МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ КРЫМ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

РЕСПУБЛИКИ КРЫМ

«КРЫМСКИЙ РЕСПУБЛИКАНСКИЙ ИНСТИТУТ ПОСТДИПЛОМНОГО

ПЕДАГОГИЧЕСКОГО ОБРАЗОВАНИЯ»

**«МЕТОДИКА ПОДГОТОВКИ ОБУЧАЮЩИХСЯ К ЕГЭ И ОГЭ**

**ПО ИНФОРМАТИКЕ И ИКТ»**

**Методические рекомендации**

г. Симферополь

2017

Методика подготовки обучающихся к ЕГЭ и ОГЭ по информатике и ИКТ. Методические рекомендации – Симферополь 2017 г. – 37 с.

Методические рекомендации предназначены для учителей информатики. В издании рассматриваются особенности изучения отдельных тем с целью подготовки обучающихся к ОГЭ и ЕГЭ по информатике и ИКТ. Особое внимание уделяется темам «Среда программирования Кумир», «Логические операции», «Операции целочисленного деления в Паскале», «Работа с массивами», с которыми крымские учителя ранее не сталкивались в школьном курсе информатики.

Составители:

*Киндра Т.В.,* методист по информатике и ИКТ центра качества образования ГБОУ ДПО РК КРИППО;

*Якубовский Ю.В.,* руководитель отдела образовательных информационных технологий ГБОУ ДПО РК КРИППО.

Одобрено на заседании Ученого советаГБОУ ДПО РК КРИППО «23» марта 2017 г. (протокол № 3)

**Оглавление**

[ВВЕДЕНИЕ 4](#_Toc477973440)

ОСНОВНАЯ ЧАСТЬ

[1. Содержание заданий экзаменационной работы основного государственного экзамена (ОГЭ) по информатике и ИКТ в 9 классе 6](#_Toc477973441)

[2. Содержание заданий экзаменационной работы единого государственного экзамена (ЕГЭ) по информатике и ИКТ в 11 классе 8](#_Toc477973442)

[3. Среда программирования КУМИР. Решение задач ЕГЭ с исполнителем «Робот» 9](#_Toc477973443)

[4. Логические операции 21](#_Toc477973444)

[4.1. Операции алгебры логики 21](#_Toc477973445)

[4.2. Изображения логических операций в виде операций над множествами 21](#_Toc477973446)

[4.3. Таблицы истинности 22](#_Toc477973447)

[5. Операции целочисленного деления в Паскале. Разбор задачи ЕГЭ № 24 29](#_Toc477973448)

[6. Работа с массивами. Разбор задачи ЕГЭ № 25 34](#_Toc477973449)

[ЗАКЛЮЧЕНИЕ 35](#_Toc477973450)

[ИНТЕРНЕТ-РЕСУРСЫ ДЛЯ ПОДГОТОВКИ К ЕГЭ И ОГЭ 36](#_Toc477973451)

[ЛИТЕРАТУРА 37](#_Toc477973452)

# ВВЕДЕНИЕ

В методических рекомендациях для учителей информатики «Методика подготовки обучающихся к ЕГЭ и ОГЭ по информатике и ИКТ» раскрывается система работы учителя по предупреждению типичных ошибок при выполнении заданий ГИА по информатике, рассматриваются особенности изучения отдельных тем с целью подготовки обучающихся к ОГЭ и ЕГЭ по информатике и ИКТ.

Краткий анализ результатов ОГЭ и ЕГЭ по информатике и ИКТ в 2016 году в Республике Крым показывает, что наибольшие затруднения вызвали темы, которые раньше не входили в учебную программу по информатике и ИКТ.

При решении задания ЕГЭ № 24 обучающиеся не смогли прочесть фрагмент программы и исправить допущенные ошибки, испытывали трудности при работе с операциями целочисленного деления, циклом с предусловием.

При решении задания ЕГЭ № 25 обучающиеся не смогли написать программу на обработку элементов массива, поиск элементов массива с заданным свойством.

При решении задания ОГЭ № 20.1 обучающиеся не смогли составить алгоритм движения робота в среде программирования Кумир.

При решении задания ОГЭ № 20.2. испытывали трудности при работе с операциями целочисленного деления, циклом с предусловием, составлением алгоритма поиска элемента в последовательности с заданным свойством.

При выполнении заданий ГИА по информатике и ИКТ по теме «Логические операции» многие обучающиеся не смогли дать правильный ответ.

Поэтому в методических рекомендациях особое внимание уделяется решению задач по темам «Среда программирования Кумир», «Логические операции», «Операции целочисленного деления в Паскале», «Работа с массивами», с которыми крымские учителя ранее не сталкивались в школьном курсе информатики. Знание этих тем необходимо при сдаче ЕГЭ и ОГЭ по информатике и ИКТ.

В методических рекомендациях рассматриваются содержание заданий экзаменационной работы основного государственного экзамена (ОГЭ) по информатике и ИКТ в 9 классе, содержание заданий экзаменационной работы единого государственного экзамена (ЕГЭ) по информатике и ИКТ в 11 классе, формат заданий и основные требования к ним, предлагаются интернет-ресурсы для подготовки обучающихся к ЕГЭ и ОГЭ по информатике и ИКТ.

# Содержание заданий экзаменационной работы основного государственного экзамена (ОГЭ) по информатике и ИКТ в 9 классе

Содержание заданий экзаменационной работы основного государственного экзамена (ОГЭ) по информатике и ИКТ в 9 классе разрабатывается по основным темам курса информатики и ИКТ, объединенных в следующие тематические блоки: «Представление и передача информации», «Обработка информации», «Основные устройства ИКТ», «Запись средствами ИКТ информации об объектах и о процессах, создание и обработка информационных объектов», «Проектирование и моделирование», «Математические инструменты, электронные таблицы», «Организация информационной среды, поиск информации. Более подробно темы рассматриваются в кодификаторе элементов содержания и требований к уровню подготовки обучающихся для проведения основного государственного экзамена по информатике и ИКТ.

Экзаменационная работа по информатике и ИКТ в 9 классе состоит из двух частей и включает в себя 20 заданий. На выполнение экзаменационной работы в 9 классе отводится 2 часа 30 минут (150 минут). Часть 1 экзаменационной работы содержит 11 заданий базового уровня сложности и 7 заданий повышенного уровня сложности. Задания части 1 являются теоретическими и выполняются обучающимися без использования компьютеров и других технических средств, использовать калькуляторы на экзаменах не разрешается.

Часть 2 содержит 2 задания высокого уровня сложности с развернутым ответом. Задания части 2 являются практическими и выполняются экзаменуемыми на компьютере. Результатом выполнения каждого из заданий является отдельный файл. После окончания экзамена организаторы экзамена осуществляют сбор файлов с выполненными заданиями и передают их на проверку экспертной комиссии.

Часть 2 экзаменационной работы содержит два задания (19 и 20), задание 20 представлено в двух вариантах (20.1 и 20.2), учащийся самостоятельно выбирает один из двух вариантов задания.

Задание 19 заключается в обработке большого массива данных с использованием электронной таблицы. Для выполнения данного задания необходимо использовать программу электронных таблиц, например, Microsoft Excel, ОpenOffice.org Calc, LibreOffice Calc или другую программу.

При выполнении задания 19 учащийся находит ответы на вопросы, сформулированные в задании, используя средства электронной таблицы: формулы, функции, операции с блоками данных, сортировку и поиск данных и записывает ответы в указанные ячейки электронной таблицы.

Задание 20.1 заключается в разработке алгоритма для учебного исполнителя «Робот». Описание команд исполнителя и синтаксиса управляющих конструкций соответствует общепринятому школьному алгоритмическому языку, также оно дано в тексте задания. Для выполнения этого задания можно использовать свободно распространяемую среду учебного исполнителя Кумир (сайт <http://www.niisi.ru/kumir/>).

Альтернативным заданием для задания 20.1 является задание 20.2, где необходимо реализовать алгоритм на языке программирования, знакомом учащимся. Задание 20.2 проверяет умения, связанные с созданием простейших программ, содержащих цикл и ветвление внутри цикла, на одном из языков программирования.

Примеры заданий 19 и 20, алгоритмы решения данных заданий представлены в методических рекомендациях по оцениванию выполнения заданий ОГЭ с развернутым ответом по информатике и ИКТ в 9 классе на сайте [www.fipi.ru](http://www.fipi.ru) в разделе «Для предметных комиссий субъектов РФ» подраздел «Методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ОГЭ 2016 года».

# Содержание заданий экзаменационной работы единого государственного экзамена (ЕГЭ) по информатике и ИКТ в 11 классе

**Содержание заданий экзаменационной работы единого государственного экзамена (ЕГЭ) по информатике и ИКТ в 11 классе** разрабатывается по основным темам курса информатики и ИКТ, объединенных в следующие тематические блоки: «Информация и ее кодирование», «Моделирование и компьютерный эксперимент», «Системы счисления», «Логика и алгоритмы», «Элементы теории алгоритмов», «Программирование», «Архитектура компьютеров и компьютерных сетей», «Обработка числовой информации», «Технологии поиска и хранения информации».

Экзаменационная работа по информатике и ИКТ в 11 классе состоит из двух частей и включает в себя 27 заданий. На выполнение экзаменационной работы в 11 классе отводится 3 часа 55 минут (235 минут).

Часть 1 экзаменационной работы содержит 23 задания с кратким ответом, из них 12 заданий относится к базовому уровню, 10 заданий к повышенному уровню сложности, 1 задание – к высокому уровню сложности.

Часть 2 экзаменационной работы содержит четыре задания (24, 25, 26, 27), относящиеся к повышенному и высокому уровню сложности. Задания этой части подразумевают запись развернутого ответа в произвольной форме. Следует отметить, что учащимся для выполнения заданий компьютеры не предоставляются. Данные задания выполняются письменно и записываются в бланк ответов.

Задание 24 заключается в умении прочесть фрагмент программы на языке программирования и исправить допущенные ошибки. Проверяет умения, связанные с созданием простейших программ, содержащих цикл и ветвление внутри цикла, знание целочисленных операций div, mod.

Задание 25 заключается в умении написать короткую (10–15 строк) простую программу обработки массива на любом языке программирования или записать алгоритм на естественном языке. Проверяет умения, связанные с применением алгоритма поиска элемента массива с заданным свойством.

Задание 26 заключается в умении построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию.

Задание 27 заключается в умении создавать собственные программы (30–50 строк) для решения задач высокого уровня сложности, знать динамическое программирование.

Примеры заданий 24-27, алгоритмы решения данных заданий представлены в методических рекомендациях по оцениванию выполнения заданий ЕГЭ с развернутым ответом по информатике и ИКТ в 11 классе на сайте [www.fipi.ru](http://www.fipi.ru) в разделе «Для предметных комиссий субъектов РФ» подраздел «Методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ЕГЭ 2016 года».

# Среда программирования КУМИР. Решение задач ЕГЭ с исполнителем «Робот»

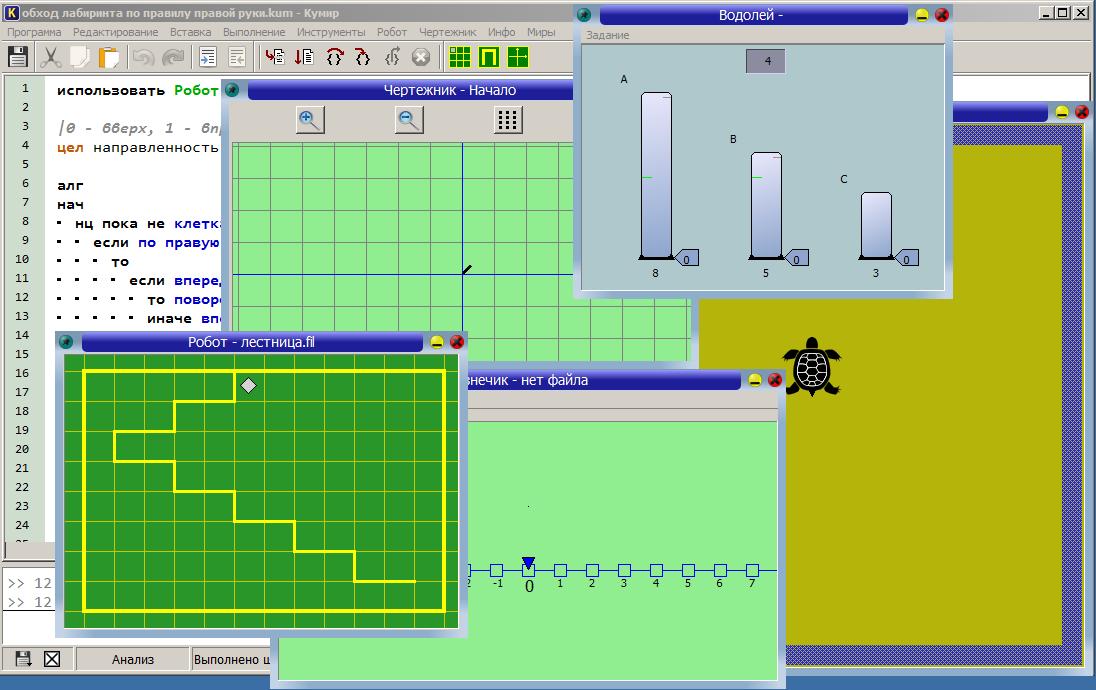
Крымские школьники при изучении информатики ранее не сталкивались со средой программирования КУМИР в школьном курсе информатики, а знание этой темы необходимо при сдаче ЕГЭ и ОГЭ по информатике, поэтому этой теме хотелось бы уделить особое внимание.

Но перед тем как работать со «Средой программирования Кумир», приведем информационную справку о данной программе [8].

Язык КУМИР - универсальный язык программирования, его прототипом послужил "школьный язык программирования", разработанный А.П. Ершовым в первой половине 80-х годов ХХ века.

Система КУМИР позволяет создавать, отлаживать и выполнять программы. Несложные программы Вы сможете начать писать и выполнять практически сразу после знакомства с системой, однако система КУМИР позволяет создавать и достаточно большие, сложные программы. Во время редактирования программы система КУМИР автоматически производит синтаксический разбор и сообщает о найденных ошибках.

В то же время КУМИР является учебной системой. В неё встроено несколько графических исполнителей, действия которых визуализируются на их "игровом поле" или арене.



Кроме обычных средств программирования Кумир имеет интерактивные средства управления исполнителями - панелью управления, которая дает выполнять определенный набор действий. Система КУМИР включает встроенные графические исполнители.

Среда программирования КУМИР может быть установлена как под Linux, так и под Windows, а также доступна в исходных кодах. Скачать ее можно на сайте НИИСИ  <http://www.niisi.ru/kumir/>

Язык КуМир ни в коем случае не должен стать конечной точкой при обучении программированию! Это язык, с которого хорошо начать, чтобы освоить основы алгоритмического подхода и процедурный стиль программирования.

Следующими языками должны стать Python, Pascal, Си или C++.

Рассмотрим пример из заданий ЕГЭ-2015 [1].

**Пример из ЕГЭ-2015 по теме «Система команд исполнителя РОБОТ»**

***(****данное задание школьники выполняют письменно без использования компьютера)*

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

|  |  |  |  |
| --- | --- | --- | --- |
| вверх | вниз | влево | вправо |

При выполнении этих команд РОБОТ перемещается на одну клетку соответственно: вверх , вниз , влево , вправо.

При выполнении команды корпус робота не поворачивается.

(Робот в лабиринте находится спиной к зрителю, т.е. команда вправо для работа равносильна команде вправо для зрителя)

Четыре команды проверяют истинность условия отсутствия стены у той клетки, где находится РОБОТ:

|  |  |  |  |
| --- | --- | --- | --- |
| сверху | снизу | слева | справа |
| свободно | свободно | свободно | свободно |

Цикл

**нц** **пока** **условие**

**команда**

**кц**

выполняется, пока условие истинно, иначе происходит переход на следующую строку.

**Задание.**

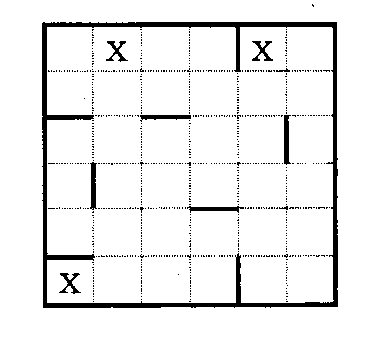
Сколько клеток приведённого лабиринта соответствует требованию, что, выполнив предложенную ниже программу, РОБОТ остановится в той же клетке, с которой он начал движение?

|  |  |
| --- | --- |
| **Лабиринт** | **Программа** |
|  | **использовать** **Робот** **алг**  **нач**  . **нц** **пока** **справа свободно**  . . **вправо**  . **кц** . **нц** **пока** **снизу свободно**  . . **вниз**  . **кц**  . **нц** **пока** **слева свободно**  . . **влево**  . **кц**  . **нц** **пока** **сверху свободно**  . . **вверх**  . **кц**  **кон** |

***Решение***

На рисунке обозначены клетки, удовлетворяющие условию, и направления движения робота при выполнении программы.

***Ответ:*** 3



Для наглядного объяснения данной темы рассмотрим решение данной задачи в среде программирования КУМИР.

**Шаг 1.**

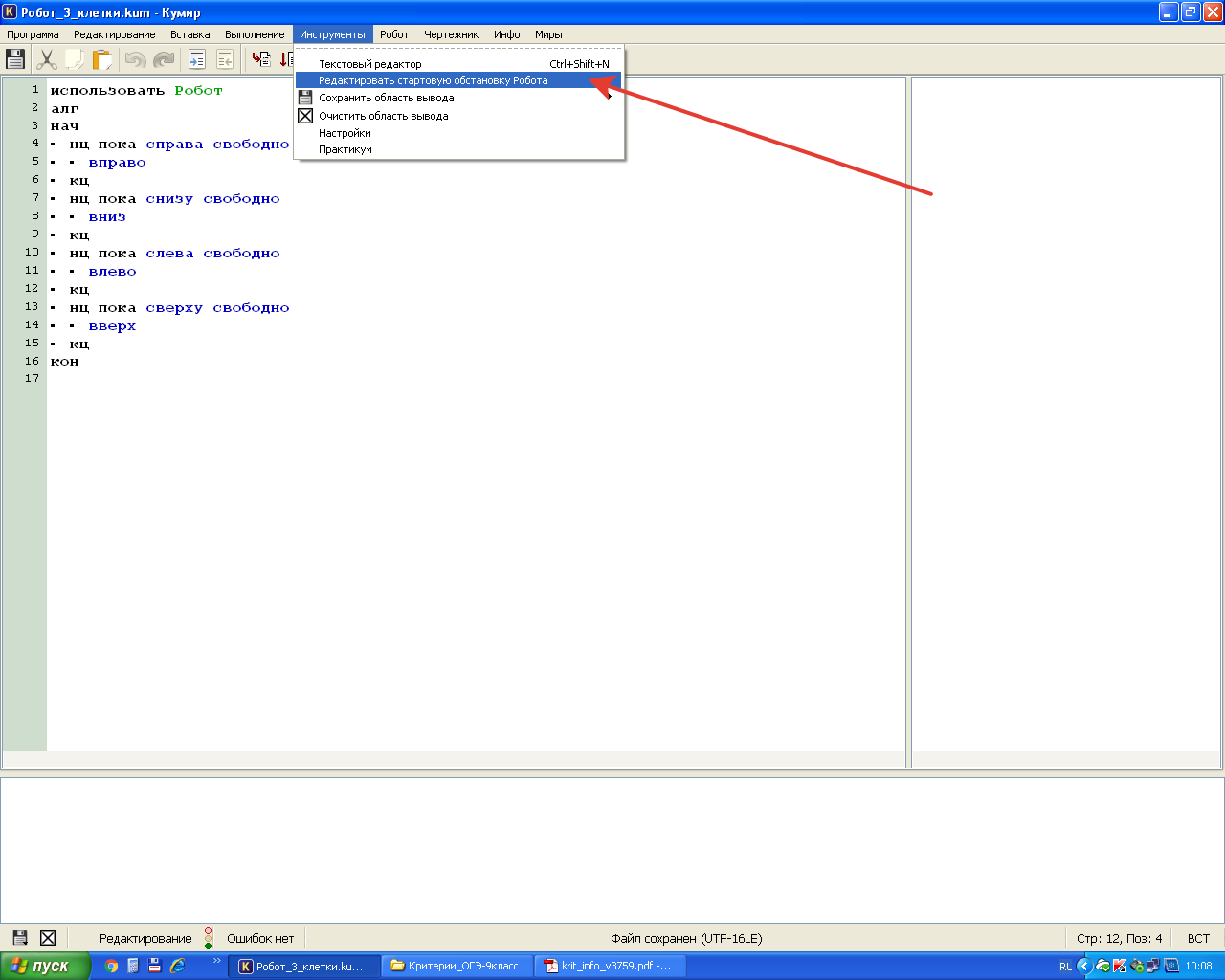
Запустить среду программирования Кумир, напечатать программу.

Обратите внимание, что структура программы в Кумире играет большую роль. Если Вы в цикле напишите в одной строке условие и команду, например, «справа свободно вправо», то это будет ошибкой

****

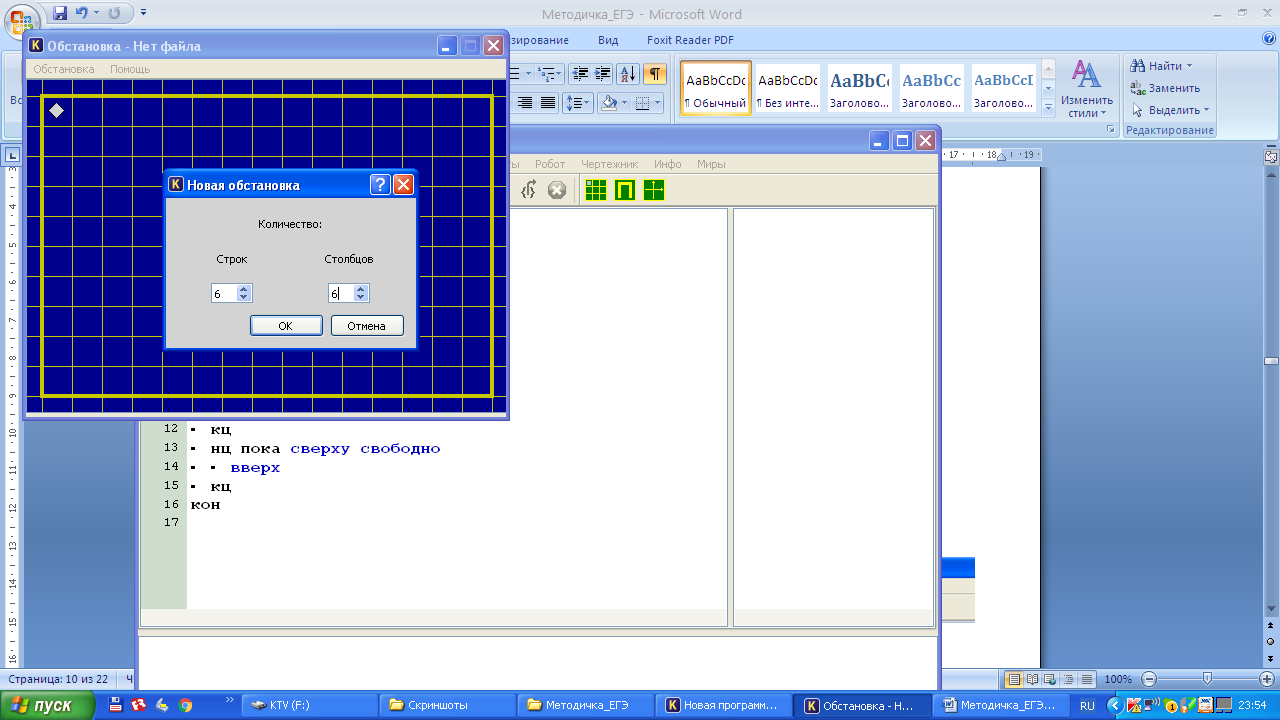
**Шаг 2.**

Создадим исходный лабиринт. Выполним команду **«Инструменты - Редактировать стартовую обстановку»**

****

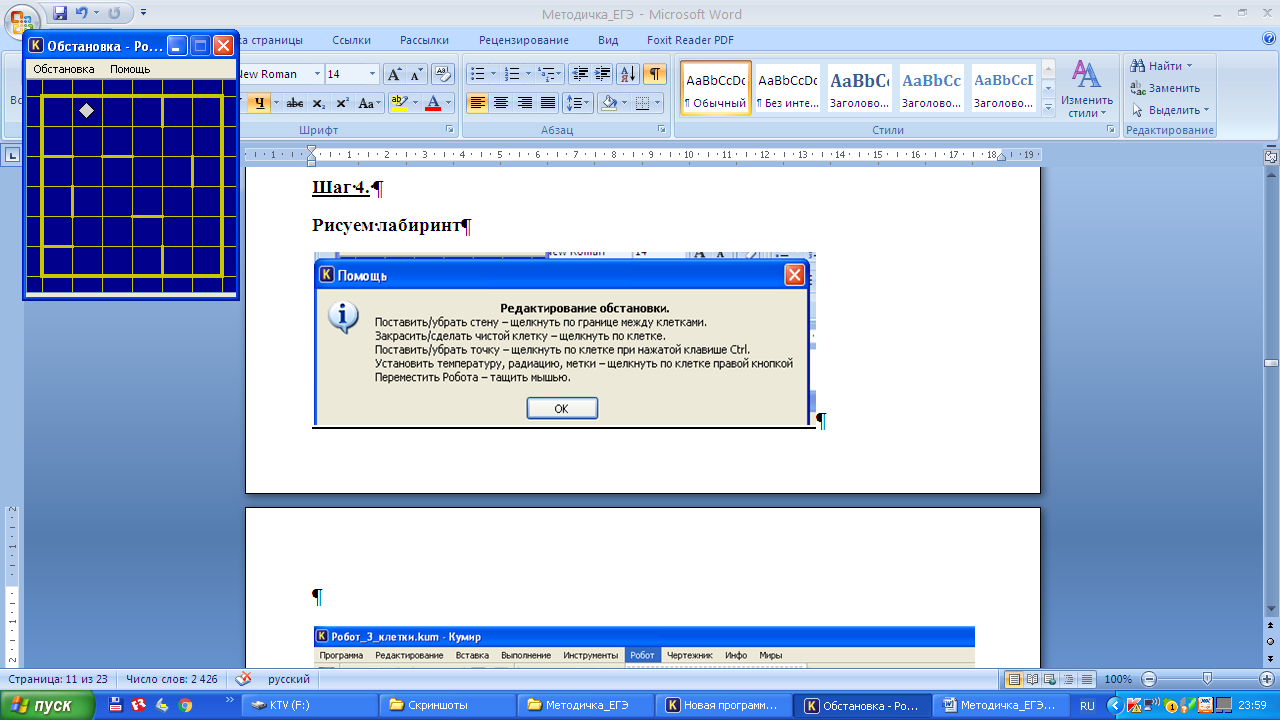
**Шаг 3.**

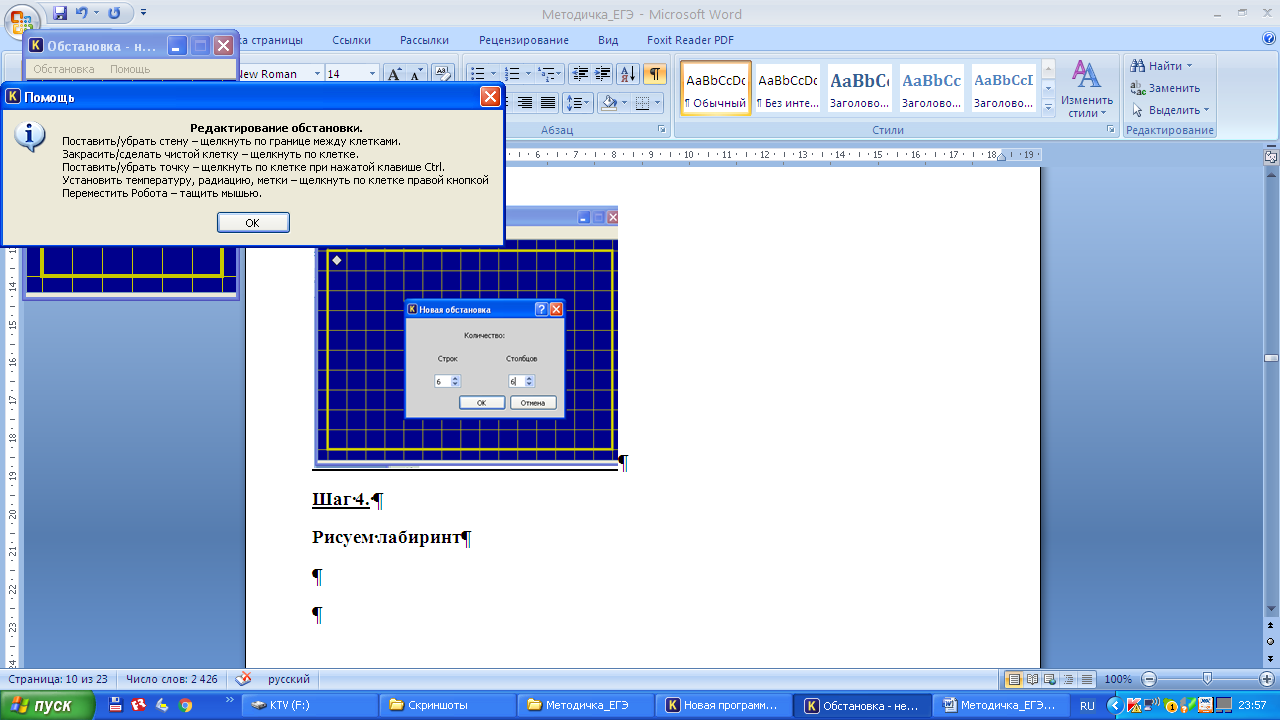
Следующая команда «Обстановка – Новая обстановка», указываем количество строк и столбцов 6.

****

**Шаг 4.**

Рисуем лабиринт

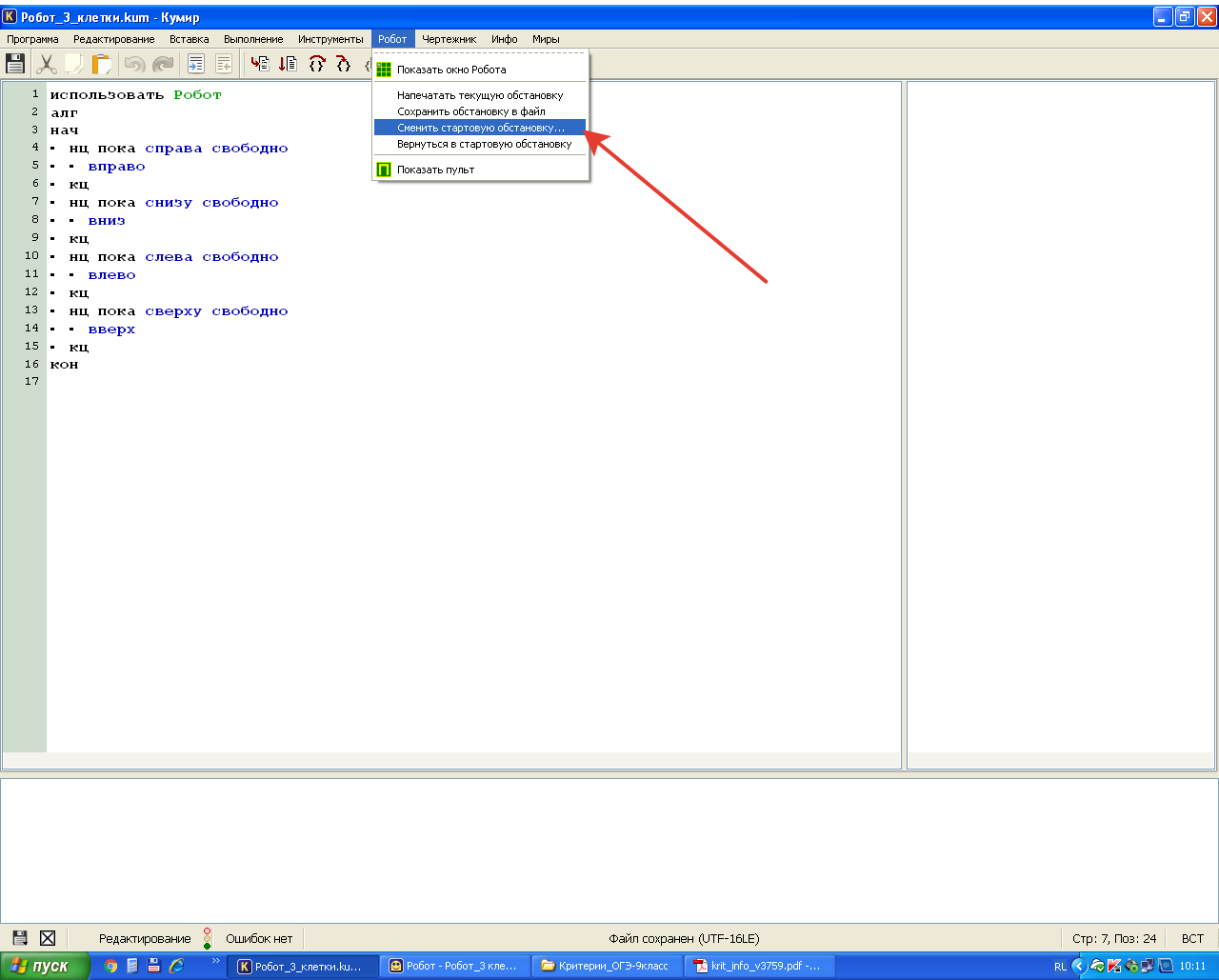
****

****

Выполняем команду «Обстановка – Сохранить»

**Шаг 5**

Выполняем команду «Робот - сменить стартовую обстановку»

****

**Шаг 6**

Выполняем команду «Робот – Показать окно Робота»

**Шаг 7**

Выполняем команду «Выполнение – Выполнить непрерывно F9»

Аналогично можно создать обстановки для оставшихся двух точек, изменить расположение робота и проверить правильность работы алгоритма.

**Рассмотрим пример из заданий ОГЭ-2016 [5].**

**Пример из ОГЭ-2016 по теме «Система команд исполнителя РОБОТ»**

Исполнитель Робот умеет перемещаться по лабиринту, начерченному   
на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Пять команд – это команды-приказы.

Четыре из них управляют перемещениями Робота:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит

команду передвижения сквозь стену, то он разрушится

Также у Робота есть команда-приказ **закрасить**,при которойзакрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «**eсли**», имеющим следующий вид:

**если** *условие* **то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность* *команд* – это одна или несколько любых команд-приказов

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «**пока**», имеющий следующий вид:

**нц пока** *условие*

*последовательность команд*

**кц**

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

**кц**

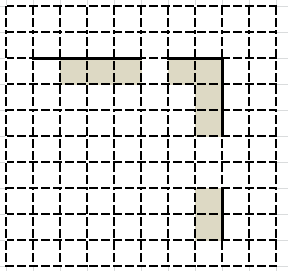
***Выполните задание.***

На бесконечном поле есть горизонтальная и вертикальная стены. Правый конец горизонтальной стены соединён с верхним концом вертикальной стены. **Длины стен неизвестны**. В каждой стене есть ровно один проход, точное место прохода и его ширина неизвестны. Робот находится в клетке, расположенной непосредственно под горизонтальной стеной у её левого конца.

На рисункеуказан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Р |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно ниже горизонтальной стены и левее вертикальной стены, кроме клетки, в которой находится Робот перед выполнением программы. Проходы должны остаться незакрашенными. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное расположение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера проходов внутри стен.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

|  |
| --- |
| **Содержание верного ответа и указания по оцениванию**  (допускаются иные формулировки ответа, не искажающие его смысла) |
| Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».  *| Пропускаем клетку, в которой стоит Робот*  **вправо**  |*Двигаемся вправо, пока не дойдём до прохода в горизонтальной стене*  |*Закрашиваем пройденные клетки*  **нц пока не сверху свободно**  **закрасить**  **вправо**  **кц**  |*Двигаемся дальше до горизонтальной стены*  **нц пока сверху свободно**  **вправо**  **кц**  |*Двигаемся вправо, пока не дойдём до вертикальной стены*  |*Закрашиваем пройденные клетки*  **нц пока справа свободно**  **закрасить**  **вправо**  **кц**  |*Двигаемся вниз, пока не дойдём до прохода в вертикальной стене*  |*Закрашиваем пройденные клетки*  **нц пока не справа свободно**  **закрасить**  **вниз**  **кц**  |*Двигаемся дальше до вертикальной стены*  **нц пока справа свободно**  **вниз**  **кц**  |*Двигаемся вниз ,до конца вертикальной стены*  |*Закрашиваем пройденные клетки*  **нц пока не справа свободно**  **закрасить**  **вниз**  **кц**  Возможны и другие варианты решения |

# Логические операции

# Операции алгебры логики

Базовыми элементами, которыми оперирует алгебра логики, являются **высказывания**. Высказывания обозначают строчными буквами — **логическими переменными** [10].

Возможные значения логической переменной:

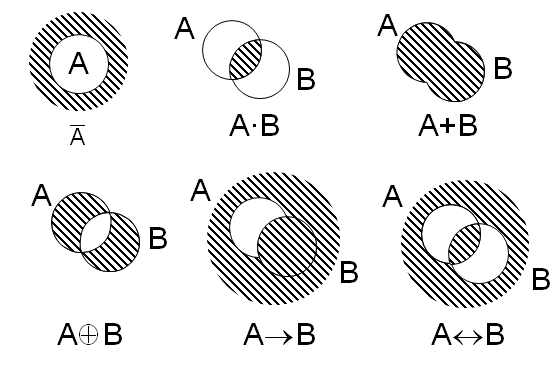
1 — *истина*, 0 — *ложь*.

Высказывания строятся из *логических переменных* и *логических констант* 1 и 0 при помощи *операций*:

1. ¬ — отрицание (унарная операция),
2. ∧ — конъюнкция, логическое умножение (бинарная),
3. ∨ — дизъюнкция, логическое сложение (бинарная),
4. ↔ — эквивалентность («тогда и только тогда, когда») (бинарная),
5. → — импликация («следовательно») (бинарная),
6. ⊕ — сложение по модулю два («исключающее или»), | — штрих Шеффера, ↓ — стрелка Пирса и другие.

**Приоритет логических операций**: если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем  – «ИЛИ», и самая последняя – «импликация». **Действия выполняются по приоритету операций слева на право**

# Изображения логических операций в виде операций над множествами



# Таблицы истинности

В алгебре логики используются следующие логически операции: НЕ, И, ИЛИ, XOR, импликация, эквивалентность.

**ОПЕРАЦИЯ НЕ**

A истинно когда «не А» ложно.

|  |  |
| --- | --- |
| a | ¬a |
| 0 | 1 |
| 1 | 0 |

**ОПЕРАЦИЯ И**

«A и B» истинно тогда и только тогда, когда А и B истинны одновременно.

И называется также логическим умножением или конъюнкцией.

|  |  |  |
| --- | --- | --- |
| a | b | a∧b |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**ОПЕРАЦИЯ ИЛИ**

«A или B» истинно, когда истинно А или B, или оба вместе.

Можно сказать, что «A или B» ложно тогда и только тогда, когда ложны А и В одновременно.

ИЛИ называют логическим сложением или дизъюнкцией.

|  |  |  |
| --- | --- | --- |
| a | b | a∨b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**ОПЕРАЦИЯ XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ)**

«A ⊕ B» истинно тогда, когда истинно А или B, но не оба одновременно.

Эту операцию также называют "сложение по модулю два".

|  |  |  |
| --- | --- | --- |
| a | b | a⊕b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**ОПЕРАЦИЯ ИМПЛИКАЦИИ**

«A → B» истинно, если из А может следовать B.

Если из истины следует ложь, то следствие ложно. При этом из лжи корректными следствиями может получиться все, что угодно, поэтому в остальных случаях импликация истинна.

|  |  |  |
| --- | --- | --- |
| a | b | a→b |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**ЭКВИВАЛЕНТНОСТЬ (ТОЖДЕСТВО)**

«A ↔ B» истинно тогда и только тогда, когда А и B равны.

|  |  |  |
| --- | --- | --- |
| a | b | a↔b |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблицы истинности можно составить и для произвольной логической функции F(a, b, c…).

В общем случае таблицы истинности имеют размер 2N строк комбинаций для N независимых логических переменных.

Поскольку таблица истинности выражения состоит из строк со всеми возможными комбинациями значений переменных, она полностью определяет значение выражения.

**Удобные обозначения логических операций.**

Обозначения логических операций И, ИЛИ и НЕ в классической математической логике (∨,∧, ¬) интуитивно непонятны, не проявляют аналогии с обычной алгеброй.

**Альтернативные обозначения**

* «НЕ» — черта сверху;
* «И» — знак умножения (**логическое умножение**);
* «ИЛИ» — знак «+» (**логическое сложение**).

**ЗАПОМНИТЕ ПРАВИЛО, КОТОРОЕ ВСЕГДА РАБОТАЕТ!!!**

**Продемонстрируем мощь альтернативных обозначений логических операций:**

|  |  |
| --- | --- |
| **1 ∧ 0 = 0 — не очевидно** | **1 ⋅ 0 = 0 — очевидно!** |
| **1 ∨ 0 = 1 — не очевидно** | **1 + 0 = 1 — очевидно!** |
| **1 ∨ 1 = 1 — не очевидно** | **1 + 1 = 1 — не очевидно, но можно смириться** |

**Итак, таблицы истинности для И и ИЛИ можно не учить, нужно запомнить только одно исключение:**

**1 + 1 = 1**

**Рассмотрим пример № 1.**

Дан фрагмент таблицы истинности выражения F.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | x6 | x7 | F |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

Какое выражение соответствует F?

x1→(x2∧x3∨x4∧x5∨x6∧x7)

x2→(x1∧x3∨x4∧x5∨x6∧x7)

x3→(x1∧x2∨x4∧x5∨x6∧x7)

x4→(x1∧x2∨x3∧x5∨x6∧x7)

**Решение**

**Вспомним, таблицу истинности для импликации.**

|  |  |  |
| --- | --- | --- |
| a | b | a→b |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Поскольку нам дан только фрагмент таблицы истинности, восстановить логическую функцию невозможно. Нужно действовать проверкой отбрасыванием негодных ответов, причем желательно это делать быстро.

Ни в коем случае не раскрывайте импликацию в логическое сложение!

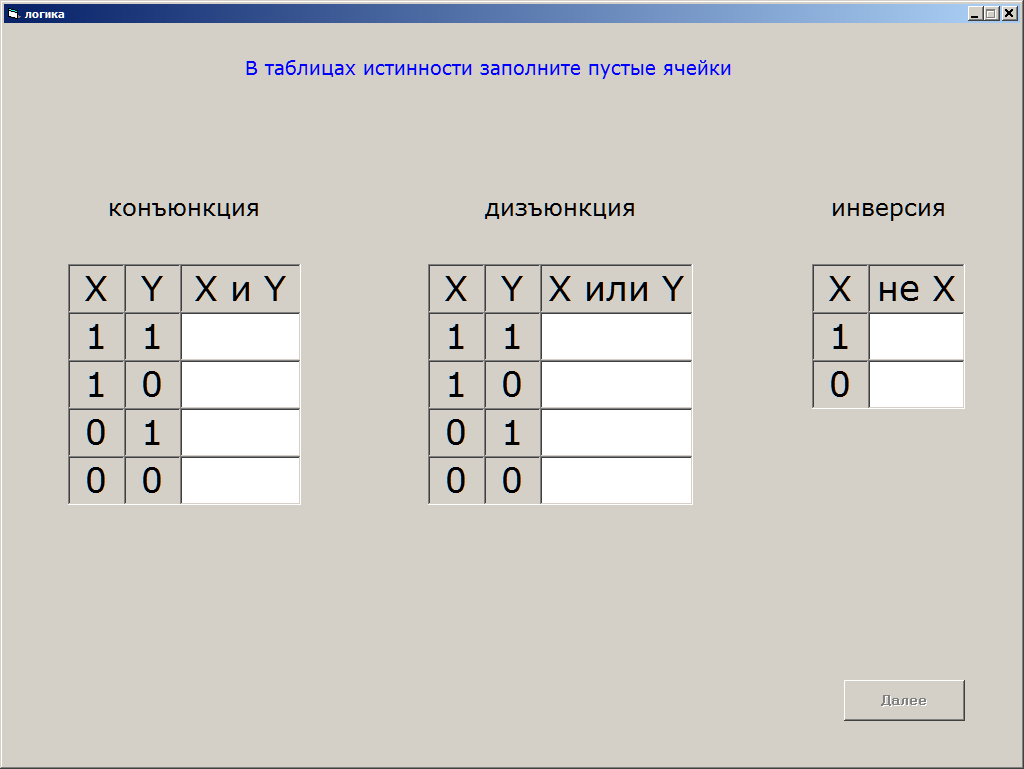
Импликация ложна только тогда, когда посылка истинна, а вывод ложен. Посылка (левый операнд) в каждом из вариантов ответов разная. Значит варианты функции, где переменная-посылка в одной из тех строк фрагмента таблицы истинности, где функция равна нулю, сама равна нулю, можно отбросить.

**x4→(x1∧x2∨x3∧x5∨x6∧x7) правильный ответ**

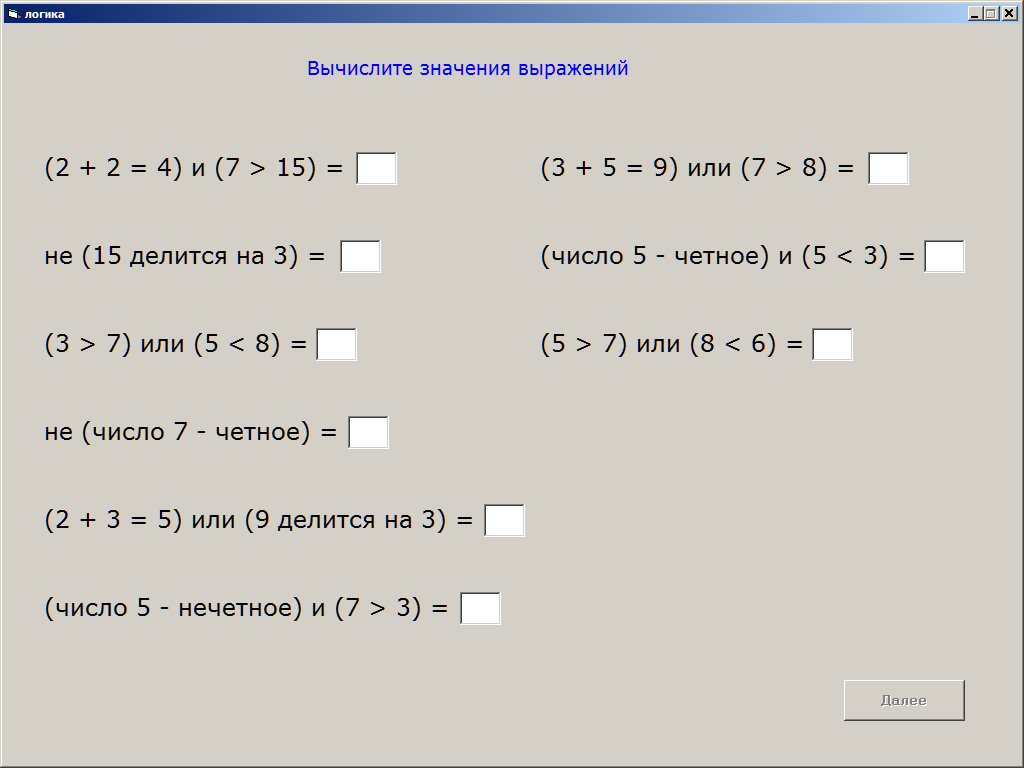
**Рассмотрим пример № 2.**

Для закрепления работы с логическими операциями рассмотрим 5 практических заданий

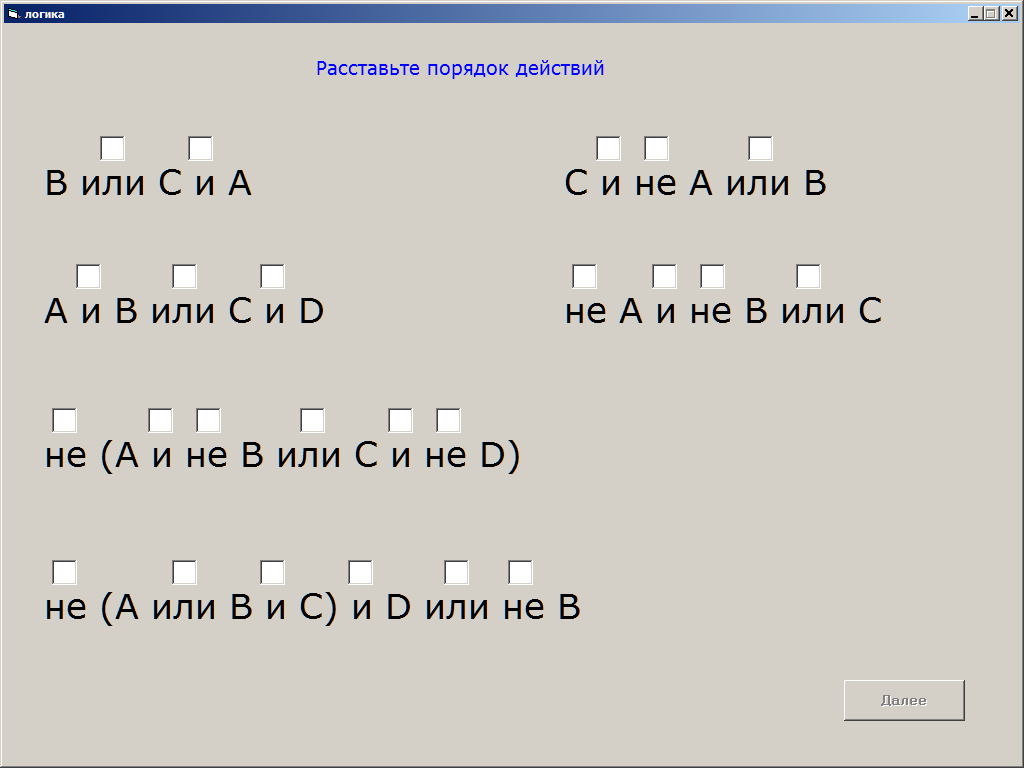
**Задание № 1. В пустых ячейках таблицы нужно поставить 0 или 1.**



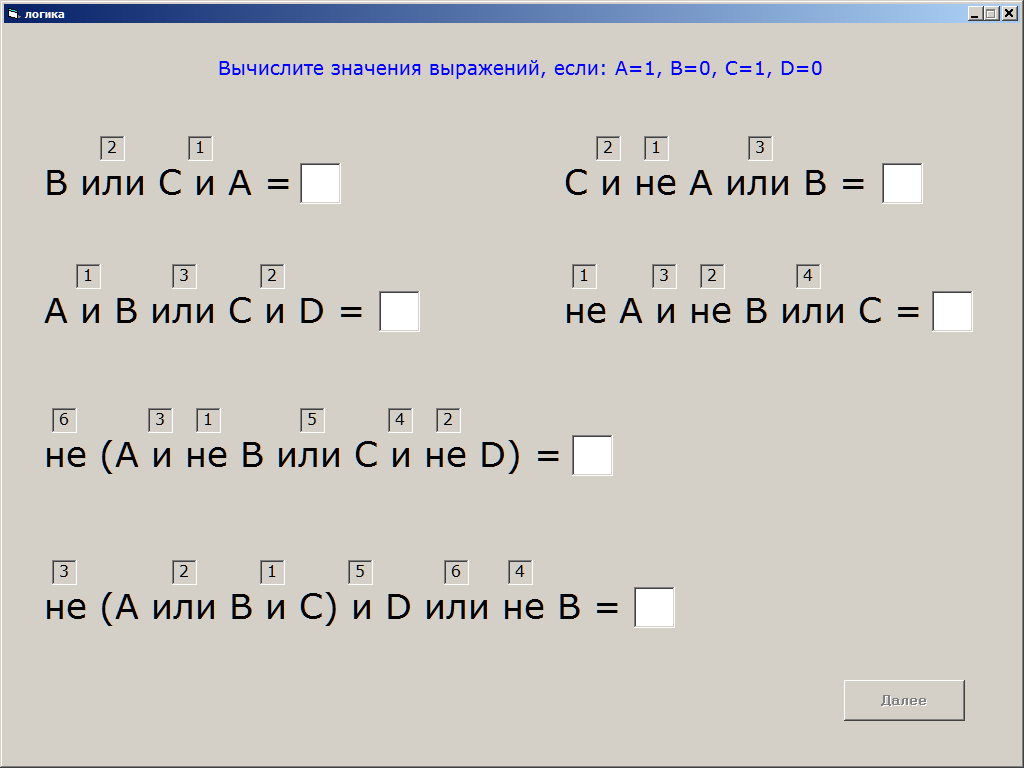
**Задание № 2. Вычислите значения выражения, нужно поставить 0 или 1.**



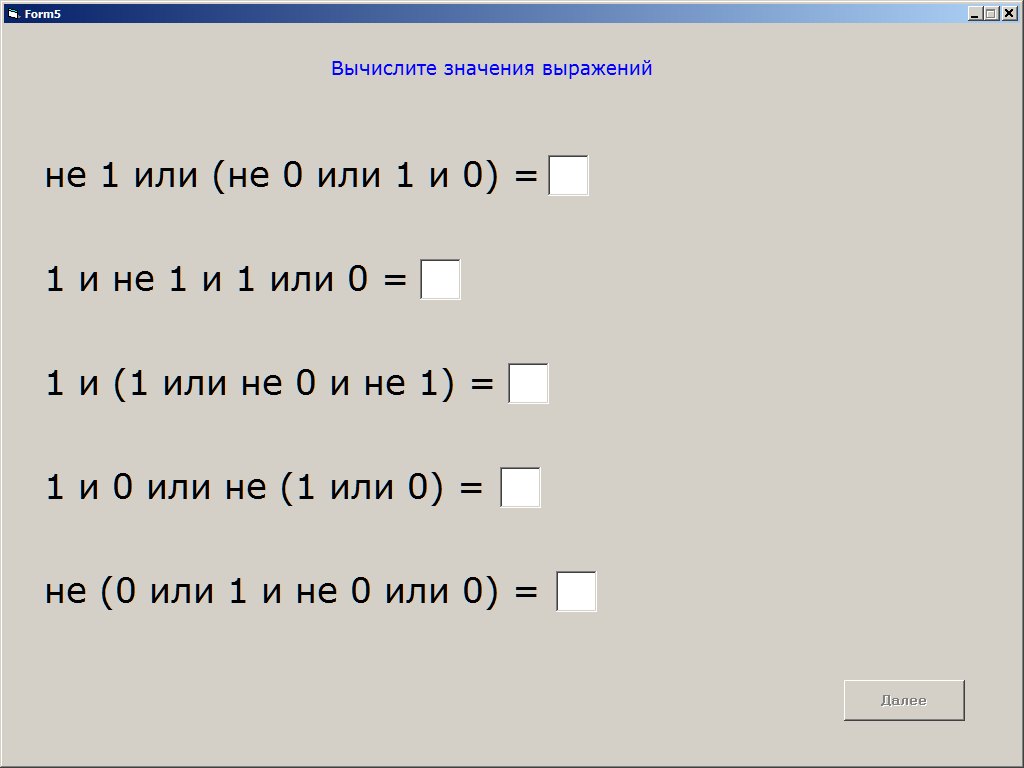
**Задание № 3. Нужно расставить порядок действий, соответственно расставить цифры 1,2,3,4,5,6**



**Задание № 4. Вычислите значения выражений, если А=1, B=0, C=1, D=0**



**Задание № 5. Вычислите значения выражений**



# Операции целочисленного деления в Паскале. Разбор задачи ЕГЭ № 24

Операции целочисленного деления div, mod предназначены только для работы целыми числами. Операция div определяет целую часть от деления, операция mod определяет остаток от деления.

Пример:

15 div 2 = 7

15 mod 2 = 1

Операции целочисленного деления div, mod играют большую роль при составлении программ.

Рассмотрим задачи, в которых используются div, mod [9]

**Задача № 1**

http://www.e-olymp.com/ru/problems/4716

**Делёж яблок - 1**

Условие:

n школьников делят k яблок поровну, неделящийся остаток остаётся в корзинке. Сколько яблок достанется каждому школьнику?

**входные данные**

Два положительных целых числа n и k, не превышающие 1500 - редко в школе бывает больше учеников, да и много яблок тоже кушать вредно...

**Выходные данные**

Вывести количество яблок, которое достанется каждому школьнику.

program pr1;

var r,k,n:integer;

begin

readln(n,k);

r:=k **div** n;

writeln(r);

end.

**Задача № 2**

**Делёж яблок - 2**

Условие:

n школьников делят k яблок поровну, неделящийся остаток остаётся в корзинке. Сколько яблок останется в корзинке?

**Входные данные**

Два положительных целых числа n и k, не больших 1500 - редко в школе бывает больше учеников, да и где найти такую корзинку?

**Выходные данные**

Вывести количество яблок, которое останется в корзинке.

program pr1;

var r,k,n:integer;

begin

readln(n,k);

r:=k **mod** n;

writeln(r);

end.

**Задача № 3. Найти сумму цифр натурального числа N, где N <109**

Program ex1;

Var N, digit, sum: longint;

begin

readln(N);

sum := 0;

while N > 0 do

begin

digit := N mod 10;

sum := sum+digit;

N := N div 10;

end;

writeln(sum)

end.

**Составим трассировочную таблицу для N=357**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **условие** | **digit** | **sum** | **N** |
| 1 | 357>0 | 357 mod 10= 7 | 0+7=7 | 357 div 10 =35 |
| 2 | 35>0 | 35 mod 10=5 | 7+5=12 | 35 div 10 =3 |
| 3 | 3>0 | 3 mod 10=3 | 12+3=15 | 3 div 10 = 0 |
| 4 | 0>0 |  |  |  |

**Решение задания ЕГЭ № 24 с использованием языка программирования Паскаль.[4]**

На обработку поступает натуральное число, не превышающее 109. Нужно написать программу, которая выводит на экран сумму цифр числа, не кратных 4. Если в числе нет цифр, не кратных 4, требуется на экран вывести «NO». Программист написал программу неправильно. Ниже приведена эта программа .

***Напоминание:*** 0 делится на любое натуральное число.

var N, digit, sum: longint;

begin

readln(N);

sum := N mod 10;

while N > 0 do

begin

digit := N mod 10;

if digit mod 4 > 0 then

sum := digit;

N := N div 10;

end;

if sum > 0 then

writeln(sum)

else

writeln('NO')

end.

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 437.

2. Приведите пример такого трёхзначного числа, при вводе которого

программа выдаёт верный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или

несколько). Известно, что каждая ошибка затрагивает только одну строку

и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант

строки.

Достаточно указать ошибки и способ их исправления для одного языка

программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе,

а не написать свою, возможно, использующую другой алгоритм решения.

Исправление ошибки должно затрагивать только строку, в которой находится

ошибка

**Решение**

1. Программа выведет число 3.

2. Программа выдаёт правильный ответ, например, для числа 407.

*Замечание для проверяющего. Программа работает неправильно из-за*

*неверной начальной инициализации суммы и неверного увеличения суммы.*

*Соответственно, программа будет работать верно, если в числе ровно одна*

*цифра не кратна 4 или таких цифр вообще нет и при этом число*

*заканчивается на 0.*

3. В программе есть две ошибки.

Первая ошибка. Неверная инициализация суммы (переменная sum).

Строка с ошибкой:

sum := N mod 10;

Верное исправление:

sum := 0;

Вторая ошибка. Неверное увеличение суммы.

Строка с ошибкой:

sum := digit;

Верное исправление:

sum := sum + digit;.

# Работа с массивами. Разбор задачи ЕГЭ № 25

Рассмотрим программу работы с массивами [4].

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число является однозначным. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 5; 10; 15; 3; 5 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Program pr1;

const

N = 20;

var

a: array [1..N] of integer;

i, j, k: integer;

begin

for i := 1 to N do

readln(a[i]);

...

end.

**Решение**

k := 0;

for i := 1 to N-1 do

if (a[i] < 10) or (a[i+1] < 10) then

inc(k);

writeln(k);

**При составлении данной программы типичной ошибкой является то, что школьники забывают поставить скобки (), в результате чего нарушается приоритет операций и программа работает неправильно.**

# ЗАКЛЮЧЕНИЕ

Одним из способов методической помощи учителям является создание и распространение методической продукции, описывающей наиболее рациональные, эффективные формы, методы организации образовательного процесса.

С целью оказания помощи учителю были подготовлены и методические рекомендации «Методика подготовки обучающихся к ЕГЭ и ОГЭ по информатике и ИКТ», в которых рассмотрена система работы учителя по предупреждению типичных ошибок при выполнении заданий ГИА по предмету, особенности решения задач по темам, которые вызывают наибольшие трудности при подготовке к ГИА по информатике и ИКТ («Среда программирования Кумир», «Логические операции», «Операции целочисленного деления в Паскале», «Работа с массивами»).

Надеемся, что данные методические рекомендации будут полезны учителям информатики в подготовке обучающихся к ОГЭ и ЕГЭ по информатике и ИКТ.

# ИНТЕРНЕТ-РЕСУРСЫ ДЛЯ ПОДГОТОВКИ К ЕГЭ И ОГЭ

<http://gia.edu.ru/>- официальный информационный портал ГИА 9 класс;

<http://www.ege.edu.ru> – официальный информационный портал ЕГЭ;

<http://fipi.ru/> – сайт Федерального института педагогических измерений;

<http://ege.yandex.ru/> – тренировочные online-тесты;

<http://kpolyakov.spb.ru> – сайт Полякова К.Ю.;

<http://inf.reshuege.ru>; <http://infoegehelp.ru> – разбор заданий, тренировочные online-тесты

# ЛИТЕРАТУРА

1. ЕГЭ 2015. Информатика. Типовые тестовые задания/В.Р. Лещинер.- М. Издательство «Экзамен», 2015 -223
2. Кирюхин В.М., Окулов С. М. Методика анализа сложных задач по информатике // Информатика и образование. 2006. №5. С. 29 – 41.
3. Кирюхин В.М., Окулов С. М. Методика решения задач по информатике. Международные олимпиады. – М.: БИНОМ. Лаборатория знаний, 2007. – 600 с.
4. Методические рекомендации по оцениванию выполнения заданий ЕГЭ с развернутым ответом по информатике и ИКТ в 11 классе на сайте [www.fipi.ru](http://www.fipi.ru)
5. Методические рекомендации по оцениванию выполнения заданий ОГЭ с развернутым ответом по информатике и ИКТ в 9 классе на сайте [www.fipi.ru](http://www.fipi.ru)
6. Окулов С.М. Основы программирования. – М.: БИНОМ. Лаборатория знаний, 2005. – 440 с.
7. Окулов С.М. Программирование в алгоритмах. – М.: БИНОМ. Лаборатория знаний. 2002. – 341 с.
8. Система программирования Кумир [**https://www.niisi.ru/kumir/**](https://www.niisi.ru/kumir/)
9. Соревнования - 29 января 2016. Соревнования младшая лига. Вступление в Cи [www.e-olymp.com](http://www.e-olymp.com)
10. Фоксфорд для учителей (онлайн курсы повышения квалификации для учителей) http://teacher.foxford.ru/