



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

ТЕМА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ:

Методика решения задач на массивы в курсе программирования для профильных  
классов

Выпускная квалификационная работа  
по направлению 44.03.05 Педагогическое образование (с двумя профилями  
подготовки)

Направленность программы бакалавриата

«Информатика. Английский язык»

Проверка на объем заимствований:

50,09 % авторского текста

Работа рекомендована к защите  
рекомендована/не рекомендована

«02» июня 2017.

и.о. зав. кафедрой И, ИТ и МОИ

А Рузаков А.А.

Выполнил:

Студент группы ОФ-513/093-5-1

Сандер Дмитрий Евгеньевич

Научный руководитель:

К.п.н., доцент кафедры ИИТиМОИ

Паршукова Н.Б.

Паршукова

Челябинск

2017



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ЮУрГГПУ»)**

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**ТЕМА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ:**

**Методика решения задач на массивы в курсе программирования для профильных  
классов**

**Выпускная квалификационная работа  
по направлению 44.03.05 Педагогическое образование (с двумя профилями  
подготовки)  
Направленность программы бакалавриата  
«Информатика. Английский язык»**

Проверка на объем заимствований:  
\_\_\_\_\_ % авторского текста

Работа \_\_\_\_\_ к защите  
рекомендована/не рекомендована

« \_\_\_ » \_\_\_\_\_ 20\_\_ г.  
и.о. зав. кафедрой И, ИТ и МОИ

\_\_\_\_\_ Рузаков А.А.

Выполнил:  
Студент группы ОФ-513/093-5-1  
Сандер Дмитрий Евгеньевич

Научный руководитель:  
К.п.н., доцент кафедры ИИТиМОИ  
Паршукова Н.Б.

**Челябинск  
2017**

## Оглавление

Введение.....	3
Глава 1. Теоретические основы изучения темы «Массивы» в курсе информатики и ИКТ.....	5
1.1. Место темы «Массивы» в курсе обучения информатике.....	5
1.2. Блок схемы .....	10
1.3. Демонстрационные материалы .....	15
1.4 Тема «Массивы» в современной учебной литературе.....	18
Выводы по 1 главе.....	27
Глава 2. Методическая поддержка курса «Массивы» при подготовке к ЕГЭ	28
2.1. Анализ ФГОС на тему «Массивы» .....	28
2.2. Примерная программа.....	30
2.3. Программно-методическая поддержка курса .....	37
2.4 Апробация результатов исследования.....	40
Выводы по 2 главе.....	41
Заключение .....	42
Список литературы .....	43
Приложение 1 .....	45

## Введение

Для изучения курса информатики в условиях внедрения ФГОС недостаточно использования традиционных средств обучения, требуется комплексное оснащение курса, включающего в себя электронный учебник, систему практических и тестовых заданий, дидактических материалов для самостоятельной работы и проектной деятельности обучающихся. Необходимость чёткого понимания темы «Массивы» объясняется её важностью при решении практических задач в случаях, когда в задаче приходится иметь дело с большим, но конечным количеством однотипных упорядоченных данных.

В разделе базового курса «Введение в программирование» структурированные величины, к которым относятся массивы, рассматриваются в ознакомительном порядке. Это обедняет содержание обучения, не дает возможность сформировать целостное представление о понятиях «информация», «данные», «обработка информации и данных». Кроме того, устойчивые знания и умения составлять программы для задач практического характера необходимы на экзаменах по информатике.

Немаловажную роль при изучении массивов в школьном курсе информатики занимает и выбор языка программирования. При выборе практических заданий обязательно нужно учитывать, что к моменту изучения данной темы «Массивы», учащиеся уже ознакомлены с одним из языков программирования: Pascal, Delphi или VisualBasic. А так же у учеников должны быть сформированы знания, умения и навыки самостоятельного выполнения некоторых задач по программированию, поэтому нужно уделить особое внимание: формированию технического задания, написанию алгоритма на определенном языке программирования, тестированию программ и модификации и усовершенствованию кода программ.

При изучении программирования школьника необходимо научить формально исполнять алгоритмы, отслеживать результаты выполнения программы, решать практические задачи с использованием массивов.

Все выше сказанное определяет актуальность темы работы «Методика изучения темы «Массивы» при изучении курса информатики и ИКТ».

**Целью работы** является совершенствование методики преподавания темы «Массивы» в школьном курсе информатика и ИКТ.

**Объектом исследования** является: изучение вопросов программирования в школьном курсе информатики.

**Предметом исследования** является: методика преподавания темы «Массивы» при изучении программирования в школьном курсе информатики.

**Задачи:**

- 1) Изучить теоретические основы решения задач по теме «Массивы»
- 2) Провести анализ изложения темы в нормативной и учебно-методической литературе
- 3) Разработать содержание уроков по решению задач по теме «Массивы»
- 4) Разработать информационную поддержку для обучения учащихся решению задач на массивы

**Методы исследования:** теоретический анализ литературных источников, педагогический эксперимент.

**Гипотеза** исследования состоит в том, что если в рамках изучения темы «Массивы» содержание и методы обучения будут определяться специально разработанной системой задач, то это позволит улучшить усвоение учениками вопросов обработки структур данных и качественнее подготовиться к решению задач по данной теме в ЕГЭ.

# **Глава 1. Теоретические основы изучения темы «Массивы» в курсе информатики и ИКТ**

## **1.1. Место темы «Массивы» в курсе обучения информатике**

Изучение программирования в школьном курсе Информатики включает в себя изучение темы «Массивы». Данная тема играет не маловажную роль, так как объединяет в себе многие основы программирования. Именно поэтому у многих учеников возникают проблемы с изучением данной темы.

Перед тем как приступить к изучению темы «массивы», ученики должны знать основы программирования, например, знать типы данных, команды ввода и вывода информации, уметь работать с блок-схемами, решать стандартные задачи на составление линейных программ, а так же составлять программы содержащие команды ветвления. Ко всему этому ученик должен освоить тему «Циклы». Уметь реализовывать циклические алгоритмы, решать стандартные задачи на циклы.

В учебнике по информатике за 10 класс, Угринович Н.Д., профильный уровень, 16 часов отведено теме «Массивы».

В данном учебнике, тема «Массивы» представлена на языке Паскаль - это язык структурного программирования. Паскаль является основой для ряда других языков, именно из-за этого лучше всего при изучении программирования в школе использовать именно его. Изучая данный язык программирования, ученики изучают множество тем. Ввод и вывод данных. Элементы языка и типы данных. Операции, функции, выражения. Операторы присваивания. Структуры алгоритмов. Программирование ветвлений. Программирование циклов. Вспомогательные алгоритмы и программы. Массивы. Типовые задачи обработки массивов. Метод последовательной детализации. Символьный тип данных. Строки символов. Комбинированный тип данных. Рекурсивные подпрограммы. Основные понятия объектно-ориентированного программирования.

Система программирования Delphi. Этапы программирования на Delphi. Программирование метода статистических испытаний. Построение графика функции.

Практикум: Операции, функции, выражения. Структуры алгоритмов и программ. Контрольная работа. Программирование линейных алгоритмов на Паскале. Программирование ветвлений. Отладка программ. Программирование циклов. Контрольная работа. Программирование циклов на Паскале. Разработка программ с использованием подпрограмм. Контрольная работа. Вспомогательные алгоритмы и процедуры. Массивы в языке Паскаль. Самостоятельная работа. Массивы. Типовые задачи обработки массивов. Типовые задачи обработки массивов. Решение задач ЕГЭ. Символьный тип данных. Строковый тип данных. Комбинированный тип данных. Рекурсивные подпрограммы. Создание консольного приложения. Создание оконного приложения. Программирование метода статистических испытаний. Построение графика функции

Учащиеся должны знать/понимать:

- 1) логическую символику;
- 2) свойства алгоритмов и основные алгоритмические конструкции;
- 3) тезис о полноте формализации понятия алгоритма;
- 4) основные конструкции языка программирования.

Учащиеся должны уметь:

- 1) вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний;
- 2) строить информационные модели объектов, систем и процессов, используя для этого типовые средства языка программирования.

Основные термины по разделу: Алгоритм. Вспомогательные алгоритмы. Комбинированный тип данных. Массивы. Метод последовательной детализации. Объектно-ориентированное

программирование. Парадигмы программирования. Рекурсивные подпрограммы. Символьный тип данных. Система программирования Delphi. Структурное программирование. Типы данных. Этапы программирования. Язык программирования Паскаль.

В учебнике по информатике за 10 класс, Семакин И.Г., профильный уровень, 10 часов отведено теме «Массивы».

#### *Дискретизация и кодирование (5 час)*

Дискретное (цифровое) представление результатов измерений, текстовой, графической, звуковой, видео информации. Цепочки (конечные последовательности символов) и операции над ними. Примеры систем двоичного кодирования различных алфавитов. Сигнал, кодирование, декодирование, сжатие. Скорость передачи информации. Зависимость скорости передачи от используемой полосы частот. Искажение информации при передаче и при сжатии.

#### *Системы, взаимодействие (3 час)*

Состояния объекта. Система, компоненты, взаимодействие компонентов. Информационное взаимодействие в системе. Графы, графы переходов, графы взаимодействия.

#### *Управление, обратная связь. (6 час)*

Управление в повседневной деятельности человека. Анализ и описание объекта с целью построения схемы управления; системы автоматического управления; задача выбора оптимальной модели управления; математические и компьютерные моделирование систем управления.

Примеры управления в социальных, технических, биологических системах. Команды управления и сигналы датчиков для учебных управляемых устройств, экранных объектов и устройств ИКТ.

#### *Моделирование и проектирование (13 час)*

Описания (информационные модели) объектов, процессов и систем,



соответствие описания реальности и целям описания. Фотографии, карты, чертежи, схемы, графы, таблицы, графики, формулы как описания. Использование описания (информационной модели) в процессах: общения, практической деятельности, исследования.

Математические модели, их использование для описания объектов и процессов живой и неживой природы и технологии, в том числе – в физике, биологии, экономике.

Связь между непрерывными моделями, их дискретными приближениями и компьютерными реализациями. Машинные представления целых и действительных чисел. Точность вычислений, интервальная арифметика.

Модели информационных процессов в технических, биологических и социальных системах. Моделирование, прогнозирование, проектирование в человеческой деятельности.

Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности. Использование инструментов автоматизированного проектирования.

#### *Логический язык (5 час)*

Имена, логические операции, кванторы, правила построения и семантика. Примеры записи утверждений на логическом языке. Логические формулы при поиске в базе данных. Дизъюнктивная нормальная форма. Логические функции. Схемы из функциональных элементов.

#### *Алгоритмический язык (6 час)*

Правила построения и выполнения алгоритмов. Разбиение задачи на подзадачи. Использование имен для алгоритмов и объектов. Примеры записи алгоритмов на алгоритмическом языке для графических и числовых исполнителей.

### *Вычислимые функции (4 час)*

Функции, вычисляемые алгоритмами. Полнота формализации понятия вычислимости. Универсальная вычислимая функция. Диагональные доказательства не существования. Индуктивные определения объектов. Задание вычислимой функции системой функциональных уравнений.

### *Детерминированные игры с полной информацией (4 час)*

Деревья. Выигрышная стратегия в игре. Игровая интерпретация логических формул.

### *Доказательства правильности (4 час)*

Соответствие алгоритма заданию (спецификации), инварианты, индуктивные доказательства.

### *Построение алгоритмов (4 час)*

Системы счисления, арифметические операции и перевод; кодирование с исправлением ошибок; генерация псевдослучайных последовательностей. Алгоритмы решения задач вычислительной математики (приближенные вычисления площади, значения функции, заданной рядом, моделирования процессов, описываемых дифференциальными уравнениями). Переборные алгоритмы. Обход дерева.

### *Типы данных (10 час)*

Основные конструкции. Матрицы (массивы). Работа с числами, матрицами, строками, списками, использование псевдослучайных чисел. Определяемые (абстрактные) типы данных.

### *Сложность описания объекта (2 час)*

Оптимальный способ описания. Алгоритмическое определение случайности.

## 1.2. Блок-схемы

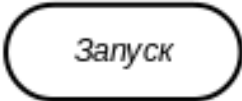

Для записи алгоритмов в школьном курсе часто используется два языка – псевдокод и язык блок-схем.

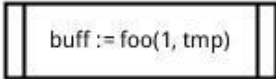
**Блок-схемы** – это способ графического представления алгоритма, в котором шаги изображаются в виде блоков различной формы, соединенных между собой стрелками.

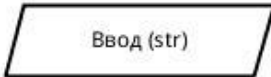
Условные обозначения в блок-схеме представлены в таблице 1

Таблица 1

Условные обозначения в блок-схеме

Название блока	Обозначение	Назначение блока
Терминатор		Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).
Процесс		Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записываются операции, например, присваивания: $a:=10*b+c$ .
Решение		отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий,

		<p>определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути. в общем случае – сравнение (три выхода: &gt;, &lt;, =); в программировании – условные операторы if (два выхода: true, false) и case (множество выходов).</p>
<p><b>Предопределенный процесс</b></p>		<p>символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и передаваемые в него данные. Например вызов процедуры или функции.</p>

<p><b>Данные</b></p>		<p>Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет носителя данных (для указания типа носителя данных используются специфические символы).</p>
----------------------	---	---

Пример составления блок-схемы показан на Рисунке 1.

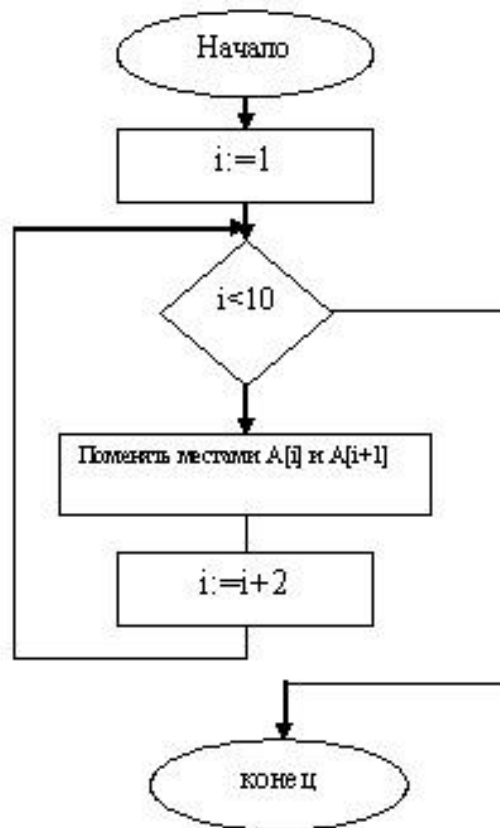


Рисунок 1. Пример блок-схемы

Любая блок-схема строится по определенному алгоритму действий, описывающих работу программы. Поэтому первоначально строится сама система. Алгоритм это последовательность операций для решения какой-либо задачи. В сущности, алгоритм это правило выполнения необходимых процессов обработки информации. Прежде чем приступить к построению алгоритма, требуется грамотно определить задачу: во-первых, что необходимо получить в результате, во-вторых, какая исходная информация

нужна, а какая уже имеется, и последнее - есть ли ограничения для ее получения. После этого составляется список определенных действий, которые необходимо осуществить для получения требуемого результата.

На практике чаще всего применяют следующие виды блок-схем:

- графическая, то есть в основе находятся геометрические символы;
- словесная: составляется с помощью обычных слов того или иного языка;
- псевдокоды: представляют собой полужформализованное описание на условно-алгоритмическом языке, которое включает в себя элементы языка программирования и фразы литературного, а также общепринятые математические символы;
- программная: для записи используются исключительно языки программирования.

Графическое представление последовательности действий включает в себя изображение алгоритма, описывающего связи функциональных блоков данной схемы, которые соответствуют выполнению одного либо нескольких действий. Блок-схема массива состоит из отдельных элементов, размеры и правила построения которых определены государственным стандартом. Для каждого типа действия (ввода данных, вычисления значений выражений, проверки условий, управления повторением действий, окончания обработки и др.) предусмотрена отдельная геометрическая фигура, представленная в виде блока. Эти символы соединяются линиями, определяющими очередность действий [8].

### 1.3. Демонстрационные материалы

В настоящее время практически невозможно найти область знаний, в которой не существовало бы электронных информационных мультимедийных энциклопедий, справочников и учебных пособий. Подавляющее большинство из них является гипермедиа-системой, сочетающей текст, статичные изображения (рисунки, фотографии), аудио и видеофрагменты.

Образовательная деятельность тесно связана с использованием всевозможных мультимедийных ресурсов. В этом случае решаются определенные задачи и учитываются различные целевые аудитории:

- обучаемые (например, в целях самостоятельного изучения предмета, в целях основного или дополнительного источника информации);
- преподаватели (например, как сырье при подготовке уроков, как источник для организации самостоятельной работы обучаемых);

Известные мультимедиа ресурсы можно классифицировать по различным признакам. Наиболее интересна градация этих документов

1. по степени доступности. В этом случае выделяют (а) ресурсы открытого доступа (открытые образовательные или справочные сайты Интернет, презентационные материалы, демонстрационные версии); (б) ресурсы ограниченного доступа (они доступны либо определенным целевым аудиториям – студентам учебных заведений, сотрудникам данной системы или корпорации; либо на определенных условиях – за плату, по запросу и пр.)

2. по материальному носителю. В этом случае можно назвать (а) ресурсы на компакт-дисках (в т.ч. на DVD); (б) ресурсы, доступные через Интернет (виртуальные? для данного пользователя).



В настоящее время число названий мультимедийных продуктов измеряется тысячами. Российский рынок мультимедийных продуктов значительно скромнее западного, хотя он, по данным экспертов, развивается довольно быстро. Если в первом издании российского справочника по CD-ROM и мультимедиа 1995 года перечислено всего 34 названия мультимедиа-продуктов образовательного назначения, то в издании 1996 года таких продуктов было уже более 112 экземпляров, а сейчас этот список составляет тысячи наименований.

В настоящее время созданы мультимедийные энциклопедии по многим учебным дисциплинам и образовательным направлениям. Разработаны игровые ситуационные тренажеры и мультимедийные обучающие системы, позволяющие организовать учебный процесс с использованием новых методов обучения.

Создается диалоговое кино, где пользователь может управлять ходом зрелища с клавиатуры дисплея посредством реплик, если к компьютеру подключено устройство распознавания речи. Технологии мультимедиа позволяют осмысленно и гармонично сочетать многие виды мультимедийной информации.

Использование мультимедийных ресурсов наиболее востребовано в настоящее время в преподавательской среде в качестве иллюстративного материала к урокам-лекциям. Безусловно, презентации, сопровождаемые красивыми изображениями или анимацией, являются визуально более привлекательными, нежели статический текст, и они могут поддерживать должный эмоциональный настрой, облегчающий восприятие и запоминание представляемого материала. Эти презентации призваны проиллюстрировать очный рассказ преподавателя.

Согласно исследованиям психологов, человек воспринимает информацию на слух, через взгляд, через моторику. Презентация, сопровождающая лекцию, позволяет максимально эффективно использовать особенности восприятия каждого обучающегося. Тем более, что

в ряде случаев проблемы материального плана могут быть решены за счет демонстрации виртуальной модели (например, химического или физического опыта). При этом очень опасно перегружать демонстрационные вспомогательные материалы урока-лекции текстами и изображениями.

Иллюстративный материал презентации может быть подготовлен самим преподавателем (сканирование статичных изображений, оцифровка аудио и видео ряда, использование специализированных или универсальных графических и анимационных программ и т.д.).

Возможно использование фрагментов готовых мультимедийных ресурсов. Ряд таких ресурсов размещен в Интернете. Соблюдение авторского права требует при использовании материалов, созданных другими авторами, указывать имена авторов или наименование источника.

Наиболее часто при подготовке презентаций используются универсальные программные продукты, входящие в состав Офиса. Но можно использовать и специализированные программы-оболочки для организации учебных гипертекстовых и гипермедиа-продуктов. Эти оболочки использует преподаватель, наполняет и настраивает их, учитывая специфику подготовки обучающихся, значимость и содержание дисциплины.

Более того, появляются полуфабрикаты, содержащие основную фактографическую и методическую канву изучаемой дисциплины, которую преподавателю следует лишь настроить.

Мультимедийные ресурсы являются перспективным и высокоэффективным инструментарием в образовательной области. Они предоставляют преподавателю массивы информации в большем объеме, чем традиционные источники информации, и в более наглядной форме. Система презентации может в интегрированном виде включать не только текст, графики, схемы, но и звук, анимацию, видео и т.п. Самым сложным для преподавателя является необходимость отбирать виды информации и

устанавливать последовательность, которая соответствует логике познания и уровню восприятия конкретного контингента обучающихся.

#### 1.4 Тема «Массивы» в современной учебной литературе

Программа по информатике для основной школы составлена на основе Федерального государственного образовательного стандарта общего образования. Требования к результатам освоения основной образовательной программы основного общего образования и необходимость разработки программы обусловлена, с одной стороны, потребностью развития информационных и коммуникационных технологий (ИКТ) и связанной с этим необходимостью уделить в курсе информатики большее внимание вопросам алгоритмизации и программирования, с другой стороны, пересмотром содержания общего образования в целом. При этом учитывается важная роль, которую играет алгоритмическое мышление в формировании личности.

В учебнике по информатике за 10 класс, Угринович Н.Д., профильный уровень, на программирование выделяется 66 часов, из них 16 часов отведено теме «Массивы».

Таблица 2

Часовое планирование темы «Массивы» по учебнику Угриновича Н.Д.

№	Тема	Основное содержание
1	Массивы. Заполнение массивов	Типы массивов и объявление массивов. Заполнение массива.
2	Проект «Заполнение массива». Создание графического интерфейса.	Постановка задачи. Создание графического интерфейса проекта.
3	Проект «Заполнение массива». Создание программного кода.	Создание программного кода проекта. Запуск проекта.

4	Поиск элемента в массивах	Поиск элемента в массиве. Поиск максимального или минимального элемента.
5	Проект «Поиск в массиве». Создание графического интерфейса.	Постановка задачи. Создание графического интерфейса проекта.
6	Проект «Поиск в массиве». Создание программного кода.	Создание программного кода проекта. Запуск проекта.
7	Сортировка числовых массивов	Алгоритмы сортировки числовых массивов.
8	Проект «Сортировка числового массива». Создание графического интерфейса.	Постановка задачи. Создание графического интерфейса проекта.
9	Проект «Сортировка числового массива». Создание программного кода.	Создание программного кода проекта. Запуск проекта.
10	Сортировка строковых массивов	Алгоритмы сортировки строковых массивов.
11	Проект «Сортировка строкового массива».	Постановка задачи. Создание графического интерфейса проекта.
12	Проект «Сортировка строкового массива».	Создание программного кода проекта. Запуск проекта.
13	Решение задач по разделу «Алгоритмизация и объектно-ориентированное программирование»	Решение задач на программирование.
14	Решение задач по разделу «Алгоритмизация и объектно-ориентированное	Решение задач на программирование.

	программирование»	
15	Решение задач по разделу «Алгоритмизация и объектно-ориентированное программирование»	Решение задач на программирование.
16	Контрольная работа «Алгоритмизация и объектно-ориентированное программирование»	Алгоритмизация и объектно-ориентированное программирование. Решение задач.

В учебнике по информатике за 10 класс, Семакин И. Г., профильный уровень, на программирование выделяется 66 часов, из них 10 часов отведено теме «Массивы» [11].

Таблица 3

Часовое планирование темы «Массивы» по учебнику Семакина И.Г.

№		Наименование разделов и тем	Все го час ов	Из них		
урока	п/п			Лаб. и практ. раб.	Конт рол. раб	Само стоят. раб.
39	2.2.2 3	Массивы в языке Паскаль. Одномерные массивы	1	0,5		
40	2.2.2 4	Массивы в языке Паскаль. Одномерные и двумерные массивы	1	0,5		
41	2.2.2 5	Урок-практикум. Массивы в языке Паскаль	1	1		
42	2.2.2 6	Самостоятельная работа на тему «Массивы»	1			1
43	2.2.2 7	Типовые задачи обработки массивов	1			

44	2.2.2 8	Типовые задачи обработки массивов	1	1		
45	2.2.2 9	Типовые задачи обработки массивов	1	1		
46	2.2.3 0	Типовые задачи обработки массивов	1	1		
47	2.2.3 1	Типовые задачи обработки массивов. Решение задач ЕГЭ	1	1		
48	2.2.3 2	Типовые задачи обработки массивов. Решение задач ЕГЭ	1	1		

Массив –это пронумерованная последовательность величин одинакового типа, обозначаемая одним именем. Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом. Каждое из значений, составляющих массив, называется его компонентой (или элементом массива).

Массив данных в программе рассматривается как переменная структурированного типа. Массиву присваивается имя, посредством которого можно ссылаться как на массив данных в целом, так и на любую из его компонент [9].

В общих чертах, массив это однородный, упорядоченный структурированный тип данных с прямым доступом к элементам. Использование массивов позволяет заменить большое количество индивидуальных имен каждого объекта одним групповым именем набора данных, вслед за которым в круглых (QBasic) или квадратных (Си, Паскаль) скобках задаются один или несколько индексов, определяющих местоположение требуемого значения. Естественно, что такая возможность упрощает и массовую обработку данных в соответствующих циклах программы.

В большинстве задач приходится иметь дело с массивами, элементами которых являются числа того или иного типа, символы и строки фиксированной длины. Однако приходится обрабатывать и битовые массивы, каждый элемент которых представлен одним или несколькими двоичными разрядами, соответствующими кодам цветности пикселей графических изображений. Некоторые системы программирования позволяют работать с массивами, состоящими из неоднородных элементов. Например, с одномерными массивами, каждый элемент которых представлен строкой переменной длины.

При решении практических задач часто приходится иметь дело с различными таблицами данных, математическим эквивалентом которых служат матрицы. Такой способ организации данных, при котором каждый элемент определяется номером строки и номером столбца, на пересечении которых он расположен, называется двумерным массивом или таблицей.

Например, данные о планетах Солнечной системы представлены следующей таблицей:

Таблица 4

Данные о планетах солнечной системы

Планета	Расст. до Солнца	Относ.объем	Относ.масса
Меркурий	57.9	0.06	0.05
Венера	108.2	0.92	0.81
Земля	149.6	1.00	1.00
Марс	227.9	0.15	0.11
Юпитер	978.3	1345.00	318.40
Сатурн	1429.3	767.00	95.20

Их можно занести в память компьютера, используя понятие двумерного массива. Положение элемента в массиве определяется двумя индексами. Они показывают номер строки и номер столбца. Индексы

разделяются запятой.

При выполнении инженерных и математических расчетов часто используются переменные более чем с двумя индексами. При решении задач на ЭВМ такие переменные представляются как компоненты соответственно трех-, четырехмерных массивов и т.д.

Однако описание массива в виде многомерной структуры делается лишь из соображений удобства программирования как результат стремления наиболее точно воспроизвести в программе объективно существующие связи между элементами данных решаемой задачи. Что же касается образа массива в памяти ЭВМ, то как одномерные, так и многомерные массивы хранятся в виде линейной последовательности своих компонент, и принципиальной разницы между одномерными и многомерными массивами в памяти ЭВМ нет. Однако порядок, в котором запоминаются элементы многомерных массивов, важно себе представлять. В большинстве алгоритмических языков реализуется общее правило, устанавливающее порядок хранения в памяти элементов массивов: элементы многомерных массивов хранятся в памяти в последовательности, соответствующей более частому изменению младших индексов.

Самой распространенной структурой, реализованной практически во всех языках программирования, является массив.

В начале изучения темы «Массивы» мотивом её освоения может являться задача, требующая большого количества ввода данных и последующей их обработки. Например, учащимся предлагается вычислить среднюю температуру за месяц, используя данные показаний за каждые сутки. Очевидно, что решение данной задачи требует использования одномерного массива в качестве хранилища данных. Совместно с учителем учащиеся формулируют понятие одномерного массива, а затем изучают его структуру и способ описания на языке программирования. Точно так же можно подвести учащихся к необходимости использования двумерного массива (задача, где данные представлены в форме таблицы).



Далее, учащиеся совместно с учителем формулируют цели изучения массивов, изучают структуру описания нового типа данных «массив», например, на языке Паскаль.

Логическим продолжением изучения данной темы может служить совместный с учащимися разбор алгоритмов заполнения массивов с помощью оператора ввода, генератора случайных чисел, используя операторы цикла для обработки каждого элемента массива. Важно дать возможность каждому учащемуся помочь выбрать приемлемый метод заполнения массивов, показать преимущества и недостатки каждого из них. Вести совместное обсуждение удобно, выведя с помощью проектора каждый новый метод заполнения на экран и проводя своеобразную презентацию (можно даже в игровой форме) каждого из них.

На следующем этапе учащиеся разбирают готовые программы и отвечают на поставленный в задании вопрос. Несколько заданий учитель разбирает вместе с учащимися. Примером конкретного набора задач этого этапа (на примере одномерных массивов) может служить следующая таблица:

Таблица 5

Примеры задач

<p><b>Задание_1.</b> Сколько элементов массива В будут иметь положительные значения?</p> <pre> <b>for</b>    <b>n:=1</b>    <b>to</b>    <b>100</b>    <b>do</b>     <b>A[n]:=n-10;</b> <b>for</b>    <b>n:=1</b>    <b>to</b>    <b>100</b>    <b>do</b>     <b>B[n]:=A[n]*n;</b> </pre> <p>1) 10 2) 50 3) <b>90</b> 4) 100</p>	<p><b>Алгоритм решения (устно):</b></p> <p>1. Составим частичную трассировочную таблицу для первого цикла:  <math>A = \{-9, -8, \dots, -1, 0, 1, 2, \dots, 90\}</math> - массив содержит 90 положительных чисел.</p> <p>2. При выполнении второго цикла количество положительных чисел не изменится, т.к. элементы массива умножаются на их</p>
--	---

	<p>порядковый номер.</p> <p><b>3. Ответ = 3</b></p>																																																		
<p><b>Задание_2.</b> Чему будут равны элементы этого массива после выполнения фрагмента программы?</p> <pre> <b>for i:=0 to 9 do A[i]:=9-i;</b> <b>for i:=0 to 4 do begin</b>   <b>k:=A[i];</b>   <b>A[i]:=A[9-i];</b>   <b>A[9-i]:=k; end;</b> </pre> <p><b>Ответ: 0 1 2 3 4 5 6 7 8 9</b></p>	<p><b>Алгоритм решения:</b></p> <p>Составим трассировочную таблицу. Во второй строке таблицы запишем исходное состояние массива. В третьей строке – новое значение массива после выполнение второго цикла. Вспомогательная переменная k используется для хранения текущего элемента массива во время обмена значениями двух переменных.</p> <table border="1" data-bbox="890 1055 1444 1377"> <tr> <td>I=0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>A[i]=</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>A[i]=</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>	I=0	1	2	3	4	5	6	7	8	9	A[i]=	8	7	6	5	4	3	2	1	0	9										A[i]=	1	2	3	4	5	6	7	8	9	0									
I=0	1	2	3	4	5	6	7	8	9																																										
A[i]=	8	7	6	5	4	3	2	1	0																																										
9																																																			
A[i]=	1	2	3	4	5	6	7	8	9																																										
0																																																			
<p><b>Задание_3.</b> Значения элементов массива A[1..15] задаются с помощью следующего фрагмента программы. Какие значения будут присвоены элементам A[5], A[8]?</p> <pre> <b>for n:=1 to 15 do</b>   <b>A[n]:=sqr(n);</b> <b>for n:=1 to 15 do begin</b>   <b>if n mod 2 &lt;&gt; 0</b>   <b>then A[n] := sqrt(A[n])</b> </pre>	<p><b>Алгоритм решения:</b></p> <p>Составим частичную трассировочную таблицу.</p> <table border="1" data-bbox="890 1574 1444 1899"> <tr> <td>N=1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td> </tr> <tr> <td>A[n]=</td><td>4</td><td>9</td><td>1</td><td>2</td><td>3</td><td>4</td><td>6</td> </tr> <tr> <td>1</td><td></td><td></td><td>6</td><td>5</td><td>6</td><td>9</td><td>4</td> </tr> <tr> <td>A[n]=</td><td></td><td></td><td></td><td>5</td><td></td><td></td><td>1</td> </tr> <tr> <td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p>Если индекс элемента является нечетным числом, то новое</p>	N=1	2	3	4	5	6	7	8	A[n]=	4	9	1	2	3	4	6	1			6	5	6	9	4	A[n]=				5			1	1																	
N=1	2	3	4	5	6	7	8																																												
A[n]=	4	9	1	2	3	4	6																																												
1			6	5	6	9	4																																												
A[n]=				5			1																																												
1																																																			

<pre>else A[n] := A[1];       end</pre>	<p>значение элемента – квадратный корень из исходного элемента массива, в противном случае – становится равным первому элементу(1).</p> <p style="text-align: center;"><b>Ответ A[5]=5, A[8]=1</b></p>
---	--

Следующий этап – знакомство с типовыми алгоритмами обработки массивов и формированием умений использовать эти алгоритмы при решении практических задач. Представленные задачи учитель разбирает вместе с учащимися, делая акцент не только на самом алгоритме, но и на его временной эффективности и объёме затрачиваемой памяти. Алгоритм решения обсуждается устно, затем оформляется программа. Дополнением может служить возможность использования этих алгоритмов в рамках других дисциплин с целью оптимизации затрат на обработку каких-либо данных (практическая значимость и межпредметная связь).

Следующий этап – решение практических задач, требующих умений применять при разработке программы различные типовые алгоритмы. Такие задачи встречаются и в экзаменационных материалах. Алгоритмы решения обсуждаются устно в форме дискуссии, находятся оптимальные пути решения каждой задачи.

Ярким примером использования такого метода обучения в рамках ФГОС как работа в группах может служить изучение различных способов сортировки массивов. Учащиеся, разбитые в группы (по 5-6 человек) получают карточки, на которых размещается какой-то способ сортировки (его программная реализация) и практическая задача, к которой этот способ следует применить. По окончании работы в группах, представители каждой из них презентуют свой способ сортировки (можно с привлечением остальных участников группы показать, как он работает), а также показывают решение данной им задачи. Происходит совместное

обсуждение данного решения, разбираются возможные ошибки и недочёты.

После завершения изучения темы «Массивы» учащимся в рамках проектной деятельности может быть предложено самостоятельное изучение массивов, применяя другой язык программирования или специальные программы.

Таким образом, на примере темы «Массивы» была показана методика их изучения в рамках реализации ФГОС с привлечением как традиционных, так и методов обучения в рамках деятельностного подхода.

### **Выводы по 1 главе**

В данной главе были рассмотрены основные понятия, касающиеся изучения темы «Массивы», и были рассмотрены блок-схемы, которые являются неотъемлемой частью изучения программирования в школе. Также были проанализированы основные курсы изучения данной темы в современных учебниках информатики, которые используются в нынешних учебных программах.

Как результат изложения материала в первой главе, были рассмотрены необходимый теоретический материал, для формирования и наполнения школьного факультативного курса «Массивы».

## **Глава 2. Методическая поддержка курса «Массивы» при подготовке к ЕГЭ**

### **2.1. Анализ ФГОС на тему «Массивы»**

В соответствии с Федеральным государственным образовательным стандартом основного общего образования (ФГОС ООО) курс информатика входит в предметную область «Математика и информатика». Курс информатики основной школы является частью непрерывного курса информатики, который включает в себя также пропедевтический курс в начальной школе и обучение информатике в старших классах (на базовом или профильном уровне). К концу обучения начальной школы (в соответствии с ФГОС начального общего образования) обучающиеся должны обладать ИКТ — компетентностью, достаточной для дальнейшего обучения. В основной школе, начиная с 5-го класса, они закрепляют полученные технические навыки и развивают их в рамках применения при изучении всех предметов. Исходя из конкретных условий, образовательное учреждение, может начинать изучение курса информатики с 5 класса за счет часов школьного учебного плана, выстраивая непрерывный курс информатики в 5–9 классах, обеспечивая его преемственность с курсом информатики начальной школы.

Общеобразовательный курс информатики – один из основных предметов, способный дать обучающимся методологию приобретения знаний об окружающем мире и о себе, а так же обеспечить эффективное развитие общеучебных умений и способов интеллектуальной деятельности на основе методов информатики, становление умений и навыков информационно-учебной деятельности на базе средств ИКТ для решения познавательных задач и саморазвития. Вместе с математикой, физикой, химией, биологией курс информатики закладывает основы естественнонаучного мировоззрения. Информатика имеет очень большое и всё возрастающее число междисциплинарных связей, причем как на

уровне понятийного аппарата, так и на уровне инструментария. Многие положения, развиваемые информатикой, рассматриваются как основа создания и использования информационных и коммуникационных технологий – одного из наиболее значимых технологических достижений современной цивилизации.

Системный характер содержания курса информатики определяется названными тремя сквозными направлениями (представленными в несколько обобщенном виде):

- области применения методов и средств информатики.
- моделирование, информационные модели;
- информация и информационные процессы;

В рамках этих направлений можно выделить следующие основные содержательные линии курса информатики:

а) в направлении «Моделирование, информационные модели»:

- алгоритмизация и программирование;
- представление информации;
- моделирование и формализация;

б) в направлении «Области применения методов и средств информатики»:

- информационные и коммуникационные технологии;
- информационные основы управления;
- информационная цивилизация.

в) в направлении «Информация, информационные процессы»:

- информационные процессы;
- информационные ресурсы;

Предмет информатики ведется в 10-11 классах (по 2 часа в неделю), всего на курс отводится 136 часов, из них на изучение программирования - 60 часов.

При изучении программирования учащиеся должны познакомиться с основами языка программирования TurboPascal, средой

программирования, структурой программы, используемыми типами данных и т. д.

Основной упор делается на отработку практических навыков при решении конкретных задач, создание и отладку небольших программ.

В курсе программирования одно из основных мест занимает тема «Массивы», так как на основе обработки табличных величин построена основная масса программ[11].

## **2.2. Примерная программа**

На основе сделанных анализов школьной литературы, была составлена рабочая программа, которая отвечает представленным ФГОС требованиям. Следует так же отметить, что данный курс уже был использован на практике в школе в курсе информатики и ИКТ в одиннадцатых классах. Так же нужно отметить различную степень подготовленности учеников: данные уроки проводились как в профильном классе, так и в классе с базовым уровнем изучения информатики и ИКТ, при этом подавляющее большинство учеников успешно освоили данный раздел. Об успешности курса можно утверждать, основываясь на результатах самостоятельных работ – средний балл по итогам проведения контрольных мероприятий 4.4.

По данной теме были разработаны 4 урока: два из них основаны на усвоении новых знаний, другие 2 для закрепления пройденного материала.

Для изучения темы массивы было взято 2 темы «Одномерные массивы» и «Двумерные массивы». Основой данного курса является рабочая программа по учебнику Семакина И.Г..

Урок №1.

**Тема урока:** Алгоритмизация и программирование. Одномерные массивы целых чисел. Описание, ввод и вывод элементов одномерного массива.

**Класс:**11.

**Тип урока:** Объяснение и первичное закрепление знаний.

**Цель урока:** создать условия для формирования у обучающихся умений и навыков по теме «Одномерные массивы целых чисел».

**Формы организации учебной деятельности обучающихся:**

- Фронтальная (при повторении материала)
- Индивидуальная (при выполнении практической работы).

**Перечень средств ИКТ, используемых на уроке:**

- АРМ учителя;
- Интерактивная доска;
- Презентация «Одномерные массивы целых чисел».

**Основное понятие:** Массив – это упорядоченное множество однотипных переменных (элементов массива), которым можно присвоить общее имя, различающихся номерами (индексами).

**ЦОРы для урока:** презентация «Одномерные массивы»

**Контрольные вопросы:**

- Что такое массив?
- Как правильно объявить массив?
- Сколько есть способов заполнить массив и в чем их особенности?

**Методические рекомендации к уроку:** так как этот урок является основополагающим при изучении темы «Массивы», обязательно нужно



провести актуализацию знаний учащихся, задав вопросы по предыдущим темам изучения программирования. Ученики должны уверенно отвечать на следующие вопросы:

- 1) Какие типы данных вы знаете?
- 2) Где находится раздел описания переменных?
- 3) Какие бывают типы циклов?

Это минимальный уровень знаний, которыми должны обладать ученики при изучении данной темы. Если же ученики затрудняются в ответе, или не могут ответить на поставленные вопросы, необходимо напомнить им, под запись в тетрадь и задать дополнительное задание на дом – выучить типы данных, разделы программы и типы циклических алгоритмов.

## Урок №2

**Тема урока:** Алгоритмизация и программирование. Одномерный массивы для целых чисел. Описание, ввод и вывод элементов одномерного массива.

**Класс:**11.

**Тип урока:**самостоятельная работа учащихся;

**Цель урока:** проверить знания, умения и навыки учеников работы с одномерными массивами.

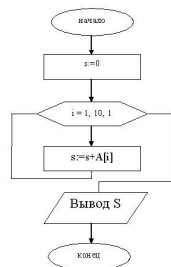
**ЦОРы для урока:** самостоятельная работа №1

**Задания для выполнения:**

<b>Задача №1.</b> Выполнить следующие действия: 1) Создать одномерный массив	<b>Задача №2.</b> Выполнить следующие действия: 1) Создать одномерный массив А
---	---

А из 10 целых чисел (числа вводит пользователь)

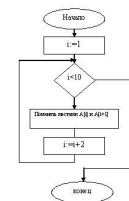
- вывести на экран массив в виде строки чисел;
- подсчитать сумму элементов массива (блок-схема изображена



на рисунке 1 файл )  
вывести сумму на экран.

из 10 целых чисел (числа вводит пользователь)

- вывести на экран массив в виде строки чисел;
- поменять местами элементы массива (блок-схема алгоритма показана на



рисунке 2 файл  
) следующим образом

- 1-ый со 2-ым
- 3-ий с 4-ым
- 5-ый с 6-ым
- 7-ой с 8-ым
- 9-ый с 10-ым

- вывести измененный массив на экран.

**Методические рекомендации к уроку:** проведение данной самостоятельной работы должно проводиться в индивидуальном порядке. Задания не должны вызывать сложности при выполнении, если же они есть, нужно напомнить ученикам необходимую информацию, так как данная самостоятельная работа включена для закрепления пройденного материала. Ученики должны хорошо освоить данную тему, это очень важно для продолжения изучения темы «Массивы»

## Урок №3

**Тема урока:** Алгоритмизация и программирование. Двумерные массивы.

**Класс:** 11.

**Тип урока:** Объяснение и первичное закрепление знаний.

**Цель урока:** создать условия для формирования у обучающихся умений и навыков по теме «Двумерные массивы».

**Формы организации учебной деятельности обучающихся:**

- Фронтальная (при повторении материала)
- Индивидуальная (при выполнении практической работы).

**Перечень средств ИКТ, используемых на уроке:**

- АРМ учителя;
- Интерактивная доска;
- Презентация «Одномерные массивы целых чисел».

**ЦОРы для урока:** презентация «Двумерные массивы»

**Контрольные вопросы:**

- Чем отличаются двумерный и одномерный массив?
- Что такое размерность массива?
- Какие действия можно совершать над двумерными массивами?

**Методические рекомендации к уроку:** при изучении темы «двумерные массивы» важной составляющей является наглядность изучения темы, очень важно приводить простые примеры из жизни, например: Зрительный зал кинотеатра – двумерный массив. Название кинотеатра – *имя массива*. Ряд, место – *индексы*. Человек на определенном месте – *элемент массива*. Это необходимо для лучшего усвоения темы, так

как наглядность при изучении любого материала играет большую роль на результаты изучения.

Урок №4

**Тема урока** Алгоритмизация и программирование. Двумерные массивы.

**Класс:** 11.

**Тип урока:** самостоятельная работа учащихся;

**Цель урока:** проверить знания, умения и навыки учеников работы с двумерными массивами.

**ЦОРы для урока:** самостоятельная работа №2

**Задания для выполнения:**

Задание	Варианты ответа
1. Укажите правильное описание массива:	<ul style="list-style-type: none"><li>• a: array {1..n} of integer;</li><li>• a: aray [1..n, 1..m] of integer.</li><li>• a: array [1..n, 1..m] of real;</li><li>• a: array [1...n, 1...m] of real;</li></ul>
2. Что определяет для массива X[1..n, 1..n] следующий алгоритм S:=0; For i:= 1 to n do For j:= 1 to n do If i =j then S: = S+X[i, j];	<ul style="list-style-type: none"><li>• нечетные строки матрицы заменить на A;</li><li>• сумма элементов главной диагонали матрицы;</li></ul>

	<ul style="list-style-type: none"> <li>• сумма элементов побочной диагонали матрицы;</li> <li>• четные столбцы матрицы заменить на A;</li> </ul>
<p>3. Что определяет для массива X[1..n] следующий алгоритм S:= 0; For i:= 1 to n do If X[i] &gt;0 then S:=S+X[i];</p>	<ul style="list-style-type: none"> <li>• количество отрицательных элементов массива;</li> <li>• максимальный элемент массива;</li> <li>• минимальный элемент массива;</li> <li>• сумму положительных элементов массива;</li> </ul>
<p>4. Что определяет для массива X[1..n, 1..m] следующий алгоритм For i:= 1 to n do For j:= 1 to m do If j mod 2 = 0 then X[i, j] = A;</p>	<ul style="list-style-type: none"> <li>• минимальный элемент массива;</li> <li>• четные столбцы матрицы заменить на A;</li> <li>• нечетные строки матрицы заменить на A.</li> <li>• четные строки матрицы заменить на A;</li> </ul>
<p>5. Что определяет для массива X[1..n] следующий алгоритм R:= 0; For i:= 1 to n do If X[i] &gt;T then R:=R+1;</p>	<ul style="list-style-type: none"> <li>• номер последнего элемента массива, большего T;</li> <li>• количество элементов массива, больших T;</li> </ul>

	<ul style="list-style-type: none"> <li>• номер первого элемента массива, меньшего T;</li> <li>• номер первого элемента массива, большего T;</li> </ul>
--	--

**Методические рекомендации к уроку:** проведение данной самостоятельной работы должно проводиться в индивидуальном порядке. Так же необходимо предоставить ученикам возможность проверять задачи в программе. Это поможет им для закрепления навыков работы с языком программирования Pscal, а так же способствует успешному закреплению знаний по данной теме.

Данный курс рассчитан на внедрение в школьную программу изучения курса массивы.

### **2.3. Программно-методическая поддержка курса**

В качестве программно-методической поддержки раздела был разработан сайт. Сайт разработан на основе WordPress. Главная страница сайта <http://t92273gn.beget.tech> представлена Рисунок 2.

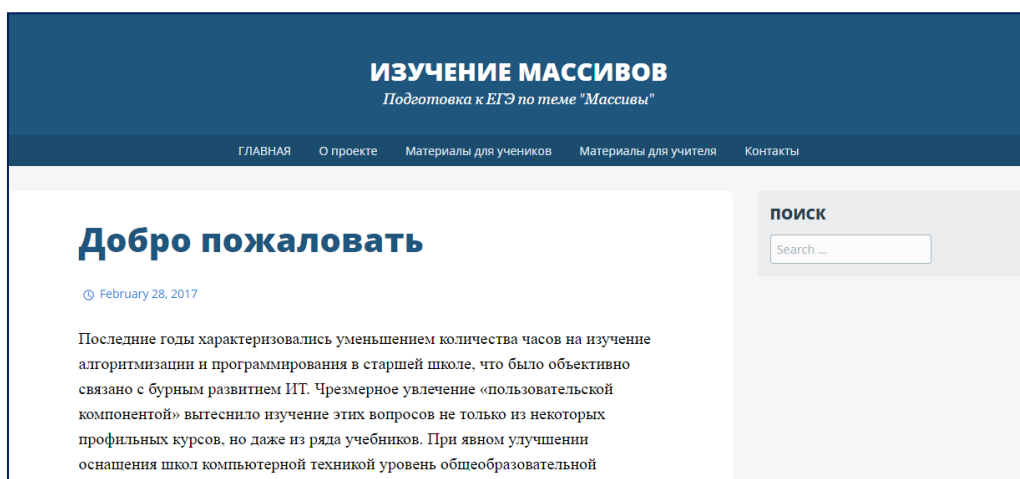


Рисунок 2. Главная страница

На главной странице сайта расположено меню, которое состоит из 5 вкладок: «Главная», «О проекте», «Материалы для учеников», «Материалы для учителя» и «Контакты».

Во вкладке «Материалы для учителя» представлены разработки уроков на темы «Одномерный массив» и «Двумерный массив», к каждой теме прилагается конспект урока с сопровождающей презентацией.

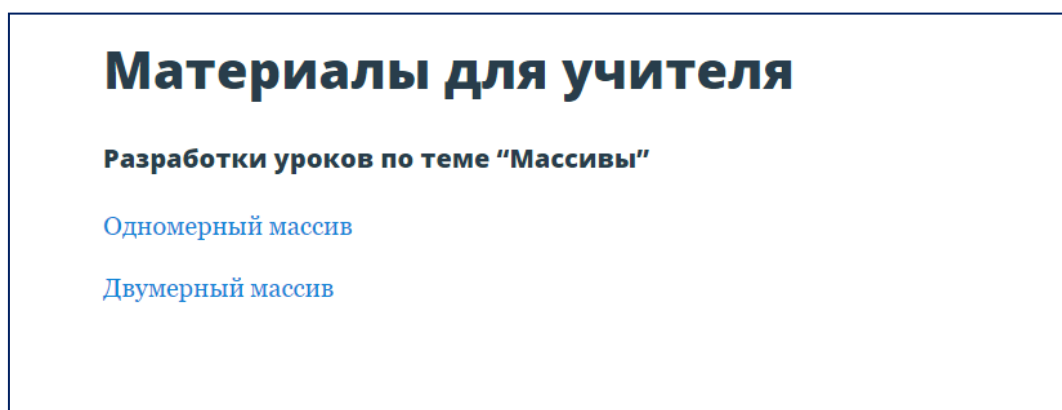


Рисунок 3. Информация для учителей

Во вкладке «Материалы для учеников» представлены разработки самостоятельных работ для учеников для контроля и проверки пройденных материалов.

# Материалы для учеников

Самостоятельная работа №1

Самостоятельная Работа №2

Рисунок 4. Раздаточный материал для учеников

Вторая самостоятельная работа была разработана с помощью бесплатного сервиса для создания приложений learningapps. Внешний вид представлен на Рисунке 5.

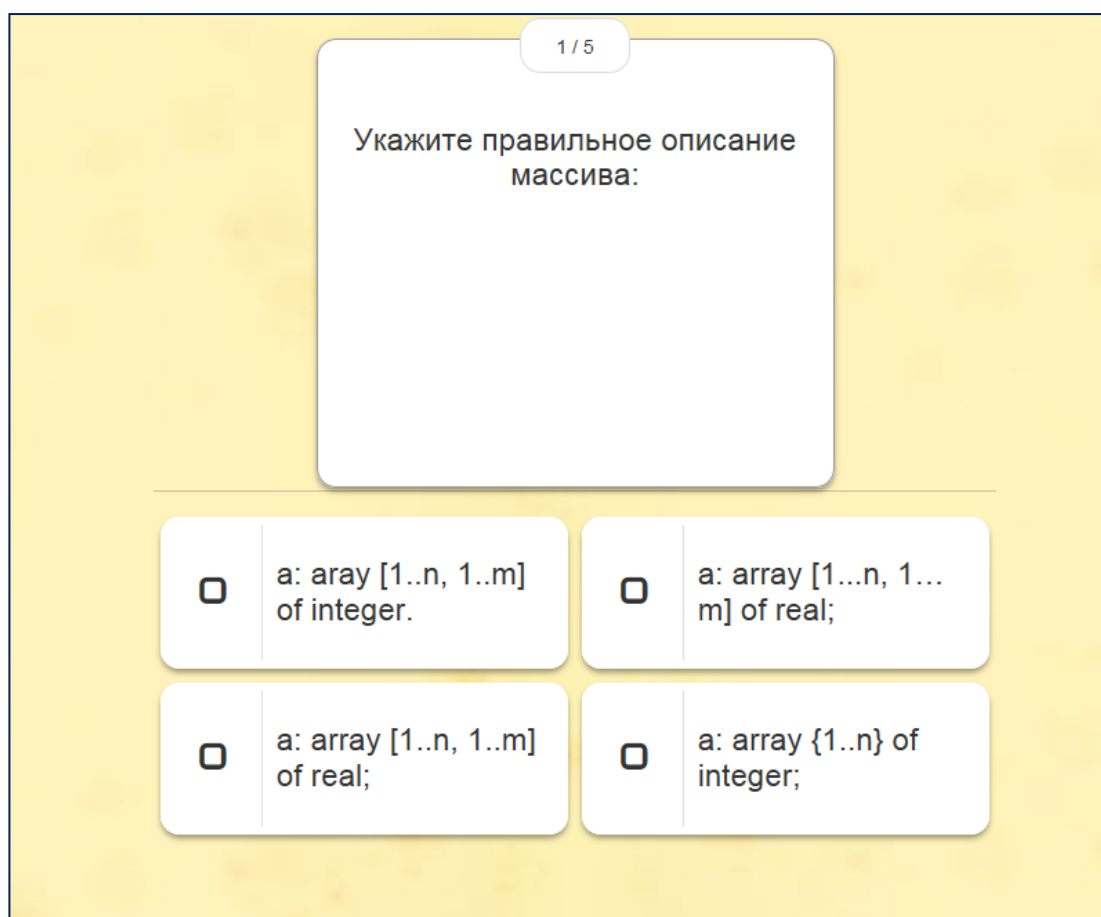


Рисунок 5. Приложение learningapps



## **2.4 Апробация результатов исследования**

Педагогическая апробация проводилась во время педагогической практики в МБОУ СОШ № 151 г. Челябинска. Модуль к разделу изучался как в профильном, так и в базовом классах. В течении двух занятий были рассмотрены темы:

1. Урок № 1 «Одномерные массивы» – 1 час.
2. Урок № 2 Самостоятельная работа №1 – 1 час.

Апробация модуля к разделу прошла успешно. Способствовала этому правильная мотивация, цели и задачи изучения темы.

## **Выводы по 2 главе**

На основе теоретических положений, изложенных в первой главе, во второй главе представлено описание темы «Массивы», выполнено в виде сайта для учителя и учеников.

Апробация курса проводилась в рамках педагогической практики в МБОУ СОШ № 151. На занятиях ученики 11 классов с удовольствием выполняли задания, активно отвечали на вопросы.

Таким образом, во второй главе исследования был разработан и апробирован модуль к разделу «Массивы» и программно-методическая поддержка к нему.

## Заключение

Подводя итоги данной работы, важно отметить, что проведенное исследование направлено на увеличение наглядности при изучении раздела «Массивы». Актуальность исследования обусловлена тем, что в нынешнее время информационные технологии имеют большое распространение в мире и при этом постоянно развиваются. Знания по теме «Массивы» помогут ученикам не только при сдаче ЕГЭ, а так же поможет в их личном развитии.

В итоге работы была достигнута цели, изучена литература по проблеме исследования. Разработан модуль к разделу «Массивы» и программно-методический комплекс.

В процессе исследования были выполнены поставленные задачи и получены следующие результаты:

- 1) Изучены теоретические основы решения задач по теме «Массивы»
- 2) Проведены анализы изложения темы в нормативной и учебно-методической литературе
- 3) Разработаны содержания уроков по решению задач по теме «Массивы»
- 4) Разработана информационная поддержка для обучения учащихся решению задач на массивы

В подтверждении гипотезы можно сказать, что данный модуль позволяет повысить наглядность при изучении раздела «Массивы».

Таким образом, поставленные задачи можно считать выполненными и можно сделать вывод о верности поставленной гипотезы.

## Список литературы

1. Андреева Е. В., Босова Л. Л., Фалина И. Н. Математические основы информатики. Элективный курс: Учебное пособие – М.: БИНОМ. Лаборатория знаний, 2005. – 328 с.
2. Бешенков С.А. Информатика. Систематический курс: учебник для 10 класса / С.А. Бешенков, Е.А. Ракитина. – М.: БИНОМ. Лаборатория знаний, 2006. – 432 с.
3. Босова Л. Л. Информатика и ИКТ. 9 кл.: учеб. для общеобразоват. учреждений . - М.: Бином, Лаборатория знаний, 2013.- 336с.: ил.
4. Босова Л.Л. Информатика и ИКТ: учебник для 9 класса часть 1 – М.: БИНОМ. Лаборатория знаний, 2012. – 244 с.: ил.
5. Босова Л.Л. Уроки информатики в 9 классе: методическое пособие / Л.Л. Босова, А.Ю. Босова. – 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2012. – 264 с.: ил.
6. Быкадоров Ю. А. Информатика и ИКТ. 9 кл.: учеб.для общеобразоват. Учреждений/Ю. А. Быкадоров. - М.: Дрофа, 2013.-336с.: ил.
7. Лапчик М.П. и др. Методика преподавания информатики: Учеб.пособие для студ. пед. Вузов [Текст] / М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер; Под общей ред. М. П. Лапчика. — М.: Издательский центр «Академия», 2012. Режим доступа [http://businessfor.ru/m/frtyh/metodika\\_prepodavaniya\\_informatiki\\_-\\_lapchik.html](http://businessfor.ru/m/frtyh/metodika_prepodavaniya_informatiki_-_lapchik.html)
8. Семакин И. Г. Информатика и ИКТ: учебник для 9 класса / И. Г. Семакин, Л. А. Залогова, С. В. Русаков, Л. В. Шестакова. -5-е изд. – М.: БИНОМ. Лаборатория знаний, 2012. – 341 с.: ил.
9. Информатика. 10–11 классы. Углубленный уровень: примерная рабочая программа / Н.Н. Самылкина, И.А. Калинин: БИНОМ. Лаборатория знаний, 2016. – 45 с.

10. Угринович Н.Д. Информатика и ИКТ. Базовый уровень: учебник для 9 класса / Н.Д. Угринович. – М.: БИНОМ. Лаборатория знаний, 2010. – 295 с.

11. Угринович Н.Д. Информатика и ИКТ. Профильный уровень: учебник для 11 класса / Н.Д. Угринович. – М.: БИНОМ. Лаборатория знаний, 2008. – 387 с.

12. Федеральный государственный образовательный стандарт среднего (полного) общего образования [Электронный ресурс] // Режим доступа <http://standart.edu.ru/catalog.aspx?CatalogId=6408>

## Приложение 1

### Разработка урока по теме “ Одномерные массивы”

#### ПЛАН-КОНСПЕКТ УРОКА

**Предмет:** Информатика и ИКТ.

**Класс:** 11.

**Тема урока:** Алгоритмизация и программирование. Одномерные массивы для целых чисел. Описание, ввод и вывод элементов одномерного массива.

**Учитель** Сандер Д.Е.

**Дата:** 22.12.2016

**Тип урока:** Объяснение и первичное закрепление знаний.

**Цель урока:** создать условия для формирования у обучающихся умений и навыков по теме «Одномерные массивы целых чисел».

#### **Задачи**

*Образовательные:*

- познакомить с понятием массив, одномерный массив, видами массивов;
- показать правила описания одномерного массива, способы ввода, вывода его элементов;
- совершенствовать умения и навыки работы с программой Pascal.

*Развивающие:*

- содействовать развитию познавательных интересов, самоконтроля.
- вовлечь в активную практическую деятельность;
- воспитать культуру учебного труда, умение распределять свое время.

*Воспитательные:*

- создать условия для развития способностей к анализу, исследовательским навыкам;
- способствовать развитию интереса учащихся к изучению предметов,
- развить умения написания программ на одномерные массивы;
- развить логическое мышление.

### **Этапы урока:**

1. Организационный момент - 1 мин
2. Объяснение нового материала – 20 мин
3. Практическая работа – 15 мин
4. Домашнее задание, рефлексия – 1 мин

### **Формы организации учебной деятельности обучающихся:**

- Фронтальная (при повторении материала)
- Индивидуальная (при выполнении практической работы).

### **Перечень средств ИКТ, используемых на уроке:**



- АРМ учителя;
- Интерактивная доска;
- Презентация «Одномерные массивы целых чисел»;

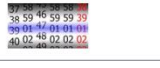

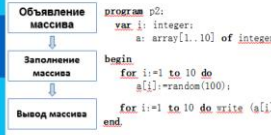
## **СТРУКТУРА И ХОД УРОКА**

<b>Деятельность учителя</b>	<b>Деятельность учащихся</b>	<b>Интерактивная доска</b>	<b>Тетрадь</b>
<b>Этап 1. Организационный момент 1 мин</b>			

<p>Приветствие.</p> <p>Сегодня мы с вами познакомимся с понятием «Массив», описанием, вводом и выводом элементов одномерного массива, а также будем решать задачи связанные с массивами.</p> <p>Но сначала давайте вспомним.</p>		<p>ОДНОМЕРНЫЕ МАССИВ ЦЕЛЫХ ЧИСЕЛ АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ</p>	
<p><b>Этап 2 Объяснение нового материала 20 мин</b></p>			
<p>На предыдущих уроках мы с вами работали только с простыми типами данных. Из элементов простых типов в языке Паскаль можно образовывать составные типы данных, так называемые структуры данных. Давайте вспомним что мы проходили на прошлом уроке</p> <p>Сегодня мы и познакомимся с одной из таких структур – с массивом.</p> <p>Открывайте тетради и</p>	<p>Отвечают на вопросы</p> <p>Записывают число и тему урока.</p>	<p><b>Слайды 2-6</b></p>	



<p>записывайте число и тему сегодняшнего урока.</p> <p>Одномерные массивы целых чисел.</p>			
<p>Что такое массив?</p> <p>Прочитайте определение.</p> <p>Запишите в тетрадь.</p>	<p>Записывают в тетрадь.</p>	 <p><b>Массив</b> Массив – это упорядоченное множество однотипных переменных (элементов массива), которым можно присвоить общее имя, различающиеся номерами (индексами).</p> <p><b>Одномерный массив</b></p> <p>Диаграмма: 1 2 3 4 5 6 7 8 9 7   -5   9   1   0   -2   4   3   6</p> <p>Элементы массива: 7, -5, 9, 1, 0, -2, 4, 3, 6 Индекс элемента массива: 1, 2, 3, 4, 5, 6, 7, 8, 9</p>	
<p>Перед тем как использовать массив в основной программе, его необходимо описать.</p> <p>Где у нас находится раздел описания данных?</p> <p>Как называется?</p> <p>Записываем в тетрадь общий вид описания массива.</p> <p>При описании в компьютере выделяется участок памяти нужного размера для хранения массива.</p>	<p>В начале программы, после заголовка.</p> <p>Var</p>	 <p><b>Описание массива</b></p> <p>Общий вид описания массива: <code>var &lt;имя_массива&gt; array [&lt;мин. знач. индекса&gt; - &lt;макс. знач. индекса&gt;] of &lt;тип_элементов&gt;;</code></p> <p>Примеры: <code>var a: array [1..10] of integer;</code> <code>const b: array [1..5] of integer = (4, 2, 3, 5, 7);</code></p> <p>Элементы слайда: Имя массива, Тип элемента массива, Минимальное значение индекса, Максимальное значение индекса, Значение элемента массива.</p>	
<p>Пример. Записываем.</p>			
<p>Массив, элементы которого имеют заданные начальные значения, может быть описан в разделе описания</p>			

<p>констант.</p> <p>В этом случае не просто выделяются последовательные ячейки памяти – в них сразу же заносятся соответствующие значения.</p>			
<p>Рассмотрим способы заполнения массива.</p> <p>Записываем в тетрадь.</p>	<p>Записывают в тетрадь</p>	<p><b>Способы заполнения массива</b></p> <p>1 способ: Ввод каждого значения с клавиатуры:  <code>for i:=1 to 10 do read (a[i]);</code></p> <p>2 способ: С помощью оператора присваивания (по формуле):  <code>for i:=1 to 10 do a[i]:=i;</code></p> <p>3 способ: С помощью оператора присваивания (случайными числами):  <code>randomize;</code>  <code>for i:=1 to 10 do a[i]:=random(100);</code></p> 	
<p>Рассмотрим способы вывода массива. С помощью какого оператора мы выводим данные?</p> <p>Записываем в тетрадь.</p>	<p>Write</p> <p>Записываем в тетрадь</p>	<p><b>Вывод массива</b></p> <p>1 способ: Элементы массива можно вывести с помощью оператора <code>write</code>, разделив их пробелом:  <code>for i:=1 to 10 do write (a[i], ' ');</code></p>  <p>2 способ: Вывод с комментариями:  <code>for i:=1 to 10 do writeln ('a[', i, ']=', a[i]);</code></p>	
<p>Разберем пример программы заполнения массива случайными числами и вывод элементов массива.</p>		<p><b>Заполнения массива A(10) случайными числами и вывод элементов массива</b></p> 	
<p><b>Этап 3. Практическая работа 15 мин</b></p>			

<p>Переходим к практической части урока.</p> <p><b>Задание 1.</b> Сформировать и вывести на экран (в столбик) последовательность из n элементов, заданных датчиком случайных чисел на интервале [0, 34]. (на доске со мной)</p> <p><b>Задание 2.</b> Сформировать массив и вывести значения из таблицы (№дня – температура). *табл</p> <pre> 1Program z1; var a: array [1..100] of integer; i, n: integer; begin write ('Введите n'); readln (n); for i:=1 to n do a[i]:=random (34); for i:=1 to n do writeln (a[i], ' '); end. </pre>	<p>Записывают задачи в тетрадь.</p>		
<pre> 2. program z2; vara: array [1..13] of integer; </pre>			

<pre> i: integer; begin for i:=1 to 13 do read (a[i]); for i:=1 to 13 do writeln (i, день = ', a[i]); end. </pre>			
<b>Этап 5. Домашнее задание, рефлексия 1 мин</b>			
Выучить конспект, РТ №71-74			

\*табл

День	1	2	3	4	5	6	7	8	9	10	11	12	13
C <sup>0</sup>	16	10	15	26	18	27	30	28	35	30	21	18	20

### Самостоятельная работа №1

Задачи по теме «Одномерные массивы»

**Задача №1.** Выполнить следующие действия:

- 3) Создать одномерный массив А из 10 целых чисел (числа вводит пользователь)
- 4) вывести на экран массив в виде строки чисел;  
подсчитать сумму элементов массива (блок-схема изображена на рисунке 1 файл [Рисунок10](#)) вывести сумму на экран.

**Задача №2.** Выполнить следующие действия:

- 5) Создать одномерный массив А из 10 целых чисел (числа вводит пользователь)
- 6) вывести на экран массив в виде строки чисел;
- 7) поменять местами элементы массива (блок-схема алгоритма показана на рисунке 2 файл [Рисунок1](#)) следующим образом
  - 1-ый со 2-ым
  - 3-ий с 4-ым
  - 5-ый с 6-ым
  - 7-ой с 8-ым
  - 9-ый с 10-ым
- 8) вывести измененный массив на экран.

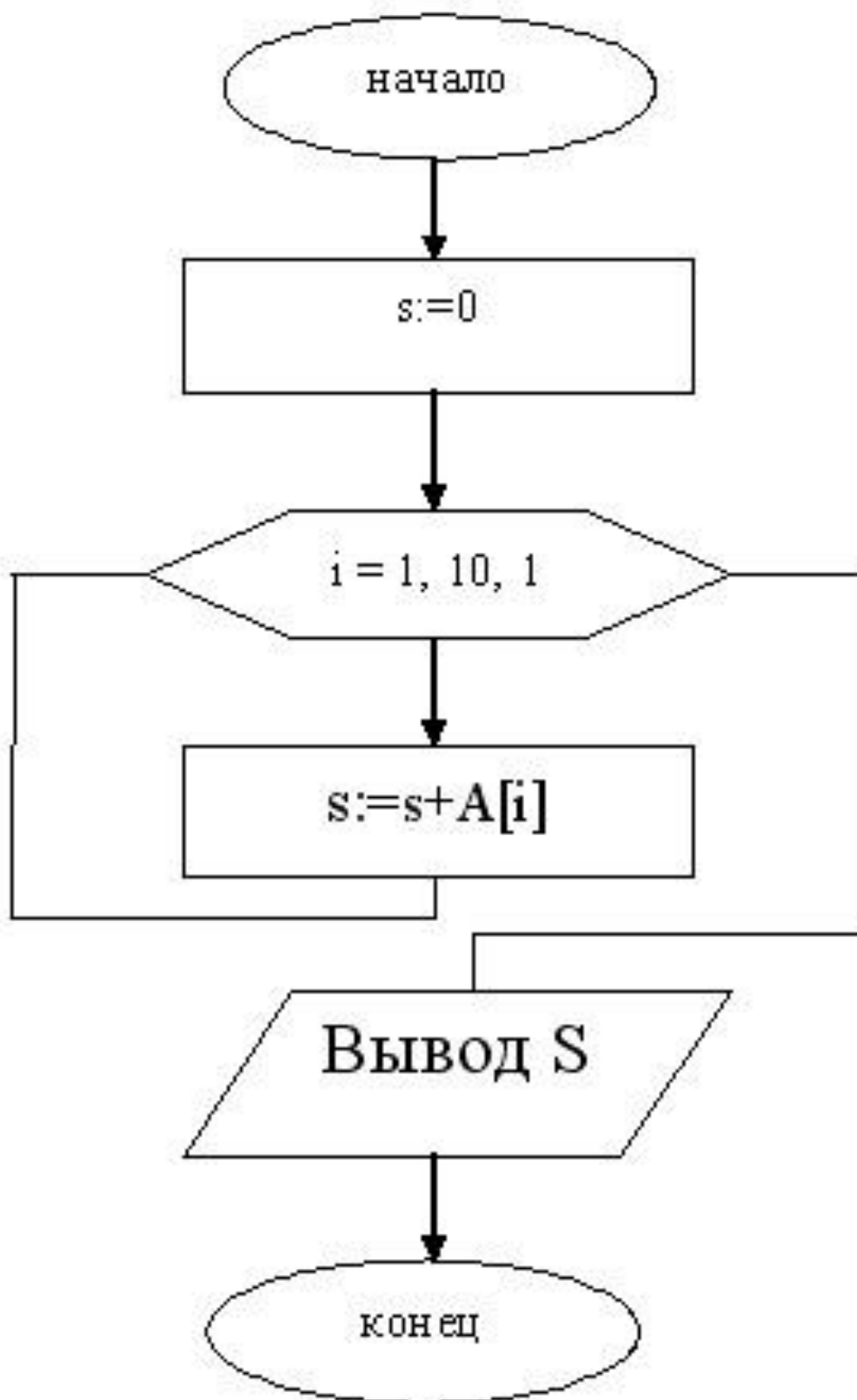
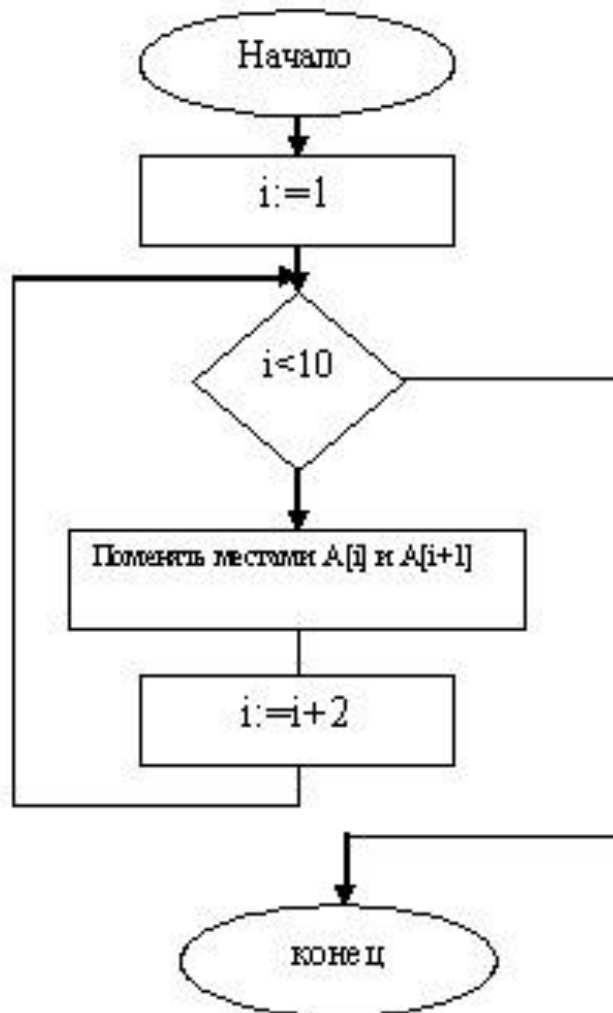


Рисунок 5



**Рисунок 6**

**Разработка урока по теме “ Двумерные массивы”**

Продолжение изучения массивов основано на изучении темы «Двумерные массивы».

## ПЛАН-КОНСПЕКТ УРОКА

**Предмет:** Информатика и ИКТ.

**Класс:** 11.

**Тема урока:** Алгоритмизация и программирование. Двумерные массивы.

**Учитель** Сандер Д.Е.

**Дата:**

**Тип урока:** Объяснение и первичное закрепление знаний.

**Цель урока:** создать условия для формирования у обучающихся умений и навыков по теме «Двумерные массивы».

### **Задачи**

*Образовательные:*

- познакомить с понятием массив, одномерный массив, видами массивов;
- показать правила описания одномерного массива, способы ввода, вывода его элементов;
- совершенствовать умения и навыки работы с программой Pascal.

*Развивающие:*

- содействовать развитию познавательных интересов, самоконтроля.
- вовлечь в активную практическую деятельность;
- воспитать культуру учебного труда, умение распределять свое время.

*Воспитательные:*



- создать условия для развития способностей к анализу, исследовательским навыкам;
- способствовать развитию интереса учащихся к изучению предметов,
- развить умения написания программ на одномерные массивы;
- развить логическое мышление.

### **Этапы урока:**

5. Организационный момент - 1 мин
6. Объяснение нового материала – 20 мин
7. Практическая работа – 15 мин
8. Домашнее задание, рефлексия – 1 мин

### **Формы организации учебной деятельности обучающихся:**

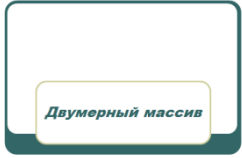
- Фронтальная (при повторении материала)
- Индивидуальная (при выполнении практической работы).

### **Перечень средств ИКТ, используемых на уроке:**

- АРМ учителя;
- Интерактивная доска;
- Презентация «Одномерные массивы целых чисел».

## **СТРУКТУРА И ХОД УРОКА**

<b>Деятельность учителя</b>	<b>Деятельность учащихся</b>	<b>Интерактивная доска</b>	<b>Тетрадь</b>
<b>Этап 1. Организационный момент 1 мин</b>			

<p>Приветствие.</p> <p>Сегодня мы с вами познакомимся с двумерными массивами, описанием, вводом и выводом элементов двумерного массива, а также будем решать задачи связанные с данной темой.</p> <p>Но сначала давайте вспомним.</p>			
<p><b>Этап 2 Объяснение нового материала 20 мин</b></p>			
<p>На предыдущих уроках мы с вами работали только с массивами. Давайте вспомним, что мы проходили на прошлом уроке</p> <p>Что такое массив?</p> <p>Как правильно описать массив?</p> <p>Какие есть способы заполнения массива?</p> <p>Открывайте тетради и записывайте число и тему</p>	<p>Отвечают на вопросы</p> <p>Записывают число и тему урока.</p>		

<p>сегодняшнего урока.</p> <p>Двумерные массивы.</p>			
<p>Прочитайте определение и запишите его в тетрадь.</p>	<p>Записывают в тетрадь.</p>	<p><b>Двумерный массив</b>  Массив, который состоит из нескольких строк и нескольких столбцов называется двумерным массивом.  Его элементы нумеруются двумя индексами – номером строки и номером столбца соответственно.  (Аналогом явл. ваше место в кинотеатре, где номер строки массива – это ряд, а номер столбца массива – это ваше место в ряду)</p>	
<p>Рассмотрим характеристики массива</p> <p>Тип –общий тип всех элементов массива;</p> <p>Размерность – количество индексов массива;</p> <p>Диапазон изменения индекса (индексов), определяет количество элементов в массиве</p>		<p><b>Характеристики массива:</b></p> <ul style="list-style-type: none"> <li>• Тип –общий тип всех элементов массива;</li> <li>• Размерность – количество индексов массива;</li> <li>• Диапазон изменения индекса (индексов), определяет количество элементов в массиве</li> </ul> <p><b>Двумерные массивы располагаются в памяти ПК по строкам: сначала все элементы первой строки, затем элементы второй строки и т.д.</b></p>	
<p>Индексы массива</p> <p>При обращении к элементу массива указывается имя массива, а в квадратных скобках через запятую индексы (номер строки, номер столбца) этого элементы массива.</p>		<p><b>Наиболее часто индексы массива – это переменные типа integer.</b></p> <p>При обращении к элементу массива указывается имя массива, а в квадратных скобках через запятую индексы (номер строки, номер столбца) этого элемента массива.</p> <p>Например A[3,2] мы обратились к элементу массива A, который располагается в третьей строке во втором столбце.</p>	

<p>Объявление двумерного массива.</p> <p>Самый простой способ описания массива в программе – это объявить переменную в разделе описания переменных var с использованием зарезервированного слова array( т.е. массив):</p> <p>var имя массива: array[нижняя граница индекс 1.. Верхняя граница индекс 1, нижняя граница индекс 2..верхняя граница индекс 2] of тип элементов;</p>		<p><b>Например:</b></p> <p>Пусть в памяти ПК расположена таблица чисел:</p> <pre>1 2 3 4 5 6 7 8 9 10 11 12</pre> <p>Тогда описание этого массива: Var b:array[1..3,1..4] of integer; (12 элементов массива – целые числа)</p>	
<p>Заполнение массива</p> <p>Для заполнения двумерного массива данными используются вложенные циклы с параметром: For...</p> <p>Внешний цикл организуется по номеру строки, а внутренний цикл – по номеру столбца. Значения элементов массива можно</p>	<p>Записывают в тетрадь</p>	<p><b>Заполнение массива данными</b></p> <p>Для заполнения двумерного массива данными используются вложенные циклы с параметром: For...</p> <p>Внешний цикл организуется по номеру строки, а внутренний цикл – по номеру столбца. Значения элементов массива можно задавать вводом данных с клавиатуры.</p> <p><b>Пример заполнения двумерного массива состоящего из 3 строк и 2 столбцов данными с клавиатуры:</b></p> <pre>for i:=1 to 3 do {внешний цикл по номеру строки} for j:=1 to 2 do {внутренний цикл по номеру столбца} Begin writeLn('Введите B[',i,',',j], ' элемент массива'); readLn(B[i,j]); End;</pre>	

<p>задавать вводом данных с клавиатуры.</p>			
<p>Рассмотрим способы вывода массива. С помощью какого оператора мы выводим данные? Записываем в тетрадь.</p>	<p>Write Записываем в тетрадь</p>	<div style="border: 1px solid black; padding: 5px;"> <p>Вывод элементов двумерного массива по строкам и столбцам:</p> <pre> For i:= 1 to 3 do begin For j:= 1 to 2 do Write(B[i,j]:5); Writeln; End;</pre> </div>	
<p>Действия над элементами двумерного массива Вычисление суммы элементов каждой строки. Поиск минимального элемента всей матрицы. Поиск максимального элемента всей матрицы.</p>			
<p><b>Этап 3. Практическая работа 15 мин</b></p>			

<p>Переходим к практической части урока.</p> <p><b>Задание 1.</b> Вычислить сумму элементов главной диагонали квадратной матрицы</p> <pre> Program Sumglav;  const n=3; var a:array [1..n,1..n] of real; i, j: integer; S: real;  Begin   S:=0; for i:=1 to n do     S:=S+a[i,i]; write('Суммаэлементовглав нойдиагонали = ',S); readln; readln end.  <b>Задание 2.</b> Нахождение наибольших элементов каждой строки массива:  Program Stroki; const n=3; m=4 var a:array [1..n,1..m] of real; max:array [1..n] of real; i, j: integer; </pre>	<p>Записываются задачи в тетрадь.</p>		
---	---------------------------------------	--	--

<pre> Begin for i:=1 to n do begin max[i]:=a[i,1]; for j:=2 to m do if max[i]&lt;a[i,j] then max[i]:=a[i,j]; end; write('Наибольшее число в массиве =&gt; '); for i:=1 to n do writeln(max[i]); readln; readln end. </pre>			
<b>Этап 5. Домашнее задание, рефлексия 1 мин</b>			
<p>Что такое двумерный массив?</p> <p>Приведите примеры двумерных массивов из жизни</p> <p>Чем двумерные массивы отличаются от одномерных?</p> <p>Домашнее задание: Выучить конспект</p>			

## Самостоятельная работа №2

1. Что определяет для массива  $X[1..n, 1..m]$  следующий алгоритм  
For  $i:= 1$  to  $n$  do For  $j:= 1$  to  $m$  do If  $j \bmod 2 = 0$  then  $X[i, j] = A$ ;

- нечетные строки матрицы заменить на  $A$ .
- четные строки матрицы заменить на  $A$ ;
- четные столбцы матрицы заменить на  $A$ ;
- минимальный элемент массива;

2. Что определяет для массива  $X[1..n]$  следующий алгоритм  
 $S:= 0$ ; For  $i:= 1$  to  $n$  do If  $X[i] > 0$  then  $S:=S+X[i]$ ;

- максимальный элемент массива;
- сумму положительных элементов массива;
- количество отрицательных элементов массива;
- минимальный элемент массива;

3. Укажите правильное описание массива:

- a: array  $[1..n, 1..m]$  of integer.
- a: array  $[1..n, 1..m]$  of real;
- a: array  $\{1..n\}$  of integer;
- a: array  $[1..n, 1..m]$  of real;

4. Что определяет для массива  $X[1..n]$  следующий алгоритм  
 $R:= 0$ ; For  $i:= 1$  to  $n$  do If  $X[i] > T$  then  $R:=R+1$ ;

- номер первого элемента массива, меньшего  $T$ ;
- номер первого элемента массива, большего  $T$ ;
- количество элементов массива, больших  $T$ ;
- номер последнего элемента массива, большего  $T$ ;

5. Что определяет для массива  $X[1..n, 1..n]$  следующий алгоритм  
 $S:=0$ ; For  $i:= 1$  to  $n$  do For  $j:= 1$  to  $n$  do If  $i = j$  then  $S:= S+X[i, j]$ ;



- сумма элементов побочной диагонали матрицы;
- четные столбцы матрицы заменить на  $A$ ;
- нечетные строки матрицы заменить на  $A$ ;
- сумма элементов главной диагонали матрицы;