

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**  
**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**Кафедра «Информационные технологии»**

**Е. Л. МИНЯЙЛОВА**

**РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ  
СРЕДСТВАМИ ИНФОРМАЦИОННЫХ  
КОМПЬЮТЕРНЫХ КОМПЛЕКСОВ:  
ЗАДАЧА  
«ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ В СИСТЕМАХ  
УПРАВЛЕНИЯ НА ТРАНСПОРТЕ»**

*Учебно-методическое пособие*

**Гомель 2012**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

Кафедра «Информационные технологии»

Е. Л. МИНЯЙЛОВА

РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ  
СРЕДСТВАМИ ИНФОРМАЦИОННЫХ  
КОМПЬЮТЕРНЫХ КОМПЛЕКСОВ:  
ЗАДАЧА  
«ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ В СИСТЕМАХ  
УПРАВЛЕНИЯ НА ТРАНСПОРТЕ»

*Одобрено научно-методическим советом механического факультета  
в качестве учебно-методического пособия*

Гомель 2012

УДК 004 (075.8)  
ББК 32.81  
М62

Рецензент – *П.Л. Чечет*, канд. техн. наук, доцент кафедры АСОИ  
УО «ГГУ им. Ф. Скорины»

**Миняйлова, Е. Л.**

М62 Решение инженерных задач средствами информационных компьютерных комплексов: задача «Логические элементы в системах управления на транспорте»: учеб.-метод. пособие / Е. Л. Миняйлова ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2012. – 37 с.  
ISBN 978-985-554-145-6

Приведены методические указания, справочные материалы, краткие теоретические сведения, примеры и порядок оформления курсовой работы по дисциплине «Информатика».

Предназначено для студентов механического факультета I курса.

**УДК 004 (075.8)**  
**ББК 32.81**

**ISBN 978-985-554-145-6**

© Миняйлова Е. Л., 2012  
© Оформление. УО «БелГУТ», 2012

## ОГЛАВЛЕНИЕ

Введение.....	4
1 Теоретический минимум .....	4
1.1. Электромеханические устройства в схемах автоматического управления.....	4
1.2. Особенности записи логических выражений.....	7
1.3. Основные аксиомы и законы алгебры логики .....	8
1.4. Примеры упрощения выражений алгебры логики .....	9
2 Пример условия и решения задачи в общем виде .....	11
3 Решение задачи на языке программирования PascalABC.....	13
3.1. Решение с использованием побитовых операций как логических .....	13
3.2. Решение с использованием логических операций .....	16
3.3. Работа с типизированными файлами.....	17
4 Решение задачи в математическом пакете MATHCAD .....	18
5 Решение задачи в табличном процессоре .....	22
6 Стилевое оформление документа в текстовом процессоре .....	23
6.1. Основные понятия .....	23
6.2. Структура и оформление курсовой работы .....	24
6.3. Создание шаблона и документа для курсовой работы.....	26
6.4. Создание стилей для курсовой работы .....	27
6.5. Автоматическое создание оглавлений .....	33
7 Условия заданий .....	34
7.1. Задание «Шагающий конвейер» .....	34
7.2. Задание «Электромеханический привод» .....	35
Список использованной и рекомендуемой литературы .....	37

## **ВВЕДЕНИЕ**

Цель курсовой работы – закрепление умений записи и вычисления логических выражений на материале логических элементов в системах управления на транспорте.

Управление разнообразными средствами механизации производства может быть автоматизировано. В таком случае технологические операции будут выполняться без непосредственного участия людей. Для описания системы управления такими технологическими операциями используют математический аппарат алгебры логики или булевой алгебры. Возможность применения булевой алгебры для решения инженерных задач обусловлена аналогией понятий этой алгебры и двоичной системы счисления.

Курсовая работа является как углублением основных понятий дисциплины «Информатика», так и пропедевтикой изучения тем «Автоматическое управление машинами циклического действия», «Формирование таблицы функций неисправностей объектов диагностирования с использованием логических диагностических моделей» и др.

Разработка содержит задание, методические указания, справочные материалы, краткие теоретические сведения, примеры и порядок оформления курсовой работы.

## **1 ТЕОРЕТИЧЕСКИЙ МИНИМУМ**

### **1.1 Электромеханические устройства в схемах автоматического управления**

Области применения электромеханических устройств разнообразны: от миниатюрных датчиков в навигационных устройствах до крупных генераторов переменного тока в энергосистемах. Учитывая широкое многообразие таких устройств, рассмотрим их общие принципы, а не особенности отдельных конструкций.

Электромеханические устройства применяются в схемах автоматического управления благодаря конструктивной простоте, надежности в работе, высокой ремонтпригодности, отсутствию требовательности к квалификации обслуживающего персонала. Общими свойствами таких аппаратов являются изменение взаимного расположения отдельных частей при подаче

электрического тока в обмотку или переключение электрических цепей под действием приложенных механических усилий.

В устройствах применяются токовые или тепловые контактные устройства, которые переключают свои контакты, например, под воздействием температуры.

Рассмотрим замыкающий и размыкающий контакты (рисунок 1).

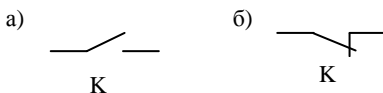


Рисунок 1 – Контакты: *а* – замыкающий; *б* – размыкающий

В схемах управления автоматическим устройством используется конструкция «ЕСЛИ ... ТО ...». Например, «ЕСЛИ *пассажир оплатил проезд*, ТО загорается лампочка».

Условие, стоящее после слова «ЕСЛИ», может быть сложным и состоять из нескольких частей. Например, «ЕСЛИ *стебли льна пойдут длиннее и их вершины окажутся над щупом*, ТО этот щуп замкнет управляемый им контакт».

Действий, следующих после слова «ТО», также может быть несколько: «ЕСЛИ *скорость движения троллейбуса во время торможения становится ниже 7 км/ч*, ТО тормозной ток в силовой цепи уменьшается до 10–15 А, и реле минимального тока срабатывает (отключается)».

Сложные условия можно реализовать с использованием логических элементов, изготовленных в виде микросхем. Логические элементы могут выполнять функции «И», «ИЛИ», «НЕ», которые являются простейшими функциями алгебры логики (булевой алгебры). Применяются также две разновидности логических элементов: «И–НЕ», «ИЛИ–НЕ». В данной работе рассмотрим только основные логические элементы.

Приведем условное графическое обозначение устройств, применяющихся в схемах автоматического управления, таблицы истинности, характеризующие зависимость выходных сигналов *Z* от сигналов на входах *X* и *Y* логических элементов, а также группы контактов с буквенно-цифровым позиционным обозначением.

Схемы изображают таким образом, что их входы располагаются слева или сверху, а выходы – соответственно справа или снизу. Входы одних элементов могут подключаться к выходам других, что позволяет реализовывать логические зависимости любой степени сложности.

**Элемент «И»** выполняет операцию логического умножения, которую называют операцией конъюнкции (рисунок 2, *а*). Из таблицы истинности (рисунок 2, *б*) следует, что выходной сигнал принимает значение 1 только при подаче на оба входа 1. Во всех остальных случаях на выходе будет 0.

Эквивалентом элемента «И» является последовательное соединение замыкающих контактов (рисунок 2, в).

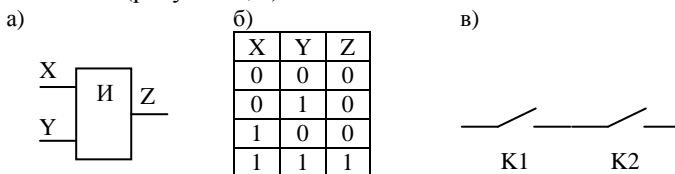


Рисунок 2 – Элемент «И»: а – условное графическое обозначение; б – таблица истинности; в – последовательное соединение замыкающих контактов

**Элемент «ИЛИ»** выполняет операцию логического сложения, которую называют операцией дизъюнкции (рисунок 3, а). Из таблицы истинности (рисунок 3, б) следует, что выходной сигнал равен 1, если хотя бы на один из входов подана 1. Эквивалентом элемента «ИЛИ» является параллельное соединение замыкающих контактов (рисунок 3, в).

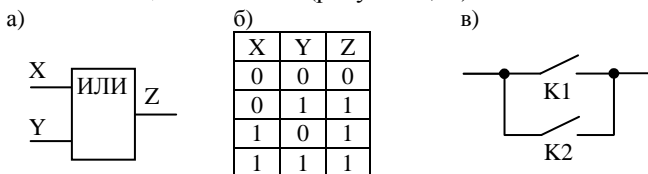


Рисунок 3 – Элемент «ИЛИ»: а – условное графическое обозначение; б – таблица истинности; в – параллельное соединение замыкающих контактов

Условно входные и выходные сигналы обозначают двоичными цифрами, которые принимают значения 1 или 0.

**Элемент «НЕ»** выполняет операцию логического отрицания, которую называют операцией инверсии (рисунок 4, а). В отличие от остальных логических элементов он имеет только один вход. Из таблицы истинности (рисунок 4, б) следует, что выходной сигнал всегда принимает значение, противоположное значению входному сигналу. Эквивалентом элемента «НЕ» является размыкающий контакт (рисунок 4, в).

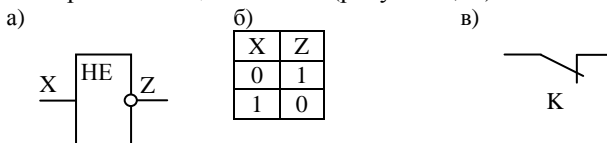


Рисунок 4 – Элемент «НЕ»: а – условное графическое обозначение; б – таблица истинности; в – размыкающий контакт

Логические элементы имеют от двух до четырех входов. Действие элементов с тремя и четырьмя входами аналогично действию элементов с двумя входами, реализующими ту же самую логическую функцию.

Например, элемент «3 И» имеет на выходе 1, если все три входных сигнала тоже 1 (рисунок 5).

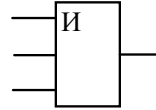


Рисунок 5 – Элемент «3 И»

## 1.2 Особенности записи логических выражений

Операции конъюнкции, дизъюнкции, инверсии могут быть записаны:

- соответствующими русскими словами (и, или, не);
- английскими словами (and, or, not);
- специальными знаками (&, |, ~);
- знаками операций булевой алгебры ( $\wedge$ ,  $\vee$ ,  $\neg$ );
- знаками умножения “ $\cdot$ ”, сложения “+”, инверсии. Инверсия

обозначается чертой над буквенным обозначением. Например,  $\overline{A}$ ,  $\overline{A \cdot B}$ .

Также отрицание (not) обозначается восклицательным знаком !; конъюнкция (and) – знаком & (амперсанд); дизъюнкция (or) – знаком | (вертикальная черта); исключаящее ИЛИ (xor) – знаком ^ (каре).

Результаты выполнения этих операций над переменными A и B логического типа приведены в таблице истинности (таблица 1).

Таблица 1 – Таблица истинности

A	B	not A	A and B	A or B
false	false		false	false
false	true	true	false	true
true	false		false	true
true	true	false	true	true

**Пример 1.** Рассмотрим фрагмент схемы, состоящей из группы контактов (рисунок 6).

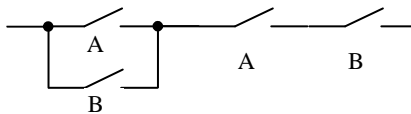


Рисунок 6 – Фрагмент схемы

Опишем данную схему с помощью логических выражений:

$$(A + B) \cdot (A \cdot B).$$

Логические операции могут записываться так:

$$(A \text{ или } B) \text{ и } (A \text{ и } B), \quad (A \text{ or } B) \text{ and } (A \text{ and } B), \quad (A | B) \& (A \& B).$$



Правильным также будет следующее выражение:  $(A \vee B) \wedge (A \wedge B)$ , где,  $\wedge$  – конъюнкция,  $\vee$  – дизъюнкция,  $\neg$  – инверсия.

При вычислении значения логического выражения (формулы) логические операции вычисляются в определенном порядке, согласно их приоритету:

- 1) инверсия;
- 2) конъюнкция;
- 3) дизъюнкция.

Операции одного приоритета выполняются слева направо. Для изменения порядка действий используются скобки.

Рассматриваемое выражение  $(A + B) \cdot (A \cdot B)$  можно сократить. Также можно будет сократить и схему. Для этого выпишем основные аксиомы и законы алгебры логики, которые можно использовать для сокращения логического выражения.

### 1.3 Основные аксиомы и законы алгебры логики

*Аксиомы:*

$$\begin{aligned}
 A \cdot 0 = 0, \quad A + 0 = A & \quad \text{– нулевого множества;} \\
 A \cdot 1 = A, \quad A + 1 = 1 & \quad \text{– универсального множества;} \\
 A \cdot A = A, \quad A + A = A & \quad \text{– повторения (тавтологии);} \\
 A \cdot \bar{A} = 0, \quad A + \bar{A} = 1 & \quad \text{– дополненности;} \\
 = & \\
 A = A & \quad \text{– двойной инверсии (двойного отрицания).}
 \end{aligned}$$

*Законы:*

коммутативные (переместительные)

$$A \cdot B = B \cdot A, \quad A + B = B + A;$$

ассоциативные (сочетательные)

$$\begin{aligned}
 A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C, \\
 A + (B + C) = (A + B) + C = A + B + C;
 \end{aligned}$$

поглощения

$$A \cdot (A + B) = A, \quad A + A \cdot B = A;$$

дистрибутивные (распределительные)

$$\begin{aligned}
 A \cdot (B + C) = A \cdot B + A \cdot C, \\
 (A + B) \cdot (A + C) = A + B \cdot C;
 \end{aligned}$$

склеивания полного

$$A \cdot B + A \cdot \overline{B} = A,$$

$$(A + B) \cdot (A + \overline{B}) = A;$$

склеивания неполного

$$A \cdot B + A \cdot \overline{B} = A + A \cdot B + A \cdot \overline{B},$$

$$(A + B) \cdot (A + \overline{B}) = A \cdot (A + B) \cdot (A + \overline{B});$$

склеивания обобщенного

$$A \cdot B + \overline{A} \cdot C + B \cdot C = A \cdot B + \overline{A} \cdot C,$$

$$(A + B) \cdot (\overline{A} + C) \cdot (B + C) = (A + B) \cdot (\overline{A} + C);$$

расширения

$$A \cdot B + \overline{A} \cdot C = A \cdot B + \overline{A} \cdot C + B \cdot C,$$

$$(A + B) \cdot (\overline{A} + C) = (A + B) \cdot (\overline{A} + C) \cdot (B + C);$$

де Моргана (законы инверсий)

$$\overline{A \cdot B} = \overline{A} + \overline{B},$$

$$\overline{A + B} = \overline{A} \cdot \overline{B};$$

Блейка – Порецкого

$$A + (\overline{A} \cdot B) = A + B,$$

$$A \cdot (\overline{A} + B) = A \cdot B.$$

#### 1.4 Примеры упрощения выражений алгебры логики

**Пример 1.** Проведем упрощение исходного выражения  $(A + B) \cdot (A \cdot B)$ .

Последовательно применяем:

$(A + B) \cdot A \cdot B$  – дистрибутивный и коммутативный законы;

$A \cdot A \cdot B + A \cdot B \cdot B$  – аксиому повторения;

$A \cdot B + A \cdot B \cdot B$  – аксиому повторения;

$A \cdot B + A \cdot B$  – аксиому повторения;

$A \cdot B$ .

Ответ после упрощения:  $A \cdot B$ .

**Пример 2.** Упростить выражение алгебры логики:

$$C \cdot D + \overline{D}.$$

Последовательно применяем:

$C \cdot D + 1 \cdot \overline{D}$  – аксиому универсального множества;

$C \cdot D + (C + 1) \cdot \overline{D}$  – дистрибутивный закон;

$C \cdot D + C \cdot \overline{D} + \overline{D}$  – дистрибутивный закон;

$C \cdot (D + \overline{D}) + \overline{D}$  – аксиому дополнительности;

$C + \overline{D}$ .

Ответ после упрощения:  $C + \overline{D}$ .

**Пример 3.** Упростить выражение алгебры логики:

$(A + B) \cdot \overline{C} \cdot B + C \cdot D + C$ .

Последовательно применяем:

$(A + B) \cdot \overline{C} \cdot B + C \cdot D + C$  – дистрибутивный закон;

$A \cdot \overline{C} \cdot B + \overline{C} \cdot B \cdot B + C \cdot D + C$  – аксиому повторения;

$A \cdot \overline{C} \cdot B + \overline{C} \cdot B + C \cdot D + C$  – аксиому универсального множества;

$A \cdot \overline{C} \cdot B + \overline{C} \cdot B + C$  – дистрибутивный закон;

$\overline{C} \cdot B \cdot (A + 1) + C$  – аксиому универсального множества;

$\overline{C} \cdot B + C$  – закон Блейка – Порецкого;

$B + C$ .

Ответ после упрощения:  $B + C$ .

**Пример 4.** Упростить выражение алгебры логики:

$B \cdot C + \overline{C}$ .

Последовательно применяем:

$B \cdot C + \overline{C}$  – аксиому универсального множества;

$B \cdot C + \overline{C} \cdot (B + 1)$  – дистрибутивный закон;

$B \cdot C + \overline{C} \cdot B + \overline{C}$  – дистрибутивный закон;

$B \cdot (C + \overline{C}) + \overline{C}$  – аксиому дополнительности;

$B + \overline{C}$ .

Ответ после упрощения:  $B + \overline{C}$ .

Правильность выполнения преобразований можно проверить путем сопоставления таблиц истинности исходной и преобразованной функции алгебры логики. Метод подбора охватывает все возможные комбинации (наборы) значений логических переменных; ведется в табличной форме путем построения и сравнения таблиц истинности (таблица 2). При  $n$  входных и

одной выходной переменной таблица истинности содержит  $2n$  строк и  $n + 1$  столбцов.

Запишем последовательность вычислений для примера 4:

1.  $B \cdot C$ .
2.  $\overline{C}$ .
3.  $B \cdot C + \overline{C}$ .

Последовательность вычислений для сокращенного варианта:

1.  $\overline{C}$ .
2.  $\overline{C} + B$ .

Таблица 2 – Доказательство табличным методом

Переменные			Первое выражение		Второе выражение
1	2	3	4	5	6
$B$	$C$	$\overline{C}$	$B \cdot C$	$B \cdot C + \overline{C}$	$\overline{C} + B$
0	0	1	0	1	1
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	1	1	1

Значения, указанные в пятой и шестой графах таблицы, совпадают, что доказывает правильность упрощения заданного выражения алгебры логики.

## 2 ПРИМЕР УСЛОВИЯ И РЕШЕНИЯ ЗАДАЧИ В ОБЩЕМ ВИДЕ

**Задание «Шагающий конвейер».** Дан фрагмент схемы управления шагающим конвейером с двумя рабочими позициями (только контактная группа), записанный в виде логического выражения. По множеству указанных сигналов в конкретный момент времени определить наименование выполняемой производственной операции.

Сигнал 0 на выходе фрагмента схемы соответствует выполнению операции очистки вагона от загрязнений механической щеткой, сигнал 1 – обмывки вагона.

Содержание работы.

1 Упростить заданное выражение с использованием аксиом и законов алгебры логики.

2 Изобразить в тетради контактную группу полученной сокращенной схемы. Предоставить работу преподавателю на проверку.

3 Графические построения выполнить средствами:

- языка программирования Паскаль;
- математического пакета MathCAD.

4 Выполнить вычисление прохождения сигнала по заданному и сокращенному выражениям. Вычисления произвести средствами:

- языка программирования Паскаль;
- математического пакета MathCAD;
- электронного процессора Excel.

5 Доказать правильность сокращения логического выражения табличным методом в электронном процессоре Excel.

6 Отчет о выполненной работе с предоставлением текстов программ и скриншотов результатов работы оформить в текстовом процессоре с помощью стилей.

Решение.

Провести упрощение выражения с использованием законов алгебры логики (см. разд. 1):

$$A \cdot A + B + A,$$

$$A + B + A,$$

$$A + B.$$

Ответ после упрощения:  $A + B$ .

Выполнить вычисления прохождения сигнала непосредственно по заданному выражению (см. разд. 3–5).

Изобразить контактную группу полученного ответа с буквенно-цифровым позиционным обозначением. Размер стороны клетки равен 20 пикселей (рисунок 7).

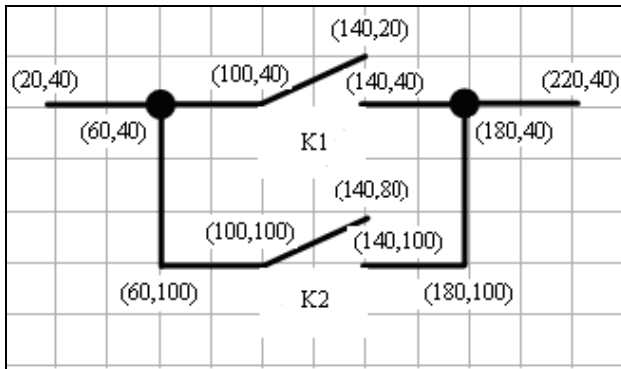


Рисунок 7 – Схематическое изображение сокращенного выражения  $A+B$  (скриншот)

Выполнить вычисления прохождения сигнала по сокращенному выражению.

Вычисления по исходному и сокращенному выражениям можно выполнить в одной программе на Паскале, на одном листе математического пакета и электронного процессора (см. разд. 3–5).

Отчет о выполненной работе с предоставлением текстов программ и скриншотов результатов работы оформить в текстовом процессоре с помощью стилей (см. разд. 6).

### 3 РЕШЕНИЕ ЗАДАЧИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PASCALABC

#### 3.1 Решение с использованием побитовых операций как логических

```
Program n1 ;
uses GraphABC;
var
  A, B, P1, P2, P3, P4 : byte;
  input, out : text;
```

*{Функция логического умножения "И". Эквивалентом является последовательное соединение замыкающих контактов K1, K2 }*

```
function konjunk (K1, K2 : byte) : byte;
begin
  konjunk := K1 and K2;
end;
```

*{Функция логического сложения "ИЛИ". Эквивалентом является параллельное соединение замыкающих контактов K1, K2 }*

```
function dizjunk (K1, K2 : byte) : byte;
begin
  dizjunk := K1 or K2;
end;
```

*{ Функция для операции инверсии "НЕ". Эквивалентом элемента "НЕ" является размыкающий контакт.*

*Операции not, and, or в Паскале могут быть логическими или побитовыми операциями. Для упрощения решения в данной работе использовались эти операции как побитовые. Поэтому переменные имеют целый тип, а не логический. Операция not (X-2) дает решение, аналогичное решению с логическим выражением, но с целым типом результата }*

```
function invers(X:byte):byte;
```

```

begin
  invers:= not (X-2);
end;
BEGIN
  assign ( input, 'input.txt' );
  assign ( out, 'output.txt' );
  reset ( input );
  rewrite ( out );
  read ( input, A, B ); {После ввода данных проверяем их!}
  if ( A = 0 ) or ( A = 1 ) or ( B = 0 ) or ( B = 1 ) then
    BEGIN
      { A * A + B + A }
      P1 := konjunk ( A, A );
      P2 := dizjunk ( P1, B );
      P3 := dizjunk ( P2, A );
      writeln ( out, P1, ' ', P2, ' ', P3 );
      if P3=1
        then writeln ( out, ' Выполняется обмывка вагона. ' )
        else writeln ( out, ' Выполняется очистка вагона от загрязнений
механической щеткой. ' );

      { После сокращения остается выражение A+B }
      { Проведем проверку сокращенного варианта }
      P4 := dizjunk ( A, B );
      Writeln ( out, P4 );
      if P4 = P3
        then writeln ( out, ' Решение верно! ' )
        else writeln ( out, ' Решение не верно! ');
    END
  else writeln(out,'Нештатная ситуация! Проверьте входные данные!');
  close ( input );
  close ( out );
  { Изобразим сокращенное выражение A+B.
  Размер стороны клетки равен 20 пикселей (см. рисунок 7) }
  Line ( 20, 40, 100, 40 );
  Line ( 100, 40, 140, 20 );
  Line ( 140, 40, 220, 40 );
  Line ( 60, 40, 60, 100 );
  Line ( 180, 40, 180, 100 );
  Line ( 60, 100, 100, 100 );
  Line ( 140, 100, 180, 100 );
  Line ( 100, 100, 140, 80 );
  TextOut ( 120, 50, ' K1 ' );
  TextOut ( 120, 110, ' K2 ' );
  { TextOut(x,y: integer; s: string); Выводит строку s в позицию (x,y).
  Точка (x,y) задает верхний левый угол прямоугольника,
  который будет содержать текст из строки s. }
END.

```

Программа содержит вычислительную и графическую части. Результаты выполнения этих частей представим на рисунках: графическая часть (рисунок 8), вычислительная часть (рисунки 9, 10).

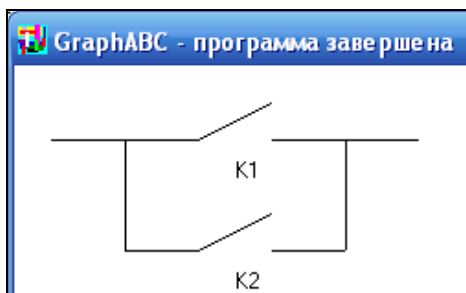


Рисунок 8 – Результат выполнения графической части программы

Так как выражение на рисунке 8 повторяет операцию логического сложения, то приведены обозначения контактов K1, K2. В функциях дизъюнкции (dizjunk) и конъюнкции (konjunk) также используются обозначения контактов K1, K2. В задачах же для самостоятельного решения в рисунках рекомендуется использовать исходные обозначения, такие как A, B, и т.д.

Входной файл input.txt содержит множество сигналов в конкретный момент времени, соответствующих входным данным программы на Паскале (см. рисунок 9).

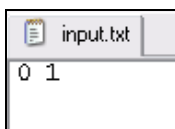


Рисунок 9 – Пример входного файла input.txt, где A=0, B=1

Результаты выполнения программы будут записаны в файл output.txt (см. рисунок 10).

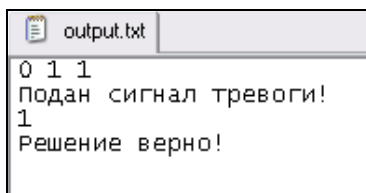


Рисунок 10 – Пример выходного файла output.txt



### 3.2 Решение с использованием логических операций

```
Program n2;
uses GraphABC;
var
  P1, P2, P3, P4 : boolean;
const
  A = false;  B = true;
  { Функция логического умножения "И".
    Эквивалентом является последовательное соединение замыкающих
    контактов K1, K2 }
  function konjunk (K1, K2 : boolean) : boolean;
  begin
    konjunk := K1 and K2;
  end;

  { Функция логического сложения "ИЛИ".
    Эквивалентом является параллельное соединение замыкающих
    контактов K1, K2 }
  function dizjunk (K1, K2 : boolean) : boolean;
  begin
    dizjunk := K1 or K2;
  end;

  { Функция для операции инверсии "НЕ".
    Эквивалентом элемента "НЕ" является размыкающий контакт }
  function invers (X : boolean) : boolean;
  begin
    invers := not (X);
  end;

BEGIN
  // A*A+B+A
  {Первый способ решения задачи}
  P1 := konjunk ( A, A );
  P2 := dizjunk ( P1, B );
  P3 := dizjunk ( P2, A );
  writeln( P1, ' ', P2, ' ', P3 );
  if P3 then writeln( 'Выполняется обмывка вагона.' )
    else writeln( 'Выполняется очистка вагона от загрязнений
    механической щеткой.' );

  { После сокращения остается выражение A+B }
  { Проведем проверку сокращенного варианта }
  P4 := dizjunk ( A, B );
  if P4 = P3
  then writeln ( 'Решение верно!' )
  else writeln ( 'Решение не верно!' );
```

```

    {Второй способ решения задачи}
writeln ( '2-й способ решения ' );
writeln ( 'До сокращения: ', A and A or B or A, 'После сокращения: ', A or B );

```

```

    {Схематически изобразим сокращенный вариант схемы}
line ( 20,40, 100,40 );
line ( 100,40, 140,20 );
line ( 140,40, 220,40 );
line ( 60,40, 60,100 );
line ( 180,40, 180,100 );
line ( 60,100, 100,100 );
line ( 140,100, 180,100 );
line ( 100,100, 140,80 );
TextOut ( 120, 50, 'K1' );
TextOut ( 120,110, 'K2' );
    { TextOut ( x,y : integer; s : string );
      Выводит строку s в позицию ( x, y )
      ( точка ( x, y ) задает верхний левый угол прямоугольника,
        который будет содержать текст из строки s). }
END.

```

Результат выполнения графической части программы (см. рисунок 8) .  
 Результаты выполнения программы приведены на рисунке 11.

```

False True True
Выполняется обмывка вагона.
Решение верно!
2-й способ решения
До сокращения: True После сокращения: True

```

Рисунок 11 – Пример вывода

### 3.3 Работа с типизированными файлами

```

Program n1;
var
  A, B, P1, P2, P3, P4 : boolean;
  input, out : file of boolean;

begin
  {Запись в типизированный файл}
  {Устанавливаем связь файловой переменной с физическим файлом на диске}
  assign ( input, ' input.txt ' );
  rewrite ( input );      {Открываем файл для записи}
  write ( input, true, false ); {Записываем последовательно данные в файл}

```

close ( input );            *{Заканчиваем работу с файлом. Теперь на диске существует файл с именем input.txt, содержащий введенные данные. На этом программу можно закончить, работу с файлом можно продолжить в другой программе, в другое время, но мы продолжим}*

*{Чтение из типизированного файла}*  
*{Устанавливаем связь файловой переменной с физическим файлом на диске}*  
 assign ( input, ' input.txt ' );  
 reset ( input );            *{Открываем файл для чтения}*  
 read ( input, A, B );      *{Читаем данные из файла}*  
 writeln ( A, B );          *{Выводим на экран второй файл}*  
 close ( input );            *{Заканчиваем работу с файлом}*

*{Запись в другой типизированный файл}*  
*{Устанавливаем связь файловой переменной с физическим файлом на диске}*  
 assign ( out, ' output.txt ' );  
 rewrite ( out );            *{Открываем файл для записи}*  
 write ( out, A, B );        *{Записываем последовательно данные в файл}*  
 close ( out );              *{Заканчиваем работу с файлом}*  
 end.

#### 4 РЕШЕНИЕ ЗАДАЧИ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ MATHCAD

Вычислительную часть работы выполним с помощью функций конъюнкции, дизъюнкции и инверсии.

Графическую часть работы выполним через построение графиков функций. Части рисунка, то есть отдельные линии, можно рассматривать как графики функций.

В общем виде отрезки можно построить через уравнение прямой, проходящей через две заданные несовпадающие точки  $(x_1, y_1)$  и  $(x_2, y_2)$ :

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1},$$

или в общем виде

$$(y_1 - y_2) \cdot x + (x_2 - x_1) \cdot y + (x_1 \cdot y_2 - x_2 \cdot y_1) = 0.$$

$$\text{Выразим отсюда } y = \frac{(x_1 \cdot y_2 - x_2 \cdot y_1) + (y_1 - y_2) \cdot x}{x_1 - x_2}.$$

Прямые линии, перпендикулярные осям координат, удобно строить через ранжированную переменную. Например,

$$x11:= 20..100 \quad f(x11):= 40.$$

Решение.

*Вычислительная часть работы.*

*Задаем функции.*

$\text{konjunk}(K1, K2) := K1 \wedge K2$

$\text{dizjunk}(K1, K2) := K1 \vee K2$

$\text{invers}(X) := \neg X$

*Вводим исходные данные.*

A := 0

B := 1

*Последовательно вычисляем операции.*

P1 := konjunk(A, A)

P1 = 0

P2 := dizjunk(P1, B)

P2 = 1

P3 := dizjunk(P2, A)

P3 = 1

if(P3, " Выполняется обмывка вагона. ", " Выполняется очистка вагона от загрязнений механической щеткой. ") = " Выполняется обмывка вагона. "

*Графическая часть работы.*

*Линия 1.*

x11 := 20 .. 100

y11(x11) := 100

*Линия 2.*

x22 := 100 .. 140

x1 := 100

y1 := 100

x2 := 140

y2 := 120

$y22(x22) := \frac{(x1 \cdot y2 - x2 \cdot y1) + (y1 - y2) \cdot x22}{x1 - x2}$

*Линия 3.*

x33 := 140 .. 220

y33(x33) := 100

*Линия 4.*

x44 := 60

y44 := 40 .. 100

Линия 5.

$$x55 := 180$$

$$y55 := 40.. 100$$

Линия 6.

$$x66 := 60.. 100$$

$$x1 := 60$$

$$y1 := 40$$

$$x2 := 100$$

$$y2 := 40$$

$$y66(x66) := \frac{(x1 \cdot y2 - x2 \cdot y1) + (y1 - y2) \cdot x66}{x1 - x2}$$

Линия 7.

$$x77 := 140.. 180$$

$$y77(x77) := 40$$

Линия 8.

$$x88 := 100.. 140$$

$$x1 := 100$$

$$y1 := 40$$

$$x2 := 140$$

$$y2 := 60$$

$$y88(x88) := \frac{(x1 \cdot y2 - x2 \cdot y1) + (y1 - y2) \cdot x88}{x1 - x2}$$

В математическом пакете оси координат соответствуют математическим представлениям. Поэтому предыдущий график (см. рисунок 7) следует изобразить с уточнением координат по оси  $y$  (рисунок 12).

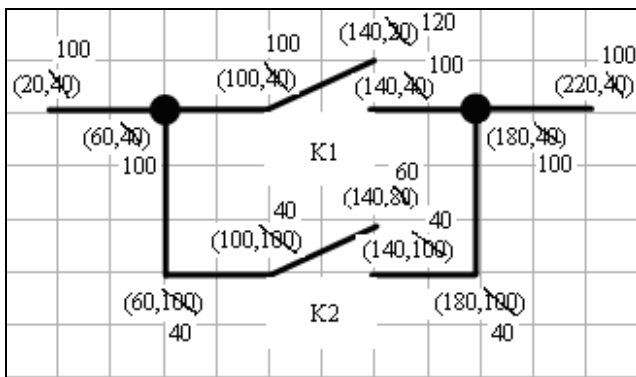


Рисунок 12 – Схематическое изображение сокращенного выражения  $A+B$  (скриншот)

Решение содержит вычислительную и графическую части. Результаты вычислительной части работы находятся непосредственно в решении. Графическая часть представлена на рисунке 13.

Двумерные графики строятся самим MathCAD на основании обработанных данных. Для отображения заданных функций  $y_{11}(x_{11})$ ,  $y_{22}(x_{22})$ ,  $y_{33}(x_{22})$ ,  $y_{44}$ ,  $y_{55}$ ,  $y_{66}(x_{66})$ ,  $y_{77}(x_{77})$ ,  $y_{88}(x_{88})$  требуется выполнить следующие действия:

1 Дать команду **Вставка (Insert) ⇒График (Graph) ⇒X-Y Зависимость (X-Y Plot)** или нажать комбинацию клавиш **Shift + @**, или щелкнуть



кнопку панели **Графики**. Появится шаблон декартового графика.

2 Ввести в средней метке под осью  $X$  имя аргумента, в средней метке слева от вертикальной оси  $Y$  – имя функции. Если аргументов или имен функций несколько (для построения сразу нескольких графиков в одном шаблоне), то их вводят через запятую.

3 Щелкнуть вне шаблона графика – для заданного диапазона изменения аргумента график будет построен.

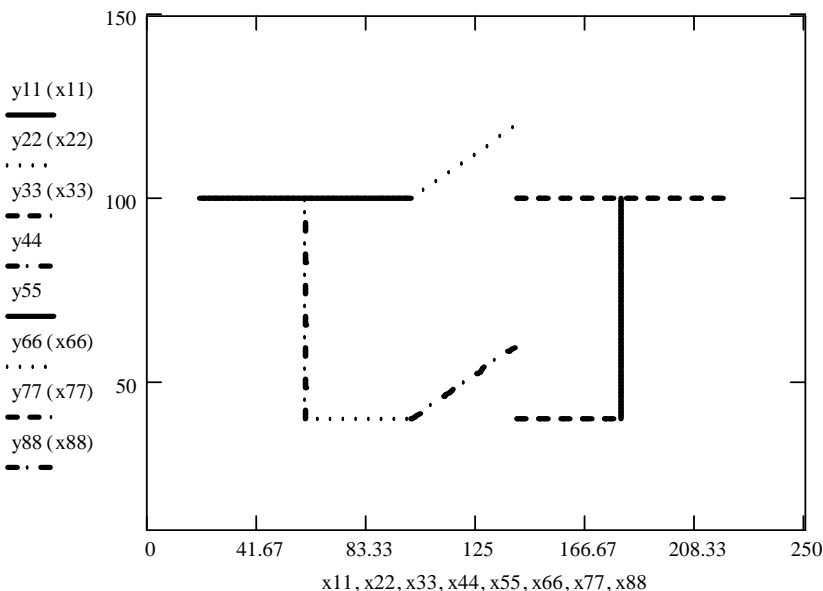


Рисунок 13 – Результат выполнения графической части решения задачи

## 5 РЕШЕНИЕ ЗАДАЧИ В ТАБЛИЧНОМ ПРОЦЕССОРЕ

Приведем решение задачи в табличном процессоре (рисунок 14).

	A	B	C	D	E
1					
2	Исходные данные				
3	A	B			
4	0	1			
5	Формула для вычисления				
6	=A4*A4+B4+A4				
7	Результат вычисления				
8	1				
9	Подан сигнал тревоги				
10	Формула для вычисления				
11	=ЕСЛИ(AВ=1;"Подан сигнал тревоги";"Все спокойно")				

Рисунок 14 – Вычислительная часть работы

Также следует привести доказательство правильности сокращения логического выражения табличным методом (рисунок 15).

	A	B	C	D	E	F	G	H
1								
2	N строки вычис- лений	Переменные			Первое выражение		Второе выражение	Проверка на совпадение
3		1	2	3	4	5	6	7
4		B	C	$\bar{C}$	$B \cdot C$	$B \cdot C + \bar{C}$	$\bar{C} + B$	
5	1	0	0	ИСТИНА	ЛОЖЬ	ИСТИНА	ИСТИНА	ИСТИНА
6	2	0	1	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ИСТИНА
7	3	1	0	ИСТИНА	ЛОЖЬ	ИСТИНА	ИСТИНА	ИСТИНА
8	4	1	1	ЛОЖЬ	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА
9						Всего истин:		4
10						Ответ:		Верно

Рисунок 15 – Доказательство табличным методом

Значения, указанные в пятой и шестой графах таблицы, совпадают, что доказывает правильность упрощения заданного выражения алгебры логики.

Кроме результатов вычисления в доказательстве табличным методом надо показать формулы вычислений (рисунок 16). Отразить формулы можно так: **Сервис – Параметры – Вид – Формулы.**

	A	B	C	D	E	F	G	H
1								
2	N ст ро ки	Переменные			Первое выражение		Второе выражение	Проверка на совпадение
		1	2	3	4	5	6	
3								
4		$B$	$C$	$\bar{C}$	$B \cdot C$	$B \cdot C + \bar{C}$	$\bar{C} + B$	
5	1	0	0	=НЕ(C5)	=И(B5;C5)	=ИЛИ(E5;D5)	=ИЛИ(D5;B5)	=ЕСЛИ(F5=G5;ИСТИНА;ЛОЖЬ)
6	2	0	1	=НЕ(C6)	=И(B6;C6)	=ИЛИ(E6;D6)	=ИЛИ(D6;B6)	=ЕСЛИ(F6=G6;ИСТИНА;ЛОЖЬ)
7	3	1	0	=НЕ(C7)	=И(B7;C7)	=ИЛИ(E7;D7)	=ИЛИ(D7;B7)	=ЕСЛИ(F7=G7;ИСТИНА;ЛОЖЬ)
8	4	1	1	=НЕ(C8)	=И(B8;C8)	=ИЛИ(E8;D8)	=ИЛИ(D8;B8)	=ЕСЛИ(F8=G8;ИСТИНА;ЛОЖЬ)
9						Всего истин:		=СЧЁТЕСЛИ(H5:H8;"=ИСТИНА")
10						Ответ:		=ЕСЛИ(A8=H9;"Верно","Упрощение НЕ ВЕРНО!")

Рисунок 16 – Формулы в доказательстве табличным методом

## 6 СТИЛЕВОЕ ОФОРМЛЕНИЕ ДОКУМЕНТА В ТЕКСТОВОМ ПРОЦЕССОРЕ

### 6.1 Основные понятия

Для правильного оформления документа в тексте следует выделить логическую структуру и уточнить смысловое назначение элементов текста, которые необходимо зрительно выделять.

**Логическая структура текста** получается разделением его на главы, разделы, подразделы, пункты, абзацы текста.

Разделы, подразделы, пункты и подпункты нумеруют арабскими цифрами с точками и записываются с нового абзаца. В конце заголовков глав, разделов и подразделов точку не ставят. Если заголовок состоит из двух или более предложений, их разделяют точкой (точками). В конце заголовка пункта ставят точку.

Номер подраздела включает номера раздела и подраздела, разделенные точкой (1.1, 1.2, 1.3 и т.д.). Номер пункта – номера раздела, подраздела и пункта, разделенные точками (1.1.1, 1.1.2, 1.1.3 и т.д.). Номер подпункта – номера раздела, подраздела, пункта и подпункта, разделенные точками (1.2.3.4, 1.2.3.5 и т.д.).

Разделы должны иметь заголовки. Подразделы при необходимости могут иметь заголовки. В заголовках не допускается перенос слов, применение римских цифр, математических знаков и греческих букв. Точка в конце заголовка не ставится.

**Назначение элементов текста** указывает на их роль в тексте: определение термина, формула, заголовок таблицы, подрисуночная подпись. А за




внешний вид этих элементов текста в текстовом процессоре будет отвечать стилю. Например, определение должно быть выделено полужирным начертанием шрифта, а формула размещена с выравниванием по центру.


Таким образом, смысловое назначение элементов текста является основанием для назначения стиля.

**Стиль** – это инструмент, помогающий быстро форматировать документ. Под стилем понимают специально созданные наборы форматирования, позволяющие использовать одновременно несколько атрибутов. Стилям присваивают имена.

Различают два типа стилей: абзаца и символа.

**Стиль абзаца** применяется целиком к абзацу. Этот стиль определяет как тип и размер шрифта, так и размещение абзаца, интервалы для всего текста.

В области задач стиль абзаца помечается значком .

**Стиль символа (знака)** применяется в дополнение к стилю абзаца к отдельному символу или последовательности символов. Стили знаков помечаются в области задач помещенным рядом значком .

**Шаблон** – это файл с расширением dot, на основе которого создаются документы. В шаблон сохранены необходимые стили.

Посмотреть с каким шаблоном вы сейчас работаете можно так: «Сервис – Шаблоны и надстройки» (рисунок 17).

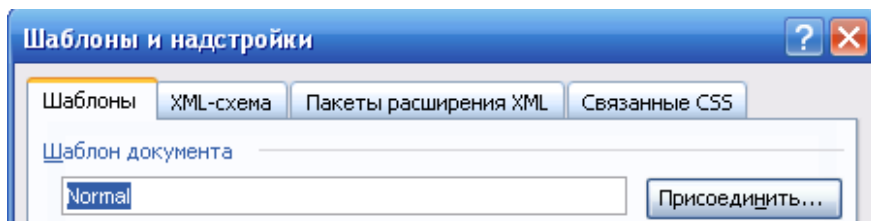


Рисунок 17 – Фрагмент диалогового окна «Шаблоны и настройки»

## 6.2 Структура и оформление курсовой работы

Курсовая работа должна содержать следующие структурные части:

- **титульный лист** – является первым листом работы, но номер на нем не ставится;
- **задание, выданное преподавателем**, – в нумерацию листов не включается, если лист задания выдан преподавателем в распечатанном виде;
- **содержание** – должно быть создано автоматически на основе стилей типа «Заголовки»;
- **перечень условных обозначений** (при необходимости);

- **введение.** Страницы нумеруют в центре нижней части листа. Во введении говорится об актуальности работы, какую цель она преследует и какие задачи решаются для достижения этой цели. Объем введения 1–1,5 с.;

- **основную часть**, разбитую на разделы. В первом разделе задача решается средствами языка PascalABC. Во втором разделе эта же задача решается в среде MathCAD. В третьем разделе задача решается средствами пакета Excel. Расчеты в Excel представляют в обычном режиме и режиме отображения формул. Окончательно работа оформляется средствами текстового процессора Word. Все созданные ранее документы объединяют в единый документ. Рисунки выполняют с помощью векторной графики;

- **заключение.** Приводят выводы из работы, а не общие слова. Фактически дают ответы на те задачи, которые были поставлены во введении. Объем 1–2 с.;

- **список использованных источников.** Список должен включать около 10 наименований печатных и электронных источников. Список формируют в алфавитном порядке фамилий первых авторов и (или) заглавий.

Пример оформления списка.

1 Гораев, О. П. Информатика. Математические и физические основы ЭВМ : учеб. пособие / О. П. Гораев, Т. Н. Модина. – Гомель : БелГУТ, 2007. – 31 с.

2 Миняйлова, Е. Л. Создание презентаций в среде PowerPoint / Е. Л. Миняйлова. – Гомель : ГГОИПК и ПРР и СО, 2003. – 28 с.

3 Национальный Интернет-портал Республики Беларусь [Электронный ресурс] / Нац. центр правовой информ. Респ. Беларусь. – Минск, 2005. – Режим доступа : <http://www.pravo.by>. – Дата доступа : 25.01.2006.

- **приложения** (при необходимости).

Каждую структурную часть следует начинать с нового листа.

К защите работы необходимо представить в электронном виде файлы с именем:

**Группа\_Фамилия\_Имя\_Курсовая.pas**

**Группа\_Фамилия\_Имя\_Курсовая.mcd**

**Группа\_Фамилия\_Имя\_Курсовая.xls**

**KURS.dot**

**Группа\_Фамилия\_Имя\_Курсовая.doc**

В итоговом документе оформление должно быть выполнено с помощью стилей.

### 6.3 Создание шаблона и документа для курсовой работы

В папке «Курсовая работа» создайте новый текстовый документ с именем `Группа_Фамилия_Имя_РГР`, например, `МЭ11_Иванов_Иван_РГР.doc`. Это можно сделать разными способами.

**Способ 1.** На основе шаблона Обычный. По умолчанию документы Word создают на базе обычного шаблона `Normal.dot`

В шаблоне `Normal.dot` сохраняют все стили по умолчанию, поэтому лучше всего оставить его неизменным, а для курсовой работы использовать переименованную копию `Normal.dot`.

**Способ 2.** На основе своего шаблона `KURS.dot`

*Шаг 1.* Создайте свой шаблон `KURS.dot` для курсовой работы с помощью последовательности действий.

- создать новый документ.
- дать команду «**Файл – Сохранить как...**». Найти свою папку, в списке «**Тип файла**» внизу окна выбрать «**Шаблон документа**», ввести имя файла `KURS`, расширение `dot` не менять, сохранить файл (рисунок 18). Вновь созданный шаблон является переименованной копией `Normal.dot`. Дальше можно изменять нужные элементы.

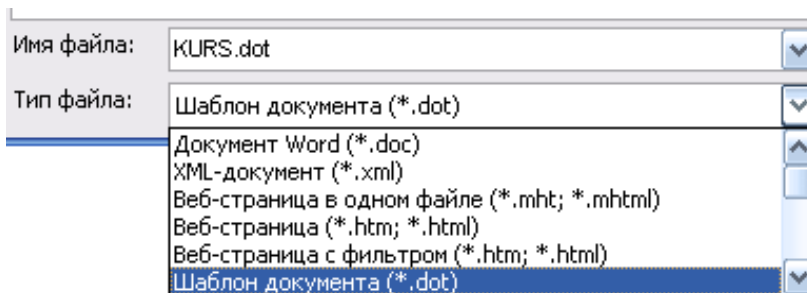


Рисунок 18 – Сохранение нового шаблона

*Шаг 2.* Создайте новый документ на основе шаблона `KURS.dot`. Откройте папку с сохраненным шаблоном. Щелкните двойным щелчком мыши на его имени, при этом откроется не сам шаблон, а новый документ, созданный на его базе.

Сохраните документ с именем `Группа_Фамилия_Имя_Курсовая`, например, `МО11_Иванов_Иван_Курсовая.doc`. Продолжайте работу в этом документе.

Запишите в тетрадь адрес хранения шаблона!

Укажите основные настройки для нового шаблона. Установите параметры страницы (рисунок 19).

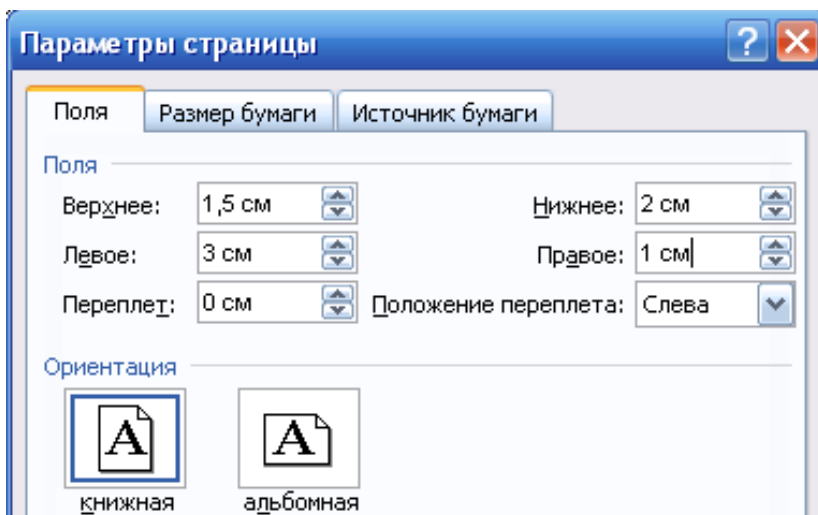


Рисунок 19 – Параметры страницы

#### 6.4 Создание стилей для курсовой работы

Командой **Формат – Стили и форматирование** откройте окно «Стили» (рисунок 20).

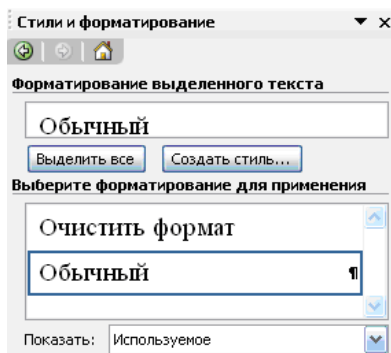


Рисунок 20 – Окно «Стили» на панели «Форматирование»

Настройте стиль абзаца на основе стандартного стиля «Обычный» (рисунки 21, 22).

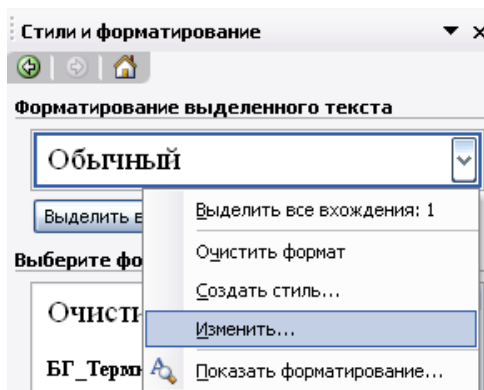


Рисунок 21 – Изменение стиля «Обычный»

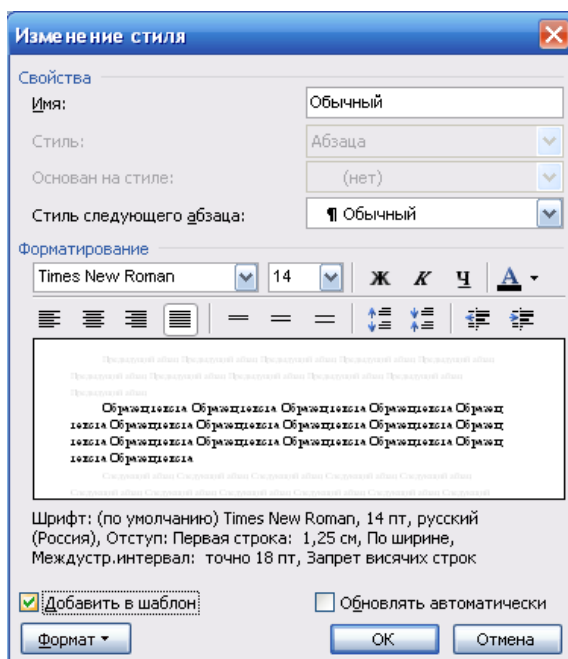


Рисунок 22 – Настройка стиля «Обычный»

Параметры для основного абзаца устанавливаются после нажатия кнопки «Формат – Абзац» в окне «Изменение стиля» (рисунок 23).

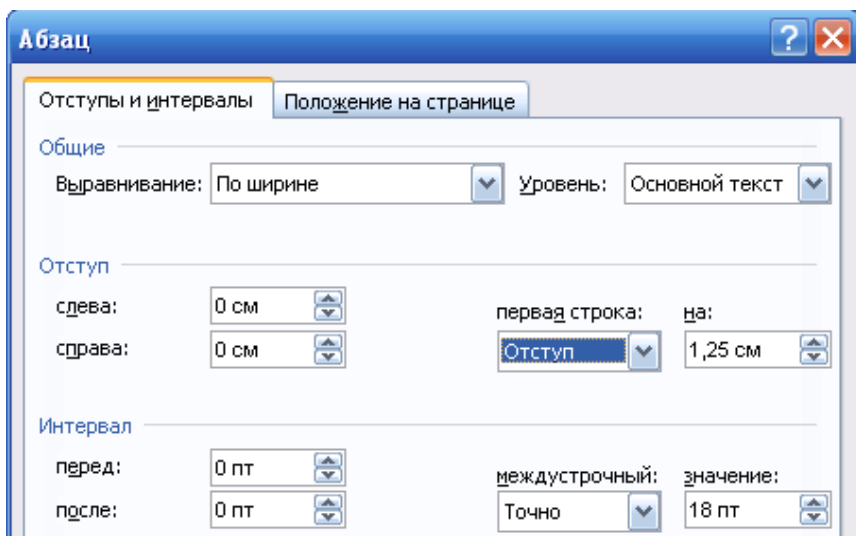


Рисунок 23 – Настройка параметров абзаца

После настройки стандартного стиля «Обычный» можно создать на его основе новые стили. Для этого достаточно щелкнуть кнопку «Создать стиль» (рисунок 24).

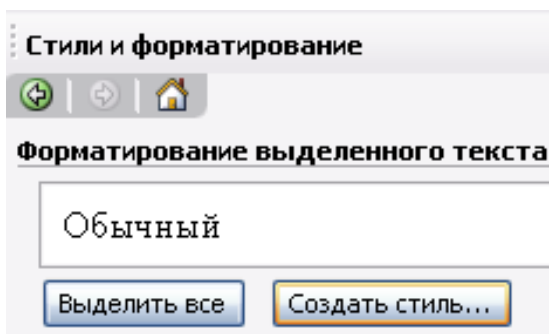


Рисунок 24 – Создание нового стиля

При создании нового стиля требуется указать его имя (например, К\_Министерство), тип стиля (абзац или знак), на каком стиле основан новый стиль (например, на стиле «Обычный»), стиль следующего абзаца. Также требуется указать все элементы форматирования: шрифт, абзац, границы. Новый стиль рекомендуется добавить в шаблон (рисунок 25).

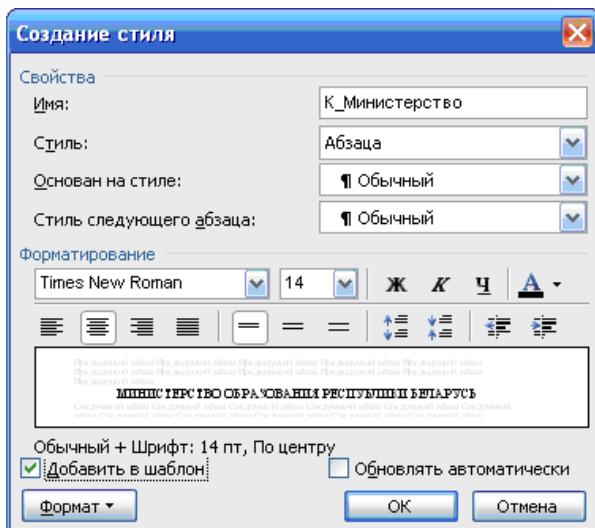


Рисунок 25 – Создание стиля К\_Министерство на основе стиля «Обычный»

Выпишем какие стили понадобятся для титульного листа (таблица 3).

Таблица 3 – Форматирование титульного листа

Имя стиля	Тип стиля	Основан на стиле	Стиль следующего абзаца	Отличие форматирования от стиля «Обычный»
К_Министерство	Абзац	Обычный	Обычный	<b>Абзац:</b> Выравнивание – по центру. Уровень – основной текст. Отступ: слева – 0, справа – 0, первая строка – отступа нет, интервал – перед 0, после – 30, междустрочный – точно 18 пт
К_Учреждение	Абзац	Обычный	Обычный	<b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет
К_Кафедра	Абзац	Обычный	Обычный	<b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет, интервал – перед 24, после – 0
К_Курсовая	Абзац	Обычный	Обычный	<b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет, интервал – перед 102, после – 0
К_Тема	Абзац	Обычный	Обычный	<b>Шрифт:</b> Размер – 20. <b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет, интервал – перед 30, после – 132

Окончание таблицы 3

Имя стиля	Тип стиля	Основан на стиле	Стиль следующего абзаца	Отличие форматирования от стиля «Обычный»
К_Выполнил	Абзац	Обычный	Обычный	<b>Шрифт:</b> Размер – 14. <b>Абзац:</b> Выравнивание – по ширине, первая строка – отступа нет, интервал – перед 0, после – 0, Поз.табуляции: 10,75 см, по левому краю
К_Гомель	Абзац	Обычный	Обычный	<b>Шрифт:</b> Размер – 14. <b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет, интервал – перед 150, после – 0

После выполнения работ по созданию стилей для титульного листа и набора соответствующего текста должно получиться так, как изображено на рисунке 26.

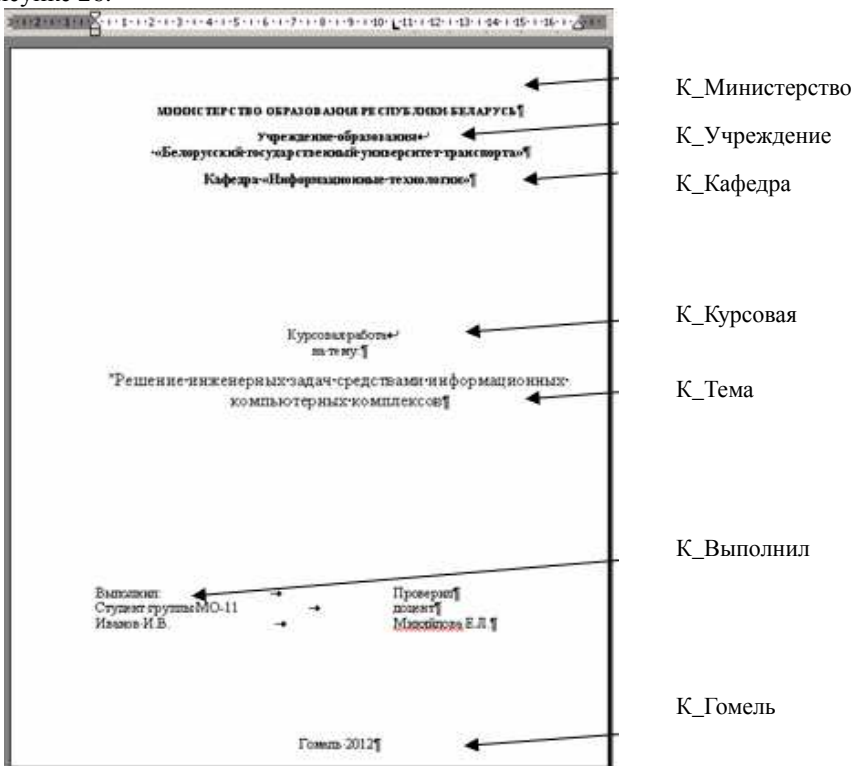


Рисунок 26 – Стилевое оформление титульного листа



Как видно из пояснений, стиль «Обычный» не применяется при наборе текста. Он используется как основа для построения стилей.

Для оформления текста курсовой работы могут понадобиться стили, приведенные в таблице 4.

Таблица 4 – Форматирование текста курсовой работы

Имя стиля	Тип стиля	Основан на стиле	Стиль следующего абзаца	Отличие форматирования от стилей «Обычный» и «Заголовок»
К_Главный	Абзац	Обычный	К_Главный	
К_Подрисуночный	Абзац	Обычный	К_Главный	<b>Шрифт:</b> Размер – 12. <b>Абзац:</b> Выравнивание – по центру. Уровень – основной текст Отступ: слева – 0, справа – 0, первая строка – отступа нет, интервал – перед 6, после – 4, не разрывать абзац
К_Рисунок	Абзац	Обычный	К_Подрисуночный	<b>Абзац:</b> Выравнивание – по центру, первая строка – отступа нет, не отрывать от следующего
К_Термин	Знак	Обычный	К_Главный	<b>Шрифт:</b> полужирный
К_Решение	Знак	Обычный	К_Главный	<b>Шрифт:</b> полужирный, курсив
К_Паскаль	Абзац	Обычный	К_Паскаль	<b>Шрифт:</b> Verdana, английский (США), размер – 10. <b>Абзац:</b> первая строка – отступа нет
К_Содержание	Абзац	Обычный	К_Главный	<b>Абзац:</b> Выравнивание – по центру. Уровень – основной текст. Отступ: первая строка – отступа нет, интервал – перед 6, после – 4, не разрывать абзац
К_Список нумерованный	Абзац	Обычный	К_Список нумерованный	Создать список, а на его основе создать стиль
К_Список маркированный	Абзац	Обычный	К_Список маркированный	Создать список, а на его основе создать стиль
К_Табл_Название	Абзац	Обычный	К_Табл_Название	<b>Абзац:</b> Выравнивание – по левому краю. Уровень – основной текст. Отступ: первая строка – отступа нет, интервал – перед 6, после – 4, не разрывать абзац

#### Окончание таблицы 4

Имя стиля	Тип стиля	Основан на стиле	Стиль следующего абзаца	Отличие форматирования от стилей «Обычный» и «Заголовок»
К_Табл_Загол_столбцов	Абзац	Обычный	К_Табл_Загол_столбцов	<b>Абзац:</b> Выравнивание – по центру. Уровень – основной текст. Отступ: первая строка – отступа нет, интервал – перед 0, после – 0
К_Табл_Осн_Текст	Абзац	Обычный	К_Табл_Осн_Текст	<b>Шрифт:</b> Размер – 13. <b>Абзац:</b> Выравнивание – по левому краю. Отступ: первая строка – отступа нет, интервал – перед 0, после – 0, не разрывать абзац
К_Заголовок 1	Абзац	Заголовок 1	К_Главный	Уровень – Уровень 1. Положение на странице: с новой страницы (рисунок 27)
К_Заголовок 2	Абзац	Заголовок 2	К_Главный	Уровень – Уровень 2

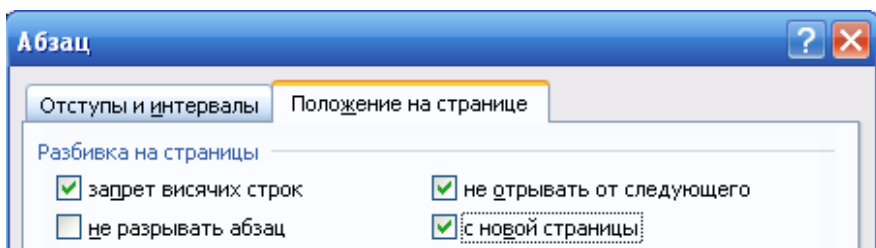


Рисунок 27 – Оформление Заголовка 1-го уровня с новой страницы

При необходимости можно создать свои стили. Основанием для назначения стиля является смысловое назначение элементов текста.

### 6.5 Автоматическое создание оглавлений

Содержание должно быть создано автоматически на основе стилей типа «Заголовок». Для этого требуется выполнить следующие действия:

- 1 Установить курсор после слова СОДЕРЖАНИЕ.
- 2 Дать команду «Вставка» – «Ссылка» – «Оглавления и указатели». Выбрать вкладку «Оглавления» и нажать ОК.
- 3 Если потребуется обновить оглавление, то это можно сделать клавишей F9 или командой быстрого меню «Обновить поле».

## 7 УСЛОВИЯ ЗАДАНИЙ

### 7.1 Задание «Шагающий конвейер»

Дан фрагмент схемы управления шагающим конвейером с двумя рабочими позициями (только контактная группа). 0 – выполняется очистка вагона от загрязнений механической щеткой, 1 – обмывка вагона. По множеству указанных сигналов в конкретный момент времени определить наименование выполняемой производственной операции (таблица 5).

Таблица 5 – Варианты задания

Вариант	Выражения алгебры логики	A	B	C	D
1	$A + B + C \cdot A + D$	0	0	0	0
2	$A \cdot C + \overline{D} + A + B$	0	0	1	1
3	$(A + B) \cdot C + \overline{D} + A + B$	0	1	0	0
4	$(A + B) \cdot B \cdot C + \overline{D} + A$	0	1	1	1
5	$(A + B) \cdot B \cdot C + A + D$	1	0	0	0
6	$(A + B) \cdot B \cdot C \cdot D + A$	1	0	1	1
7	$(A + B) \cdot B \cdot B \cdot C \cdot D + A$	1	1	0	0
8	$(A + B) \cdot A \cdot B \cdot B \cdot C + D$	1	1	1	1
9	$(A + B) \cdot B \cdot C + A \cdot B + D$	0	0	0	0
10	$(B + D) \cdot B \cdot C + A \cdot B + D$	0	0	1	1
11	$(B + D) \cdot B \cdot C + B \cdot D + D$	0	1	0	0
12	$(B + D) \cdot B \cdot C + B \cdot D + A + D$	0	1	1	1
13	$(\overline{D} + B) \cdot B \cdot C + B \cdot D + A + D$	1	0	0	0
14	$(\overline{D} + B) \cdot B \cdot C + B \cdot D + \overline{D} + A$	1	0	1	1
15	$(\overline{D} + B) \cdot B \cdot C + \overline{B \cdot D} + \overline{D} + A$	1	1	0	0
16	$(B + D) \cdot B \cdot C + B \cdot D + \overline{D} + A$	1	1	1	1
17	$(A + B) \cdot B \cdot C + B \cdot D + \overline{D} + A$	0	0	0	0
18	$(\overline{D} \cdot C + D) \cdot \overline{B} \cdot C + (C + D) \cdot A + A$	0	0	1	1
19	$(A + B) \cdot \overline{C} \cdot B + C \cdot D + A + D$	0	1	0	0

Окончание таблицы 5

Вариант	Выражения алгебры логики	A	B	C	D
20	$(A + B) \cdot \bar{C} \cdot B + C \cdot D + A + C$	0	1	1	1
21	$(A + B) \cdot \bar{C} \cdot B + C \cdot D + \bar{A} + C$	1	0	0	0
22	$B \cdot C + C \cdot D + \overline{(A + B)} + \bar{A} + C$	1	0	1	1
23	$B \cdot C + \overline{(A + B)} + \bar{A} + C + C + D$	1	1	0	0
24	$(C + D) \cdot \overline{(A + B)} + B \cdot C + \bar{A} + C$	1	1	1	1
25	$(C + D) \cdot \overline{(A + B)} + B \cdot C + \bar{B} + C$	0	0	0	0
26	$(C + D) \cdot \overline{(A + B)} + A \cdot C + \bar{B} + A$	0	0	1	1
27	$(C + D) \cdot \overline{(A + B)} + A \cdot C + \bar{D} + A$	0	1	0	0
28	$(C + D) \cdot \overline{(B + D)} + A \cdot C + \bar{D} + A$	0	1	1	1
29	$(C + D) \cdot \overline{(B + D)} + B \cdot C + \bar{D} + A$	1	0	0	0
30	$(C + D) \cdot \overline{(C + D)} + B \cdot C + A + D$	1	0	1	1
31	$(C + D) \cdot (C + D) + B \cdot C + A + D$	1	1	0	0
32	$(C + D) \cdot (C + D) \cdot B + \bar{B} \cdot C + A$	1	1	1	1
33	$(\bar{D} \cdot C + D) \cdot (C + D) \cdot B + \bar{B} \cdot C + A$	0	0	1	1
34	$(\bar{D} \cdot C + D) \cdot (C + D) \cdot A + \bar{B} \cdot C + A$	0	1	0	0
35	$(A + B) \cdot B \cdot C + C \cdot D + \bar{D} + A$	0	1	1	1

### 7.2 Задание «Электромеханический привод»

Дан фрагмент схемы управления электромеханическим приводом (только контактная группа), записанный в виде логического выражения. По множеству указанных сигналов в конкретный момент времени определить сигнал для хода подвижной рамы: 1 – соответствует прямому ходу рамы, 0 – соответствует обратному ходу рамы (таблица 6).

Таблица 6 – Варианты задания

Вариант	Выражения алгебры логики	A	B
1	$A \cdot B \cdot B \cdot A \cdot A + B \cdot A$	0	0
2	$(A \cdot B + B) \cdot A \cdot A \cdot B$	0	1
3	$(\overline{A + B + B}) \cdot A \cdot A \cdot B$	1	0
4	$(A + B) \cdot A \cdot A \cdot B \cdot B$	1	1

Окончание таблицы 6

Вариант	Выражения алгебры логики	A	B
5	$(A + B) \cdot B \cdot B + A$	0	0
6	$(A + B) \cdot B + A + B$	0	1
7	$(A + B) \cdot B + \bar{B} \cdot A + A$	1	0
8	$\bar{B} \cdot A + (A + B) \cdot B + A$	1	1
9	$\bar{B} \cdot A + A \cdot B + (A + B) \cdot \bar{B}$	0	0
10	$\bar{A} \cdot A \cdot B + \bar{B} \cdot A + (A + B) \cdot \bar{B}$	0	1
11	$\bar{B} \cdot A \cdot B + \bar{B} \cdot A + (A + B) \cdot \bar{B}$	1	0
12	$(A + B) \cdot \bar{B} \cdot A + \bar{B} \cdot A$	1	1
13	$(A + B) \cdot \bar{B} \cdot B + \bar{B} \cdot A$	0	0
14	$(A + B) \cdot \bar{B} \cdot A \cdot B + \bar{B} \cdot A$	0	1
15	$(\bar{A} \cdot A + B) \cdot B \cdot A \cdot B + A \cdot B$	1	0
16	$(\bar{B} \cdot A + B) \cdot B \cdot A \cdot B + A \cdot B$	1	1
17	$(\bar{B} \cdot A + B) \cdot A \cdot B \cdot B + A \cdot B$	0	0
18	$(\bar{B} \cdot A + B) \cdot A \cdot B \cdot B + \bar{A} \cdot B$	0	1
19	$((\bar{B} \cdot A + B) \cdot B \cdot B + \bar{A}) \cdot A \cdot B$	1	0
20	$((\bar{B} \cdot A + B) \cdot B \cdot B + \bar{A}) \cdot A \cdot A$	1	1
21	$((\bar{B} \cdot A + B) \cdot A \cdot B + \bar{A}) \cdot A$	0	0
22	$((\bar{A} \cdot A + B) \cdot A \cdot B + A) \cdot B$	0	1
23	$((\bar{A} \cdot A + B) \cdot A \cdot B + A) \cdot B + A + B$	1	0
24	$((\bar{A} \cdot A + B) \cdot A \cdot B + A) \cdot B + A$	1	1
25	$((\bar{A} \cdot A + B) \cdot A \cdot B + A) \cdot B + B$	0	0
26	$(\bar{A} \cdot B + B) \cdot A \cdot A \cdot B$	0	1
27	$(\bar{A} \cdot B + B) \cdot A \cdot A \cdot B + \bar{B} \cdot A$	1	0
28	$(\bar{A} \cdot B + B) \cdot A \cdot A \cdot B + \bar{A} \cdot \bar{B}$	1	1
29	$(A \cdot B + B) \cdot \bar{A} \cdot A \cdot B + \bar{A} \cdot \bar{B}$	0	1
30	$(A \cdot B + B) \cdot \bar{A} \cdot A \cdot B + \bar{B} \cdot A$	1	0
31	$(A \cdot B + B) \cdot \bar{A} \cdot A \cdot B + \bar{B}$	0	1
32	$(A \cdot B + B) \cdot A \cdot A \cdot B + \bar{B}$	1	0

Окончание таблицы 6

Вариант	Выражения алгебры логики	A	B
33	$((A + B) \cdot A \cdot B + A) \cdot B + \bar{B}$	1	1
34	$(A + B) \cdot A \cdot B + \bar{B} + A + A$	0	1
35	$((A + B + B) \cdot A + \bar{B} + A) \cdot A$	1	0

### СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 **Гораев, О. П.** Текстовый процессор MS WORD : практикум по компьютерным технологиям / О. П. Гораев, Ю. П. Лыч. – Гомель : БелГУТ, 2003. – 68 с.

2 **Гурский, Д. А.** Вычисления в MathCAD 12 / Д. А. Гурский, Е. С. Турбина. – СПб. : Питер, 2006. – 544 с.

3 Информатика. Базовый курс / под ред. С. В. Симоновича ; М-во образования Рос. Федерации. – 2-е изд. – СПб. : Питер, 2006. – 640 с.: ил.

4 Информатика. Программирование на языке Паскаль : практикум по лабораторным работам. Ч. I. / А. П. Кейзер [и др.]. – Гомель : БелГУТ, 2001. – 46 с.

5 **Макаров, Е. Г.** Инженерные расчеты в MathCAD : учеб. курс / Е. Г. Макаров. – СПб. : Питер, 2005. – 448 с.: ил.

6 **Миняйлова, Е. Л.** Работа с библиотечными модулями в Паскале (на примере объекта «Множество») : учеб.-метод. пособие / Е. Л. Миняйлова, Д. А. Вербовиков, Л. Н. Хойер. – Гомель : ГГОИПК и ПРР и СО, 2008. – 56 с.

7 **Разон, В. Ф.** Автоматическое управление машинами циклического действия : учеб.-метод. пособие / В. Ф. Разон ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2009. – 313 с.

8 **Фаронов, В. В.** Турбо Паскаль 7.0: практика программирования / В. В. Фаронов. – М. : Нолидж, 2000. – 724 с.

Учебное издание

*МИНЯЙЛОВА Елена Леонидовна*

РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ СРЕДСТВАМИ  
ИНФОРМАЦИОННЫХ КОМПЬЮТЕРНЫХ КОМПЛЕКСОВ:  
ЗАДАЧА «ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ  
В СИСТЕМАХ УПРАВЛЕНИЯ НА ТРАНСПОРТЕ»

Учебно-методическое пособие

Редактор *Т. М. Ризевская*

Технический редактор *В. Н. Кучерова*

Подписано в печать 22.12.2012 г. Формат 60×84  $\frac{1}{16}$   
Бумага офсетная. Гарнитура Times. Печать на ризографе.  
Усл. печ. л. 2,32. Уч.-изд. л. 1,93. Тираж 150 экз.  
Зак. № . Изд. № 163.

Издатель и полиграфическое исполнение  
Белорусский государственный университет транспорта:  
ЛИ № 02330/0552508 от 09.07.2009 г.  
ЛП № 02330/0494150 от 03.04.2009 г.  
246653, г. Гомель, ул. Кирова, 34