слово и проверяется.

Суть машины в том, чтобы сопоставить количество символов ‘a’,’b’,’c’ в отношении 1:2:4 соответственно в порядке a,b,c. Если слово подходит в конце ставим «+» , иначе «-». Для работы данного алгоритма была составлена таблица команд одноленточной МТ(Рисунок 2.1).

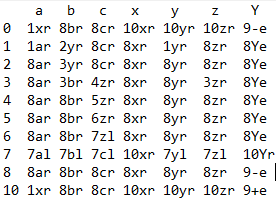


Рисунок 2.1 – Таблица команд

Подробное описание команд:

Q0- проверка слова на пустоту если не пустое замена первого «а» на «х» и переход в Q1. иначе переход в Q8.

Q1-поиск «b» и замена найденного «b» на «y», и переход вправо переход в Q2. Иначе переход в Q8.

Q2-если «b», то меняем его на «y», и переходим в Q3. Иначе переход в Q8.

Q3- поиск «с» и замена найденного «c» на «z», и переход вправо переход в Q4. иначе переход в Q8.

Q4- если «с», то меняем его на «z» и переходим в Q5. Иначе переход в Q8.

Q5- если «с», то меняем его на «z» и переходим в Q6. Иначе переход в Q8.

Q6- если «с», то меняем его на «z» и переходим в Q7. Иначе переход в Q8.

Q7-Возвращаемся влево до «x» и переходим в Q10.

Q8-Ставим «-» в конце слова.

Q10-Замена «a» на «x».

# 2.2 Протоколы работы МТ

# 2.2.1 Протоколы многоленточной МТ

Первый пример: “aabbbcc” (таблица 2.2)

Таблица 2.2 – Контрольный пример многоленточной №1

|  |  |
| --- | --- |
| Первая лента: Y0aabbbccY  Вторая лента: Y0Y  Первая лента: Ya0abbbccY  Вторая лента: Ya0Y  Первая лента: Yaa0bbbccY  Вторая лента: Yaa0Y  Первая лента: Yaa1bbbccY  Вторая лента: Ya1aY  Первая лента: Yaab2bbccY  Вторая лента: Ya2aY | Первая лента: Yaabb1bccY  Вторая лента: Y1aaY  Первая лента: Yaabbb2ccY  Вторая лента: Y2aaY  Первая лента: Yaabbbc8cY  Вторая лента: Y8aaY  Первая лента: Yaabbbcc8Y  Вторая лента: Y8aaY  Результат:  Первая лента: Yaabbbcc-  Вторая лента: YaaY |

Второй пример: “aaabbbbbbcccccccccccc” (таблица 2.3)

Таблица 2.3 – Контрольный пример многоленточной №2

|  |  |
| --- | --- |
| Первая лента: Y0aaabbbbbbccccccccccccY  Вторая лента: Y0Y  Первая лента: Ya0aabbbbbbccccccccccccY  Вторая лента: Ya0Y  Первая лента: Yaa0abbbbbbccccccccccccY  Вторая лента: Yaa0Y  Первая лента: Yaaa0bbbbbbccccccccccccY  Вторая лента: Yaaa0Y  Первая лента: Yaaa1bbbbbbccccccccccccY  Вторая лента: Yaa1aY  Первая лента: Yaaab2bbbbbccccccccccccY  Вторая лента: Yaa2aY  Первая лента: Yaaabb1bbbbccccccccccccY  Вторая лента: Ya1aaY  Первая лента: Yaaabbb2bbbccccccccccccY  Вторая лента: Ya2aaY  Первая лента: Yaaabbbb1bbccccccccccccY  Вторая лента: Y1aaaY  Первая лента: Yaaabbbbb2bccccccccccccY  Вторая лента: Y2aaaY  Первая лента: Yaaabbbbbb1ccccccccccccY  Вторая лента: 1YaaaY  Первая лента: Yaaabbbbbb4ccccccccccccY  Вторая лента: Y4aaaY  Первая лента: Yaaabbbbbbc5cccccccccccY  Вторая лента: Y5aaaY | Первая лента: Yaaabbbbbbcc6ccccccccccY  Вторая лента: Y6aaaY  Первая лента: Yaaabbbbbbccc7cccccccccY  Вторая лента: Y7aaaY  Первая лента: Yaaabbbbbbcccc4ccccccccY  Вторая лента: Ya4aaY  Первая лента: Yaaabbbbbbccccc5cccccccY  Вторая лента: Ya5aaY  Первая лента: Yaaabbbbbbcccccc6ccccccY  Вторая лента: Ya6aaY  Первая лента: Yaaabbbbbbccccccc7cccccY  Вторая лента: Ya7aaY  Первая лента: Yaaabbbbbbcccccccc4ccccY  Вторая лента: Yaa4aY  Первая лента: Yaaabbbbbbccccccccc5cccY  Вторая лента: Yaa5aY  Первая лента: Yaaabbbbbbcccccccccc6ccY  Вторая лента: Yaa6aY  Первая лента: Yaaabbbbbbccccccccccc7cY  Вторая лента: Yaa7aY  Первая лента: Yaaabbbbbbcccccccccccc4Y  Вторая лента: Yaaa4Y  Результат:  Первая лента: Yaaabbbbbbcccccccccccc+  Вторая лента: YaaaY |

Третий пример: “abbcccc” (таблица 2.4)

Таблица 2.4 – Контрольный пример многоленточной №3

|  |  |
| --- | --- |
| Первая лента: Y0abbccccY  Вторая лента: Y0Y  Первая лента: Ya0bbccccY  Вторая лента: Ya0Y  Первая лента: Ya1bbccccY  Вторая лента: Y1aY  Первая лента: Yab2bccccY  Вторая лента: Y2aY  Первая лента: Yabb1ccccY  Вторая лента: 1YaY  Первая лента: Yabb4ccccY | Вторая лента: Y4aY  Первая лента: Yabbc5cccY  Вторая лента: Y5aY  Первая лента: Yabbcc6ccY  Вторая лента: Y6aY  Первая лента: Yabbccc7cY  Вторая лента: Y7aY  Первая лента: Yabbcccc4Y  Вторая лента: Ya4Y  Результат:  Первая лента: Yabbcccc+  Вторая лента: YaY |

Четвертый пример: “ λ ” (таблица 2.5)

Таблица 2.5 – Контрольный пример многоленточной №4

|  |  |
| --- | --- |
| Первая лента: Y0Y  Вторая лента: Y0Y  Первая лента: Y8Y | Вторая лента: Y8Y  Результат:  Первая лента: Y-  Вторая лента: YY |

# 2.2.2 Протоколы одноленточной МТ

Первый пример: “aabbbcc” (таблица 2.6)

Таблица 2.6 – Контрольный пример одноленточной №1

|  |  |
| --- | --- |
| Y0aabbbccY  Yx1abbbccY  Yxa1bbbccY  Yxay2bbccY  Yxayy3bccY | Yxayyb3ccY  Yxayybz4cY  Yxayybzz5Y  Yxayybzz8Y  Yxayybzz- |

Второй пример: “aaabbbbbbcccccccccccc” (таблица 2.7)

Таблица 2.7 – Контрольный пример одноленточной №2

|  |  |
| --- | --- |
| Y0aaabbbbbbccccccccccccY  Yx1aabbbbbbccccccccccccY  Yxa1abbbbbbccccccccccccY  Yxaa1bbbbbbccccccccccccY  Yxaay2bbbbbccccccccccccY  Yxaayy3bbbbccccccccccccY  Yxaayyb3bbbccccccccccccY  Yxaayybb3bbccccccccccccY  Yxaayybbb3bccccccccccccY  Yxaayybbbb3ccccccccccccY  Yxaayybbbbz4cccccccccccY  Yxaayybbbbzz5ccccccccccY  Yxaayybbbbzzz6cccccccccY  Yxaayybbbbzz7zzccccccccY  Yxaayybbbbz7zzzccccccccY  Yxaayybbbb7zzzzccccccccY  Yxaayybbb7bzzzzccccccccY  Yxaayybb7bbzzzzccccccccY  Yxaayyb7bbbzzzzccccccccY  Yxaayy7bbbbzzzzccccccccY  Yxaay7ybbbbzzzzccccccccY  Yxaa7yybbbbzzzzccccccccY  Yxa7ayybbbbzzzzccccccccY  Yx7aayybbbbzzzzccccccccY  Y7xaayybbbbzzzzccccccccY  Yx10aayybbbbzzzzccccccccY  Yxx1ayybbbbzzzzccccccccY  Yxxa1yybbbbzzzzccccccccY  Yxxay1ybbbbzzzzccccccccY  Yxxayy1bbbbzzzzccccccccY  Yxxayyy2bbbzzzzccccccccY  Yxxayyyy3bbzzzzccccccccY  Yxxayyyyb3bzzzzccccccccY  Yxxayyyybb3zzzzccccccccY  Yxxayyyybbz3zzzccccccccY  Yxxayyyybbzz3zzccccccccY  Yxxayyyybbzzz3zccccccccY  Yxxayyyybbzzzz3ccccccccY  Yxxayyyybbzzzzz4cccccccY  Yxxayyyybbzzzzzz5ccccccY  Yxxayyyybbzzzzzzz6cccccY  Yxxayyyybbzzzzzz7zzccccY  Yxxayyyybbzzzzz7zzzccccY | Yxx10ayyyybbzzzzzzzzccccY  Yxxx1yyyybbzzzzzzzzccccY  Yxxxy1yyybbzzzzzzzzccccY  Yxxxyy1yybbzzzzzzzzccccY  Yxxxyyy1ybbzzzzzzzzccccY  Yxxxyyyy1bbzzzzzzzzccccY  Yxxxyyyyy2bzzzzzzzzccccY  Yxxxyyyyyy3zzzzzzzzccccY  Yxxxyyyyyyz3zzzzzzzccccY  Yxxxyyyyyyzz3zzzzzzccccY  Yxxxyyyyyyzzz3zzzzzccccY  Yxxxyyyyyyzzzz3zzzzccccY  Yxxxyyyyyyzzzzz3zzzccccY  Yxxxyyyyyyzzzzzz3zzccccY  Yxxxyyyyyyzzzzzzz3zccccY  Yxxxyyyyyyzzzzzzzz3ccccY  Yxxxyyyyyyzzzzzzzzz4cccY  Yxxxyyyyyyzzzzzzzzzz5ccY  Yxxxyyyyyyzzzzzzzzzzz6cY  Yxxxyyyyyyzzzzzzzzzz7zzY  Yxxxyyyyyyzzzzzzzzz7zzzY  Yxxxyyyyyyzzzzzzzz7zzzzY  Yxxxyyyyyyzzzzzzz7zzzzzY  Yxxxyyyyyyzzzzzz7zzzzzzY  Yxxxyyyyyyzzzzz7zzzzzzzY  Yxxxyyyyyyzzzz7zzzzzzzzY  Yxxxyyyyyyzzz7zzzzzzzzzY  Yxxxyyyyyyzz7zzzzzzzzzzY  Yxxxyyyyyyz7zzzzzzzzzzzY  Yxxxyyyyyy7zzzzzzzzzzzzY  Yxxxyyyyy7yzzzzzzzzzzzzY  Yxxxyyyy7yyzzzzzzzzzzzzY  Yxxxyyy7yyyzzzzzzzzzzzzY  Yxxxyy7yyyyzzzzzzzzzzzzY  Yxxxy7yyyyyzzzzzzzzzzzzY  Yxxx7yyyyyyzzzzzzzzzzzzY  Yxx7xyyyyyyzzzzzzzzzzzzY  Yxxx10yyyyyyzzzzzzzzzzzzY  Yxxxy10yyyyyzzzzzzzzzzzzY  Yxxxyy10yyyyzzzzzzzzzzzzY  Yxxxyyy10yyyzzzzzzzzzzzzY  Yxxxyyyy10yyzzzzzzzzzzzzY  Yxxxyyyyy10yzzzzzzzzzzzzY |

Продолжение таблицы 2.7

|  |  |
| --- | --- |
| Yxxayyyybbzzzz7zzzzccccY  Yxxayyyybbzzz7zzzzzccccY  Yxxayyyybbzz7zzzzzzccccY  Yxxayyyybbz7zzzzzzzccccY  Yxxayyyybb7zzzzzzzzccccY  Yxxayyyyb7bzzzzzzzzccccY  Yxxayyyy7bbzzzzzzzzccccY  Yxxayyy7ybbzzzzzzzzccccY  Yxxayy7yybbzzzzzzzzccccY  Yxxay7yyybbzzzzzzzzccccY  Yxxa7yyyybbzzzzzzzzccccY  Yxx7ayyyybbzzzzzzzzccccY  Yx7xayyyybbzzzzzzzzccccY | Yxxxyyyyyy10zzzzzzzzzzzzY  Yxxxyyyyyyz10zzzzzzzzzzzY  Yxxxyyyyyyzz10zzzzzzzzzzY  Yxxxyyyyyyzzz10zzzzzzzzzY  Yxxxyyyyyyzzzz10zzzzzzzzY  Yxxxyyyyyyzzzzz10zzzzzzzY  Yxxxyyyyyyzzzzzz10zzzzzzY  Yxxxyyyyyyzzzzzzz10zzzzzY  Yxxxyyyyyyzzzzzzzz10zzzzY  Yxxxyyyyyyzzzzzzzzz10zzzY  Yxxxyyyyyyzzzzzzzzzz10zzY  Yxxxyyyyyyzzzzzzzzzzz10zY  Yxxxyyyyyyzzzzzzzzzzzz10Y  Yxxxyyyyyyzzzzzzzzzzzz+ |

Третий пример: “abbcccc” (таблица 2.8)

Таблица 2.8 – Контрольный пример одноленточной №3

|  |  |
| --- | --- |
| Y0abbccccY  Yx1bbccccY  Yxy2bccccY  Yxyy3ccccY  Yxyyz4cccY  Yxyyzz5ccY  Yxyyzzz6cY  Yxyyzz7zzY  Yxyyz7zzzY  Yxyy7zzzzY  Yxy7yzzzzY | Yx7yyzzzzY  Y7xyyzzzzY  Yx10yyzzzzY  Yxy10yzzzzY  Yxyy10zzzzY  Yxyyz10zzzY  Yxyyzz10zzY  Yxyyzzz10zY  Yxyyzzzz10Y  Yxyyzzzz+ |

Четвертый пример: “ λ ” (таблица 2.9)

Таблица 2.9 – Контрольный пример одноленточной №4

|  |  |
| --- | --- |
| Y0Y | Y- |

# 2.3 Программная модель МТ

# 2.3.1 Одноленточная МТ

В программе одноленточная МТ реализована с помощью класса MT. Лента реализована с помощью массива объектов класса Character workStringChras. Таблица команд же реализована с помощью двумерного массива matritsaMT, с количеством столбцов равным количеству букв в алфавите + лямбда + все дополнительно использованные символы, элементами которого являются объекты класса ChtoDelatOdnolentochnaya. В классе ChtoDelatOdnolentochnaya три поля toSostoyanie, toCharacter, toNapravlenie которые определяют в какое состояние перейти, какой символ поставить и куда передвинуть каретку соответственно. Обращение к массиву matritsaMT осуществляется с использованием текущего состояния и символа, стоящим под кареткой. Строка – это текущее состояние, а столбец зависит от текущего символа. Алгоритм реализован с использованием while, switch, if. Таким образом программа работает пока не достигнет финального состояния.

Преимущества такой реализации заключаются в том, что ее легко можно изменить или модифицировать.

Для подсчета количества шагов используется переменная o.

# 2.3.2 Многоленточная МТ

В программе многоленточная МТ реализована с помощью класса mnogolentochnayaMT. Ленты реализованы с помощью массива объектов класса Character workStringCharsFirst(основная лента) и динамического массива объектов класса Character workStringCharsSecond(вспомогательная лента). Таблица команд же реализована с помощью двух двумерных массивов matritsaMTY и matritsaMTa, с количеством столбцов равным количеству букв в алфавите + лямбда + все дополнительно использованные символы, элементами которых являются объекты класса ChtoDelatMnogolentochnaya. В классе ChtoDelatMnogolentochnaya пять полей toSostoyanie, toCharacter1, toCharacter2, toNapravlenie1, toNapravlenie2, которые определяют в какое состояние перейти, какой символ поставить на главной ленте, какой символ поставить на вспомогательной ленте, куда передвинуть каретку на основной ленте и куда передвинуть каретку на вспомогательной ленте соответственно. Обращение к массиву matritsaMTa или matritsaMTY осуществляется в зависимости от символа, стоящего под кареткой на вспомогательной ленте. Если на второй ленте каретка на символе a, то используется matritsaMTa, если λ, то используется matritsaMTY. После определения с каким массивом работать используется значение текущего состояния и внутрисистемный индекс символа, стоящим под кареткой на главной ленте. Строка – это текущее состояние, а столбец зависит от текущего символа. Алгоритм реализован с использованием while, switch, if. Таким образом программа работает пока не достигнет финального состояния.

Преимущества такой реализации заключаются в том, что ее легко можно изменить или модифицировать.

Для подсчета количества шагов используется переменная o.

# 2.4 Протоколы работы машины Тьюринга, построенные программно

Для вывода листинга программы используется текстовый файл Listing.txt.

# 2.4.1 Листинг многоленточной МТ

Первый пример: “aabbbcc” (таблица 2.10)

Таблица 2.10 – Пример листинга многоленточной №1

|  |  |
| --- | --- |
| Y0aabbbccY  Y0Y  Ya0abbbccY  Ya0Y  Yaa0bbbccY  Yaa0Y  Yaa1bbbccY  Ya1aY  Yaab2bbccY  Ya2aY | Yaabb1bccY  Y1aaY  Yaabbb2ccY  Y2aaY  Yaabbbc8cY  Y8aaY  Yaabbbcc8Y  Y8aaY  Yaabbbcc-  YaaY |

Второй пример: “aaabbbbbbcccccccccccc” (таблица 2.11)

Таблица 2.11 – Пример листинга многоленточной №2

|  |  |
| --- | --- |
| Y0aaabbbbbbccccccccccccY  Y0Y  Ya0aabbbbbbccccccccccccY  Ya0Y  Yaa0abbbbbbccccccccccccY  Yaa0Y  Yaaa0bbbbbbccccccccccccY  Yaaa0Y  Yaaa1bbbbbbccccccccccccY  Yaa1aY  Yaaab2bbbbbccccccccccccY  Yaa2aY  Yaaabb1bbbbccccccccccccY  Ya1aaY  Yaaabbb2bbbccccccccccccY  Ya2aaY  Yaaabbbb1bbccccccccccccY  Y1aaaY  Yaaabbbbb2bccccccccccccY  Y2aaaY  Yaaabbbbbb1ccccccccccccY  1YaaaY  Yaaabbbbbb4ccccccccccccY  Y4aaaY  Yaaabbbbbbc5cccccccccccY  Y5aaaY | Yaaabbbbbbcc6ccccccccccY  Y6aaaY  Yaaabbbbbbccc7cccccccccY  Y7aaaY  Yaaabbbbbbcccc4ccccccccY  Ya4aaY  Yaaabbbbbbccccc5cccccccY  Ya5aaY  Yaaabbbbbbcccccc6ccccccY  Ya6aaY  Yaaabbbbbbccccccc7cccccY  Ya7aaY  Yaaabbbbbbcccccccc4ccccY  Yaa4aY  Yaaabbbbbbccccccccc5cccY  Yaa5aY  Yaaabbbbbbcccccccccc6ccY  Yaa6aY  Yaaabbbbbbccccccccccc7cY  Yaa7aY  Yaaabbbbbbcccccccccccc4Y  Yaaa4Y  Yaaabbbbbbcccccccccccc+  YaaaY |

Третий пример: “abbcccc” (таблица 2.12)

Таблица 2.12 – Пример листинга многоленточной №3

|  |  |
| --- | --- |
| Y0abbccccY  Y0Y  Ya0bbccccY  Ya0Y  Ya1bbccccY  Y1aY  Yab2bccccY  Y2aY  Yabb1ccccY  1YaY  Yabb4ccccY  Y4aY | Yabbc5cccY  Y5aY  Yabbcc6ccY  Y6aY  Yabbccc7cY  Y7aY  Yabbcccc4Y  Ya4Y  Yabbcccc+  YaY |

Четвертый пример: “ λ ” (таблица 2.13)

Таблица 2.13 – Пример листинга многоленточной №4

|  |  |
| --- | --- |
| Y0Y  Y0Y  Y8Y  Y8Y | Y-  YY |

# 2.4.2 Листинг одноленточной МТ

Первый пример: “aabbbcc” (таблица 2.14)

Таблица 2.14 – Пример листинга одноленточной №1

|  |  |
| --- | --- |
| Y0aabbbccY  Yx1abbbccY  Yxa1bbbccY  Yxay2bbccY  Yxayy3bccY | Yxayyb3ccY  Yxayybz4cY  Yxayybzz5Y  Yxayybzz8Y  Yxayybzz- |

Второй пример: “aaabbbbbbcccccccccccc” (таблица 2.15)

Таблица 2.15 – Пример листинга одноленточной №2

|  |  |
| --- | --- |
| Y0aaabbbbbbccccccccccccY  Yx1aabbbbbbccccccccccccY  Yxa1abbbbbbccccccccccccY  Yxaa1bbbbbbccccccccccccY  Yxaay2bbbbbccccccccccccY  Yxaayy3bbbbccccccccccccY  Yxaayyb3bbbccccccccccccY  Yxaayybb3bbccccccccccccY  Yxaayybbb3bccccccccccccY  Yxaayybbbb3ccccccccccccY  Yxaayybbbbz4cccccccccccY  Yxaayybbbbzz5ccccccccccY  Yxaayybbbbzzz6cccccccccY  Yxaayybbbbzz7zzccccccccY  Yxaayybbbbz7zzzccccccccY  Yxaayybbbb7zzzzccccccccY  Yxaayybbb7bzzzzccccccccY  Yxaayybb7bbzzzzccccccccY  Yxaayyb7bbbzzzzccccccccY  Yxaayy7bbbbzzzzccccccccY  Yxaay7ybbbbzzzzccccccccY  Yxaa7yybbbbzzzzccccccccY  Yxa7ayybbbbzzzzccccccccY  Yx7aayybbbbzzzzccccccccY  Y7xaayybbbbzzzzccccccccY  Yx10aayybbbbzzzzccccccccY  Yxx1ayybbbbzzzzccccccccY  Yxxa1yybbbbzzzzccccccccY  Yxxay1ybbbbzzzzccccccccY  Yxxayy1bbbbzzzzccccccccY  Yxxayyy2bbbzzzzccccccccY  Yxxayyyy3bbzzzzccccccccY  Yxxayyyyb3bzzzzccccccccY  Yxxayyyybb3zzzzccccccccY  Yxxayyyybbz3zzzccccccccY  Yxxayyyybbzz3zzccccccccY  Yxxayyyybbzzz3zccccccccY  Yxxayyyybbzzzz3ccccccccY  Yxxayyyybbzzzzz4cccccccY  Yxxayyyybbzzzzzz5ccccccY  Yxxayyyybbzzzzzzz6cccccY  Yxxayyyybbzzzzzz7zzccccY  Yxxayyyybbzzzzz7zzzccccY | Yxx10ayyyybbzzzzzzzzccccY  Yxxx1yyyybbzzzzzzzzccccY  Yxxxy1yyybbzzzzzzzzccccY  Yxxxyy1yybbzzzzzzzzccccY  Yxxxyyy1ybbzzzzzzzzccccY  Yxxxyyyy1bbzzzzzzzzccccY  Yxxxyyyyy2bzzzzzzzzccccY  Yxxxyyyyyy3zzzzzzzzccccY  Yxxxyyyyyyz3zzzzzzzccccY  Yxxxyyyyyyzz3zzzzzzccccY  Yxxxyyyyyyzzz3zzzzzccccY  Yxxxyyyyyyzzzz3zzzzccccY  Yxxxyyyyyyzzzzz3zzzccccY  Yxxxyyyyyyzzzzzz3zzccccY  Yxxxyyyyyyzzzzzzz3zccccY  Yxxxyyyyyyzzzzzzzz3ccccY  Yxxxyyyyyyzzzzzzzzz4cccY  Yxxxyyyyyyzzzzzzzzzz5ccY  Yxxxyyyyyyzzzzzzzzzzz6cY  Yxxxyyyyyyzzzzzzzzzz7zzY  Yxxxyyyyyyzzzzzzzzz7zzzY  Yxxxyyyyyyzzzzzzzz7zzzzY  Yxxxyyyyyyzzzzzzz7zzzzzY  Yxxxyyyyyyzzzzzz7zzzzzzY  Yxxxyyyyyyzzzzz7zzzzzzzY  Yxxxyyyyyyzzzz7zzzzzzzzY  Yxxxyyyyyyzzz7zzzzzzzzzY  Yxxxyyyyyyzz7zzzzzzzzzzY  Yxxxyyyyyyz7zzzzzzzzzzzY  Yxxxyyyyyy7zzzzzzzzzzzzY  Yxxxyyyyy7yzzzzzzzzzzzzY  Yxxxyyyy7yyzzzzzzzzzzzzY  Yxxxyyy7yyyzzzzzzzzzzzzY  Yxxxyy7yyyyzzzzzzzzzzzzY  Yxxxy7yyyyyzzzzzzzzzzzzY  Yxxx7yyyyyyzzzzzzzzzzzzY  Yxx7xyyyyyyzzzzzzzzzzzzY  Yxxx10yyyyyyzzzzzzzzzzzzY  Yxxxy10yyyyyzzzzzzzzzzzzY  Yxxxyy10yyyyzzzzzzzzzzzzY  Yxxxyyy10yyyzzzzzzzzzzzzY  Yxxxyyyy10yyzzzzzzzzzzzzY  Yxxxyyyyy10yzzzzzzzzzzzzY |

Продолжение таблицы 2.7

|  |  |
| --- | --- |
| Yxxayyyybbzzzz7zzzzccccY  Yxxayyyybbzzz7zzzzzccccY  Yxxayyyybbzz7zzzzzzccccY  Yxxayyyybbz7zzzzzzzccccY  Yxxayyyybb7zzzzzzzzccccY  Yxxayyyyb7bzzzzzzzzccccY  Yxxayyyy7bbzzzzzzzzccccY  Yxxayyy7ybbzzzzzzzzccccY  Yxxayy7yybbzzzzzzzzccccY  Yxxay7yyybbzzzzzzzzccccY  Yxxa7yyyybbzzzzzzzzccccY  Yxx7ayyyybbzzzzzzzzccccY  Yx7xayyyybbzzzzzzzzccccY | Yxxxyyyyyy10zzzzzzzzzzzzY  Yxxxyyyyyyz10zzzzzzzzzzzY  Yxxxyyyyyyzz10zzzzzzzzzzY  Yxxxyyyyyyzzz10zzzzzzzzzY  Yxxxyyyyyyzzzz10zzzzzzzzY  Yxxxyyyyyyzzzzz10zzzzzzzY  Yxxxyyyyyyzzzzzz10zzzzzzY  Yxxxyyyyyyzzzzzzz10zzzzzY  Yxxxyyyyyyzzzzzzzz10zzzzY  Yxxxyyyyyyzzzzzzzzz10zzzY  Yxxxyyyyyyzzzzzzzzzz10zzY  Yxxxyyyyyyzzzzzzzzzzz10zY  Yxxxyyyyyyzzzzzzzzzzzz10Y  Yxxxyyyyyyzzzzzzzzzzzz+ |

Третий пример: “abbcccc” (таблица 2.16)

Таблица 2.16 – Пример листинга одноленточной №3

|  |  |
| --- | --- |
| Y0abbccccY  Yx1bbccccY  Yxy2bccccY  Yxyy3ccccY  Yxyyz4cccY  Yxyyzz5ccY  Yxyyzzz6cY  Yxyyzz7zzY  Yxyyz7zzzY  Yxyy7zzzzY  Yxy7yzzzzY | Yx7yyzzzzY  Y7xyyzzzzY  Yx10yyzzzzY  Yxy10yzzzzY  Yxyy10zzzzY  Yxyyz10zzzY  Yxyyzz10zzY  Yxyyzzz10zY  Yxyyzzzz10Y  Yxyyzzzz+ |

Четвертый пример: “ λ ” (таблица 2.17)

Таблица 2.17 – Пример листинга одноленточной №4

|  |  |
| --- | --- |
| Y0Y | Y- |

# 2.5 Расчет временной сложности

Программа имеет возможность строить графики временной сложности для одноленточной и многоленточной МТ в реальном времени. Пользователь вводит максимальную длину слова и программа перебирает все возможные комбинации текущей и меньшей длины, использует для них алгоритм и находит максимальное количество шагов, которое соответствует определенной длине слова.

Для того, чтобы избежать слишком долгой работы алгоритма рассчета временной сложности было введено ограничение на ввод максимальной длины слова до 17. Также для того, чтобы было возможно использование системы во время построения графика все рассчеты связанные с графиком временной сложности происходят в отдельных потоках.

Графики изображены на рисунках 2.2 и 2.3

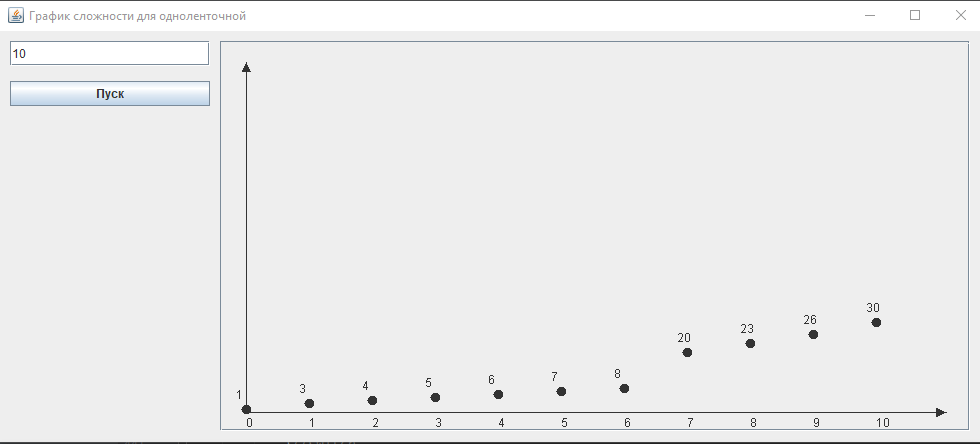


Рисунок 2.2 – График временной сложности для одноленточной МТ

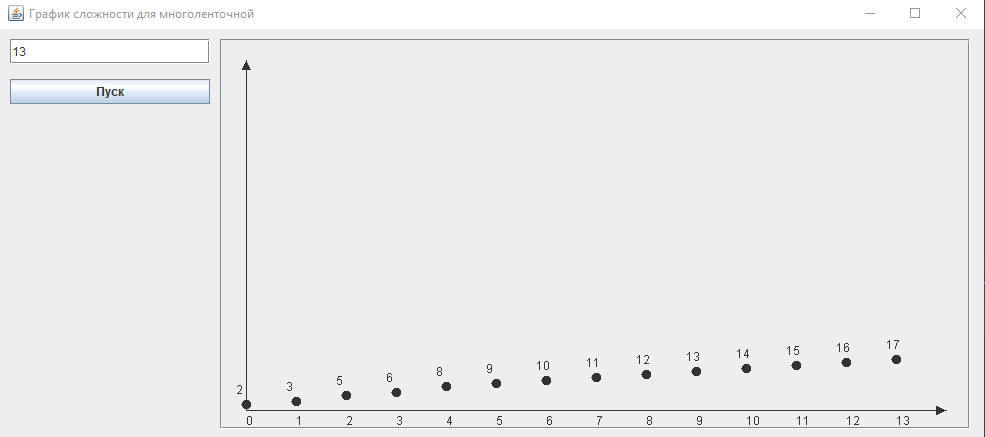


Рисунок 2.3 – График временной сложности для многоленточной МТ

# РАЗРАБОТКА АНАЛИТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА С ИСПОЛЬЗОВАНИЕМ НОРМАЛЬНЫХ АЛГОРИТМОВ МАРКОВА

Норма́льный алгори́тм Ма́ркова — один из стандартных способов формального определения понятия алгоритма. Нормальный алгоритм описывает метод переписывания строк, похожий по способу задания на формальные грамматики. НАМ — Тьюринг-полный язык, что делает его по выразительной силе эквивалентным машине Тьюринга и, следовательно, современным языкам программирования. [2]

Рассмотрим алгоритм, использующий нормальные алгоритмы Маркова, над алфавитом A={\*}, исключающий в слове последнюю звездочку(рис 3.1).

|  |
| --- |
| y\*→\*y  \*y →.  y→.  →y |

Рисунок 3.1 – Замены для данного алгоритма

Запишем контрольные примеры для данного алгоритма(таблица 3.1)

Таблица 3.1 – Контрольные примеры

|  |  |  |
| --- | --- | --- |
| Пример:\* | Пример:\*\*\* | Пример:\*\*\*\*\* |
| \*  y\*  \*y  . | \*\*\*  y\*\*\*  \*y\*\*  \*\*y\*  \*\*\*y  \*\*. | \*\*\*\*\*  y\*\*\*\*\*  \*y\*\*\*\*  \*\*y\*\*\*  \*\*\*y\*\*  \*\*\*\*y\*  \*\*\*\*\*y  \*\*\*\*. |

# ОПИСАНИЕ ФОРМАЛЬНОЙ МОДЕЛИ АЛГОРИТМА НА ОСНОВЕ РЕКУРСИВНЫХ ФУНКЦИЙ

Рекурси́вная фу́нкция — это числовая функция f(n){\displaystyle f(n)} числового аргумента, которая в своей записи содержит себя же. Такая запись позволяет вычислять значения f(n){\displaystyle f(n)} {\displaystyle f(n)} на основе значений f(n-1) ,f(n-2), …{\displaystyle f(n)}{\displaystyle f(n)} {\displaystyle f(n-1),f(n-2),\ldots }, подобно рассуждению по индукции. Чтобы вычисление завершалось для любого n{\displaystyle n}, необходимо, чтобы для некоторых n{\displaystyle n} функция была определена нерекурсивно(например, для n=0,1{\displaystyle n=0,1}) [3].

Рассмотрим функцию, которая ищет количество простых чисел меньше или равных n:

(4.1)

n – входное число

fmod(a,b) – нахождение остатка от деления

k – переменная для перебора чисел от 2 до n

i – переменная для перебора чисел от 2 до k

Рассмотрим контрольный пример с входным числом 5

Как видим, пример был решен правильно, функция нашла количество простых чисел <=5 (2,3,5).

# 

# ВЫВОДЫ

Результат работы – программа, реализующая распознающую много- и одноленточную МТ, которая выполняет алгоритм над словами языка.

К Преимуществам программы относятся: реализованы как одноленточная так и многоленточная машины Тьюринга. Результаты работы программы и пошаговое выполнение дублируются как на экран, так и в файл листинга программы. При записи пошагового выполнения пишется номер текущего состояния и наглядно показывается расположение каретки на ленте. График временной сложности при вычислении не останавливает работу системы, т.е. пользователь может параллельно работать с другими элементами системы. Все сообщения об ошибках понятны и выводятся в виде всплывающих окон, что обеспечивает осведомленность пользователя об этих ошибках. Предусмотрены все исключительные ситуации и проверки на корректность вводимых данных, благодаря чему программа работает без сбоев.

Недостатки: при вычислении временной сложности для слов с длиной более 15 требуется слишком много времени для рассчетов, из-за этого пользователь может подумать, что программа зависла или не работает.

# ПЕРЕЧЕНЬ ССЫЛОК

1. Машина Тьюринга [Электронный ресурс] // Определение машины тьюринга. – Режим доступа: [https://ru.wikipedia.org/wiki/Машина\_Тьюринга](https://ru.wikipedia.org/wiki/Машина_Тьюринга%20), свободный. – Загл. с экрана.

2. Нормальные алгоритмы [Электронный ресурс] // Нормальные алгоритмы Маркова. – Режим доступа: https://ru.wikipedia.org/wiki/Нормальный\_алгоритм, свободный. – Загл. с экрана.

3. Рекурсивная функция [Электронный ресурс] // Рекурсивная функция. – Режим доступа: https://ru.wikipedia.org/wiki/Рекурсивная\_функция, свободный. – Загл. с экрана.

# ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра программной инженерии

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ: «Построение аналитических моделей алгоритмов и

оценка их сложности »

ПО КУРСУ: «Теория алгоритмов и формальных языков»

Выдано:

студенту группы ПИ-18б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Донецк – 2019

1 Основанием для разработки является задание на курсовую работу, выданное кафедрой программной инженерии.

2 Целью разработки является создание программной модели машины Тьюринга, распознающий язык \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, расчет и экспериментальная проверка расчета временной сложности МТ.

3 Требования к программе:

– при проверке слова на принадлежность языку необходимо запретить ввод с клавиатуры символов не из входного алфавита заданного языка;

– при проверке слова на принадлежность языку выводить на экран каждый шаг работы машины Тьюринга;

– сохранять протокол работы машины Тьюринга в текстовом файле;

– при построении графика временной сложности работы машины Тьюринга значения для графика получить практически, с помощью созданной программной модели машины Тьюринга; для генерации слов длиной n использовать метод полного перебора;

– построение графика временной сложности работы МТ выполняется в отдельном потоке, значения на график заносятся по мере расчёта; построение графика останавливается по требованию пользователя.

4 Требования к программной документации:

пояснительная записка;

руководство пользователя.

5 Этапы разработки

|  |  |  |
| --- | --- | --- |
| № этапа | Наименование этапа | Срок выполнения |
| 1 | Выдача задания, составление ТЗ и его утверждение | 2 недели |
| 2 | Построение формальных моделей алгоритмов. | 2-4 недели |
| 3 | Определение структур данных для представления МТ, построение алгоритмов. | 4-5 недели |
| 4 | Написание программы | 5-9 недели |
| 5 | Отладка программы | 9-12 недели |
| 6 | Написание пояснительной записки | 9-13 недели |
| 7 | Защита курсовой работы | 13-14 недели |

# ПРИЛОЖЕНИЕ Б. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске программы открывается интерфейс, которой состоит из четырех окон. Каждое из окон имеет по одному полю для ввода текста и по одной кнопке, запускающей алгоритм, что обеспечивает простоту в использовании. Два окна отвечают за работу одноленточной и многоленточной МТ, для их работы необходимо ввести слово, состоящее из символов «a», «b», «c» два остальных отвечают за простоение графиков временной сложности одно- и многоленточной МТ. Каждое из окон называется соответственно его функции. При работе с МТ на экран выводится листинг программы. При работе с графиками нужно только длину слова, до которого нужно работать и график будет строиться по мере высчитывания программой вершин.

Для повторного использования необходимо заполнить поле заново и нажать кнопку в соответствующем окне.

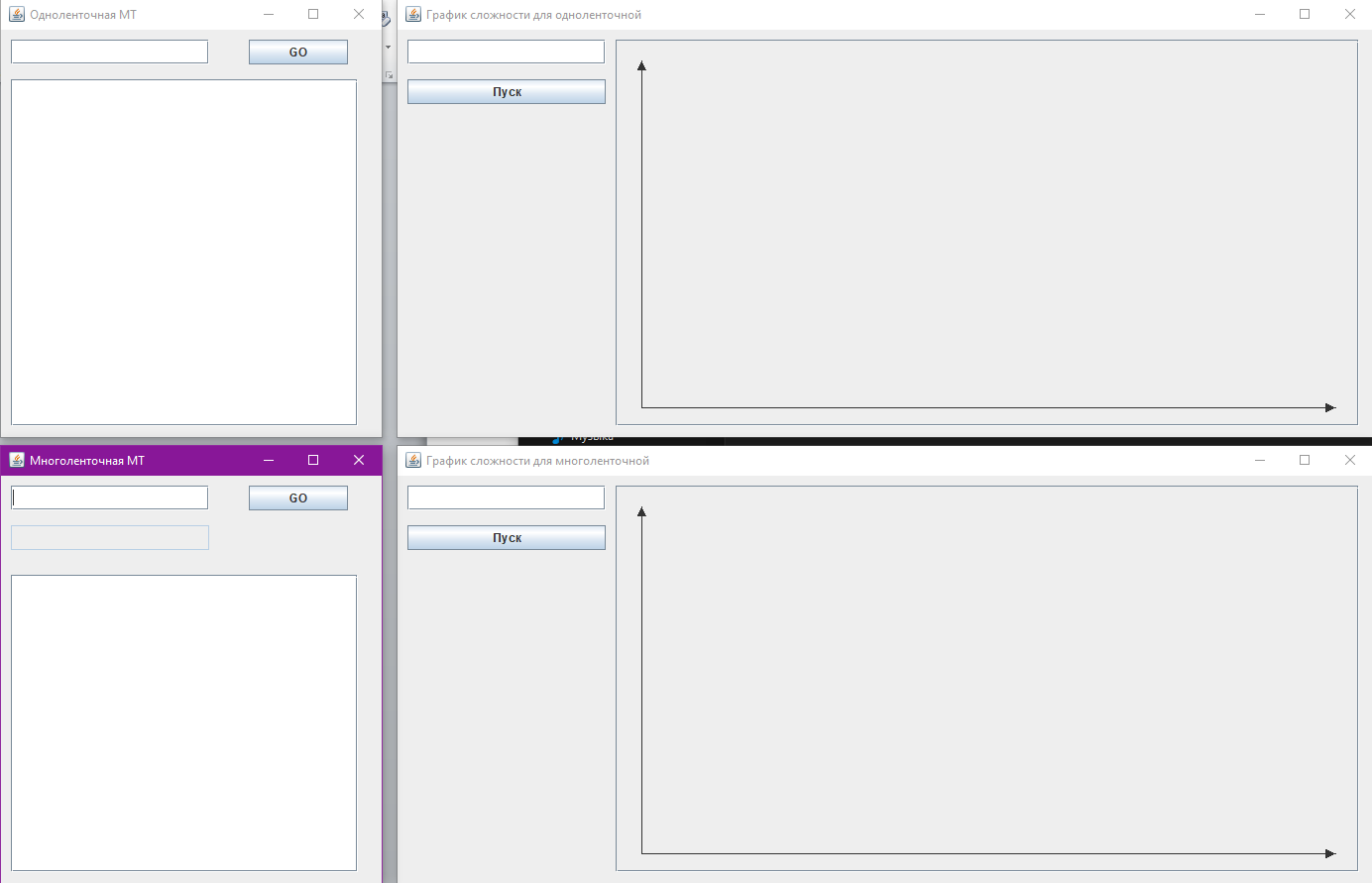


Рисунок Б.1 – Интерфейс программы

# ПРИЛОЖЕНИЕ В. КОД ПРОГРАММЫ

Kursovaya.java

import java.awt.\*;

import java.io.IOException;

public class Kursovaya {

public static void main(String[] args) throws Exception {

EventQueue.invokeLater(() -> {

try {

new Thread(new MT()).start();

} catch (IOException e) {

e.printStackTrace();

}

new GrafikOdnolentochnayaFrame();

try {

new Thread(new MnogolentochnayaMT()).start();

} catch (Exception e) {

e.printStackTrace();

}

new GrafikMnogolentochnayaFrame();

});

}//main

public static char[] insert(char[] a){

for(int i=a.length-1;i>=0;i--){

a[i]=plus1(a[i]);

if(a[i]!='a') {

return a;

}

}

return a;

}

public static char plus1(char a){

switch (a){

case 'a':

a='b';

return a;

case 'b':

a='c';

return a;

case 'c':

a='a';

return a;

}

return a;

}

}//end

MT.java

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

public class MT implements Runnable{

public Character[] alphabet = new Character[]{'a', 'b', 'c'};

final public static Character[] RABOCHIE\_PEREMENNYE = {'x', 'y', 'z'};

final public Character LYAMBDA = 'Y';

public Integer[] sostoyaniya = new Integer[]{0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10};

public static ChtoDelatOdnolenochnaya[][] matritsaMT;

public static Character[] symbols;

public static int karetka = 1, indeksSimvola,indeksSimvola1, curSostoyanie = 0,karetka1=1,curSostoyanie1=0;

public static String vyvodString;

public MT() throws IOException {

symbols = new Character[RABOCHIE\_PEREMENNYE.length + alphabet.length + 2];

System.arraycopy(alphabet, 0, symbols, 1, alphabet.length);

System.arraycopy(RABOCHIE\_PEREMENNYE, 0, symbols, alphabet.length + 1, RABOCHIE\_PEREMENNYE.length);

symbols[symbols.length - 1] = LYAMBDA;

matritsaMT = new ChtoDelatOdnolenochnaya[sostoyaniya.length][symbols.length];

zapolnenieMatritsy();

}//MT constuctor

public void run() {

for (int i = 0; i <11 ; i++) {

if(i!=9) {

for (int j = 1; j < 8; j++) {

print(MT.matritsaMT[i][j]);

System.out.print(" ");

}

System.out.println();

}

}

System.out.println("MT: "+Thread.currentThread().getName());

JFrame frame = new JFrame();

frame.setTitle("Одноленточная МТ");

JTextField inTextField = new JTextField(10);

frame.setSize(400, 450);

frame.setLocation(50,50);

frame.setLayout(null);

inTextField.setBounds(10, 10, 200, 25);

inTextField.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

super.keyTyped(e);

char c = e.getKeyChar();

if ((c != 'a' & c != 'b' & c != 'c') && (c != KeyEvent.VK\_BACK\_SPACE))

e.consume();

}

});

JButton goButton = new JButton();

goButton.setBounds(250, 10, 100, 25);

goButton.setText("GO");

JTextArea textArea = new JTextArea();

JScrollPane scroll = new JScrollPane(textArea);

scroll.setBounds(10, 50, 350, 350);

textArea.setEditable(false);

frame.add(scroll);

frame.add(goButton);

frame.add(inTextField);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setVisible(true);

goButton.addActionListener(new ActionListener() {

@Override

synchronized public void actionPerformed(ActionEvent e) {

File ofile = new File("Odnolentochnaya.txt");

FileWriter ofilew = null;

try {

ofilew = new FileWriter(ofile);

} catch (IOException ex) {

ex.printStackTrace();

}

BufferedWriter ofbuf = new BufferedWriter(ofilew);

if (!inTextField.getText().equals("")) {

if (inTextField.getText().charAt(0) != 'Y') {

curSostoyanie = 0;

karetka = 1;

}else {

JOptionPane.showMessageDialog(null, "Нельзя использовать еще раз");

return;

}

}else{

curSostoyanie = 0;

karetka = 1;

}

if (!inTextField.getText().equals(""))

if (inTextField.getText().charAt(inTextField.getText().length() - 1) == '+' || inTextField.getText().charAt(inTextField.getText().length() - 1) == '-') {

JOptionPane.showMessageDialog(null, "Нельзя использовать уже обработанное слово");

return;

}

//Timer timer = null;

vyvodString = "";

try {

//timer = new Timer(0, new ActionListener() {

String workString = "Y" + inTextField.getText() + "Y";

char[] workStringChars = workString.toCharArray();

String curString;

int o = 1;

//@Override

//public void actionPerformed(ActionEvent e) {

while (true) {

inTextField.setText(workStringChars.toString());//

curString = "";

System.out.println("Шаг " + o + "=========================");

vyvodString += "Шаг " + o + "=========================\n";

o++;

for (int i = 0; i < workStringChars.length; i++) {

if (i == karetka) {

curString += curSostoyanie;

System.out.print(curSostoyanie);

vyvodString += "" + curSostoyanie;

try {

ofbuf.write("" + curSostoyanie);

} catch (IOException ex) {

ex.printStackTrace();

}

}

curString += workStringChars[i];

vyvodString += workStringChars[i];

System.out.print(workStringChars[i]);

try {

ofbuf.write(workStringChars[i]);

} catch (IOException ex) {

ex.printStackTrace();

}

}

inTextField.setText(curString);//

textArea.setText(vyvodString);

try {

ofbuf.newLine();

} catch (IOException ex) {

ex.printStackTrace();

}

vyvodString += '\n';

System.out.println();

switch (workStringChars[karetka]) {

case 'a':

indeksSimvola = 1;

break;

case 'b':

indeksSimvola = 2;

break;

case 'c':

indeksSimvola = 3;

break;

case 'x':

indeksSimvola = 4;

break;

case 'y':

indeksSimvola = 5;

break;

case 'z':

indeksSimvola = 6;

break;

case 'Y':

indeksSimvola = 7;

break;

}//switch

int curcursost = curSostoyanie;

curSostoyanie = matritsaMT[curcursost][indeksSimvola].toSostoyanie;

workStringChars[karetka] = matritsaMT[curcursost][indeksSimvola].toCharacter;

switch (matritsaMT[curcursost][indeksSimvola].toNapravlenie) {

case 'r':

karetka++;

break;

case 'l':

karetka--;

break;

case 'e':

break;

}//switch

if (curSostoyanie == 9) {

System.out.println("Результат: ");

System.out.println(workStringChars);

String outString = new String(workStringChars);

vyvodString += "Результат: \n" + outString;

try {

ofbuf.write("" + outString);

} catch (IOException ex) {

ex.printStackTrace();

}

//Timer timer = (Timer) e.getSource();

//timer.stop();

inTextField.setText(outString); textArea.setText(vyvodString);

try {

ofbuf.flush();

} catch (IOException ex) {

ex.printStackTrace();

}

return;

}

}//while

// }

//});

} catch (Exception ex) {

ex.printStackTrace();

}

if (inTextField.getText().equals("")){

System.out.println("Пустое множество");

}

// timer.start();

else if (inTextField.getText().charAt(0) != 'Y') {

// timer.start();

}

}

});

}//run

public static void zapolnenieMatritsy() {

//q0

matritsaMT[0][1] = new ChtoDelatOdnolenochnaya(1, 'x', 'r');

matritsaMT[0][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[0][3] = new ChtoDelatOdnolenochnaya(8, 'c', 'r');

matritsaMT[0][4] = new ChtoDelatOdnolenochnaya(10, 'x', 'r');

matritsaMT[0][5] = new ChtoDelatOdnolenochnaya(10, 'y', 'r');

matritsaMT[0][6] = new ChtoDelatOdnolenochnaya(10, 'z', 'r');

matritsaMT[0][7] = new ChtoDelatOdnolenochnaya(9, '-', 'e');

//q1//

matritsaMT[1][1] = new ChtoDelatOdnolenochnaya(1, 'a', 'r');

matritsaMT[1][2] = new ChtoDelatOdnolenochnaya(2, 'y', 'r');

matritsaMT[1][3] = new ChtoDelatOdnolenochnaya(8, 'c', 'r');

matritsaMT[1][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[1][5] = new ChtoDelatOdnolenochnaya(1, 'y', 'r');

matritsaMT[1][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[1][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q2//

matritsaMT[2][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[2][2] = new ChtoDelatOdnolenochnaya(3, 'y', 'r');

matritsaMT[2][3] = new ChtoDelatOdnolenochnaya(8, 'c', 'r');

matritsaMT[2][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[2][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[2][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[2][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q3//

matritsaMT[3][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[3][2] = new ChtoDelatOdnolenochnaya(3, 'b', 'r');

matritsaMT[3][3] = new ChtoDelatOdnolenochnaya(4, 'z', 'r');

matritsaMT[3][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[3][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[3][6] = new ChtoDelatOdnolenochnaya(3, 'z', 'r');

matritsaMT[3][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q4//

matritsaMT[4][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[4][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[4][3] = new ChtoDelatOdnolenochnaya(5, 'z', 'r');

matritsaMT[4][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[4][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[4][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[4][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q5//

matritsaMT[5][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[5][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[5][3] = new ChtoDelatOdnolenochnaya(6, 'z', 'r');

matritsaMT[5][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[5][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[5][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[5][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q6//

matritsaMT[6][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[6][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[6][3] = new ChtoDelatOdnolenochnaya(7, 'z', 'l');

matritsaMT[6][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[6][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[6][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[6][7] = new ChtoDelatOdnolenochnaya(8, 'Y', 'e');

//q7//

matritsaMT[7][1] = new ChtoDelatOdnolenochnaya(7, 'a', 'l');

matritsaMT[7][2] = new ChtoDelatOdnolenochnaya(7, 'b', 'l');

matritsaMT[7][3] = new ChtoDelatOdnolenochnaya(7, 'c', 'l');

matritsaMT[7][4] = new ChtoDelatOdnolenochnaya(10, 'x', 'r');

matritsaMT[7][5] = new ChtoDelatOdnolenochnaya(7, 'y', 'l');

matritsaMT[7][6] = new ChtoDelatOdnolenochnaya(7, 'z', 'l');

matritsaMT[7][7] = new ChtoDelatOdnolenochnaya(10, 'Y', 'r');

//q8//

matritsaMT[8][1] = new ChtoDelatOdnolenochnaya(8, 'a', 'r');

matritsaMT[8][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[8][3] = new ChtoDelatOdnolenochnaya(8, 'c', 'r');

matritsaMT[8][4] = new ChtoDelatOdnolenochnaya(8, 'x', 'r');

matritsaMT[8][5] = new ChtoDelatOdnolenochnaya(8, 'y', 'r');

matritsaMT[8][6] = new ChtoDelatOdnolenochnaya(8, 'z', 'r');

matritsaMT[8][7] = new ChtoDelatOdnolenochnaya(9, '-', 'e');

//q10//

matritsaMT[10][1] = new ChtoDelatOdnolenochnaya(1, 'x', 'r');

matritsaMT[10][2] = new ChtoDelatOdnolenochnaya(8, 'b', 'r');

matritsaMT[10][3] = new ChtoDelatOdnolenochnaya(8, 'c', 'r');

matritsaMT[10][4] = new ChtoDelatOdnolenochnaya(10, 'x', 'r');

matritsaMT[10][5] = new ChtoDelatOdnolenochnaya(10, 'y', 'r');

matritsaMT[10][6] = new ChtoDelatOdnolenochnaya(10, 'z', 'r');

matritsaMT[10][7] = new ChtoDelatOdnolenochnaya(9, '+', 'e');

}

synchronized public static int countOfMoves(int n) {

int count = 0;

karetka1=1;

char[] charMas = new char[n];

for (int o = 0; o < charMas.length; o++) {

charMas[o] = 'a';

}

for (int e=0;e<(Math.pow(3,n));e++) {

//System.out.println(String.valueOf(charMas));

String workString="Y"+String.valueOf(charMas)+"Y";

char[] workStringChars = workString.toCharArray();

int p = 0;

while (true) {

p++;

switch (workStringChars[karetka1]) {

case 'a':

indeksSimvola1 = 1;

break;

case 'b':

indeksSimvola1 = 2;

break;

case 'c':

indeksSimvola1 = 3;

break;

case 'x':

indeksSimvola1 = 4;

break;

case 'y':

indeksSimvola1 = 5;

break;

case 'z':

indeksSimvola1 = 6;

break;

case 'Y':

indeksSimvola1 = 7;

break;

}//switch

int curcursost = curSostoyanie1;

curSostoyanie1 = matritsaMT[curcursost][indeksSimvola1].toSostoyanie;

workStringChars[karetka1] = matritsaMT[curcursost][indeksSimvola1].toCharacter;

switch (matritsaMT[curcursost][indeksSimvola1].toNapravlenie) {

case 'r':

karetka1++;

break;

case 'l':

karetka1--;

break;

case 'e':

break;

}//switch

if (curSostoyanie1 == 9) {

// System.out.println(p);

if (count<p) count=p;

break;

}

}//while

Kursovaya.insert(charMas);

curSostoyanie1=0;

karetka1=1;

}//for

return count;

}//countOfMoves;

public static void print(ChtoDelatOdnolenochnaya a) {

System.out.print(a.toSostoyanie + "" + a.toCharacter + "" + a.toNapravlenie);

}

}//MT

MnogolentochnayaMT.java

import jdk.nashorn.internal.scripts.JO;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

public class MnogolentochnayaMT implements Runnable {

public Character[] alphabet = new Character[]{'a', 'b', 'c'};

final public Character LYAMBDA = 'Y';

public Integer[] sostoyaniya = new Integer[]{0, 1, 2, 3, 4, 5, 6, 7, 8,9};

public static ChtoDelatMnogolenochnaya[][] matritsaMTa,matritsaMTY;

public static Character[] symbols;

public static int karetka1=1,karetka2=1,indeksSimvola1,indeksSimvola2,curSostoyanie=0;

public static int karetka12=1,karetka22=1,indeksSimvola12,indeksSimvola22,curSostoyanie1=0;

public static String vyvodString;

public MnogolentochnayaMT() throws Exception {

symbols = new Character[alphabet.length + 2];

System.arraycopy(alphabet, 0, symbols, 1, alphabet.length);

symbols[symbols.length - 1] = LYAMBDA;

matritsaMTa = new ChtoDelatMnogolenochnaya[sostoyaniya.length][symbols.length];

matritsaMTY = new ChtoDelatMnogolenochnaya[sostoyaniya.length][symbols.length];

zapolnenieMatritsy();

}//MT constuctor

public void run() {

System.out.println("MMT: "+Thread.currentThread().getName());

JFrame frame = new JFrame();

frame.setTitle("Многоленточная МТ");

JTextField inTextField1 = new JTextField(10);

frame.setSize(400, 450);

frame.setLocation(50,500);

frame.setLayout(null);

JTextField inTextField2 = new JTextField(10);

inTextField1.setBounds(10, 10, 200, 25);

inTextField1.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

super.keyTyped(e);

char c = e.getKeyChar();

if ((c != 'a' & c != 'b' & c != 'c') && (c != KeyEvent.VK\_BACK\_SPACE))

e.consume();

}

});

inTextField2.setBounds(10, 50, 200, 25);

inTextField2.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

super.keyTyped(e);

char c = e.getKeyChar();

if ((c != 'a' & c != 'b' & c != 'c') && (c != KeyEvent.VK\_BACK\_SPACE))

e.consume();

}

});

frame.add(inTextField2);

JButton goButton = new JButton();

goButton.setBounds(250, 10, 100, 25);

goButton.setText("GO");

JTextArea textArea = new JTextArea();

JScrollPane scroll = new JScrollPane(textArea);

scroll.setBounds(10, 100, 350, 300);

textArea.setEditable(false);

frame.add(scroll);

inTextField2.setEditable(false);

frame.add(goButton);

frame.add(inTextField1);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setVisible(true);

goButton.addActionListener(new ActionListener() {

@Override

synchronized public void actionPerformed(ActionEvent e) {

File ofile = new File("Mnogolentochnaya.txt");

FileWriter ofilew = null;

try {

ofilew = new FileWriter(ofile);

} catch (IOException ex) {

ex.printStackTrace();

}

BufferedWriter ofbuf = new BufferedWriter(ofilew);

if (!inTextField1.getText().equals(""))

if (inTextField1.getText().charAt(0) != 'Y') {

curSostoyanie = 0;

karetka1 = 1;

karetka2=1;

}

if (!inTextField1.getText().equals(""))

if (inTextField1.getText().charAt(inTextField1.getText().length() - 1) == '+' || inTextField1.getText().charAt(inTextField1.getText().length() - 1) == '-') {

JOptionPane.showMessageDialog(null, "Нельзя использовать уже обработанное слово");

return;

}

vyvodString = "";

inTextField1.setEditable(true);

String workStringFirst;

if(inTextField1.getText().equals("")){

workStringFirst="YY";

}else

workStringFirst = "Y" + inTextField1.getText() + "Y";

char[] workStringCharsFirst = workStringFirst.toCharArray();

String curString1,curString2;

ArrayList<Character> workStringCharsSecond = new ArrayList<>();

workStringCharsSecond.add('Y');

workStringCharsSecond.add('Y');

int o = 1;

curSostoyanie = 0;

karetka1 = 1;

karetka2=1;

while (true) {

inTextField1.setText(workStringCharsFirst.toString());

curString1="";

curString2="";

System.out.println("Шаг " + o + "===============");

vyvodString+="Шаг " + o + "===============\n"+"Первая лента: ";

o++;

System.out.print("Первая лента: ");

for (int i = 0; i < workStringCharsFirst.length; i++) {

if (i == karetka1) {

curString1+=curSostoyanie;

vyvodString+=""+curSostoyanie;

System.out.print(curSostoyanie);

try {

ofbuf.write("" + curSostoyanie);

} catch (IOException ex) {

ex.printStackTrace();

}

}

curString1 += workStringCharsFirst[i];

vyvodString += workStringCharsFirst[i];

System.out.print(workStringCharsFirst[i]);

try {

ofbuf.write(workStringCharsFirst[i]);

} catch (IOException ex) {

ex.printStackTrace();

}

}

System.out.println();

try {

ofbuf.newLine();

} catch (IOException ex) {

ex.printStackTrace();

}

System.out.print("Вторая лента: ");

vyvodString+="\nВторая лента: ";

for (int i = 0; i < workStringCharsSecond.size(); i++) {

if (i == karetka2) {

curString2+=curSostoyanie;

vyvodString+=""+curSostoyanie;

System.out.print(curSostoyanie);

try {

ofbuf.write("" + curSostoyanie);

} catch (IOException ex) {

ex.printStackTrace();

}

}

curString1 += workStringCharsSecond.get(i);

vyvodString += workStringCharsSecond.get(i);

System.out.print(workStringCharsSecond.get(i));

try {

ofbuf.write(workStringCharsSecond.get(i));

} catch (IOException ex) {

ex.printStackTrace();

}

}

System.out.println();

inTextField1.setText(curString1);//

inTextField2.setText(curString2);//

textArea.setText(vyvodString);

try {

ofbuf.newLine();

} catch (IOException ex) {

ex.printStackTrace();

}

vyvodString += '\n';

System.out.println();

switch (workStringCharsFirst[karetka1]) {

case 'a':

indeksSimvola1 = 1;

break;

case 'b':

indeksSimvola1 = 2;

break;

case 'c':

indeksSimvola1 = 3;

break;

case 'Y':

indeksSimvola1 = 4;

break;

}//switch

switch (workStringCharsSecond.get(karetka2)) {

case 'a':

indeksSimvola2 = 1;

break;

case 'Y':

indeksSimvola2 = 4;

break;

}//switch

if (indeksSimvola2 == 1) {

int curcursost = curSostoyanie;

curSostoyanie = matritsaMTa[curcursost][indeksSimvola1].toSostoyanie;

workStringCharsFirst[karetka1] = matritsaMTa[curcursost][indeksSimvola1].toCharacter1;

workStringCharsSecond.set(karetka2, matritsaMTa[curcursost][indeksSimvola1].toCharacter2);

switch (matritsaMTa[curcursost][indeksSimvola1].toNapravlenie1) {

case 'r':

karetka1++;

break;

case 'l':

karetka1--;

break;

case 'e':

break;

}//switch

switch (matritsaMTa[curcursost][indeksSimvola1].toNapravlenie2) {

case 'r':

karetka2++;

break;

case 'l':

karetka2--;

break;

case 'e':

break;

}//switch

} else {

int curcursost = curSostoyanie;

curSostoyanie = matritsaMTY[curcursost][indeksSimvola1].toSostoyanie;

workStringCharsFirst[karetka1] = matritsaMTY[curcursost][indeksSimvola1].toCharacter1;

workStringCharsSecond.set(karetka2, matritsaMTY[curcursost][indeksSimvola1].toCharacter2);

switch (matritsaMTY[curcursost][indeksSimvola1].toNapravlenie1) {

case 'r':

karetka1++;

break;

case 'l':

karetka1--;

break;

case 'e':

break;

}

switch (matritsaMTY[curcursost][indeksSimvola1].toNapravlenie2) {

case 'r':

karetka2++;

break;

case 'l':

karetka2--;

break;

case 'e':

break;

}//switch

}

if (workStringCharsSecond.get(workStringCharsSecond.size() - 1) != 'Y')

workStringCharsSecond.add('Y');

if (curSostoyanie == 9) {

System.out.println("Результат: ");

System.out.print("Первая лента: ");

System.out.println(workStringCharsFirst);

String outString = new String(workStringCharsFirst);

inTextField1.setText(outString);

vyvodString+="Результат: \nПервая лента: " + outString+"\n";

outString=new String(workStringCharsSecond.toString());

char[] out = new char[workStringCharsSecond.size()];

for (int i = 0; i < workStringCharsSecond.size(); i++) {

out[i] = workStringCharsSecond.get(i);

}

vyvodString+="Вторая лента: "+String.valueOf(out);

textArea.setText(vyvodString);

inTextField2.setText(""+String.valueOf(out));

System.out.print("Вторая лента: ");

System.out.println(out);

try {

ofbuf.write(workStringCharsFirst);

ofbuf.newLine();

ofbuf.write(out);

ofbuf.flush();

} catch (IOException ex) {

ex.printStackTrace();

}

return;

}

}

}

});

}//run

public static void zapolnenieMatritsy() {

//q0

matritsaMTY[0][1]=new ChtoDelatMnogolenochnaya(0,'a','a','r','r');

matritsaMTY[0][2]=new ChtoDelatMnogolenochnaya(1,'b','Y','e','l');

matritsaMTY[0][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[0][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

//q1

matritsaMTY[1][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[1][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[1][3]=new ChtoDelatMnogolenochnaya(4,'c','Y','e','r');

matritsaMTY[1][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[1][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[1][2]=new ChtoDelatMnogolenochnaya(2,'b','a','r','e');

matritsaMTa[1][3]=new ChtoDelatMnogolenochnaya(8,'c','a','r','e');

matritsaMTa[1][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q2

matritsaMTY[2][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[2][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[2][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[2][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[2][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[2][2]=new ChtoDelatMnogolenochnaya(1,'b','a','r','l');

matritsaMTa[2][3]=new ChtoDelatMnogolenochnaya(8,'c','a','r','e');

matritsaMTa[2][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q3

matritsaMTY[3][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[3][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[3][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[3][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[3][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[3][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[3][3]=new ChtoDelatMnogolenochnaya(8,'c','a','r','e');

matritsaMTa[3][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q4

matritsaMTY[4][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[4][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[4][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[4][4]=new ChtoDelatMnogolenochnaya(9,'+','Y','e','l');

matritsaMTa[4][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[4][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[4][3]=new ChtoDelatMnogolenochnaya(5,'c','a','r','e');

matritsaMTa[4][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q5

matritsaMTY[5][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[5][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[5][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[5][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[5][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[5][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[5][3]=new ChtoDelatMnogolenochnaya(6,'c','a','r','e');

matritsaMTa[5][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q6

matritsaMTY[6][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[6][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[6][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[6][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[6][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[6][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[6][3]=new ChtoDelatMnogolenochnaya(7,'c','a','r','e');

matritsaMTa[6][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q7

matritsaMTY[7][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[7][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[7][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[7][4]=new ChtoDelatMnogolenochnaya(8,'Y','Y','e','e');

matritsaMTa[7][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[7][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[7][3]=new ChtoDelatMnogolenochnaya(4,'c','a','r','r');

matritsaMTa[7][4]=new ChtoDelatMnogolenochnaya(8,'Y','a','e','e');

//q8

matritsaMTY[8][1]=new ChtoDelatMnogolenochnaya(8,'a','Y','r','e');

matritsaMTY[8][2]=new ChtoDelatMnogolenochnaya(8,'b','Y','r','e');

matritsaMTY[8][3]=new ChtoDelatMnogolenochnaya(8,'c','Y','r','e');

matritsaMTY[8][4]=new ChtoDelatMnogolenochnaya(9,'-','Y','e','e');

matritsaMTa[8][1]=new ChtoDelatMnogolenochnaya(8,'a','a','r','e');

matritsaMTa[8][2]=new ChtoDelatMnogolenochnaya(8,'b','a','r','e');

matritsaMTa[8][3]=new ChtoDelatMnogolenochnaya(8,'c','a','r','e');

matritsaMTa[8][4]=new ChtoDelatMnogolenochnaya(9,'-','a','e','e');

}

synchronized public static int countOfMoves(int n) {

int count = 0;

karetka12 = 1;

karetka22 = 1;

curSostoyanie1 = 0;

char[] charMas = new char[n];

for (int o = 0; o < charMas.length; o++) {

charMas[o] = 'a';

}

for (int e=0;e<(Math.pow(3,n));e++) {

String workStringFirst;

if (String.valueOf(charMas).equals("")) {

workStringFirst = "YY";

} else

workStringFirst = "Y" + String.valueOf(charMas) + "Y";

char[] workStringCharsFirst = workStringFirst.toCharArray();

ArrayList<Character> workStringCharsSecond = new ArrayList<>();

workStringCharsSecond.add('Y');

workStringCharsSecond.add('Y');

int o = 1;

while (true) {

o++;

switch (workStringCharsFirst[karetka12]) {

case 'a':

indeksSimvola12 = 1;

break;

case 'b':

indeksSimvola12 = 2;

break;

case 'c':

indeksSimvola12 = 3;

break;

case 'Y':

indeksSimvola12 = 4;

break;

}//switch

switch (workStringCharsSecond.get(karetka22)) {

case 'a':

indeksSimvola22 = 1;

break;

case 'Y':

indeksSimvola22 = 4;

break;

}//switch

if (indeksSimvola22 == 1) {

int curcursost = curSostoyanie1;

curSostoyanie1 = matritsaMTa[curcursost][indeksSimvola12].toSostoyanie;

workStringCharsFirst[karetka12] = matritsaMTa[curcursost][indeksSimvola12].toCharacter1;

workStringCharsSecond.set(karetka22, matritsaMTa[curcursost][indeksSimvola12].toCharacter2);

switch (matritsaMTa[curcursost][indeksSimvola12].toNapravlenie1) {

case 'r':

karetka12++;

break;

case 'l':

karetka12--;

break;

case 'e':

break;

}//switch

switch (matritsaMTa[curcursost][indeksSimvola12].toNapravlenie2) {

case 'r':

karetka22++;

break;

case 'l':

karetka22--;

break;

case 'e':

break;

}//switch

} else {

int curcursost = curSostoyanie1;

curSostoyanie1 = matritsaMTY[curcursost][indeksSimvola12].toSostoyanie;

workStringCharsFirst[karetka12] = matritsaMTY[curcursost][indeksSimvola12].toCharacter1;

workStringCharsSecond.set(karetka22, matritsaMTY[curcursost][indeksSimvola12].toCharacter2);

switch (matritsaMTY[curcursost][indeksSimvola12].toNapravlenie1) {

case 'r':

karetka12++;

break;

case 'l':

karetka12--;

break;

case 'e':

break;

}

switch (matritsaMTY[curcursost][indeksSimvola12].toNapravlenie2) {

case 'r':

karetka22++;

break;

case 'l':

karetka22--;

break;

case 'e':

break;

}//switch

}

if (workStringCharsSecond.get(workStringCharsSecond.size() - 1) != 'Y')

workStringCharsSecond.add('Y');

if (curSostoyanie1 == 9) {

if (count < o) count = --o;

break;

}

}//while

Kursovaya.insert(charMas);

karetka12 = 1;

karetka22 = 1;

curSostoyanie1 = 0;

}

return count;

}//countOfMoves;

public static void print(ChtoDelatOdnolenochnaya a) {

System.out.print(a.toSostoyanie + "" + a.toCharacter + "" + a.toNapravlenie);

}

}

ChtoDelatOdnolenochnaya.java

public class ChtoDelatOdnolenochnaya {

public int toSostoyanie;

public char toCharacter;

public char toNapravlenie;

public ChtoDelatOdnolenochnaya(int sost, char charac, char naprav) {

toSostoyanie = sost;

toCharacter = charac;

toNapravlenie = naprav;

}

}

ChtoDelatMnogolenochnaya.java

public class ChtoDelatMnogolenochnaya {

public int toSostoyanie;

public char toCharacter1;

public char toCharacter2;

public char toNapravlenie1;

public char toNapravlenie2;

public ChtoDelatMnogolenochnaya(int sost, char charac1,char charac2, char naprav1, char naprav2) {

toSostoyanie = sost;

toCharacter1 = charac1;

toCharacter2 = charac2;

toNapravlenie1 = naprav1;

toNapravlenie2 = naprav2;

}

}

GrafikOdnolentochnayaFrame.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

public class GrafikOdnolentochnayaFrame extends JFrame {

public GrafikOdnolentochnayaFrame() {

System.out.println("Frame: " + Thread.currentThread().getName());

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setBounds(450, 50, 1000, 450);

setLayout(null);

setTitle("График сложности для одноленточной");

GrafikOdnolentochnayaPanel grafic = new GrafikOdnolentochnayaPanel();

JTextField toCount = new JTextField();

toCount.setBounds(10, 10, 200, 25);

toCount.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

super.keyTyped(e);

char c = e.getKeyChar();

if ((c<'0'||c>'9') && (c != KeyEvent.VK\_BACK\_SPACE))

e.consume();

}

});

JButton goButton = new JButton("Пуск");

//goButton.setBackground(Color.green);

goButton.setBounds(10, 50, 200, 25);

goButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (Integer.parseInt(toCount.getText()) > 17) {

JOptionPane.showMessageDialog(null, "Рассчет займет лишком много времени. Введите число до 17");

return;

}

grafic.vershiny.clear();

VershinaOdnolentochnaya.number=-1;

for (int i = 0; i <= Integer.parseInt(toCount.getText()); i++) {

Runnable r = () -> {

synchronized (this) {

if (VershinaOdnolentochnaya.number++ < Integer.parseInt(toCount.getText())) {

VershinaOdnolentochnaya v = new VershinaOdnolentochnaya(MT.countOfMoves(VershinaOdnolentochnaya.number)); grafic.add(v);

grafic.paint(grafic.getGraphics());

}

}

};

Thread t = new Thread(r);

t.start();

}

}

});

JScrollPane scroll = new JScrollPane(grafic);

scroll.setBounds(220, 10, 750, 390);

add(scroll);

add(toCount);

add(goButton);

}

}

GrafikOdnolentochnayaPanel.java

import javax.swing.\*;

import java.awt.\*;

import java.io.IOException;

import java.util.ArrayList;

public class GrafikOdnolentochnayaPanel extends JPanel{

public static int toCount=0;

public static int[] a={0,0};

public java.util.List<VershinaOdnolentochnaya> vershiny = new ArrayList<>();

public void add(VershinaOdnolentochnaya v){

vershiny.add(v);

}

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

System.out.println("GrafikOdnolentochnayaPanel: " + Thread.currentThread().getName());

drawArrowLine(g, 25, 370, 25, 20, 10, 5);//начало координат x=25,y=370

drawArrowLine(g, 25, 370, 725, 370, 10, 5);//width=700 height=350

//System.out.println(cur);

if (vershiny.size() != 0){

for (int i = 0; i < vershiny.size(); i++) {

int dx;

if (i == 0) {

dx = 0;

} else {

dx = 700 / vershiny.size();

}

int dy = 350 / 100;

System.out.println("Painting component "+i);

g.fillOval(25 + dx \* (i)-5, 370 - dy \* vershiny.get(i).getY()-5, 10, 10);

g.drawString(Integer.toString(vershiny.get(i).getY()), 25 + dx \* (i) - 10, 370 - dy \* vershiny.get(i).getY() - 10);

g.drawString(""+i,25+dx\*(i),385);

}

}else System.out.println("Size = 0");

}//paintComponent

public static void drawArrowLine(Graphics g, int x1, int y1, int x2, int y2, int d, int h) {

int dx = x2 - x1, dy = y2 - y1;

double D = Math.sqrt(dx\*dx + dy\*dy);

double xm = D - d, xn = xm, ym = h, yn = -h, x;

double sin = dy / D, cos = dx / D;

x = xm\*cos - ym\*sin + x1;

ym = xm\*sin + ym\*cos + y1;

xm = x;

x = xn\*cos - yn\*sin + x1;

yn = xn\*sin + yn\*cos + y1;

xn = x;

int[] xpoints = {x2, (int) xm, (int) xn};

int[] ypoints = {y2, (int) ym, (int) yn};

g.drawLine(x1, y1, x2, y2);

g.fillPolygon(xpoints, ypoints, 3);

}

}

GrafikMnogolentochnayaPanel.java

import javax.swing.\*;

import java.awt.\*;

import java.util.ArrayList;

public class GrafikMnogolentochnayaPanel extends JPanel {

public static int toCount=0;

public static int[] a={0,0};

public java.util.List<VershinaMnogolentochnaya> vershiny = new ArrayList<>();

public void add(VershinaMnogolentochnaya v){

vershiny.add(v);

}

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

System.out.println("GrafikMnogolentochnayaPanel: " + Thread.currentThread().getName());

drawArrowLine(g, 25, 370, 25, 20, 10, 5);//начало координат x=25,y=370

drawArrowLine(g, 25, 370, 725, 370, 10, 5);//width=700 height=350

//System.out.println(cur);

if (vershiny.size() != 0){

for (int i = 0; i < vershiny.size(); i++) {

int dx;

if (i == 0) {

dx = 0;

} else {

dx = 700 / vershiny.size();

}

int dy = 350 / 100;

System.out.println("Painting component "+i);

g.fillOval(25 + dx \* (i)-5, 370 - dy \* vershiny.get(i).getY()-5, 10, 10);

g.drawString(Integer.toString(vershiny.get(i).getY()), 25 + dx \* (i) - 10, 370 - dy \* vershiny.get(i).getY() - 10);

g.drawString(""+i,25+dx\*(i),385);

}

}else System.out.println("Size = 0");

}//paintComponent

public static void drawArrowLine(Graphics g, int x1, int y1, int x2, int y2, int d, int h) {

int dx = x2 - x1, dy = y2 - y1;

double D = Math.sqrt(dx\*dx + dy\*dy);

double xm = D - d, xn = xm, ym = h, yn = -h, x;

double sin = dy / D, cos = dx / D;

x = xm\*cos - ym\*sin + x1;

ym = xm\*sin + ym\*cos + y1;

xm = x;

x = xn\*cos - yn\*sin + x1;

yn = xn\*sin + yn\*cos + y1;

xn = x;

int[] xpoints = {x2, (int) xm, (int) xn};

int[] ypoints = {y2, (int) ym, (int) yn};

g.drawLine(x1, y1, x2, y2);

g.fillPolygon(xpoints, ypoints, 3);

}

}

GrafikMnogolentochnayaFrame.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.util.ArrayList;

import javax.swing.\*;

import java.awt.\*;

import java.io.IOException;

import java.util.ArrayList;

public class GrafikMnogolentochnayaFrame extends JFrame{

public GrafikMnogolentochnayaFrame() {

System.out.println("Frame: " + Thread.currentThread().getName());

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setBounds(450, 500, 1000, 450);

setLayout(null);

setTitle("График сложности для многоленточной");

GrafikMnogolentochnayaPanel grafic = new GrafikMnogolentochnayaPanel();

JTextField toCount = new JTextField();

toCount.setBounds(10, 10, 200, 25);

toCount.addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

super.keyTyped(e);

char c = e.getKeyChar();

if ((c<'0'||c>'9') && (c != KeyEvent.VK\_BACK\_SPACE))

e.consume();

}

});

JButton goButton = new JButton("Пуск");

//goButton.setBackground(Color.green);

goButton.setBounds(10, 50, 200, 25);

goButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (Integer.parseInt(toCount.getText())>50) { JOptionPane.showMessageDialog(null, "Рассчет займет лишком много времени. Введите число до 17");

return;

}

grafic.vershiny.clear();

VershinaMnogolentochnaya.number=-1;

for (int i = 0; i <= Integer.parseInt(toCount.getText()); i++) {

Runnable r = () -> {

synchronized (this) {

if (VershinaMnogolentochnaya.number++ < Integer.parseInt(toCount.getText())) { VershinaMnogolentochnaya v = new VershinaMnogolentochnaya(MnogolentochnayaMT.countOfMoves(VershinaMnogolentochnaya.number)); grafic.add(v); grafic.paint(grafic.getGraphics());

}

}

};

Thread t = new Thread(r);

t.start();

}

}

});

JScrollPane scroll = new JScrollPane(grafic);

scroll.setBounds(220, 10, 750, 390);

add(scroll);

add(toCount);

add(goButton);

}

}

VershinaMnogolentochnaya.java

public class VershinaMnogolentochnaya {

private int y;

public static int number = -1;

public VershinaMnogolentochnaya(int y) {

this.y = y;

}

public int getY(){return y;}

}

VershinaOdnolentochnaya.java

public class VershinaOdnolentochnaya {

private int y;

public static int number = -1;

public VershinaOdnolentochnaya(int y) {

this.y = y;

}

public int getY(){return y;}

}