

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Преподаватель департамента  
программной инженерии  
Национального исследовательского  
университета «Высшая школа  
экономики», кандидат физико-  
математических наук

\_\_\_\_\_ А. А. Каленкова  
«\_\_» \_\_\_\_\_ 2015 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»

\_\_\_\_\_ В. В. Шилов  
«\_\_» \_\_\_\_\_ 2015 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

**ПРОГРАММА ВЗАИМНОГО ПРЕОБРАЗОВАНИЯ КОНЕЧНЫХ АВТОМАТОВ  
И РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

RU.17701729.503100-01 81 01-1-ЛУ

Исполнитель  
студент группы 101ПИ

\_\_\_\_\_ Т. Р. Бадретдинов  
«\_\_» \_\_\_\_\_ 2015 г.

УТВЕРЖДЕН  
RU.17701729.503100-01 81 01-1-ЛУ

ПРОГРАММА ВЗАИМНОГО ПРЕОБРАЗОВАНИЯ КОНЕЧНЫХ АВТОМАТОВ И  
РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

**Пояснительная записка**

RU.17701729.503100-01 81 01-1

Листов 37

## АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «Finite Automata Builder.exe» («Программа взаимного преобразования конечных автоматов и регулярных выражений»), предназначенной для построения КА и регулярных выражений на основе введенного пользователем регулярного выражения.

В данном программном документе, в разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В данном программном документе, в разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
- описание и обоснование выбора метода организации входных и выходных данных;
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться со списком сокращений, приведенным в Приложении 1 настоящей пояснительной записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ .....	5
1.1.	Наименование программы .....	5
1.2.	Документы, на основании которых ведётся разработка .....	5
2.	НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ .....	6
2.1.	Назначение программы .....	6
2.1.1.	Функциональное назначение .....	6
2.1.2.	Эксплуатационное назначение .....	6
2.2.	Краткая характеристика области применения .....	6
3.	ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....	7
3.1.	Постановка задачи на разработку программы .....	7
3.2.	Описание алгоритма и функционирования программы .....	7
3.3.	Описание и обоснование выбора метода организации входных и выходных данных .....	7
3.3.1.	Описание метода организации входных и выходных данных .....	7
3.3.2.	Обоснования выбора метода организации входных и выходных данных .....	7
3.4.	Описание и обоснование выбора состава технических и программных средств .....	7
3.4.1.	Состав технических и программных средств .....	7
3.4.2.	Обоснование выбора технических и программных средств .....	8
4.	ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....	9
4.1.	Предполагаемая потребность .....	9
4.2.	Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами .....	9
5.	ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ .....	10
	ПРИЛОЖЕНИЕ 1 .....	11
	ПРИЛОЖЕНИЕ 2 .....	12
	ПРИЛОЖЕНИЕ 3 .....	13

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. ВВЕДЕНИЕ

### 1.1. Наименование программы

Наименование программы — «Программа взаимного преобразования конечных автоматов и регулярных выражений».

Краткое наименование программы — «Finite Automata Builder».

### 1.2. Документы, на основании которых ведётся разработка

Разработка ведётся на основании приказа Национального исследовательского университета «Высшая школа экономики» № 6.18.1-02/1912-10 от 19.12.14.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Назначение программы

#### 2.1.1. Функциональное назначение

Функциональным назначением программы является построение КА на основе регулярных выражений и обратных преобразований автоматов в регулярное выражение.

#### 2.1.2. Эксплуатационное назначение

Программа может быть полезна всем, кто изучает регулярные выражения и КА, а также тем, кому нужно проверить построенный вручную КА.

### 2.2. Краткая характеристика области применения

«Программа взаимного преобразования конечных автоматов и регулярных выражений» — программа, позволяющая строить КА на основе регулярных выражений и преобразовывать полученные КА обратно в регулярные выражения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку программы

Разрабатываемая программа должна:

- 1) строить ДКА и НКА на основе регулярных выражений;
- 2) минимизировать построенное ДКА
- 3) строить ДКА из НКА
- 4) получать регулярные выражения из КА
- 5) рисовать КА в отдельном поле с возможностью масштабирования полученного рисунка

#### 3.2. Описание алгоритма и функционирования программы

Используемые в программе алгоритмы описаны в [12] и [13]. Их выбор был обоснован относительной простотой реализацией и относительно высокому времени выполнения алгоритма.

#### 3.3. Описание и обоснование выбора метода организации входных и выходных данных

##### 3.3.1. Описание метода организации входных и выходных данных

Входные данные могут быть внесены пользователем вручную в форму ввода регулярного выражения.

Выходные данные представляются в двух вариантах:

- 1) КА — в виде рисунков в отдельном окне;
- 2) Регулярные выражения — в текстовом виде в отдельном окне.

##### 3.3.2. Обоснования выбора метода организации входных и выходных данных

Так как основная цель программы — создание КА для исследовательских целей, выходные данные должны быть прежде всего удобно отображаться и иметь гибкие возможности настройки. Поэтому в программе были созданы условия для отображения КА. Для удобства пользователь может менять размеры окна, а также менять размер и перемещать КА внутри окна. Всё это позволяет подробно изучить полученные в процессе работы программы КА.

Регулярные выражения выводятся в отдельном окне в текстовом виде с возможностью копировать текст. Для удобства текст доступен только для чтения.

#### 3.4. Описание и обоснование выбора состава технических и программных средств

##### 3.4.1. Состав технических и программных средств

Для работы программы необходим следующий состав программных средств:

- 1) Операционная система Microsoft Windows XP или более поздняя версия;
- 2) Установленный Microsoft .NET Framework 3.5 или более поздняя версия;

Для работы программы необходим следующий состав технических средств:

- 1) Мышь;
- 2) Клавиатура;
- 3) Монитор;
- 4) Процессор (тактовая частота не менее 1 ГГц);
- 5) ОЗУ (ёмкость не менее 512 МБ);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 6) Жёсткий диск (ёмкость не менее 2 ГБ);
- 7) CD-ROM привод;
- 8) звуковая плата;
- 9) динамики или наушники.

### 3.4.2. Обоснование выбора технических и программных средств

При реализации программы использована LINQ-выражения. Данная функция была представлена в составе пространства имён System.Linq в версии Microsoft .NET Framework 3.5.

Microsoft .NET Framework 3.5 в свою очередь требует [16]:

- 1) Операционную систему Windows XP;
- 2) Процессор с тактовой частотой не менее 400 МГц (рекомендуется не менее 1 ГГц);
- 3) 96 МБ оперативной памяти или больше (рекомендуется 256 МБ или больше);
- 4) 0,28 гигабайт (ГБ) (для 32-разрядной системы) или 0,61 ГБ (для 64-разрядной системы) пространства на жёстком диске и выше.

Для установки Windows XP [15] необходимы:

- 1) процессор Pentium с частотой 233 МГц или более быстрый (рекомендуется не менее 300 МГц);
- 2) не менее 64 МБ оперативной памяти (рекомендуется не менее 128 МБ);
- 3) не менее 1,5 ГБ свободного места на жестком диске;
- 4) дисковод для компакт- или DVD-дисков;
- 5) клавиатура;
- 6) мышь;
- 7) видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 800x600 точек;
- 8) звуковая плата;
- 9) динамики или наушники.

Наибольшее из окон программы имеет разрешение 575x323 точек, поэтому разрешение монитора должно быть не менее чем 575x323 точек.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 4.1. Предполагаемая потребность

Программа может быть полезна всем, кто занимается исследовательской деятельностью: студентам, профессорам, учёным.

### 4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Существующие в открытом доступе программы с похожим функционалом едва ли можно назвать конкурентами. Большинство из них сильно ограничено в возможностях. Многие из программ не удобны для повседневного использования. Во всех из найденных решений отсутствует возможность масштабировать полученные рисунки КА.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ Р 7.02-2006 Консервация документов на компакт-дисках. Общие требования. – М.: ИПК Издательство стандартов, 2006.
11. 18. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. В. А. Серебряков, М. П. Галочкин. Основы конструирования компиляторов. 3.3. Алгоритмы построения конечных автоматов
13. Sarel Har-Peled and Madhusudan Parthasarathy. Lecture 8: From DFAs/NFAs to Regular Expressions
14. Dick Grune and Criel J.H. Jacobs. Parsing Techniques: A Practical Guide
15. Системные требования ОС Windows XP. [Электронный ресурс]// URL: <http://support.microsoft.com/kb/314865/ru> (Дата обращения: 12.05.2015, режим доступа: свободный).
16. Требования к системе для .NET Framework. [Электронный ресурс]// URL:[https://msdn.microsoft.com/en-us/library/bb882520\(v=vs.90\).aspx](http://https://msdn.microsoft.com/en-us/library/bb882520(v=vs.90).aspx) (Дата обращения: 12.05.2015, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1  
ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

КА — конечный автомат;

НКА — недетерминированный конечный автомат;

ДКА — детерминированный конечный автомат;

МДКА — минимизированный детерминированный конечный автомат.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 2

### ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 9.1 — описание и функциональное назначение классов

Класс	Назначение
Converter	Статический вспомогательный класс для преобразования КА и регулярных выражений.
DFAToMDFA	Класс, который позволяет минимизировать ДКА.
FAToDot	Класс, который позволяет преобразовать КА в описание графа на языке Dot.
FAToRegEx	Класс, который позволяет преобразовать КА в регулярное выражение.
NFAToDFA	Класс, который позволяет преобразовать НКА в ДКА.
RegExToDFA	Класс, который позволяет преобразовать регулярное выражение в ДКА.
RegExToNFA	Класс, который позволяет преобразовать регулярное выражение в НКА.
Drawing	Класс, который позволяет рисовать КА с помощью PictureBox.
Edge	Класс, представляющий структуру ребра графа.
Geometry	Класс, содержащий реализацию геометрических алгоритмов.
GraphViz	Класс, представляющий собой оболочку над библиотекой GraphViz.
Node	Класс, представляющий структуру вершины графа.
Edge	Класс, представляющий структуру ребра КА.
FiniteAutomaton	Класс, представляющий структуру КА.
State	Класс, представляющий структуру состояния КА.
Transition	Класс, представляющий структуру перехода КА.
ExtendedNode	Класс, представляющий структуру вершины дерева строк.
EdtendedTree	Класс, представляющий структуру дерева строк.
RegEx	Класс, представляющий структуру регулярного выражения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 3 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 10.1 — описание полей, методов и свойств класса Converter.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
FreeTransition	internal	char	Константа, которая содержит символ свободного перехода.	
FreeTransitionString	internal	string	Константа, которая содержит символ свободного перехода.	
FreeTransitionDecoded	internal	string	Константа, которая содержит символ свободного перехода для перевода графа в язык Dot.	
EndMarker	Internal	Char	Константа, которая содержит конечный символ.	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
GetCharacterList	internal	List<string>	FiniteAutomaton automaton	Метод возвращает список символов КА.

Таблица 10.2 — описание полей, методов и свойств класса DFAToMDFA.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Convert	public	FiniteAutomaton	FiniteAutomaton dfa	Метод для получения МДКА
Divide	private	List<List<State>>	FiniteAutomaton dfa, List<List<State>> set	Метод для получения списков, полученных в результате деления на группы.
DivideGroup	private	bool	FiniteAutomaton dfa, List<List<State>> set, ref List<List<State>> pseudoGroup	Метод для разделения группы состояний на более мелкие
FormMDFASStates	private	List<State>	List<List<State>> pseudoStateList	Метод для создания списка состояний МДКА
FormMDFATransitions	private	List<Transition>	FiniteAutomaton dfa, List<List<State>> pseudoStateList, List<State> stateList	Метод для создания списка переходов МДКА
FormPSet	private	void	FiniteAutomaton dfa, ref List<List<State>> pSet	Метод для создания Р набора
GetFinalStateList	private	List<State>	FiniteAutomaton dfa, List<List<State>> pseudoStateList, List<State> stateList	Метод для получения списка конечных состояний
GetGroup	private	int	List<List<State>> set, State state	Метод для получения номера группы
GetIndexList	private	List<int>	int[] index	Метод для получения списка индексов
GetTransition	private	Transition	FiniteAutomaton automaton, State state, string character	Метод для получения перехода
RecalculateIndex	private	void	ref int[] groupIndex	Метод для повторного вычисления индекса

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 10.3 — описание полей, методов и свойств класса FAtToDot.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
ConvertToArray	private	string[]	FiniteAutomaton fa	Метод представляет КА в формате Dot в виде массива строк
ConvertToString	public	string	FiniteAutomaton fa	Метод представляет КА в формате Dot в виде строки

Таблица 10.4 — описание полей, методов и свойств класса FAtToRegEx.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
ConcatenateTransitions	private	string	List<string> transitionList	Метод объединяет полученные переходы в строку
Convert	public	string	FiniteAutomaton fa	Метод преобразует КА в регулярное выражение
GetSelfTransition	private	Transition	ref List<Transition> transitionList, State state	Метод создаёт переход из состояния в само себя
RemoveFreeTransitions	private	string	Transition transition	Метод удаляет переход
RemoveFreeTransitions	private	List<string>	List<Transition> transitionList	Метод удаляет все переходы
RemoveState	private	void	ref List<Transition> transitionList, State	Метод удаляет состояние

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

			state	
--	--	--	-------	--

Таблица 10.5 — описание полей, методов и свойств класса NFAToDFA.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Convert	public	FiniteAutomaton	FiniteAutomaton nfa	Метод преобразует НКА в ДКА
EClosure	private	List<State>	FiniteAutomaton nfa, State state	Метод производит замыкание по пустому переходу
Equals	private	bool	List<State> list1, List<State> list2	Метод проверяет равенство двух списков состояний
GetEChildren	private	List<State>	FiniteAutomaton nfa, List<State> currentChildren, State state	Метод получает список дочерних состояний
IndexOf	private	int	List<List<State>> pseudoStateList, List<State> pseudoState	Метод находит индекс состояния
MarkState	private	void	FiniteAutomaton automaton, ref List<State> realStateList, ref List<Transition> transitionList, ref List<State> finalStateList, ref List<List<State>> pseudoStateList, ref List<bool> marked,	Метод помечает состояние

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



			List<string> characterList, List<State> pseudoState	
Move	private	List<State> >	FiniteAutomaton nfa, List<State> currentChildren, State state, string moveCharacter	Метод выполняет сдвиг по ленте

Таблица 10.6 — описание полей, методов и свойств класса RegExToDFA.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
BuildNewTree	private	ExtendedTree	ExtendedTree tree	Метод строит новое дерево
CalculateFirstPos	private	void	ExtendedNode node, List<ExtendedNode> positionList, ref List<ExtendedNode>[] firstpos, bool[] nullable	Метод вычисляет первую позицию у каждой вершины
CalculateFollowPos	private	void	List<ExtendedNode> positionList, ref List<ExtendedNode>[] followpos, List<ExtendedNode>[] firstpos, List<ExtendedNode>[] lastpos	Метод вычисляет следующую позицию у каждой вершины
CalculateFunctions	private	List<ExtendedNode>[]	ExtendedNode node, out List<ExtendedNode> positions, out	Метод вычисляет функции

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

			List<ExtendedNode> initialPseudoState	
CalculateLastPos	private	void	ExtendedNode node, List<ExtendedNode> positionList, ref List<ExtendedNode>[ ] lastpos, bool[] nullable	Метод вычисляет последнюю позицию у каждой вершины
CalculateNullable	private	void	ExtendedNode node, List<ExtendedNode> positionList, ref bool[] nullable	Метод вычисляет пустые позиции у каждой вершины
CalculateOtherNodes	private	void	ExtendedNode node, ref List<ExtendedNode> positions	Метод вычисляет остальные вершины
CalculatePositions	private	void	ExtendedNode node, ref List<ExtendedNode> positions	Метод вычисляет позиции у каждой вершины
Convert	public	FiniteAutomaton	ExtendedTree regex	Метод преобразует регулярное выражение в ДКА
Equals	private	bool	List<ExtendedNode> list1, List<ExtendedNode> list2	Метод проверяет равенство двух списков вершин
GetCharacterList	private	void	ExtendedNode node, ref List<string> characterList	Метод создаёт список вершин в виде списка строк
IndexOf	private	int	List<List<ExtendedNode>> pseudoStateList, List<ExtendedNode> pseudoState	Метод получает индекс вершины в списке вершин

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

MarkState	private	void	List<ExtendedNode>[ ] followpos, List<ExtendedNode> positions, ref List<State> realStateList, ref List<Transition> transitionList, ref List<State> finalStateList, ref List<List<ExtendedN ode>> pseudoStateList, ref List<bool> marked, List<string> characterList, List<ExtendedNode> pseudoState	Метод помечает вершину
-----------	---------	------	--	---------------------------

Таблица 10.7 — описание полей, методов и свойств класса RegExToNFA.cs

Методы				
Имя	Модифи катор доступа	Тип	Аргументы	Назначение
Convert	public	FiniteAuto maton	ExtendedTree regex	Метод преобразует регулярное выражение в НКА
MakeChoiceAuto maton	private	FiniteAuto maton	FiniteAutomaton first, FiniteAutomaton second	Метод создаёт пересечение двух НКА
MakeCycleAuto maton	private	FiniteAuto maton	FiniteAutomaton automaton	Метод создаёт цикл из НКА
MakeUnionAuto maton	private	FiniteAuto maton	FiniteAutomaton first, FiniteAutomaton second	Метод создаёт объединение двух НКА

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

NodeToNFA	private	FiniteAutomaton	ExtendedNode node	
-----------	---------	-----------------	-------------------	--

Таблица 10.8 — описание полей, методов и свойств класса Converter.cs

Поля			
Имя	Модификатор доступа	Тип	Назначение
ArrowBrush	private	Brush	Кисть для рисования стрелок
ArrowPen	private	Pen	Карандаш для рисования стрелок
EdgeBrush	private	Brush	Кисть для рисования рёбер КА
EdgeFont	private	Font	Шрифт для подписи рёбер КА
_edgeList	private	List<Edge>	Список рёбер
EdgePen	private	Pen	Карандаш для рисования рёбер КА
_endPoint	private	PointF	Конечная точка
_isInverted	private	bool	Логическая переменная, которая принимает значение true, когда рисунок расположен горизонтально; false, когда рисунок расположен вертикально
Multiplier	public	double	Масштаб рисунка
NodeBrush	private	Brush	Кисть для рисования вершин КА
NodeFont	private	Font	Шрифт для подписи вершин КА

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

_nodeList	private	List<Node>	Список вершин	
NodePen	private	Pen	Карандаш для рисования вершин КА	
TransparentBrush	private	Brush	Прозрачная кисть	
AdditionalSpace	private	int	Размер отступа	
ArrowSize	private	double	Размер стрелок	
FinalStatePen	private	double	Карандаш для рисования конечных состояний	
MultiplierRate	public	double	Коэффициент изменения размера рисунка	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Draw	public	void	object sender, PaintEventArgs e	Метод рисует КА
DrawEdges	private	void	PaintEventArgs e	Метод рисует рёбра
Drawing	public	конструктор	string path, bool isInverted	Конструктор
DrawNodes	private	void	PaintEventArgs e	Метод рисует вершины
FindNode	private	Node	int id	Метод находит вершину по её идентификатору
PossiblySwap	private	void	ref double a, ref double b	Метод меняет местами координаты, если это необходимо
SetEdges	private	void	string[] graph	Метод устанавливает
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

				рёбра
SetEndPoint	private	void	string header	Метод устанавливает конечную точку
SetNodes	private	void	string[] graph	Метод устанавливает вершины

Таблица 10.9 — описание полей, методов и свойств класса Edge.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
ChildId	public	int	Идентификатор вершины-ребёнка	
Label	public	string	Метка ребра	
LabelPosition	public	PointF	Позиция метки	
ParentId	public	int	Идентификатор вершины-родителя	
Path	public	List<PointF>	Список точек, через которые проходит ребро	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Edge	public	конструктор	int parentId, int childId, List<PointF> path, string label, PointF labelPosition	Конструктор

Таблица 10.10 — описание полей, методов и свойств класса Geometry.cs

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

FindLineCircleIntersections	public	int	float cx, float cy, float radius, PointF point1, PointF point2, out PointF intersection1, out PointF intersection2	Метод нахождения пересечения окружностей
GetArrowPoint	public	PointF[ ]	PointF startPoint, PointF circleIntersectionPoint, double arrowSize	Метод получения расположения стрелки
GetPerpendicularLine	private	Line	PointF point, Line line	Метод нахождения перпендикулярной прямой
GetY	private	bool	Line line, double x, out double y	Метод получения ординаты
SetPoint	private	void	double distance, PointF startPoint, PointF endPoint, out PointF point	Метод устанавливает точку
SetPoints	private	void	Line line, double distance, PointF startPoint, out PointF point1, out PointF point2	Метод устанавливает точки

Таблица 10.11 — описание полей, методов и свойств класса GraphViz.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
LibGraph	private	string	Название файла библиотеки Graph	
LibGvc	private	string	Название файла библиотеки Gvc	
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Success	private	int	Код успешного создания рисунка	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
agclose	private	void	IntPtr g	Метод освобождает память
agmemread	private	IntPtr	string data	Метод читает граф из строки
GetLayout	public	byte[]	string source, string layout, string format	Метод строит граф
gvContext	private	IntPtr	—	Метод создаёт новый контекст
gvFreeContext	private	int	IntPtr gvc	Метод освобождает ресурсы контекста
gvFreeLayout	private	int	IntPtr gvc, IntPtr g	Метод освобождает ресурсы графа
gvLayout	private	int	IntPtr gvc, IntPtr g, string engine	Метод строит граф используя движок
gvRenderData	private	int	IntPtr gvc, IntPtr g, string format, out IntPtr result, out int length	Метод создаёт граф в памяти

Таблица 10.12 — описание полей, методов и свойств класса Node.cs

Поля				
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Имя	Модификатор доступа	Тип	Назначение	
Center	public	PointF	Центр вершины	
D	public	float	Диаметр вершины	
Id	public	int	Идентификатор вершины	
IsFinal	public	bool	Логическая переменная, которая принимает значение true, если вершина конечная	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Node	public	конструктор	int id, PointF center, float d, bool isFinal	Конструктор

Таблица 10.13 — описание полей, методов и свойств класса Edge.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Text	public	string	Текст	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Edge	public	конструктор	string text	Конструктор

Таблица 10.14 — описание полей, методов и свойств класса FiniteAutomaton.cs

Поля				
Имя		Модификатор	Тип	Назначение
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	доступа			
FinalStates	public	List<State>		Конечные состояния
InitialState	public	State		Начальное состояние
States	public	List<State>		Состояния
Transitions	public	List<Transition>		Переходы
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
FiniteAutomaton	public	конструктор	List<State> states, List<Transition> transitions, State initialState, List<State> finalStates	Конструктор
GetChildren	public	List<State>	State state	Метод получает список дочерних состояний

Таблица 10.15 — описание полей, методов и свойств класса State.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Id	public	int	Идентификатор состояния	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
State	public	конструктор	int id	Конструктор

Таблица 10.16 — описание полей, методов и свойств класса Transition.cs

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Child	public	State	Дочернее состояние	
Parent	public	State	Родительское состояние	
TransitionText	public	Edge	Текст перехода	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Transition	public	конструктор	State parent, State child, Edge transitionText	Конструктор

Таблица 10.17 — описание полей, методов и свойств класса ExtendedNode.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Children	public	List<ExtendedNode>	Список дочерних вершин	
Value	public	string	Значение вершины	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
ExtendedNode	public	конструктор	string value, List<ExtendedNode> children	Конструктор

Таблица 10.18 — описание полей, методов и свойств класса ExtendedTree.cs

Поля				
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Имя	Модификатор доступа	Тип	Назначение	
StartNode	public	ExtendedNode	Начальная вершина	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
ExtendedTree	public	конструктор	ExtendedNode startNode	Конструктор

Таблица 10.19 — описание полей, методов и свойств класса RegEx.cs

Поля				
Имя		Модификатор доступа	Тип	Назначение
ExtendedGrammar		private	Grammar	Расширенная грамматика
ExtendedRegExGrammar		private	string[]	Расширенная грамматика в виде массива строк
Grammar		private	Grammar	Грамматика
RegExGrammar		private	string[]	Грамматика в виде массива строк
ExtendedRegExTableFilePath		private	string	Путь к таблице расширенной грамматики
RegExTableFilePath		private	string	Путь к таблице грамматики
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Convert	private	ExtendedTree	Tree tree	Метод конвертирует дерево
ConvertNode	private	ExtendedNode	Node node	Метод конвертирует вершину
FormRuleList	private	List<Rule>	string[] grammarRules	Метод формирует список правил
InitializeExtendedRegExGrammar	private	void	—	Метод инициализирует расширенную грамматику
InitializeRegExGrammar	private	void	—	Метод инициализирует грамматику
Parse	public	Tree	string str	Метод преобразует строку содержащую регулярное выражение
ParseExtended	public	Tree	string str	Метод преобразует строку содержащую регулярное выражение
Regex	private	конструктор	—	Конструктор
ReplaceExtendedNode	private	ExtendedNode	Node oldNode	Метод заменяет вершину дерева
ReplaceExtendedNodes	private	ExtendedTree	Tree oldTree	Метод заменяет все вершины дерева
ReplaceNode	private	Node	Node oldNode	Метод заменяет вершину дерева
ReplaceNodes	private	Tree	Tree oldTree	Метод заменяет все вершины дерева

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Simplify	private	Tree	Tree oldTree	Метод упрощает дерево
SimplifyNode	private	Node	Node oldNode	Метод упрощает вершину
TransformExtendedTree	public	ExtendedTree	Tree oldTree	Метод преобразует расширенное дерево
TransformTree	public	ExtendedTree	Tree oldTree	Метод преобразует дерево

Таблица 10.20 — описание полей, методов и свойств класса Form1.cs

Поля				
Имя		Модификатор доступа	Тип	Назначение
button1		private	Button	Кнопка 1
button2		private	Button	Кнопка 2
button3		private	Button	Кнопка 3
components		private	IContainer	Компоненты
_dfaAutomaton		private	FiniteAutomaton	ДКА
label1		private	Label	Метка 1
label2		private	Label	Метка 2
_nfaAutomaton		private	FiniteAutomaton	НКА
radioButton1		private	RadioButton	Радио-кнопка 1
radioButton2		private	RadioButton	Радио-кнопка 2
_terminalIsACharacter		private	bool	Логическая переменная, которая говорит,
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

			является ли терминал символом	
textBox1	private	TextBox	Поля для ввода текста	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
button1_Click	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при нажатии на кнопку 1
button2_Click	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при нажатии на кнопку 2
button3_Click	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при нажатии на кнопку 3
Dispose	protected	void	bool disposing	Метод освобождает ресурсы
Form1	public	конструктор	—	Конструктор без параметров
InitializeComponent	private	void	—	Метод инициализирует форму

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

radioButton1_CheckedChanged	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при нажатии на радиокнопку 1
radioButton2_CheckedChanged	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при нажатии на радиокнопку 2
textBox1_TextChanged	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает при изменении поля для ввода текста

Таблица 10.21 — описание полей, методов и свойств класса Form2.cs

Поля			
Имя	Модификатор доступа	Тип	Назначение
_clickPosition	private	Point	Позиция клика
components	private	IContainer	Компоненты
_graph	private	Drawing	Граф
_isLeftClick	private	bool	Логическая переменная, принимающая значение true, если нажата левая кнопка

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



_moveMode	private	bool	Логическая переменная, принимающая значение true, если зажата кнопка мыши	
panel1	private	Panel	Панель	
pictureBox1	private	PictureBox	Поле для рисования	
_scrollPosition	private	Point	Позиция скrolла	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Dispose	protected	void	bool disposing	Метод освобождает ресурсы
Form2	private	конструктор	—	Конструктор без параметров
Form2	public	конструктор	FiniteAutomaton automaton	Конструктор
InitializeButtons	protected	void	EventHandler button1Handler, string button1Text, EventHandler button2Handler, string button2Text	Метод инициализирует кнопки
InitializeComponent	private	void	—	Метод инициализирует форму
pictureBox1_DoubleClick	private	void	object sender, EventArgs e	Метод-событие, которое срабатывает, когда кнопка

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

				мышь нажимается дважды
pictureBox1_MouseDown	private	void	object sender, EventArgs e	Метод- событие, которое срабатывает, когда кнопка мышь отжимается
pictureBox1_MouseMove	private	void	object sender, EventArgs e	Метод- событие, которое срабатывает, когда мышь передвигается
pictureBox1_MouseUp	private	void	object sender, EventArgs e	Метод- событие, которое срабатывает, когда кнопка мышь нажимается
pictureBox1_Paint	private	void	object sender, PaintEventArgs e	Метод- событие, которое срабатывает, когда поле для рисования обновляется
SetDrawing	private	void	FiniteAutomaton automaton	Метод запускает рисование

Таблица 10.22 — описание полей, методов и свойств класса FormDFA.cs

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Поля				
Имя	Модификатор доступа	Тип	Назначение	
components	private	IContainer	Компоненты	
_dfa	private	FiniteAutomaton	ДКА	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Dispose	protected	void	bool disposing	Метод освобождает ресурсы
FormDFA	public	конструктор	FiniteAutomaton automaton	Конструктор
InitializeComponent	private	void	—	Метод инициализирует форму
ToMDFA	private	void	object sender, EventArgs e	Минимизация ДКА
ToRegEx	private	void	object sender, EventArgs e	Перевод в регулярное выражение

Таблица 10.23 — описание полей, методов и свойств класса FormNFA.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
components	private	IContainer	Компоненты	
_dfa	private	FiniteAutomaton	ДКА	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Dispose	protected	void	bool disposing	Метод освобождает ресурсы
FormNFA	public	конструктор	FiniteAutomaton automaton	Конструктор
InitializeComponent	private	void	—	Метод инициализирует форму
ToDFA	private	void	object sender, EventArgs e	Перевод в ДКА
ToRegEx	private	void	object sender, EventArgs e	Перевод в регулярное выражение

Таблица 10.24 — описание полей, методов и свойств класса FormRegEx.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
components	private	IContainer	Компоненты	
textBox1	private	TextBox	Поле для ввода текста	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Dispose	protected	void	bool disposing	Метод освобождает ресурсы
FormRegEx	public	конструктор	string regex	Конструктор
InitializeComponent	private	void	—	Метод инициализирует форму

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. И дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.503100-01				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата