

**ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ УКООПСОЮЗА
«ПОЛТАВСКИЙ УНИВЕРСИТЕТ ЭКОНОМИКИ И ТОРГОВЛИ»**

**ИНСТИТУТ ЭКОНОМИКИ, УПРАВЛЕНИЯ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

**ФОРМА ОБУЧЕНИЯ ДНЕВНАЯ
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И
СОЦИАЛЬНОЙ ИНФОРМАТИКИ**

Допускается к защите

Заведующий кафедрой _____ О.А. Емец
(подпись)

« _____ » _____ 20__ р.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К БАКАЛАВРСКОЙ РАБОТЕ**

на тему

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ТРЕНАЖЕРА
ПО ТЕМЕ «МАГАЗИННЫЕ АВТОМАТЫ» ДИСТАНЦИОННОГО
УЧЕБНОГО КУРСА «ТЕОРИЯ ПРОГРАММИРОВАНИЯ»**

**по специальности 122 «Компьютерные науки и информационные
технологии»**

Исполнитель работы Рзаев Эмин Фархад оглы

_____ « ____ » _____ 2020г.
(подпись)

Научный руководитель к.ф.-м.н., доц. Черненко Оксана Алексеевна

_____ « ____ » _____ 2020г.
(подпись)

ПОЛТАВА 2020г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ПОСТАНОВКА ЗАДАЧИ.....	5
2. ИНФОРМАЦИОННЫЙ ОБЗОР.....	6
2.1. Составляющие дистанционного образования.....	6
2.2. Обзор программного обеспечения дистанционного обучения.....	8
3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	10
3.1. Материал по теме работы.....	10
3.2. Алгоритмизация задачи по теме работы.....	16
3.3. Разработка блок-схемы.....	26
4. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	29
4.1. Обоснование выбора программных средств.....	29
4.2. Описание процесса программной реализации.....	31
4.3. Инструкция по использованию тренажера.....	32
ВЫВОДЫ.....	37
СПИСОК ЛИТЕРАТУРЫ.....	38
ПРИЛОЖЕНИЕ А. КОД ПРОГРАММЫ.....	40

ВВЕДЕНИЕ

Современные телекоммуникационные, информационные и компьютерные технологии, в первую очередь - электронные тренажеры широко используются для обучения. Поэтому разработка тренажеров актуальна сейчас.

Программы-тренажеры используются для тренировки навыков по решению разных задач.

Они обеспечивают:

- последовательное выведение на экран задач заданной сложности с выбранной темой;
- контроль за действиями пользователя по решению предложенной задачи;
- мгновенную реакцию на неправильные действия;
- исправление ошибок пользователя;
- демонстрацию правильного решения задачи;
- вывод итогового сообщения о результатах работы пользователя (возможно, с рекомендациями или советами).

С помощью тренажеров возможна реализация индивидуального подхода в организации обучения. Так, к преимуществам данной формы работы относим следующее:

- генерация учебных задач и примеров;
- возможность пошагового контроля выполнения задания;
- возможность проведения работы с тренажером во вне-учебное время;
- формирование задач различного уровня сложности;
- отсутствие контроля со стороны преподавателя.

Целью работы является создание тренажера по теме «Магазинные автоматы» для дистанционного курса «Теория программирования».

Объектом исследования в данной работе является дистанционные технологии обучения.

Предметом исследования является программирование учебного тренажера.

Тренажер реализован в среде программирования NetBeans на языке программирования Java.

Главная задача работы: разработать алгоритм работы программы-тренажера и его реализовать.

Работа состоит из четырех разделов. В первом разделе рассмотрена постановка задачи. Во втором разделе описаны составляющие дистанционного образования и сделан обзор программного обеспечения дистанционного обучения. В третьем разделе рассмотрены основные определения и понятия, описан алгоритм работы тренажера, представлено блок-схему алгоритма. В четвертом разделе - описано обоснование выбора программных средств, процесса программной реализации, инструкцию по использованию тренажера.

Объем пояснительной записки 51 стр., в т.ч. основная часть - 38 стр., источники - 10 названий.

1. ПОСТАНОВКА ЗАДАЧИ

Для выполнения работы нужно разработать алгоритм работы тренажера и его программную реализацию.

Программа-тренажер должна содержать:

- стартовую страницу;
- страницы с вопросами;
- сообщение об ошибке;
- страницу с результатами прохождения тренажера.

Для этого нужно реализовать следующие функции:

- переход между основными панелями:
 - стартовая страница;
 - страница с вопросами;
 - страница с результатами;
- переход к следующему вопросу;
- переход к предыдущему вопросу;
- проверка выбранного ответа на вопрос;
- вывод сообщения об ошибке;
- подсчет результатов прохождения тренажера.

Для удобства также нужно реализовать вывод условия выбранного примера при его прохождении.

После вывода результатов прохождения тренажера необходимо сделать перенаправление на стартовую страницу, где можно выбрать другой пример и ознакомиться с ним.

2. ИНФОРМАЦИОННЫЙ ОБЗОР

2.1. Составляющие дистанционного образования

Дистанционное образование является одной из самых перспективных форм образования на сегодняшний день. Являясь формой заочной формы образования, дистанционное образование гораздо эффективнее, поскольку предполагает гораздо больше очное общение между преподавателем и студентом, даже при том, если они разделены территориально. Это стало возможным благодаря современным телекоммуникационным технологиям и Интернет. Именно это позволяет значительно повысить эффективность обучения и привлечь значительно большее количество студентов для обучения.

Любое обучение требует определенной организационно-информационной поддержки. Составляющими дистанционного образования являются:

- учебный центр, осуществляющий необходимые функции организационной поддержки;
- информационные ресурсы - учебные курсы, справочные, методические и другие материалы;
- средства обеспечения технологии дистанционного обучения (организационные, технические, программные и др.)
- преподаватели-консультанты, которые курируют дистанционные курсы и называются тьюторами;
- студенты.

Для организации и правильного функционирования системы дистанционного образования необходимо выполнять следующие основные функции:

- поддержка учебных курсов;
- доставка учебного материала студентам;

- поддержка справочных материалов (библиотека)
- консультации;
- контроль знаний;
- организация общения студентов (коллективные формы обучения).

Элементы дистанционного учебного курса:

- информационные ресурсы;
- средства общения;
- система тестирования;
- система администрирования.

Важнейшим компонентом дистанционного курса являются ресурсы, так как в них сосредоточена содержательная часть - контент.

Контент включает:

- учебный материал (конспекты лекций, демонстрационные материалы и т.д.);
- дополнительные информационные материалы (комментарии преподавателя, ответы на вопросы, часто задаваемые и т.д.);
- библиотеку ресурсов (рекомендуемая литература, списки Web-ресурсов по теме курса и т.д.);
- предметный и / или тематический словарь (глоссарий)
- программу обучения (академический календарь).

Все эти компоненты могут быть реализованы различными средствами. Это и применение мощных языков программирования и применение специализированных авторских систем и др. Также дистанционное образование может использоваться в дневной форме обучения. Например, для студентов, что пропустили занятия или не могут присутствовать на занятиях по какой-то причине и т.д., а также как дополнительный учебный ресурс [3].

2.2. Обзор программного обеспечения дистанционного обучения

На сегодняшний день в учебных заведениях Украины используется широкая номенклатура систем дистанционного обучения как с открытым кодом (условно бесплатных), так и платных, как широко используемых и специфически ориентированных.

Так, программный пакет eLearning Server 3000 имеет следующие основные характеристики:

- пакет создан в открытом коде PHP, MySQL, Apache;
- пакет поддерживает все современные форматы данных;
- пакет интегрируется в любую операционную систему: Windows, Linux, Free BSD, Unix и тому подобное;
- пакет достаточно легко воспринимается с точки зрения программирования, имеет подробный комментарий, хорошо подготовлен к изменениям.

Но большинство настроек доступны только при наличии квалифицированного PHP- и SQL-программиста. Кроме того, система не имеет графического интерфейса для гибкой настройки отчетов или изменения структуры шаблонов портала. Изменения параметров системы могут значительно усложнить ее дальнейшее использование.

В веб-ориентированной системы управления обучением ATutor проблем с гибкостью нет. Программный продукт прост в установке, настройке и поддержке для системных администраторов: преподаватели (инструкторы) могут достаточно легко создавать и переносить учебные материалы и запускать свои онлайн-курсы. Поскольку система является модульной, то есть состоит из отдельных функциональных единиц - модулей, то она открыта для модернизации и расширения функциональных возможностей. Данная система является кроссплатформенной. К недостаткам можно отнести слабо развитую систему отчетности.

Система LAMS предлагает преподавателю интуитивно понятный интерфейс для создания учебного контента, который может включать в себя различные индивидуальные задания, задания для групповой работы и фронтальную работу с учебной группой. Но данная платформа имеет частичную поддержку русского / украинского языка, а также вместе с ATutor, имеют ограниченную систему проверки знаний. Также к недостаткам следует отнести отсутствие поддержки модульности, то есть LAMS - система-монолит.

Moodle (модульная объектно-ориентированная среда дистанционного обучения) - бесплатная, открытая (Open Source) система дистанционного обучения. Система ориентирована прежде всего на организацию взаимодействия между преподавателем и учениками, но подходит и для организации традиционных дистанционных курсов, а также поддержки очного обучения. Moodle переведена на десятки языков, в том числе есть и частичный перевод на украинский. Преимуществами Moodle над выше указанными системами являются:

- возможность расширения за счет внешних модулей;
- достаточно развитая система отчетности;
- поддержка внешних тестов;
- высокая степень надежности;
- отсутствие ограничений по количеству слушателей.

Недостатками системы являются высокие требования к квалификации специалистов для настройки и поддержки системы, необходимость использования мощного сервера и широкого канала.

Все четыре рассмотрены программные продукты распространяются на основе GNU General Public License (GPL), которая, в частности, позволяет свободно использовать, изменять и дополнять программу [4-5].

3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

3.1. Материал по теме работы

Автоматы с магазинной памятью, или магазинные автоматы (сокращенно — МП-автоматы), образуют класс распознающих моделей для КС-языков точно так же, как конечные автоматы являются распознающими моделями в классе регулярных языков.

Понятие МП-автомата является частным случаем общего интуитивного понятия автомата. Как и всякий автомат, МП-автомат — это устройство, имеющее блок управления, входную ленту, головку и блок внутренней памяти в виде магазина МП-автомата (или стека).

Блок управления в каждый момент времени находится в одном из конечного множества Q состояний.

Входная лента предполагается "полубесконечной", т.е. она имеет начало ("левый край"), но не имеет конца. Лента разделена на ячейки, пронумерованные от крайней левой ячейки натуральными числами, начиная с единицы; в каждой ячейке может быть записан символ алфавита V , называемого входным алфавитом МП-автомата. В каждый момент времени читающая головка обозревает (или читает) некоторую ячейку входной ленты.

Если в этой ячейке записан некоторый входной символ, т.е. символ входного алфавита, то его называют обозреваемым (в данный момент) входным символом (рис. 3.1).

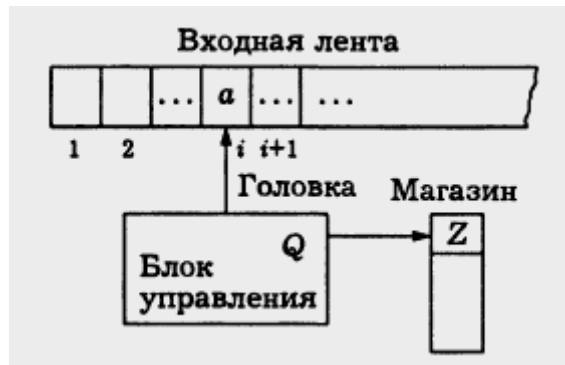


Рисунок 3.1 – Магази́нный автомат

Предполагается, что если на ленте записана некоторая непустая цепочка $x = x(1)x(2)\dots x(k)$ во входном алфавите, то ее символы последовательно записаны в ячейках от первой до k -й, без пропусков, так, что символ $x(i)$ записан в i -й ячейке (рис. 3.2).

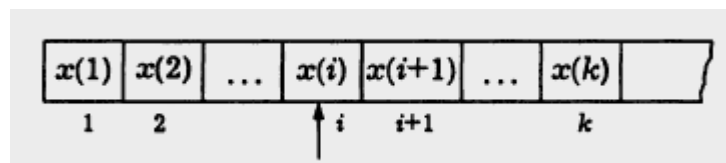


Рисунок 3.2 – На ленте записана некоторая непустая цепочка

МП-автомат может читать цепочку x , продвигая головку только в одном направлении, слева направо, по шагам, причем за каждый шаг головка переходит от обозреваемой ячейки к следующей. Если в какой-то момент времени обозревается символ $x(i)$, $1 \leq i \leq k$, записанной на ленте цепочки x , то непрочитанной частью входной цепочки x будет подцепочка $x(i+1)\dots x(k)$. В частности, при $i=k$ непрочитанная часть совпадает с пустой цепочкой, и тогда говорят, что цепочка x полностью прочитана МП-автоматом (рис. 3.3).

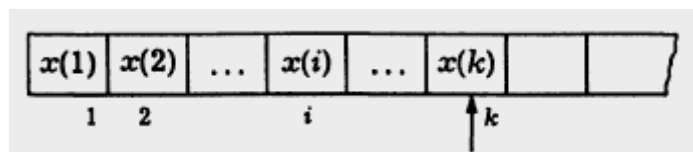


Рисунок 3.3 – Цепочка полностью прочитана МП-автоматом

Магазин МП-автомата устроен и работает следующим образом. Как и входная лента, магазин является полубесконечным и разделенным на пронумерованные ячейки, в каждой из которых может быть записан символ алфавита Γ , называемого магазинным алфавитом МП-автомата. Входной и магазинный алфавиты МП-автомата могут пересекаться и даже совпадать друг с другом. Символы алфавита Γ называют магазинными символами. Первую ячейку магазина называют его верхней ячейкой (иногда просто верхом магазина). Символ, в данный момент записанный в верхней ячейке магазина, называют верхним символом магазина.

Единственная операция, которую МП-автомат может реализовать с магазином, состоит в следующем: верхний символ Z магазина заменяется некоторой цепочкой γ магазинных символов. При этом если цепочка γ непустая, то она записывается в первых m ячейках, где $m = |\gamma|$ (длине цепочки γ), так, что ее первый символ становится верхним символом магазина.

Если до замены Z цепочкой γ под верхней ячейкой (т.е. в ячейках, начиная со второй) была записана какая-то цепочка α , то после замены она сдвигается "в глубь" магазина и оказывается записанной уже в ячейках, начиная с $(m+1)$ -й (рис. 3.4).

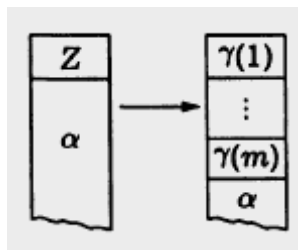


Рисунок 3.4 – Сдвиг "в глубь" магазина

Если же верхний символ Z заменяется пустой цепочкой λ , то после такой замены верхним символом магазина становится первый символ цепочки α (записанной под верхней ячейкой магазина), т.е. цепочка α "поднимается" на одну ячейку.

В случае, когда α — пустая цепочка, замена верхнего символа магазина пустой цепочкой γ приводит к опустошению магазина (ни в одной из ячеек магазина не записан магазинный символ). Заметим, что, по определению, с пустым магазином МП-автомат не может производить никаких операций.

Описанная выше операция с магазином составляет основу поведения МП-автомата, его, образно говоря, "динамику". Эта "динамика" определяется системой команд МП-автомата, которая, аналогично системе команд конечного автомата, определяется как конечное множество δ команд, каждая из которых записывается в виде

$$q\alpha Z \rightarrow r\gamma \quad (3.1)$$

где q и r — состояния из множества Q ; a — входной символ или пустая цепочка; Z — магазинный символ; γ — цепочка магазинных символов (может быть и пустая).

МП-автомат задается упорядоченной семеркой

$$M = (Q, V, \Gamma, q_0, F, Z_0, \delta),$$

где Q — конечное множество состояний; V — алфавит, называемый входным алфавитом; Γ — алфавит, называемый магазинным алфавитом; $q_0 \in Q$ — начальное состояние; $F \subseteq Q$ — подмножество заключительных состояний; $Z_0 \in \Gamma$ — начальный магазинный символ; δ — система команд, определенная как конечное множество команд;

$$q, r \in Q; \alpha \in V \cup \{\lambda\}; Z \in \Gamma; \gamma \in \Gamma.$$

Конфигурацией МП-автомата M называют упорядоченную тройку вида (q, x, β) , где $q \in Q$ — состояние, $x \in V$ — входная цепочка, β — магазинная цепочка.

Конфигурацию $C' = (r, w, \gamma\alpha)$ называют непосредственно выводимой из конфигурации $C = (q, aw, Z\alpha)$, если в множестве команд МП-автомата есть команда (3.1). Отношение непосредственной выводимости на множестве конфигураций МП-автомата M обозначаем \vdash_M , используя

запись $C \vdash_M C'$ (и опуская, если это не вредит пониманию, обозначение самого МП-автомата).

Итак, непосредственная выводимость

$$(q, aw, Z\alpha) \vdash_M (r, w, \gamma\alpha) \quad (3.2)$$

имеет место тогда и только тогда, когда в системе δ команд МП-автомата M есть команда (3.2), которую в этом случае называют применимой к конфигурации $(q, aw, Z\alpha)$.

Таким образом, неформально конфигурация показывает состояние блока управления, неп прочитанную часть входной цепочки (первый символ этой цепочки, если она не пуста, является обозреваемым) и содержимое магазина (первый символ магазинной цепочки β , если она не пуста, есть верхний символ магазина). Если вторая компонента конфигурации — пустая цепочка, то это означает, что входная цепочка прочитана полностью. Если же пуста третья компонента, то это означает, что пуст магазин.

Любую конфигурацию вида (q_0, x, Z_0) называют начальной, а любую конфигурацию вида (q_f, λ, λ) , где $q_f \in F$, — заключительной.

Выводом на множестве конфигураций МП-автомата M называют последовательность $C_0, C_1, \dots, C_n, \dots$ (конечную или бесконечную) таких его конфигураций, что для любого $i \geq 0$ имеет место $C_i \vdash C_{i+1}$, если C_{i+1} существует.

Если вывод конечен и C_n — его последняя конфигурация, то число n называют длиной вывода. В этом случае будем говорить, что вывод C_0, C_1, \dots, C_n связывает конфигурацию C_0 с конфигурацией C_n .

Конфигурацию C' называют выводимой из конфигурации C , обозначая это как $C \vdash^* C'$, если существует вывод, связывающий C с C' , т.е. вывод C_0, C_1, \dots, C_n такой, что $C_0 = C$, а $C_n = C'$.

В общем случае говорят, что конфигурация C' выводится из конфигурации C за n шагов, записывая при этом $C \vdash^n C'$.

Пример. Пусть МП-автомат $M=(\{q_0, q_1, q_2\}, \{a, b\}, \{Z, a, b\}, q_0, \{q_0\}, Z, \delta_2)$ определен множеством команд δ_2 :

$$q_0aZ \rightarrow q_0aZ \quad (1) \quad q_0ab \rightarrow q_0ab \quad (5) \quad q_1aa \rightarrow q_1\lambda \quad (9)$$

$$q_0bZ \rightarrow q_0bZ \quad (2) \quad q_0ba \rightarrow q_0ba \quad (6) \quad q_1bb \rightarrow q_1\lambda \quad (10)$$

$$q_0aa \rightarrow q_0aa \quad (3) \quad q_0bb \rightarrow q_0bb \quad (7) \quad q_1\lambda Z \rightarrow q_2\lambda \quad (11)$$

$$q_0aa \rightarrow q_0\lambda \quad (4) \quad q_0bb \rightarrow q_1\lambda \quad (8)$$

Можно доказать, что этот МП-автомат допускает зеркальный язык $\{xx^R \in \{a, b\}^*\}$, где x^R обозначает инверсию цепочки x .

Приведем допускающую последовательность конфигураций для цепочки $abba$:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ) \vdash_{(8)} \\ \vdash_{(8)} (q_1, a, aZ) \vdash_{(9)} (q_1, \lambda, Z) \vdash_{(11)} (q_2, \lambda, \lambda)$$

К начальной конфигурации $(q_0, abba, Z)$ применима только команда (1). После выполнения этой команды первый символ входной цепочки будет прочитан, а в магазине вместо одного символа Z окажется цепочка aZ , т.е. символ a станет верхним символом магазина.

К полученной конфигурации (q_0, bba, aZ) применима только команда (6), в результате выполнения которой будет прочитан еще один символ входной цепочки, т.е. ее второй символ b , и в магазине окажется цепочка baZ .

К третьей конфигурации записанного выше вывода применимы две команды: (7) и (8). Выбирая команду (8), мы тем самым читаем третий символ входной цепочки, а верхний символ магазина, совпавший с указанным символом входной цепочки, "выталкиваем" из магазина, после чего верхним символом магазина окажется уже символ a , т.е. возникнет конфигурация (q_1, a, aZ) . Заметим также, что после выполнения команды (8) МП-автомат M_2 окажется в новом состоянии q_1 .

Применив к конфигурации (q_1, a, aZ) единственную применимую к ней команду (9), получим конфигурацию (q_1, λ, Z) , т.е. входная цепочка прочитана полностью, а в магазине остался символ Z .

Применяя опять-таки единственно возможную для такой конфигурации команду (11), мы убираем символ Z из магазина, тем самым опустошая его, и переходим в заключительное состояние q_2 .

В то же время, если бы после третьей конфигурации было выбрано "неудачное продолжение", т.е. была бы применена команда (7), и автомат продолжал бы записывать в магазин вместо того, чтобы сравнивать магазин с лентой, то получился бы вывод

$$(q_0, ba, baZ) \vdash_{(7)} (q_0, a, bbaZ) \vdash_{(5)} (q_0, \lambda, abbaZ).$$

Последняя конфигурация в этом выводе характеризуется тем, что входная цепочка прочитана, магазин не пуст, состояние не является заключительным и не применима ни одна команда, т.е. полученная конфигурация $(q_0, \lambda, abbaZ)$ МП-автомата M_2 является тупиковой [6-7].

3.2. Алгоритмизация задачи по теме работы

На стартовом окне пользователю представляется вся информация о тренажере, а также надается выбор пройти теоретическую или практическую часть. После прохождения тестов появляется результат и обратно предлагается, какую часть пройти следующей, или завершить тестирование.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Шаг 1. Выводится условие, задание и варианты ответа.

Условие: Магазинные автоматы образуют класс распознающих моделей для КС-языков точно так же, как конечные автоматы являются распознающими моделями в классе регулярных языков.

Задание: Выберите другие названия магазинных автоматов.

Варианты ответа:

- Автоматы с магазинной памятью;
- Конечные автоматы;
- МП-автоматы;
- Машина Тьюринга.

Если ответ правильный, выбрано первый и третий вариант, то переход к следующему шагу.

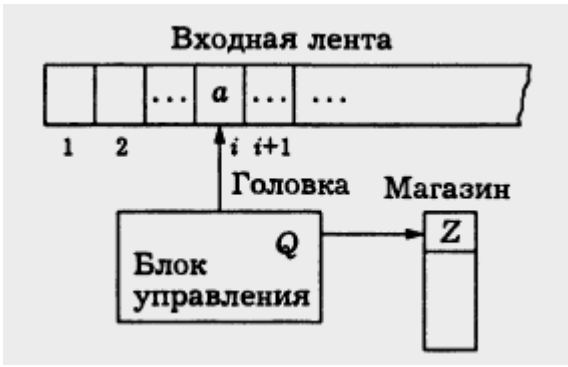
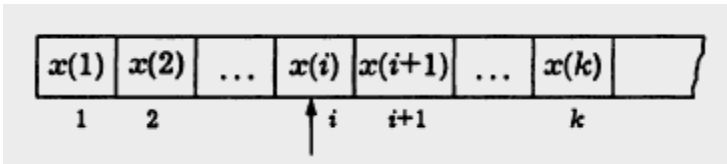
В противном случае – вывод сообщения об ошибке: Ошибка! Автоматы с магазинной памятью, или магазинные автоматы (сокращенно — МП-автоматы).

Шаг 2. Выводится условие, задание и варианты ответа.

Условие: Как и всякий автомат, МП-автомат — это устройство, имеющее блок управления, входную ленту, головку и блок внутренней памяти в виде магазина МП-автомата (или стека).

Задание: Выберите рисунок магазинного автомата.

Варианты ответа:

- ;
- .

Если ответ правильный, выбрано первый вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка!

Магазинный автомат:



Шаг 3. Выводится условие, задание и варианты ответа.

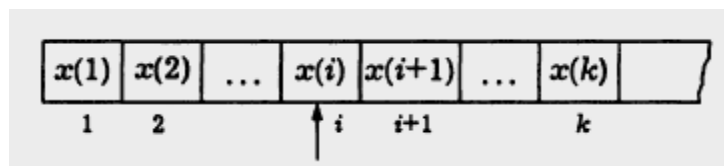
Условие: Предполагается, что если на ленте записана некоторая непустая цепочка $x = x(1)x(2)\dots x(k)$ во входном алфавите, то ее символы последовательно записаны в ячейках от первой до k -й, без пропусков, так, что символ $x(i)$ записан в i -й ячейке.

Задание: Выберите рисунок, на котором указано что на ленте записана некоторая непустая цепочка.

Варианты ответа:



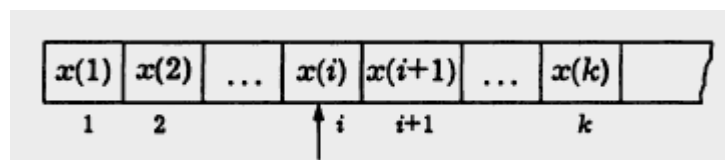
•



•

Если ответ правильный, выбрано второй вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! На ленте записана некоторая непустая цепочка:



Шаг 4. Выводится условие, задание и варианты ответа.

Условие: "Динамика" МП-автомата определяется системой команд МП-автомата, которая, аналогично системе команд конечного автомата, определяется как конечное множество δ команд, каждая из которых записывается в виде

$$q\alpha Z \rightarrow r\gamma$$

Задание: Выберите обозначения q и r .

Варианты ответа:

- q и r — состояния из множества Q ;
- q и r — входной символ или пустая цепочка;
- q и r — магазинный символ;
- q и r — цепочка магазинных символов.

Если ответ правильный, выбрано первый вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! q и r — состояния из множества Q .

Шаг 5. Выводится задание и варианты ответа.

Задание: Выберите обозначение a .

Варианты ответа:

- a — состояния из множества Q ;
- a — входной символ или пустая цепочка;
- a — магазинный символ;
- a — цепочка магазинных символов.

Если ответ правильный, выбрано второй вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! a — входной символ или пустая цепочка.

Шаг 6. Выводится задание и варианты ответа.

Задание: Выберите обозначение Z .

Варианты ответа:

- Z — состояния из множества Q ;
- Z — входной символ или пустая цепочка;
- Z — магазинный символ;
- Z — цепочка магазинных символов.

Если ответ правильный, выбрано третий вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! Z — магазинный символ.

Шаг 7. Выводится задание и варианты ответа.

Задание: Выберите обозначение γ .

Варианты ответа:

- γ — состояния из множества Q ;
- γ — входной символ или пустая цепочка;
- γ — магазинный символ;
- γ — цепочка магазинных символов.

Если ответ правильный, выбрано четвертый вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! γ — цепочка магазинных символов.

Шаг 8. Выводится условие, задание и варианты ответа.

Условие: **МП-автомат** задается упорядоченной семеркой

$$M = (_, _, _, _, _, _, _),$$

где Q — конечное множество состояний; V — алфавит, называемый входным алфавитом; Γ — алфавит, называемый магазинным алфавитом; $q_0 \in Q$ — начальное состояние; $F \subseteq Q$ — подмножество заключительных состояний; $Z_0 \in \Gamma$ — начальный магазинный символ; δ — система команд, определенная как конечное множество команд;

$$q, r \in Q; \alpha \in V \cup \{\lambda\}; Z \in \Gamma; \gamma \in \Gamma.$$

Задание: Выберите правильную формулу.

Варианты ответа:

- $M = (V, Q, \Gamma, q_0, F, Z_0, \delta)$;
- $M = (Q, V, \Gamma, Z_0, F, q_0, \delta)$;
- $M = (Q, V, \Gamma, q_0, \delta, Z_0, F)$;
- $M = (Q, V, \Gamma, q_0, F, Z_0, \delta)$.

Если ответ правильный, выбрано четвертый вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! МП-автомат задается упорядоченной семеркой

$$M = (Q, V, \Gamma, q_0, F, Z_0, \delta).$$

Шаг 9. Выводится условие, задание и варианты ответа.

Условие: **Конфигурацией МП-автомата M** называют упорядоченную тройку вида (q, x, β) .

Задание: Выберите обозначение $q \in Q$.

Варианты ответа:

- $q \in Q$ – состояние;
- $q \in Q$ – входная цепочка;
- $q \in Q$ – магазинная цепочка;
- $q \in Q$ – конечное множество состояний.

Если ответ правильный, выбрано первый вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! $q \in Q$ — состояние.

Шаг 10. Выводится задание и варианты ответа.

Задание: Выберите обозначение $x \in V$.

Варианты ответа:

- $x \in V$ – состояние;

- $x \in V$ – входная цепочка;
- $x \in V$ – магазинная цепочка;
- $x \in V$ – конечное множество состояний.

Если ответ правильный, выбрано второй вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! $x \in V$ — входная цепочка.

Шаг 11. Выводится задание и варианты ответа.

Задание: Выберите обозначение β .

Варианты ответа:

- β – состояние;
- β – входная цепочка;
- β – магазинная цепочка;
- β – конечное множество состояний.

Если ответ правильный, выбрано второй вариант, то переход к следующему шагу.

В противном случае – вывод сообщения об ошибке: Ошибка! β — магазинная цепочка.

Шаг 12. Выводится результат, дается выбор пройти теоретическую часть заново или перейти к практической, или завершить тестирование.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Шаг 1. Выводится условие, задание и список команд.

Условие: Пусть МП-автомат $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{Z, a, b\}, q_0, \{q_0\}, Z, \delta_2)$ определен множеством команд δ_2 :

$$q_0aZ \rightarrow q_0aZ \quad (1) \quad q_0ab \rightarrow q_0ab \quad (5) \quad q_1aa \rightarrow q_1\lambda \quad (9)$$

$$q_0bZ \rightarrow q_0bZ \quad (2) \quad q_0ba \rightarrow q_0ba \quad (6) \quad q_1bb \rightarrow q_1\lambda \quad (10)$$

$$q_0aa \rightarrow q_0aa \quad (3) \quad q_0bb \rightarrow q_0bb \quad (7) \quad q_1\lambda Z \rightarrow q_2\lambda \quad (11)$$

$$q_0aa \rightarrow q_0\lambda \quad (4) \quad q_0bb \rightarrow q_1\lambda \quad (8)$$

Можно доказать, что этот МП-автомат допускает зеркальный язык $\{xx^R \in \{a,b\}^*\}$, где x^R обозначает инверсию цепочки x .

Привести допускающую последовательность конфигураций для цепочки $abba$.

Задание: Выберите команду применимую к начальной конфигурации $(q_0, abba, Z)$.

Список команд:

- $q_0aZ \rightarrow q_0aZ \quad (1)$

- $q_0bZ \rightarrow q_0bZ \quad (2)$

- $q_0aa \rightarrow q_0aa \quad (3)$

- $q_0aa \rightarrow q_0\lambda \quad (4)$

- $q_0ab \rightarrow q_0ab \quad (5)$

- $q_0ba \rightarrow q_0ba \quad (6)$

- $q_0bb \rightarrow q_0bb \quad (7)$

- $q_0bb \rightarrow q_1\lambda \quad (8)$

- $q_1aa \rightarrow q_1\lambda \quad (9)$

- $q_1bb \rightarrow q_1\lambda \quad (10)$

- $q_1\lambda Z \rightarrow q_2\lambda \quad (11)$

Если ответ правильный, выбрано первую команду, то переход к следующему шагу.

В противном случае – вывод сообщения: К начальной конфигурации $(q_0, abba, Z)$ применима только команда (1). После выполнения этой команды первый символ входной цепочки будет прочитан, а в магазине

вместо одного символа Z окажется цепочка aZ , т.е. символ a станет верхним символом магазина.

Шаг 2. Выводится последовательность конфигураций, задание и список команд.

Последовательность конфигураций:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ)$$

Задание: Выберите команду применимую к полученной конфигурации (q_0, bba, aZ) .

Если ответ правильный, выбрано шестую команду, то переход к следующему шагу.

В противном случае – вывод сообщения: К полученной конфигурации (q_0, bba, aZ) применима только команда (6), в результате выполнения которой будет прочитан еще один символ входной цепочки, т.е. ее второй символ b , и в магазине окажется цепочка baZ .

Шаг 3. Выводится последовательность конфигураций, задание и список команд.

Последовательность конфигураций:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ)$$

Задание: Выберите команду применимую к полученной конфигурации (q_0, ba, baZ) .

Если выбрано восьмую команду, то переход к шагу 4. Если седьмую – то переход к шагу 5.

В противном случае – вывод сообщения: К третьей конфигурации записанного выше вывода применимы две команды: (7) и (8). Выбирая команду (8), мы тем самым читаем третий символ входной цепочки, а верхний символ магазина, совпавший с указанным символом входной цепочки, "выталкиваем" из магазина, после чего верхним символом магазина окажется уже символ a , т.е. возникнет конфигурация (q_1, a, aZ) . Заметим также, что после выполнения команды (8) МП-автомат M_2 окажется в новом состоянии q_1 .

Шаг 4. Выводится последовательность конфигураций, результат.

Последовательность конфигураций:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ) \vdash_{(7)} \\ \vdash_{(7)} (q_0, a, bbaZ) \vdash_{(5)} (q_0, \lambda, abbaZ).$$

Результат: Автомат продолжал бы записывать в магазин вместо того, чтобы сравнивать магазин с лентой, то получился бы вывод

$$(q_0, ba, baZ) \vdash_{(7)} (q_0, a, bbaZ) \vdash_{(5)} (q_0, \lambda, abbaZ).$$

Последняя конфигурация в этом выводе характеризуется тем, что входная цепочка прочитана, магазин не пуст, состояние не является заключительным и не применима ни одна команда, т.е. полученная конфигурация $(q_0, \lambda, abbaZ)$ МП-автомата M_2 является тупиковой.

Переход к шагу 8.

Шаг 5. Выводится последовательность конфигураций, задание и список команд.

Последовательность конфигураций:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ) \vdash_{(8)} \\ \vdash_{(8)} (q_1, a, aZ)$$

Задание: Выберите команду применимую к полученной конфигурации (q_1, a, aZ) .

Если ответ правильный, выбрано девятую команду, то переход к следующему шагу.

В противном случае – вывод сообщения: Применив к конфигурации (q_1, a, aZ) единственную применимую к ней команду (9), получим конфигурацию (q_1, λ, Z) , т.е. входная цепочка прочитана полностью, а в магазине остался символ Z .

Шаг 6. Выводится последовательность конфигураций, задание и список команд.

Последовательность конфигураций:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ) \vdash_{(8)} \\ \vdash_{(8)} (q_1, a, aZ) \vdash_{(9)} (q_1, \lambda, Z)$$

Задание: Выберите команду применимую к полученной конфигурации (q_1, λ, Z) .

Если ответ правильный, выбрано одиннадцатую команду, то переход к следующему шагу.

В противном случае – вывод сообщения: Применяя опять-таки единственно возможную для такой конфигурацию команду (11), мы убираем символ Z из магазина, тем самым опустошая его, и переходим в заключительное состояние q_2 .

Шаг 7. Выводится результат.

Результат: Получено последовательность конфигураций для цепочки *abba*:

$$(q_0, abba, Z) \vdash_{(1)} (q_0, bba, aZ) \vdash_{(6)} (q_0, ba, baZ) \vdash_{(8)} \\ \vdash_{(8)} (q_1, a, aZ) \vdash_{(9)} (q_1, \lambda, Z) \vdash_{(11)} (q_2, \lambda, \lambda)$$

Переход к следующему шагу.

Шаг 8. Надается выбор пройти практическую часть заново или перейти к теоретической, или завершить тестирование.

3.3. Разработка блок-схемы

На рисунке 3.5 изображена блок-схема алгоритма работы тренажера.

На рисунке 3.6 изображена блок-схема алгоритма работы теоретической части тренажера.

На рисунке 3.7 изображена блок-схема алгоритма работы практической части тренажера.

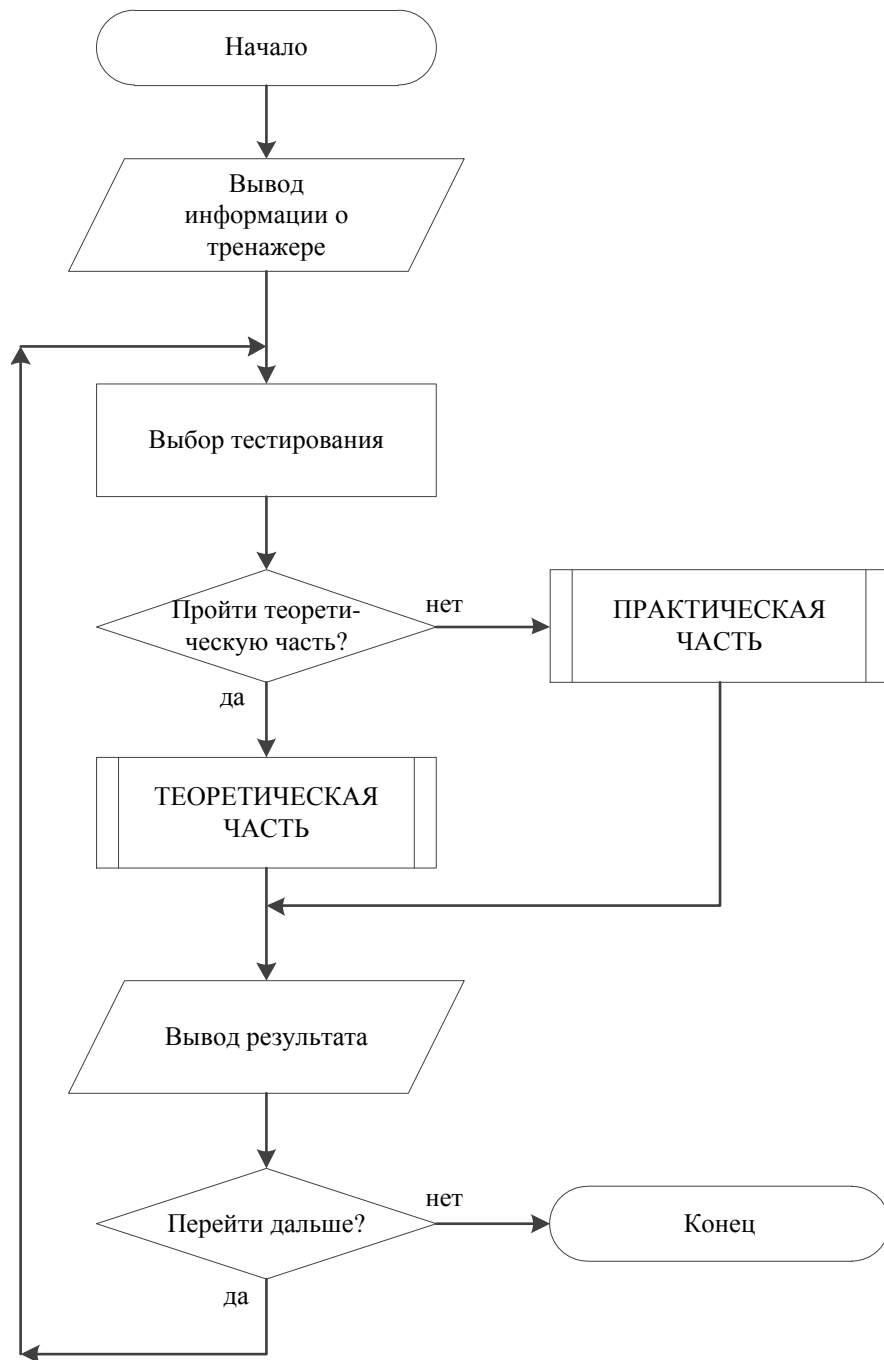


Рисунок 3.5 – Блок-схема алгоритма работы тренажера

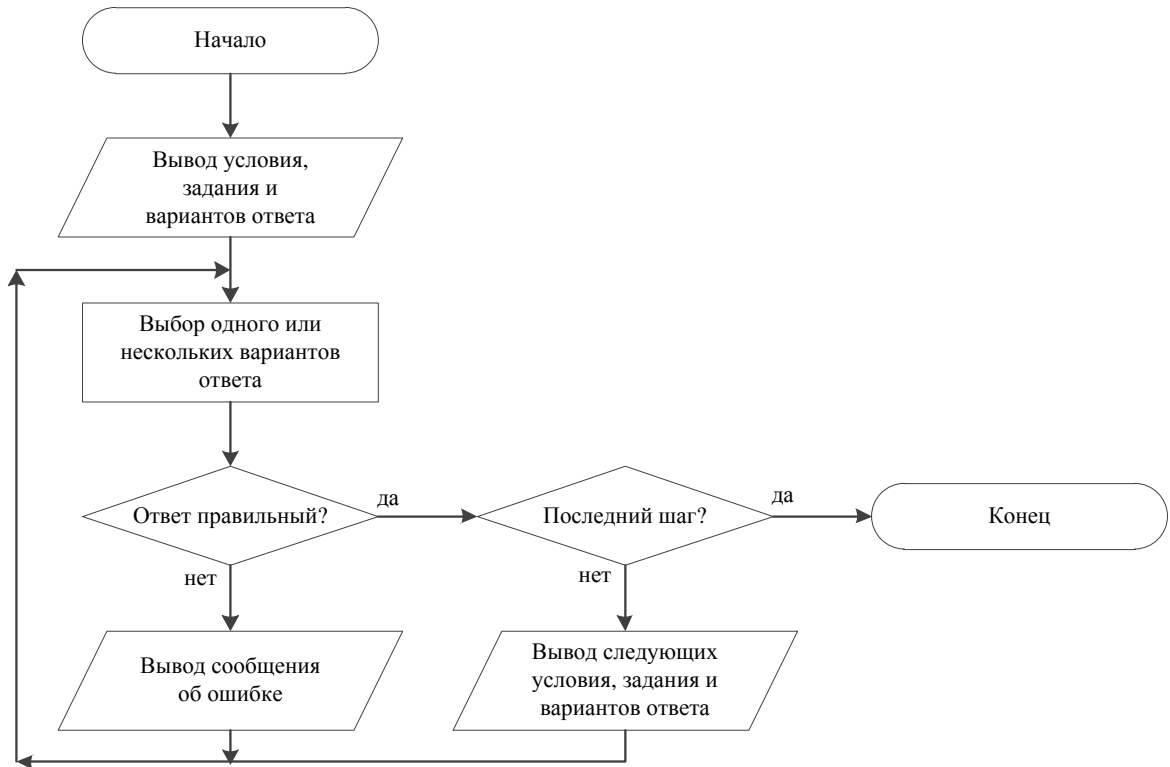


Рисунок 3.6 – Блок-схема алгоритма работы теоретической части

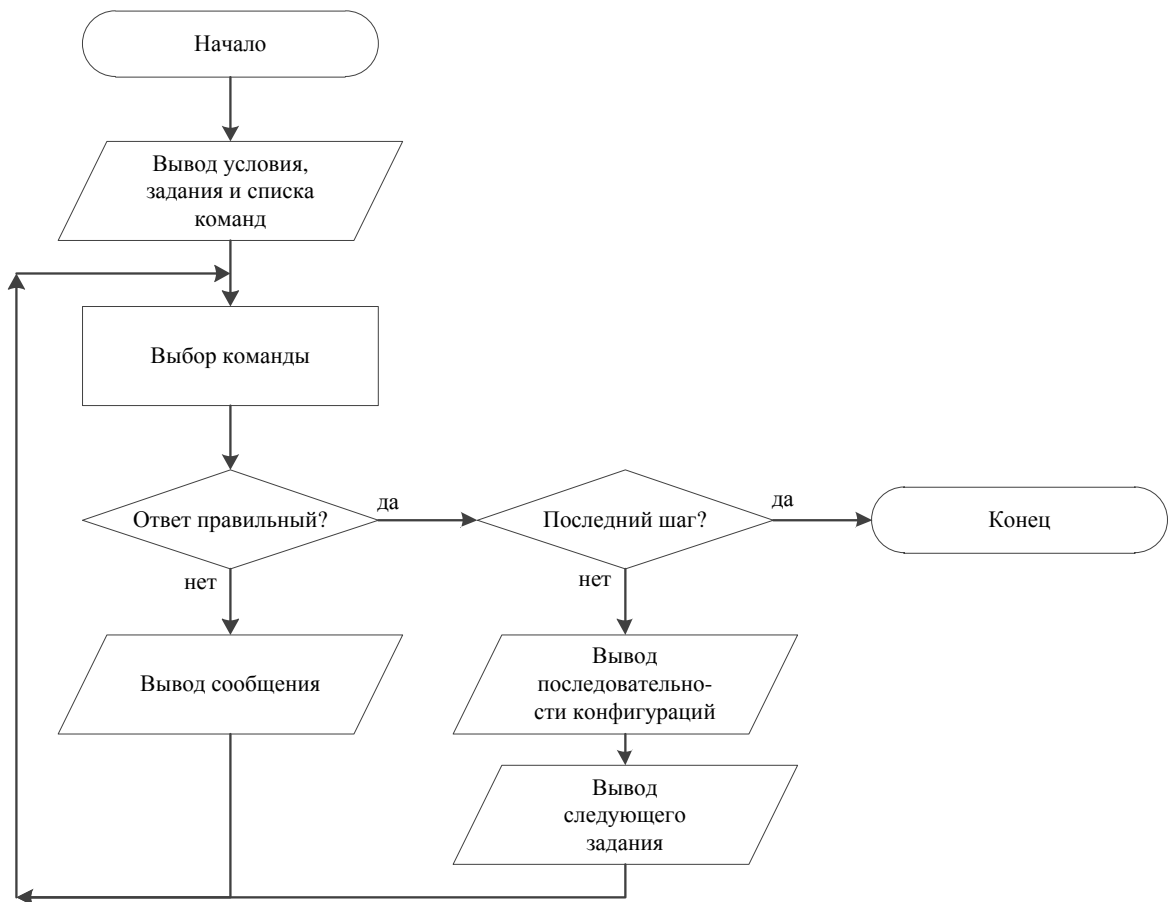


Рисунок 3.7 – Блок-схема алгоритма работы практической части

4. ПРАКТИЧЕСКАЯ ЧАСТЬ

4.1. Обоснование выбора программных средств

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process, язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года. На 2019 год Java — один из самых популярных языков программирования.

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят снижение производительности. Ряд усовершенствований несколько увеличил скорость выполнения программ на Java:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
- обширное использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
- аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами архитектуры ARM).

По данным сайта shootout.alioth.debian.org, для семи разных задач время выполнения на Java составляет в среднем в полтора-два раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в отдельных случаях в 7 раз медленнее. С другой стороны, для большинства из них потребление памяти Java-машиной было в 10—30 раз больше, чем программой на C/C++. Также примечательно исследование, проведённое компанией Google, согласно которому отмечается существенно более низкая производительность и большее потребление памяти в тестовых примерах на Java в сравнении с аналогичными программами на C++.

Идеи, заложенные в концепцию и различные реализации среды виртуальной машины Java, вдохновили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. Эти идеи нашли также выражение в спецификации общезыковой инфраструктуры CLI, заложенной в основу платформы .NET компанией Microsoft.

Java-апплет — прикладная программа, чаще всего написанная на языке программирования Java в форме байт-кода. Java-апплеты выполняются в веб-обозревателе с использованием виртуальной Java машины (JVM), или в Sun's AppletViewer, автономном средстве для

испытания апплетов. Java-апплеты были внедрены в первой версии языка Java в 1995 году. Java-апплеты обычно пишутся на языке программирования Java, но могут быть написаны и на других языках, которые компилируются в байт-код Java, таких, как Jython [8-9].

4.2. Описание процесса программной реализации

Сначала создано переменные:

```
int step = 1;
boolean error = false;
int er_count = 0;
```

Кнопка «ТЕОРЕТИЧЕСКАЯ ЧАСТЬ» открывает тестирование теоретической части тренажера.

```
private void TeoriaActionPerformed(java.awt.event.ActionEvent evt) {
    CardLayout cl =(CardLayout) Panel.getLayout();
    cl.show(Panel, "theory");
    cl =(CardLayout) jPanel1.getLayout();
    cl.show(jPanel1, "tstep1");
    buttonGroup1.clearSelection();
    step = 1;
    er_count = 0;
}
```

Кнопка «ПРАКТИЧЕСКАЯ ЧАСТЬ» открывает тестирование практической части.

```
private void PracticaActionPerformed(java.awt.event.ActionEvent evt)
{
    CardLayout cl =(CardLayout) Panel.getLayout();
    cl.show(Panel, "practic");
```

```

cl =(CardLayout) jPanel2.getLayout();
cl.show(jPanel2, "pstep1");
buttonGroup1.clearSelection();
step = 1;
}

```

Кнопка «Заккрыть» закрывает тренажер.

```

private void CloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

```

Кнопки «Ответить» в обеих частях тестирования обрабатывают результат выбора пользователя согласно алгоритма (см. Приложение А).

```

switch(step) {
    case 1:
        if(jRadioButton41.isSelected()) {
            cl.show(jPanel2, "pstep2");
            step++;
        } else {
            cl.show(jPanel2, "perror1");
        }
        break;
    ...
}

```

4.3. Инструкция по использованию тренажера

На главной странице отображается тема тренажера, информация о студенте. Предлагается выбрать пройти теоретическую часть или практическую (рис. 4.1).

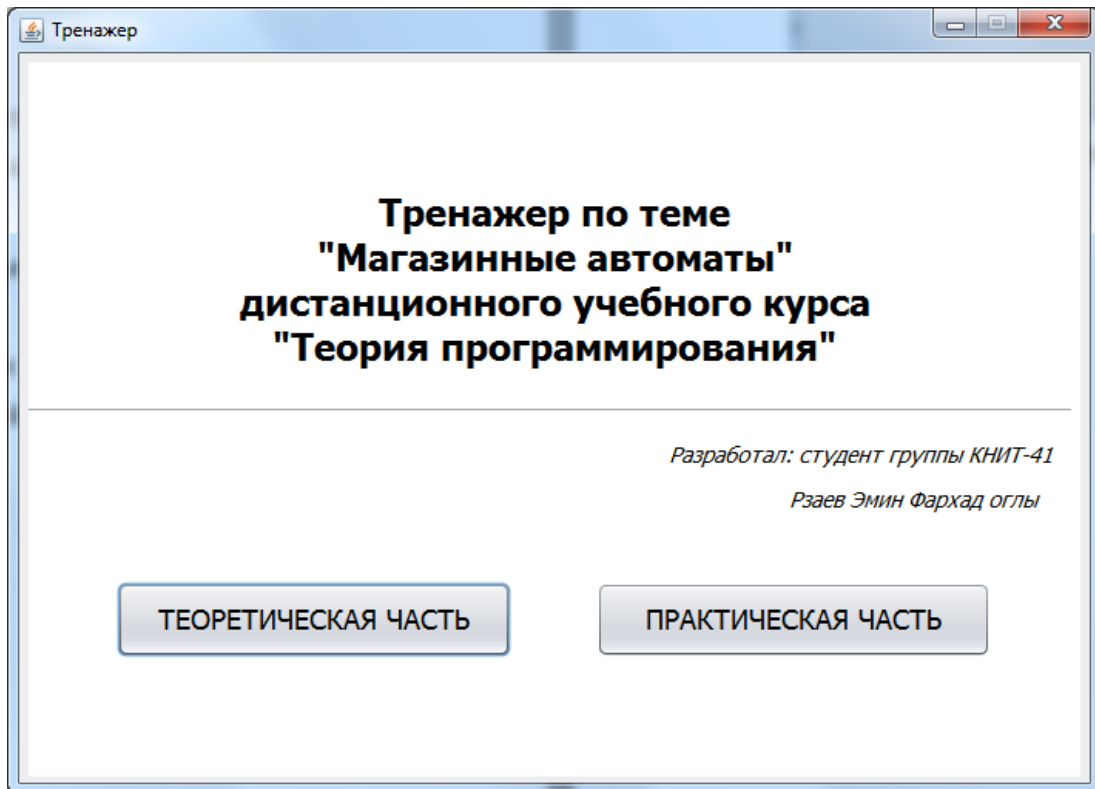


Рисунок 4.1 – Главная страница

Если нажать кнопку «ТЕОРЕТИЧЕСКАЯ ЧАСТЬ», то выводится первый вопрос тестирования теоретической части тренажера (рис. 4.2).

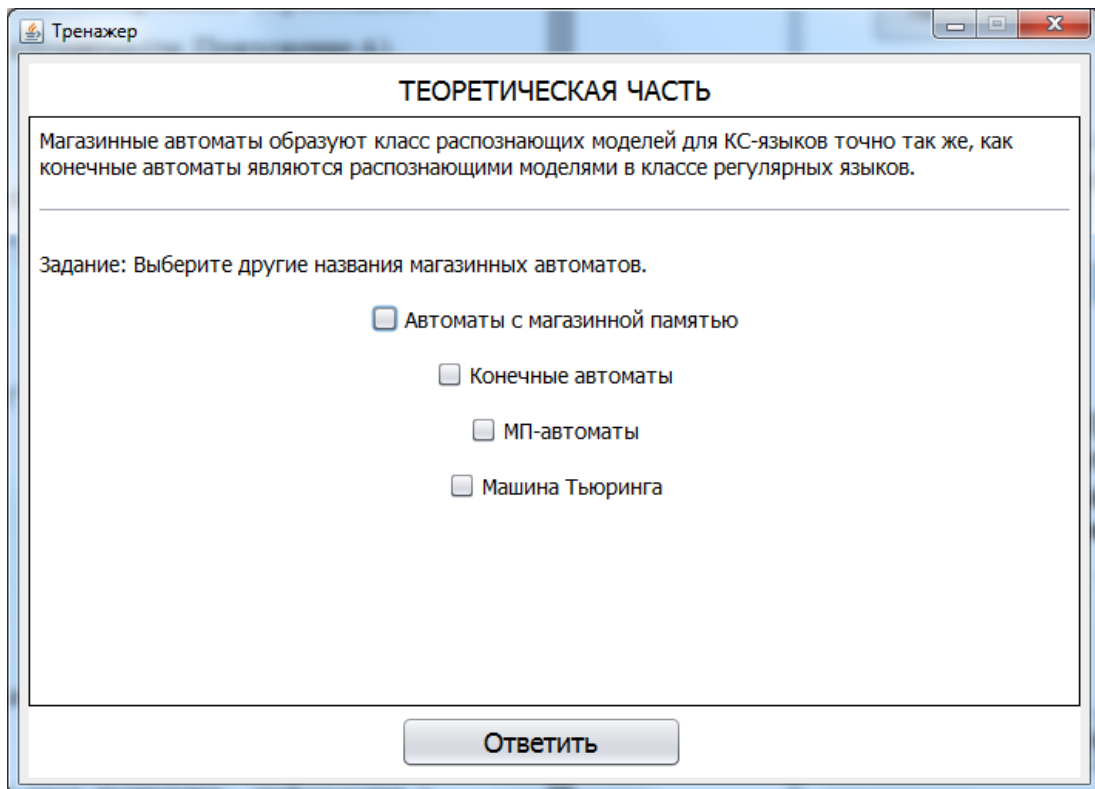


Рисунок 4.2 – Первый вопрос теоретической части

Если ответить неправильно, то отображается сообщение об ошибке. После этого нужно обратно нажать «Ответить» (рис. 4.3).

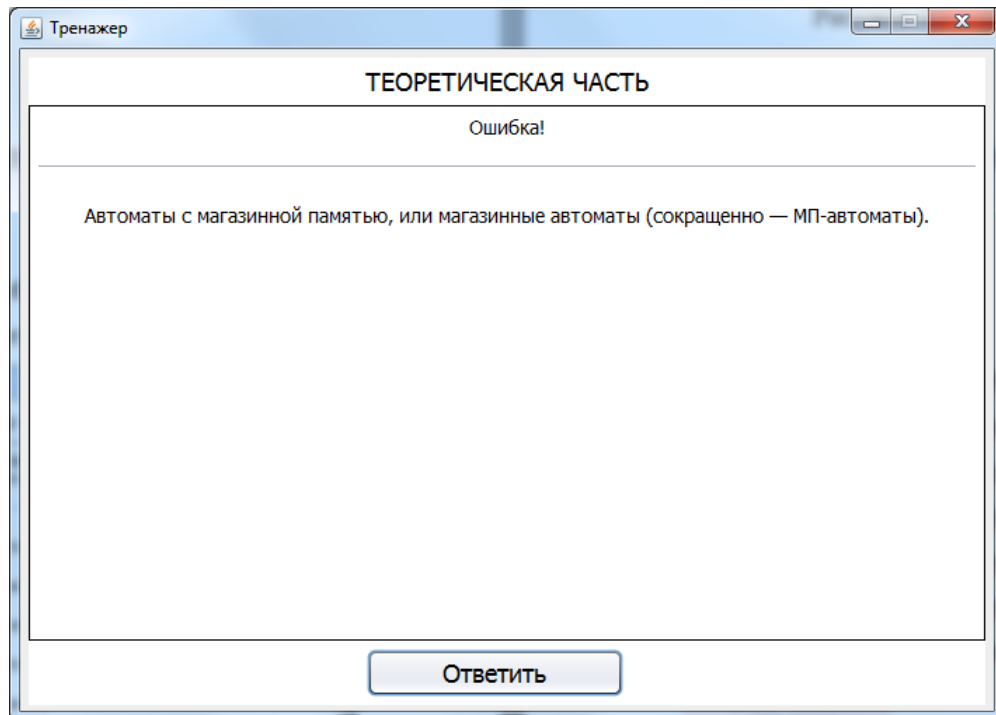


Рисунок 4.3 – Сообщение об ошибке в теоретической части

В конце показывается количество допущенных ошибок, также снова предлагается выбрать тестирование (рис. 4.4).

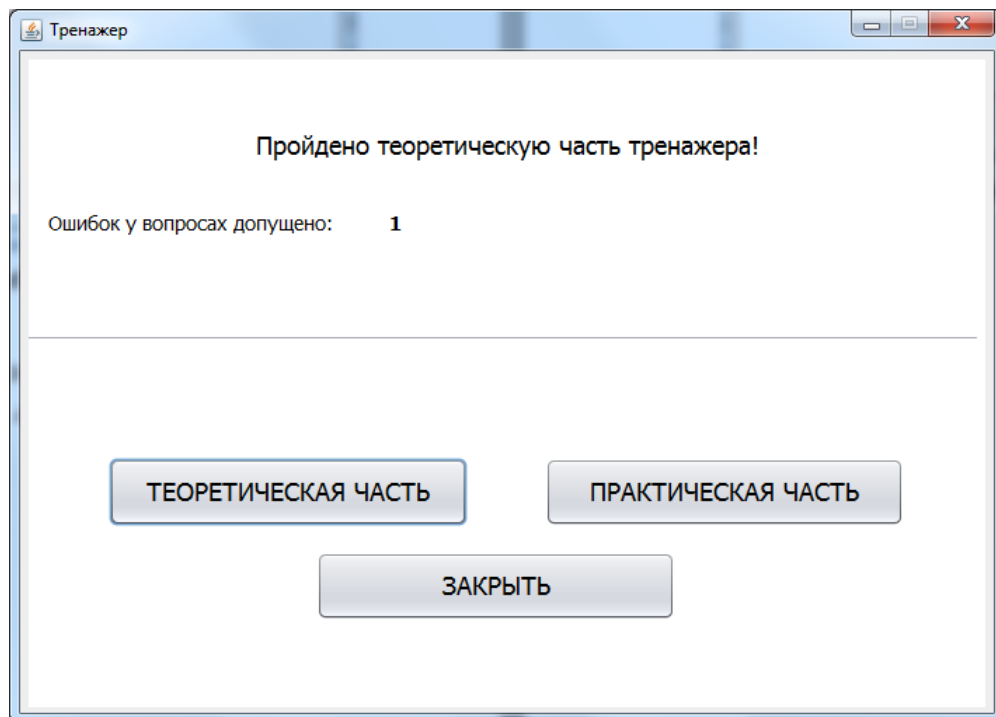


Рисунок 4.4 – Результат теоретической части

Если нажать кнопку «ПРАКТИЧЕСКАЯ ЧАСТЬ», то выводится первый вопрос тестирования практической части тренажера (рис. 4.5).

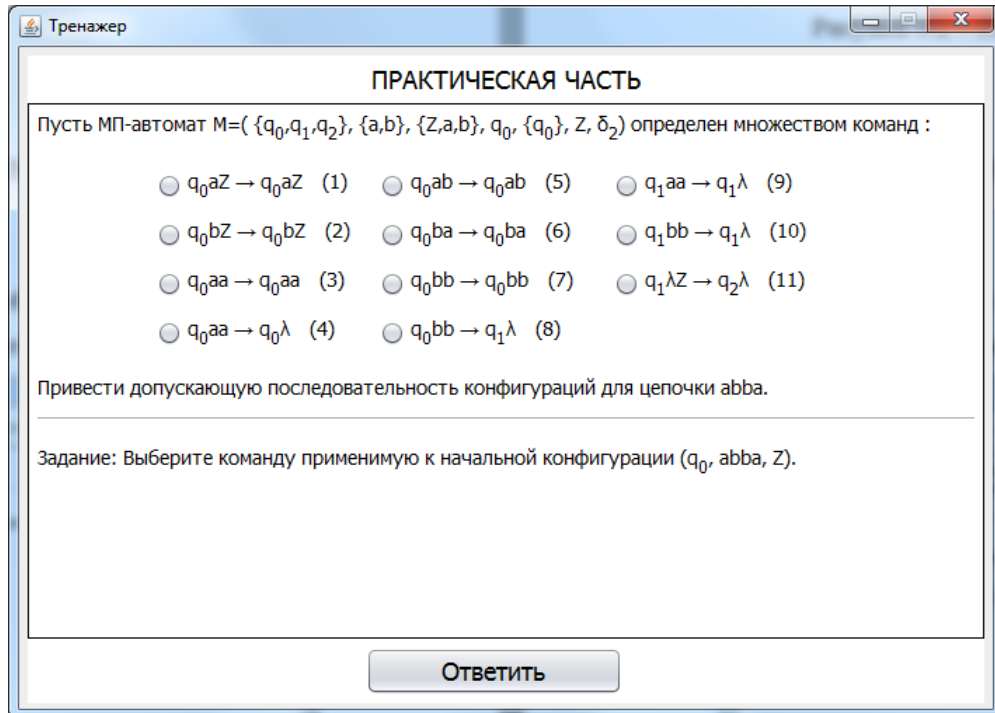


Рисунок 4.5 – Первый вопрос практической части

Если ответить неправильно, то отображается подсказка (рис.4.6).

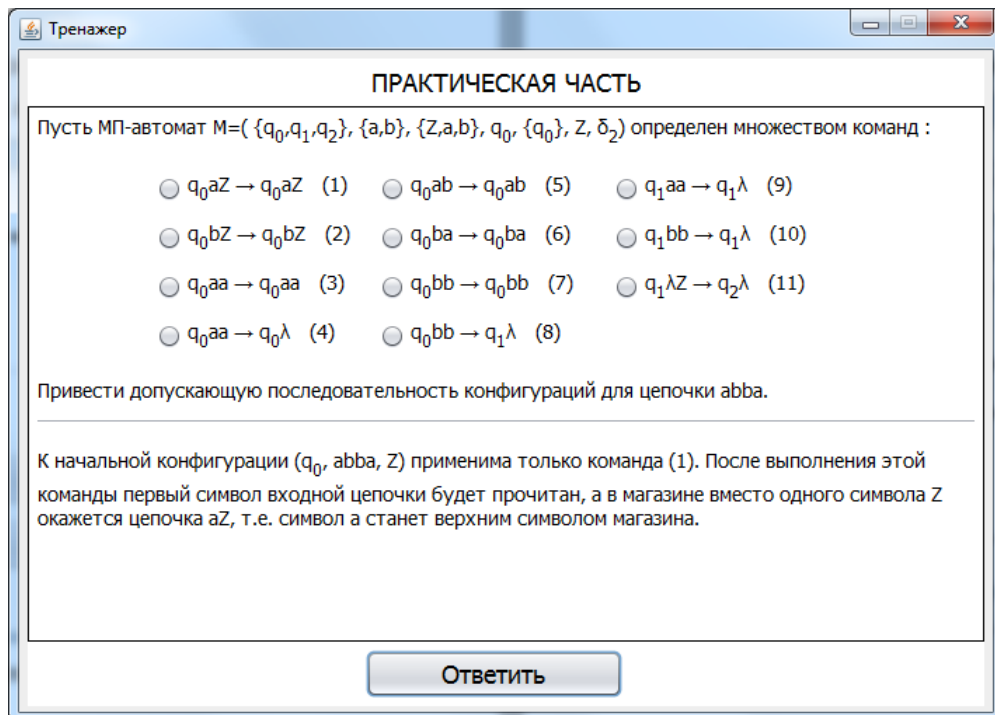


Рисунок 4.6 – Подсказка в практической части

В конце показывается результат (рис. 4.7).

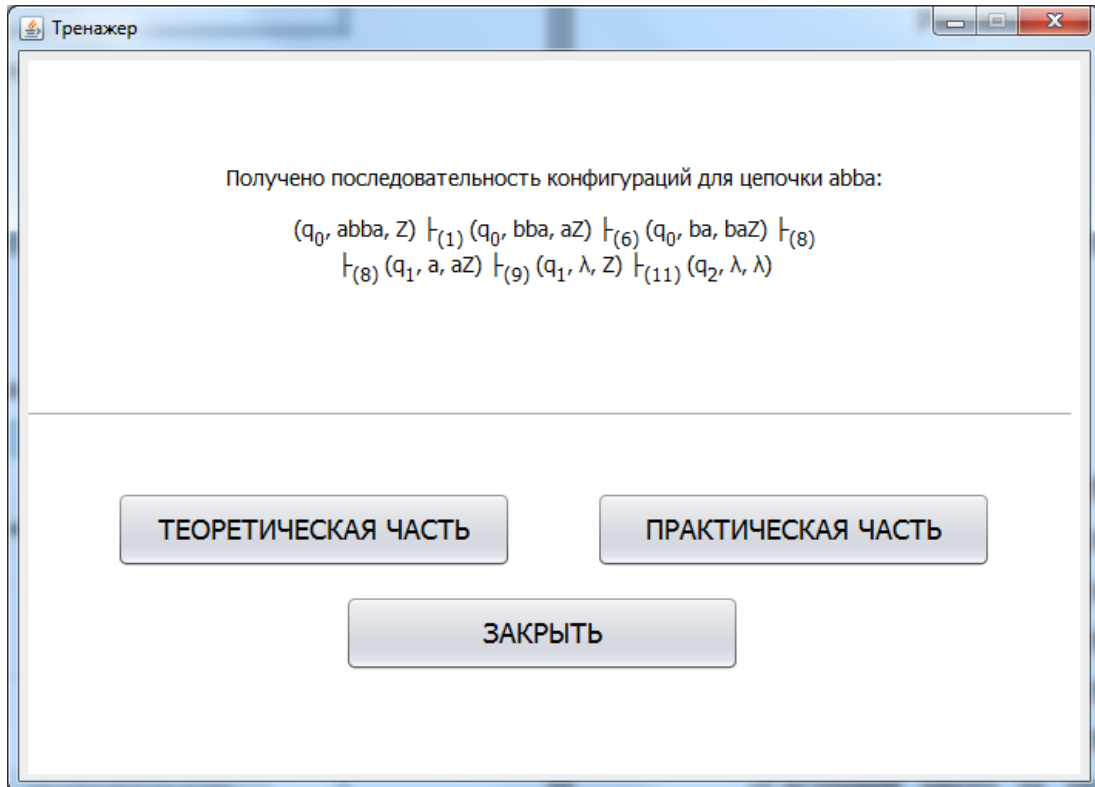


Рисунок 4.7 – Результат практической части

ВЫВОДЫ

Целью выполненной работы является созданный тренажер с темы «Магазинные автоматы» для дистанционного курса «Теория программирования».

Программа-тренажер содержит:

- стартовую страницу;
- страницы с вопросами;
- сообщения об ошибке;
- страницы с результатами прохождения тренажера.

Было реализовано следующие функции:

- переход между основными панелями:
 - стартовая страница;
 - страница с вопросами;
 - страница с результатами;
- переход к следующему вопросу;
- проверка выбранного ответа на вопрос;
- вывод сообщения об ошибке;
- подсчет результатов прохождения тренажера.

Для удобства также реализован вывод условия выбранного примера при его прохождении.

На стартовом окне пользователю представляется вся информация о тренажере, а также надается выбор пройти теоретическую или практическую часть. После прохождения тестов появляется результат и обратно предлагается, какую часть пройти следующей, или завершить тестирование.

Если ответить неправильно на любой вопрос в теоретической части, то отображается сообщение об ошибке. В практической – выводится подсказка.

СПИСОК ЛІТЕРАТУРИ

1. Ємець О. О. Методичні рекомендації до виконання дипломної роботи для студентів ступеня магістра спеціальності 122 «Комп'ютерні науки» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2018. – 35 с.
2. Ємець О. О. Методичні рекомендації щодо оформлення пояснювальних записок до курсових проектів (робіт) для студентів напряму підготовки «Інформатика» і спеціальності «Соціальна інформатика» / О. О. Ємець, Ол-ра О. Ємець. – Полтава : РВВ ПУЕТ, 2014. – 68 с.
3. Дистанционное образование: плюсы и минусы / Юлия Юткина // Дистанционное обучение - информационный портал [Электронный ресурс]. – Режим доступа:
<http://www.distance-learning.ru/db/el/doc.html>
4. Гладир А.І. Системи дистанційного навчання – огляд програмних платформ / А.І. Гладир, Зачепа Н.В., Мотруніч О.О // Проблеми вищої школи. Інновації в освіті та виробництві. Комп'ютерні технології в освіті та виробництві. – Кременчук : КНУ ім. М. Остроградського. – с. 43-44.
5. 6 систем дистанционного обучения: какую выбрать школе, репетитору, тренеру? / Сергей Золотухин - // EduNeo - актуальные методики преподавания, новые технологии и тренды в образовании, практический педагогический опыт. [Электронный ресурс]. – Режим доступа: <https://www.eduneo.ru/3-besplatnye-sistemy-distancionnogo-obucheniya-obzor/>
6. Мартыненко Б.К. Языки и трансляции: Учеб. пособие Изд. 2-е, испр. и доп. / Б.К. Мартыненко – СПб.: Изд-во С.-Петербур. ун-та, 2008. 257 с.
7. Магази́нные автоматы (автомат с магази́нной памятью) // Математический форум Math Help Planet - Обсуждение и решение задач по математике, физике, химии, экономике [Электронный ресурс].

– Режим доступа:

<http://mathhelpplanet.com/static.php?p=magazinnyye-avtomaty>

8. Java // Материал из Википедии — свободной энциклопедии [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Java>
9. Java-апплет // Материал из Википедии — свободной энциклопедии [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Java-апплет>
10. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

ПРИЛОЖЕНИЕ А. КОД ПРОГРАММЫ