

## ПРАКТИЧЕСКИЙ ТУР

### Эмулятор исполнителя «Кентавр Тьюринга»

ФИО: \_\_\_\_\_

## 1 Правила проведения

1. Задание выполняется на компьютере, выделенном участнику;
2. Разрешено использование материалов, доступных на компьютере;
3. Проект с решением участника размещается на рабочем столе, имя проекта должно иметь формат: «Фамилия\_Имя»;
4. При необходимости проект снабжается файлом «README.txt», где можно оставить дополнительные комментарии по работе;
5. Запрещено использование интернета, дополнительных электронных устройств;

## 2 Введение

Задача – создать приложение, эмулирующее работу машины Тьюринга. Пользователь должен сначала задать алфавит, а далее задать программу для Кентавра Тьюринга и задать строку, на которой исполнитель запускается. Исполнитель должен обрабатывать строку медленно, со скоростью человека. Каретка должна плавно перемещаться по ленте.

## 3 Принцип работы исполнителя «Кентавр Тьюринга»

Исполнитель «Кентавр Тьюринга» представляет собой бесконечную ленту, разделённую на ячейки одинакового размера. В каждой ячейке может быть записан ровно один символ или ничего (пустую ячейку будем обозначать  $\lambda$ ). Кентавр перемещается вдоль ленты, в каждый момент времени обозревая ровно одну ячейку.

Один такт исполнителя состоит в следующем: Кентавр обозревает текущую ячейку, а далее выполняет до трёх действий:

### 1. Записать в ячейку новый символ.

При выполнении этого действия старый символ из ячейки исчезает. Данное действие будем обозначать просто новым символом.

### 2. Сдвинуться на одну ячейку влево или вправо.

Данные действия будем обозначать заглавными буквами L (движение влево) и R (движение вправо).

### 3. Изменить состояние.

Под состоянием понимается набор реакций Кентавра при виде каждого символа. Состояния будем обозначать буквой q с индексом. Например,  $q_0$ ,  $q_1$ ,  $q_2$  и т.д.

Особым видом этого действия является переход в состояние остановки. Выполнив данное действие, Кентавр завершает свою работу. Данное состояние будем обозначать символом «!» (восклицательный знак).

Кентавр на каждом такте при необходимости может пропустить одно или два любых действия. Например, Кентавр может просто сдвинуться, не меняя символ в ячейке и не меняя состояние. Однако, Кентавр не может изменять порядок действий – сначала он должен записать символ, потом сдвинуться, а потом поменять состояние.

Программа для данного исполнителя представляет из себя таблицу, в которой каждая строка является состоянием, а каждый столбец отвечает за реакцию на один вид символов. В ячейках таблицы пишутся действия, которые нужно выполнить при виде определённого символа, находясь в определённом состоянии.

Изначально на ленте записано некоторое входное слово, не содержащее внутри себя пустых ячеек. Кентавр в начальный момент времени обозревает крайнюю левую ячейку входного слова и находится в состоянии  $q_0$ . Далее он выполняет предложенную программу и останавливается, когда выполнит команду «!».

Естественно, Кентавр умнее какой-то ленты и может записывать на неё вообще любые символы, даже те, которых на ленте изначально не было. Однако в этом случае он должен предусмотреть реакцию на новые символы, если вдруг решит пройти по ним, например, обратно.

## 4 Интерфейс приложения

Интерфейс приложения состоит из демонстрационной части – ленты, головки и управляющей части, с которой может взаимодействовать пользователь.

**Элементы управляющей части** (названия кнопок являются условными – можно использовать значки):

1. Окна для ввода алфавита ленты и алфавита доп. символов.
2. Кнопка «Задать алфавиты».
3. Таблица для записи программы (становится активна после нажатия кнопки «Задать алфавиты»)
4. Окно для ввода входного слова (становится активно после нажатия кнопки «Задать алфавиты»)
5. Кнопка «Задать строку».
6. Кнопка «Запустить исполнителя».
7. Кнопка «Остановить исполнителя».
8. Кнопка «Выполнить один шаг».
9. Кнопка «Сбросить выполнение программы».
10. Кнопки «Ускорить/Замедлить выполнение программы»

## 4.1 Работа с таблицей

Таблица для записи программы изначально недоступна пользователю и должна активироваться после первого нажатия кнопки «задать алфавиты». Если пользователь решит сменить что-либо в алфавитах, то все изменения не принимаются, пока он не нажмёт снова кнопку «Задать алфавиты». При повторном нажатии этой кнопки таблица должна автоматически перестроиться, при этом допустимо два варианта перестройки:

**Easy:** Всё содержимое программы-таблицы исчезает, и её нужно полностью заполнять заново.

**True:** Проверяется, как именно изменились алфавиты. Если изменение чисто расширяющее (только добавились новые символы), то содержимое таблицы никак не меняется, просто добавляются новые столбцы. Если же, хотя бы один из старых символов исчез, то таблицу нужно полностью очищать.

Также, в таблице должны быть предусмотрены кнопки «добавить/удалить состояние» - можно обозначить плюсом и минусом.

## 4.2 Ввод строки

Окно для ввода строки изначально недоступно пользователю и должно активироваться после первого нажатия кнопки «Задать алфавиты».

При нажатии на кнопку «Задать строку» должна выполняться первичная проверка корректности входной строки. Далее либо она помещается на ленту, либо выдаётся сообщение об ошибке, и программа возвращается в режим ожидания действий пользователя.

## 4.3 Описание кнопок

При нажатии на кнопку «Запустить машину» необходимо блокировать внесение каких-либо изменений в любое окно вплоть до нажатия на кнопку «Остановить машину» («Сбросить выполнение программы») или пока она сама не остановится.

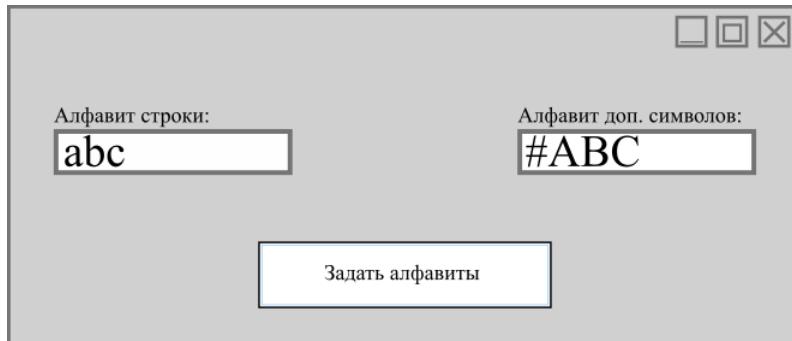
Кнопка «Сбросить выполнение программы» отменяет все изменения, произошедшие с лентой, возвращает её к исходному состоянию и помещает головку под крайний левый символ. Эта кнопка не должна блокироваться во время выполнения программы.

Кнопка «Выполнить один шаг» делает один шаг программы и блокируется, если программа запущена.

Кнопки «Ускорить/замедлить выполнение программы» должны ускорять или замедлять движение головки и внесение соответствующих изменений на ленту. Эти кнопки, очевидно, не блокируются при запуске программы.

## 5 Пример оформления

### 5.1 Окно 1



### 5.2 Окно 2



**Примечание:** В данном интерфейсе по нажатию кнопки «Изменить алфавиты» снова отображается Окно 1.

## 6 Требования

- Проверка корректности:

1. не позволять пользователю на ленту вводить символы не из алфавита строки;

2. проверять корректность вводимых в ячейки таблицы действий;
  3. не запускать машину, если нет ни единой остановки.
- Лента и каретка:
    1. каретка движется плавно по ленте, без резких скачков;
    2. если каретка дошла до левого или правого края экрана и должна продолжить движение за экран, то необходимо сдвинуть ленту вместе с кареткой (можно резко) на треть или четверть длины видимой части ленты, чтобы позволить каретке дальше ехать.
  - Управляющая таблица:
    1. обозначение пустого символа:  $\Lambda$ ,  $\lambda$ ,  $/\backslash$ , (пробел),  $ctrl +$  (пробел), ...;
    2. нужна возможность добавлять и удалять состояния;
    3. в процессе исполнения машины должно подсвечиваться текущее состояние.

## 7 Этапы выполнения задания

### 7.1 Первый этап

На первом этапе ставятся следующие требования:

- грамотно продуманный интерфейс;
- программа работает корректно;
- лента считается конечной, не более длины интерфейса;
- размеры таблицы состояний ограничены;
- алфавит задается только один раз;
- ввод пользователя считается корректным;
- анимации нет, ответ выдается со скоростью исполнения машины.

### 7.2 Второй этап

На втором этапе ставятся следующие требования:

- добавлена бесконечная лента;
- добавлена возможность менять алфавиты.

### 7.3 Третий этап

На третьем этапе ставятся следующие требования:

- добавлена проверка ввода;
- добавлена плавная анимация при работе исполнителя.

### 7.4 Четвертый этап

На четвертом этапе ставятся следующие требования:

- сняты ограничения на размер таблицы, есть прокрутка.