

# МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ УРОКОВ В 11 КЛАССЕ

## Урок 1. Табличный процессор. Основные сведения

### Планируемые образовательные результаты:

- *предметные*: владение приёмами ввода и редактирования данных в электронных таблицах; умение использовать электронные таблицы для выполнения учебных заданий из различных предметных областей;
- *метапредметные*: умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;
- *личностные*: наличие мировоззрения, соответствующего современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) обсуждение роли информатики, информационных и коммуникационных технологий в современном обществе;
- 2) актуализация знаний об объектах табличного процессора и их свойствах;
- 3) актуализация ранее изученных приёмов ввода и редактирования данных в ячейках электронных таблиц, знакомство с новыми приёмами;
- 4) актуализация навыков работы с относительными, абсолютными и смешанными ссылками.

**Основные понятия:** табличный процессор, рабочая книга, лист, электронная таблица, ячейка, диапазон ячеек, ссылка, принцип относительной адресации, формула, автоматический пересчёт.

### Электронное приложение к учебнику:

- презентация «Табличный процессор. Основные сведения»;
- онлайн-тест № 1 «Табличный процессор. Основные сведения».

### Дополнительные интернет-ресурсы:

- РЭШ (<https://resh.edu.ru>). Урок 14. Обработка информации в электронных таблицах.

### Особенности изложения содержания темы урока

Изложение материала строится в форме интерактивной лекции с использованием презентации.

Большая часть информации в презентации известна учащимся. На первый вопрос, приведённый в презентации (слайд 4), в самой презентации ответа нет.

*Вопрос* (слайд 4). Какой столбец идёт после столбца BDZ?

*Ответ.* В алфавите системы счисления, используемой для нумерации столбцов, задействованы латинские буквы. Z – «старшая» цифра этой системы счисления. После столбца BDZ следует столбец BEA.

Возможно, новой для учащихся окажется информация о другом стиле ссылок, когда столбцы и строки имеют числовую нумерацию (слайд 4).

Слайды 5, 6, 7 содержат необходимую информацию об объектах электронных таблиц, форматах данных и возможных ошибках при вводе данных.

В качестве примера записи формулы (слайд 7) в ЭТ приводится пример формулы, содержащей операцию конкатенации (склейки) текста — &. Таким образом, формула  $=(\$A\$1&C1)^2$ , ссылаясь на числовые данные, включает в себя два преобразования типа данных – двух чисел в строку и строки в число. К содержимому ячейки A1 приписывается справа содержимое ячейки C1, а полученная строка, став числом, возводится в квадрат:  $(1 \& 2)^2 = 12^2 = 144$ . Операция & не обладает свойством коммутативности.

На примере составления таблицы квадратов двузначных чисел (слайд 8) демонстрируется многовариантность решения одной и той же задачи. После заполнения диапазонов с десятками и единицами, появляются три альтернативных способа заполнения таблицы формулами квадратов. В двух предлагаемых решениях используются смешанные ссылки, но возводимое в квадрат число получается по-разному – сумма разрядных слагаемых (способ I) и операция & (способ II). Ещё один вариант (способ III) требует предварительного именованя диапазонов десятков и единиц.

На слайде «Вопросы и задания» (слайд 10) ответы спрятаны за кнопками-триггерами.

Практическую часть урока можно организовать, предложив ученикам выполнить задания № 14, 16, 20 и 21 к § 1.

При наличии времени в конце урока можно предложить ученикам выполнить онлайн-тест № 1.

### **Онлайн-тест № 1. Табличный процессор. Основные сведения**

1. Диапазон – это:
  - все ячейки одной строки;
  - ◎ **совокупность клеток, образующих в таблице область прямоугольной формы;**
  - все ячейки одного столбца;
  - множество допустимых значений.
2. Какие типы данных можно ввести в ячейки электронной таблицы?
  - числа и формулы;
  - числа и текст;
  - ◎ **числа, текст и формулы;**
  - формулы и текст.
3. При перемещении или копировании в ЭТ абсолютные ссылки:

- преобразуются в зависимости от длины формулы;
- преобразуются в зависимости от нового положения формулы;
- преобразуются вне зависимости от нового положения формулы;
- **не изменяются.**

4. Установите соответствие между сообщениями об ошибках и их причинах:

- |             |  |
|-------------|--|
| 1) #####    | А) ячейка, ссылка на которую используется в формуле, не существует;                                      |
| 2) #ДЕЛ/0!  | Б) в формуле для математических вычислений имеется ссылка на ячейку, содержанием которой является текст; |
| 3) #ЗНАЧ!   | В) столбец недостаточно широкий для отображения числа;   |
| 4) #ССЫЛКА! | Г) попытка деления на ноль.  |

Ответ:

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>В</b>	<b>Г</b>	<b>Б</b>	<b>А</b>

5. Какое число будет записано в ячейку В2 после ввода формулы?

	А	В
1	4	=A1*A2
2	2	=A1+B1^2

Ответ: **68**.

6. В электронной таблице выделен диапазон из 21 ячеек, расположенный в столбцах F, G и H. Правый нижний угол диапазона находится в ячейке Н8. Запишите адрес левого верхнего угла диапазона.

Ответ: **F2**.

**Домашнее задание:** § 1, вопросы и задания № 1–13, 15, 17–19.  
*Дополнительное задание:* № 22.

**Указания, комментарии, ответы и решения**

№ 15. 1)  $=(A1+A2+A3)/3$ ;  
 2)  $=5*A1^3+4*A2^2-3*A3$ .

№ 17. 40.

№ 18.  $=C\$2+\$D2+2$ .

## Урок 2. Редактирование и форматирование в табличном процессоре

**Планируемые образовательные результаты:**

- *предметные:* владение приёмами редактирования и форматирования в электронных таблицах; умение использовать электронные таблицы для выполнения учебных заданий из различных предметных областей;

- *метапредметные*: умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;
- *личностные*: наличие мировоззрения, соответствующего современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

#### **Решаемые учебные задачи:**

- 1) актуализация и систематизация знаний и умений редактирования книги электронной таблицы;
- 2) актуализация, расширение и систематизация знаний и умений о форматировании объектов электронной таблицы.

**Основные понятия:** рабочая книга, лист, электронная таблица, ячейка, диапазон ячеек, формат ячеек, форматирование ячеек, форматирование электронной таблицы.

#### **Электронное приложение к учебнику:**

- презентация «Редактирование и форматирование в табличном процессоре»;
- онлайн-тест № 2 «Редактирование и форматирование в табличном процессоре».

#### **Дополнительные интернет-ресурсы:**

- РЭШ (<https://resh.edu.ru>). Урок 14. Обработка информации в электронных таблицах.

#### **Особенности изложения содержания темы урока**

Изложение материала строится в форме интерактивной лекции с использованием презентации.

В презентации большинство иллюстраций на слайдах с объяснением и с вопросами – снимки экрана. Возможно, в используемой учителем на уроке версии программного обеспечения некоторые действия будут немного отличаться. Вопросы по ходу демонстрации снабжены ответами. Часть ответов появляется после вопроса, а часть — как отметка верно/неверно к выбранному ответу. На слайде 6 последовательно подсвечиваются диапазоны ячеек разных типов, затем демонстрируется тип диапазона. Большинство случаев безошибочно определяются учащимися. Возможная ошибка в определении типа данных для ячеек с данными:

т12045
т356907
т457869

Буква «т» в каждой ячейке «не позволит» работать с содержимым ячейки как с числом. Диапазон текстовый.

Информацию о защите и закреплении областей (слайд 7) можно проиллюстрировать на таблице, заполненной большим количеством информации (например, из файла task19.xls ОГЭ по информатике).

*Вопросы и задания* (слайд 9).

*Вопрос 1.* Выбрать действие с электронной таблицей.

*Комментарий к ответу.* Очистка содержимого привела бы к появлению пустого столбца С, а все остальные ячейки остались бы без изменения и в тех же позициях. Факт того, что данные скрыты, а не удалены, можно установить по заголовочной строке – никакие действия не изменят последовательность имён столбцов при редактировании или их удалении. Пользователь может только скрыть столбец/строку, но имена столбцов и строк останутся без изменения. В последовательности заголовков столбцов фрагмента «после изменения» нет столбца С, значит, он скрыт.

*Вопрос 2.* Какое значение будет в ячейке после применения к ячейке формата Числовой?

*Ответ:* Формат изменяемой ячейки – процентный. При переводе в числовой формат результат надо разделить на 100. То есть значение числа будет – 0,068. Такого ответа нет, но при отображении двух знаков после запятой табличный процессор выполнит округление перед отображением (добавление количества знаков после запятой отразит число таким, каким оно хранится в ЭТ).

Практическую часть урока можно организовать, предложив ученикам выполнить задания № 10 и 14 к § 2.

При наличии времени в конце урока можно предложить ученикам выполнить онлайн-тест № 2.

## **Онлайн-тест № 2. Редактирование и форматирование в табличном процессоре**

1. Редактирование электронной таблицы состоит в:
  - добавлении в неё диапазонов ячеек;**
  - добавлении в неё новых строк или столбцов;**
  - удалении из неё имеющихся листов;
  - удалении из неё имеющихся строк или столбцов;**
  - перемещении существующих листов;
  - удалении из неё диапазонов ячеек.**
2. Отметьте верное утверждение:
  - При форматировании данных изменяются не только сами данные, но и их внешний вид.
  - При форматировании данных изменяются не сами данные, а лишь их внешний вид.**
  - При форматировании данных изменяются только сами данные, их внешний вид остаётся без изменений.

- Форматирование данных не связано с изменением данных и их внешнего вида.
3. Установите соответствие между форматом числа и тем, когда он используется.
- |              |  |
|--------------|--|
| 1) Общий;    | А) для установки значений с добавлением к числу обозначения денежной единицы;                |
| 2) Числовой; | Б) для представления чисел в большинстве случаев так, как они были введены;                  |
| 3) Денежный; | В) для представления числа в виде даты определённого типа;                                   |
| 4) Дата.     | Г) для представления числа в виде десятичной дроби с заданным количеством десятичных знаков. |

Ответ:

1	2	3	4
Б	Г	А	В

4. При форматировании ячеек электронной таблицы можно устанавливать:
- абсолютную или относительную адресацию в формуле;
  - защиту ячейки, режим скрытия формул;**
  - относительную адресацию в формуле;
  - формат числовых данных (числовой формат);**
  - смешанную адресацию в формуле.
5. Дан фрагмент электронной таблицы:

	A	B	C	D
1	1	2	3	
2	4	5	6	
3	7	8	9	

В ячейку D1 введена формула  $=A\$1*B1+C2$ , а затем скопирована в ячейку D2. Какое значение в результате появится в ячейке D2?

Ответ: **14.**

**Домашнее задание:** § 2, вопросы и задания № 1–9, 11–13, 15, 16.

### **Уроки 3–4. Встроенные функции и их использование. Логические функции**

#### **Планируемые образовательные результаты:**

- *предметные:* владение приёмами организации вычислений в электронных таблицах с использованием встроенных функций; умение использовать электронные таблицы для выполнения учебных заданий из различных предметных областей;
- *метапредметные:* умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением

требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;

- *личностные*: наличие мировоззрения, соответствующего современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

#### **Решаемые учебные задачи:**

- 1) актуализация знаний о встроенных функциях, навыков их использовании при организации вычислений в электронных таблицах;
- 2) расширение представлений о различных категориях встроенных функций, знакомство с финансовыми и текстовыми функциями;
- 3) закрепление навыков использования логических функций.

**Основные понятия:** встроенная функция, имя функции, аргумент функции, результат работы функции, математические функции, статистические функции, логические функции, текстовые функции, финансовые функции.

#### **Электронное приложение к учебнику:**

- презентация «Встроенные функции и их использование»;
- онлайн-тест № 3 «Встроенные функции и их использование».

#### **Дополнительные интернет-ресурсы:**

- РЭШ (<https://resh.edu.ru>). Урок 14. Обработка информации в электронных таблицах.

#### **Особенности изложения содержания темы урока**

Основные аспекты рассматриваемого материала отражены в презентации. Рассмотрим её содержание.

Электронные таблицы предоставляют возможность использования огромного количества функций. Для облегчения поиска функции сгруппированы по их назначению. Следует обратить внимание учащихся на возможность получения подробной информации о выбранной функции и примерах её использования. Мастер построения функции сопровождает подсказками каждый шаг пользователя. В презентации к части перечисленных в ней функций рассмотрены подробные примеры. Задачи для примеров подобны заданиям ЕГЭ разных лет.

Понятия «имя функции», «аргументы функции» и «значение функции» раскрываются на слайде 3. Можно напомнить учащимся, что форма представления информации на этом слайде – денотатный граф. В качестве первого примера применения функции взята функция ОКРУГЛ(). По этой же функции задаются *вопросы* (слайд 3): «В чем отличие результата применения функции ОКРУГЛ и действия, назначенного кнопке уменьшения разрядности? Можно ли считать эти два варианта дублирующими друг друга действиями?»

*Ответ:* функция ОКРУГЛ изменяет значение числа, кнопка изменения разрядности может повлиять только на запись числа. Существуют случаи, когда результат работы этих двух действий одинаковый.

Можно задать *дополнительный вопрос:* «В ячейку записали результат работы функции =ОКРУГЛ(35,172; 1) и установили точность – 3 знака после запятой. Что увидит пользователь?»

*Ответ:* 35,200.

Далее последовательность показа слайдов зависит от учителя. Примеры функций расположены на скрытых слайдах и не все обязательны к просмотру. Во время демонстрации примеров используется цветовая подсветка и стрелки, иллюстрирующие зависимость формул от ячеек. В табличном процессоре можно продемонстрировать учащимся полезную возможность показать/спрятать стрелки отображения зависимостей ячеек (кнопки на вкладке Формулы).

Математические функции (слайд 5).

LOG(), ОКРУГЛВВЕРХ() — слайд 6.

*Задача.* При регистрации каждому пользователю выдаётся пароль, состоящий из 9 символов. Каждый пароль должен содержать хотя бы 1 десятичную цифру, как прописные, так и строчные латинские буквы, а также не менее 1 дополнительного символа: «#», «\*», «!», «@». Для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом все символы кодируют одинаковым и минимально возможным количеством бит. Сколько байт занимает информация о паролях 20 пользователей.

*Примечание.* В латинском алфавите 26 букв.

*Пояснение.* В вычислении используется формула Шеннона. При демонстрации этого примера следует учесть, что по математике понятие логарифма по разным программам изучают в разное время и запись формулы потребует дополнительных разъяснений. Функция ОКРУГЛВВЕРХ() в этом задании используется дважды – при определении информационного веса символа и информационного объёма пароля.

ДЕС() — слайд 7.

*Задача.* Все 5-буквенные слова, составленные из букв К, Л, Р записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ККККК
2. ККККЛ
3. ККККР
4. КККЛК
5. КККЛЛ

...

Под каким номером стоит слово ЛКРКЛ?

*Пояснение.* Так как в алфавите три разных буквы и по списку можно установить их старшинство, то можно в записи перейти к привычной цифровой записи, установив взаимно однозначное соответствие «слово – число» в системе счисления с основанием 3. После перевода полученного



числа в десятичную систему счисления прибавляется 1, так как нумерация списка начинается с 1, а первое слово в списке соответствует числу 0.

#### Статистические функции (слайд 8).

Список самых популярных статистических функций расширен на две функции с похожими названиями – ПЕРЕСТ() и ПЕРЕСТА(). Действие этих функций также демонстрируется на примере заданий ЕГЭ. Функции используются при решении комбинаторных задач.

ПЕРЕСТА() — слайд 9.

*Задача.* Вася составляет 5-буквенные слова, в которых встречаются только буквы А, Б, В, Г, причём буква А появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

ПЕРЕСТ() — слайд 10.

*Задача.* Вася составляет 5-буквенные слова, в которых встречаются только буквы А, Б, В, Г, Д, Е, Ж. Причём буква А появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в слове не более одного раза. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

*Пояснение.* Различие двух заданий не только в количестве букв, которые не влияют на используемую функцию, а и в разрешении использовать букву повторно.

#### Логические функции (слайд 11).

Все действия, которые могут быть выполнены функциями этого перечня, должны быть хорошо знакомы учащимся. Операций конъюнкции, дизъюнкции и отрицания достаточно для того, чтобы записать любую логическую операцию. Заметим, что операцию эквиваленции не обязательно записывать через функции И, ИЛИ, НЕ, так как этой операции соответствует формула  $A1=B1$  (ячейки A1 и B1 содержат логические значения ИСТИНА или ЛОЖЬ).

В разобранном примере по заданию ЕГЭ (слайд 12) используются все логические функции, предоставляемые табличным процессором. Приведённое поэтапное решение основано на построении полной таблицы истинности и может только проиллюстрировать задание, но не может рассматриваться как алгоритм решения задания ЕГЭ, так как является затратным по времени.

#### Текстовые функции (слайд 13).

Пример задачи на использование текстовых функций (слайд 14) рассматривает преобразование строки-результата диагностического тестирования в таблицу, что значительно упрощает дальнейшую статистическую обработку результата. Дополнив таблицу диаграммами и статистическими функциями, получаем инструмент для создания

объективной картины готовности учащихся к итоговой аттестации на текущий момент.

#### Инженерные функции (слайд 15).

В этом разделе можно продемонстрировать две задачи ЕГЭ.

В первой используются функции перевода в десятичную систему счисления (слайд 16).

Вторая задача — на применение функции БИТ.И() (слайд 17).

*Задача.* Для какого наименьшего неотрицательного целого десятичного числа  $a$  выражение  $(x \& 10 \neq 0) \rightarrow ((x \& 19 = 0) \rightarrow (x \& A \neq 0))$  тождественно истинно (т. е. принимает значение 1 при любом неотрицательном целом значении десятичной переменной  $x$ )?

*Пояснение.* Решение задачи начинается с преобразования выражения. Выражение должно содержать только операции отрицания, конъюнкции и дизъюнкции, так как они реализованы через логические функции табличного процессора. При заполнении таблицы использовались именованные диапазоны, что значительно упрощает чтение формул. Особенность примера заключается в том, что ответа в явном виде построенная таблица не даёт, но этот пример полезен для того, чтобы сформировать у учащихся понимание смысла данного задания.

Для поиска ответа на поставленный вопрос анализируется результат логического выражения при всех возможных значениях  $A$  и всех возможных значениях  $x$ .

*Вопрос.* Что является ответом?

*Ответ.* В 34-ой строке таблицы настроена формула подсчёта того, сколько раз при всех рассматриваемых значениях  $x$  данное значение  $A$  обращает выражение в ЛОЖЬ. Ответ будем искать среди множества значений  $A$ , при котором ни разу не встретится значение ЛОЖЬ. По условию задачи необходимо найти наименьшее значение. Значения  $A$  растёт слева направо, значит  $A$  из самого левого столбца, в котором в 34-ой строке стоит 0, является ответом.

*Вопрос.* Как подобрать достаточный для анализа диапазон значений  $X$  и  $A$  в этом задании?

*Ответ.* По условию требуется перебрать все возможные значения  $A$  и все возможные значения  $X$ . Перебрать абсолютно все значения не получится. Число в компьютере отличается от числа в математике: множество целых чисел ограничено и существует самое большое целое число. В условии задачи перечислены числа 10 и 19. Необходимо, чтобы перебираемые числа обеспечили все возможные сочетания 1 и 0 в двоичном представлении разрядов с единицами в числах 10 и 19. Двоичное представление числа  $19 = 10011_2$ . В числах 10 и 19 единицы встречаются в пяти разрядах. Можно ограничить зону поиска ответа в таблице —  $32 \times 32$ . Если до 32 ответ не найдётся, то далее его точно не будет.

### Функции даты и времени (слайд 18).

Примера использования функций даты и времени не добавлено в презентацию. Не трудно реализовать его самостоятельно, решая задачу, поставленную в презентации.

*Вопрос.* Настроить формулы электронной таблицы для обратного отсчёта «До ЕГЭ осталось: \_\_\_\_\_ мес. \_\_\_\_\_ дней».

*Пояснение.* Для того чтобы формулы читались проще, добавим ячейкам имена: СТАРТ – текущая дата и ЭКЗАМЕН – дата экзамена. Одна ячейка – утверждённая дата экзамена, другая ячейка – значение функции СЕГОДНЯ().

Самый простой вариант решения – считать, что все месяцы по 30 дней. Если определить целую часть от деления на 30 числа дней между двумя датами, узнаем количество месяцев. Формула: =ЦЕЛОЕ(ДНИ(ЭКЗАМЕН; СТАРТ)/30). Количество дней можно определить по формуле =ДЕНЬ(ЭКЗАМЕН)–ДЕНЬ(СТАРТ). Количество месяцев практически всегда будет давать правильный ответ. Но количество дней может стать отрицательным. Поэтому лучше для вычисления количества дней использовать формулу:

=ЕСЛИ(ДЕНЬ(ЭКЗАМЕН)>ДЕНЬ(СТАРТ); ДЕНЬ(ЭКЗАМЕН)–  
ДЕНЬ(СТАРТ); 30–ДЕНЬ(ЭКЗАМЕН)–ДЕНЬ(СТАРТ))

Это решение тоже не всегда будет верным: в зависимости от количества дней в текущем месяце в формуле надо использовать 30 или 31. Но такое решение более трудоёмко; рекомендовать его поиск можно только избранным учащимся.

На уроках по рассматриваемой теме преобладают практические формы работы. Так, уже после обсуждения содержания трёх–четырёх кадров презентации учащимся можно предложить представленную ниже практическую работу, выполнение которой позволит им актуализировать умения, сформированные при изучении электронных таблиц в основной школе.

### **Практическая работа**

Откройте файл Результаты.xlsx с электронной таблицей (ссылка «Файл-заготовка для практикума»). На основании данных, содержащихся в этой таблице, найдите ответы на следующие вопросы.

1. Сколько учащихся Майского района приняли участие в тестировании?  
Ответ на этот вопрос запишите в ячейку E2 таблицы.
2. Сколько всего учащихся набрали больше 80 баллов по математике?  
Ответ на этот вопрос запишите в ячейку E3 таблицы.
3. Сколько учащихся Кировского района набрали больше 75 баллов по физике? Ответ на этот вопрос запишите в ячейку E4 таблицы.
4. Сколько учащихся набрали более 70 баллов по каждому из предметов? Ответ на этот вопрос запишите в ячейку E5 таблицы.
5. Сколько учащихся Центрального района набрали более 70 баллов по каждому из предметов? Ответ на этот вопрос запишите в ячейку E6 таблицы.

6. Сколько учащихся Центрального района набрали более 140 баллов в сумме по двум предметам? Ответ на этот вопрос запишите в ячейку E7 таблицы.
7. Чему равна разница между максимальным и минимальным баллами по физике среди всех учащихся? Ответ на этот вопрос запишите в ячейку E8 таблицы.
8. Чему равна наименьшая сумма баллов у учеников Подгорного района? Ответ на этот вопрос запишите в ячейку E9 таблицы.
9. Сколько участников тестирования набрали одинаковое количество баллов по математике и физике? Ответ на этот вопрос запишите в ячейку E10 таблицы.

*Пояснения, указания, ответы и комментарии к практической работе*  
Возможные варианты выполнения работы представлены ниже.

1. Введём в ячейку E2 формулу:  
=СЧЁТЕСЛИ(B2:B1001; "Майский").
2. Введём в ячейку G2 формулу:  
=ЕСЛИ(C2>80;1;0).  
Скопируем формулу из G2 во все ячейки диапазона G3:G1001. Чтобы посчитать количество таких учащихся, достаточно в ячейку E3 записать  
=СУММ(G2:G1001).
3. Введём в ячейку E4 формулу:  
=СЧЁТЕСЛИМН(B2:B1001;"Кировский"; D2:D1001;">75").
4. Введём в ячейку I2 формулу:  
=ЕСЛИ(И(C2>70;D2>70);1;0).  
Скопируем формулу из I2 во все ячейки диапазона I3:I1001. Чтобы посчитать количество таких учащихся, достаточно в ячейку E5 записать  
=СУММ(I2:I1001).
5. Введём в ячейку J2 формулу:  
=ЕСЛИ(И(B2="Центральный";C2>70;D2>70);1;0).  
Скопируем формулу из J2 во все ячейки диапазона J3:J1001. Чтобы посчитать количество таких учащихся, достаточно в ячейку E6 записать  
=СУММ(J2:J1001).
6. Введём в ячейку K2 формулу:  
=ЕСЛИ(И(B2="Центральный";C2+D2>140);1;0).  
Скопируем формулу из K2 во все ячейки диапазона K3:K1001. Чтобы посчитать количество таких учащихся, достаточно в ячейку E7 записать  
=СУММ(K2:K1001).
7. Введём в ячейку L2 формулу:  
=МАКС(D2:D1001).  
Введем в ячейку L3 формулу  
=МИН(D2:D1001).  
В ячейку E8 запишем разность максимального и минимального баллов:  
=L2-L3.
8. Введём в ячейку M2 формулу:

=ЕСЛИ(B2="Подгорный";C2+D2;"").

Скопируем формулу из M2 во все ячейки диапазона M3:M1001. Чтобы найти наименьшую сумму баллов, в ячейку E9 запишем формулу:

=МИН(M2:M1001).

9. Введём в ячейку N2 формулу:

=ЕСЛИ(C2=D2;1;0).

Скопируем формулу из N2 во все ячейки диапазона N3:N1001. Сумма значений ячеек диапазона N2:N1001 даст нам искомое количество совпадений баллов (ячейка E10):

=СУММ(N2:N1001).

*Ответы на вопросы 1–9:*

№	1	2	3	4	5	6	7	8	9
<b>Ответ</b>	391	198	66	152	23	32	93	21	52

После выполнения всей или части предлагаемой практической работы можно продолжить работу с презентацией.

При наличии времени в конце второго из уроков можно предложить ученикам выполнить онлайн-тест № 3.

### **Онлайн-тест № 3. Встроенные функции и их использование**

1. Заполните пропуски в предложении.

Встроенные функции имеют ..., вызываются с некоторыми ... и возвращают единственное значение – ... .

- тип, условиями, результат;
- имя, аргументами, результат;**
- тип, аргументами, адрес;
- имя, условиями, адрес.

2. Перерасчёт результатов по заданным в ячейках формулам происходит:

- при сохранении на диске;
- автоматически при изменении данных;**
- автоматически при переносе ячейки с формулой в другое место;
- при переходе на другой рабочий лист таблицы.

3. Установите соответствие между именем функции и её результатом:

- |               |  |
|---------------|--|
| 1) МИН        | А) среднее арифметическое тех чисел из указанного диапазона, которые                                 |
| 2) СРЗНАЧЕСЛИ | удовлетворяют заданному условию;   |
| 3) СОВПАД     | Б) количество дней между двумя датами;   |
| 4) ДНИ        | В) наименьший среди указанных аргументов;  |
|               | Г) если две строки в точности совпадают, то возвращается значение ИСТИНА, в противном случае – ЛОЖЬ. |

Ответ:

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>В</b>	<b>А</b>	<b>Г</b>	<b>Б</b>

4. В электронной таблице значение формулы =СРЗНАЧ(B1:D1) равно 4. Чему равно значение ячейки A1, если значение формулы =СУММ(A1:D1) равно 9?

- 3;     
 5;     
 1;     
 3.

5. На рисунке приведён фрагмент электронной таблицы. В ячейку B2 записали формулу =(\$A2\*10+B\$1)^2 и скопировали её вниз на 2 строчки, в ячейки B3 и B4. Какое число появится в ячейке B4?

	A	B	C	D
1		0	1	1
2	1			
3	2			
4	3			
5				

Ответ: **900**.

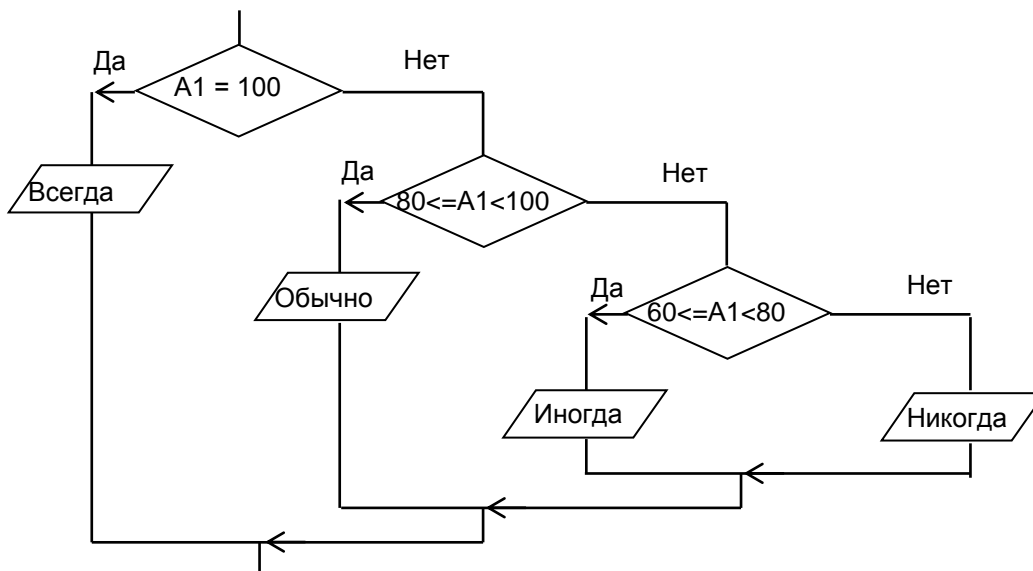
**Домашнее задание:** § 3, вопросы и задания № 1–10 (первый урок), 11–13 (второй урок).

**Указания, комментарии, ответы и решения**

№ 7.

	A	B
1	-10	10
2	ЛОЖЬ	ЛОЖЬ
3	ЛОЖЬ	ИСТИНА
4	ИСТИНА	ИСТИНА
5	ЛОЖЬ	ИСТИНА

№ 8.



№ 9. Пусть значение  $x$  записано в ячейке A1, тогда формула будет иметь вид: =ЕСЛИ(A1<=-5; SIN(A1); ЕСЛИ(A1<5; A1^2; 1/(A1^2-4\*A1))).

№ 11. Уменьшится на 6,25%.

## Урок 5. Инструменты анализа данных

### Планируемые образовательные результаты:

- *предметные*: владение инструментами анализа данных в электронных таблицах; умение использовать электронные таблицы для выполнения учебных заданий из различных предметных областей;
- *метапредметные*: умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;
- *личностные*: наличие мировоззрения, соответствующего современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) актуализация знаний о диаграммах – основном инструменте анализа данных;
- 2) рассмотрение возможностей сортировки и фильтрации данных в электронных таблицах;
- 3) знакомство с возможностями условного форматирования, подбора параметра.

**Основные понятия:** диаграмма, график, область диаграммы, название диаграммы, ось категорий, ось значений, ряды данных, легенда, сортировка данных, фильтрация данных, условное форматирование, подбор параметра.

### Электронное приложение к учебнику:

- презентация «Инструменты анализа данных»;
- онлайн-тест № 4 «Инструменты анализа данных».

### Дополнительные интернет-ресурсы:

- РЭШ (<https://resh.edu.ru>). Урок 14. Обработка информации в электронных таблицах.

### Особенности изложения содержания темы урока

Основные аспекты рассматриваемого материала отражены в презентации. Рассмотрим её содержание.

Самый известный инструмент для анализа данных – диаграмма. К уже популярным видам диаграмм (слайд 4) добавлен новый тип диаграммы (Excel 2016) – «дерево». Эта диаграмма отображает иерархическое представление данных и упрощает выявление тенденций. Например, можно быстро понять, какие запросы встречаются чаще. Ветви дерева представлены прямоугольниками, а дочерние ветви показаны как прямоугольники меньшего размера. На диаграмме «дерево» категории обозначаются цветом и

расположением. На слайде 5 предлагается вопрос о выборе оптимального типа диаграммы для иллюстрации данных таблицы. При выборе неверных ответов отображается комментарий, описывающий недостатки выбранного типа.

На слайде 6 предлагается подобрать диапазон значений таблицы, которые можно отразить с помощью круговой диаграммы. Заметим, что изображение диаграммы не имеет никакого отношения к данным таблицы и является просто условным обозначением типа диаграммы (круговая). Этот тип используется для иллюстрации значений, представляющих часть целого.

	A	B	C	D	E	F
1		Пашни	Леса	Пастбища	Прочие земли	Всего
2	Сев. Америка	3,1	7,5	4,1	9,6	24,3
3	Юж. Америка	1,4	9,4	4,6	2,3	17,8
4	Европа	3,1	3,4	1,9	2,0	10,5
5	Африка	1,9	7,1	7,9	13,4	30,3

*Комментарий к ответам:*

B2 : C3	Диапазон содержит не все типы земель и включает два материка. Нет термина, обозначающего эти данные как нечто целое
B4 : F4	Значение в ячейке F4 – сумма диапазона B4 : E4. Если учащиеся выбирают этот ответ, можно пояснить, что круговая диаграмма, включающая целое (сумму других значений), всегда будет иметь долю равную 50% от целого круга
B5 : E5	Правильный ответ. Диаграмма отражает распределение земель Африки
D2 : D5	Ответ неверный, так как нет объединяющего термина

После представленных выше слайдов можно организовать небольшую практическую работу, предложив ученикам выполнить задание № 9 и хотя бы одно из заданий № 12, 13 к § 4 учебника.

Далее в презентации рассматриваются инструменты **Условное форматирование** (слайд 9) и **Подбор параметра** (слайд 11). С этих слайдов можно перейти к примерам использования инструментов с комментариями (слайды 10 и 12).

*Вопросы и задания* (слайд 15). Для построения графика функции  $\sin(x)$  заполнили таблицу. Как можно объяснить то, что график не похож на синусоиду?

*Ответ.* Шаг значений аргумента функции (2,5 радиан) слишком велик, чтобы наблюдать реальное изменение функции. Между двумя соседними значениями таблицы непрерывная математическая функция  $\sin(x)$  может иметь одну точку экстремума.

При наличии времени можно предложить ученикам выполнить онлайн-тест № 4.



## Онлайн-тест № 4. Инструменты анализа данных

1. Заполните пропуски в предложении.

В отличие от фильтрации, условное ... не скрывает ячейки, значения в которых не удовлетворяют заданному ..., а лишь выделяет заданным образом те ячейки, ... в которых удовлетворяют заданному условию.

- редактирование, аргументу, значения;
- форматирование, условию, значения;**
- форматирование, аргументу, адреса;
- редактирование, условию, адреса.

2. Установите соответствие между названием диаграммы и её возможным видом.

1) точечная;

2) линейчатая;

3) круговая;

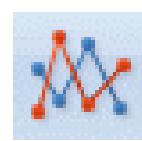
4) график;

5) гистограмма;

А)



Г)



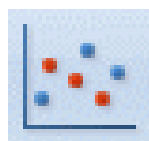
Б)



Д)



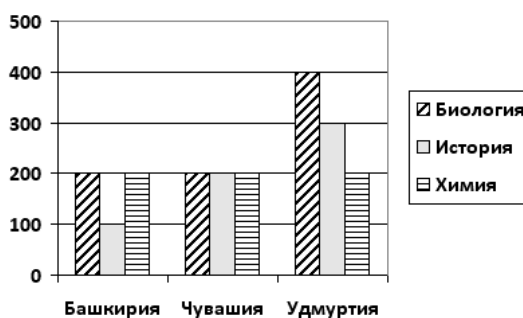
В)



Ответ:

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>В</b>	<b>Д</b>	<b>А</b>	<b>Г</b>	<b>Б</b>

3. На диаграмме представлено количество участников тестирования в разных регионах России:



Какая из диаграмм правильно отражает соотношение количества участников тестирования по биологии в регионах?



4. Фирма продаёт стройматериалы – цемент, песок, щебень. Объёмы продаж измеряются в кубометрах. На диаграмме 1 показаны суммарные продажи по всем трём типам стройматериалов, а на диаграмме 2 – годовое распределение объёма продаж по стройматериалам:

Диаграмма 1

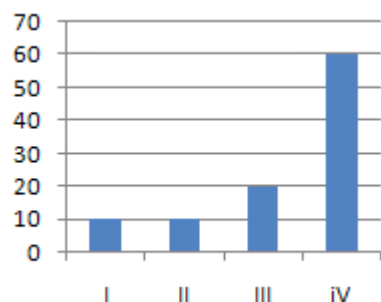
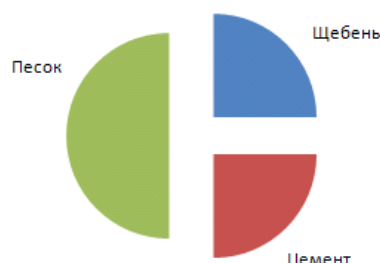


Диаграмма 2



Какое из этих утверждений ПРОТИВОРЕЧИТ информации, показанной на диаграммах?

- в первом квартале продавался только щебень, а во втором – только цемент;
- во втором квартале продавался только песок;
- весь щебень был продан в третьем квартале;
- в первом квартале был продан хотя бы один кубометр песка.

5. Дан фрагмент электронной таблицы:

	A	B	C
1	1	4	?
2	$=1+(B1-A1)*3$	$=B1/2+C1*4$	$=(A1+B1)*4$

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку? Известно, что все значения диапазона, по которым построена диаграмма, имеют один и тот же знак.



Ответ: 2.

Домашнее задание: § 4, вопросы и задания № 1–8, 10–11, 15–19.

Дополнительное задание: № 14, 20.

Указания, комментарии, ответы и решения

№ 9. Возможный вид таблицы:

	A	B	C	D
1	<b>Распределение суши и воды по поверхности земного шара</b>			
2		Северное полушарие, тыс. кв. км	Южное полушарие, тыс. кв. км.	Итого:
3	Суша, тыс. кв. км	99464,04	48456,84	148940
4	Вода, тыс. кв. км	155571,96	206579,16	361132
5	Итого:	255036	255036	510072

№ 10. 9.

## Урок 6. Обобщение и систематизация изученного материала по теме «Обработка информации в электронных таблицах»

### Планируемые образовательные результаты:

- *предметные*: умение использовать электронные таблицы для выполнения учебных заданий из различных предметных областей;
- *метапредметные*: умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности; самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью.
- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) обобщить представления учащихся о функционале электронных таблиц;
- 2) проверить умения учащихся, связанные с использованием электронных таблиц.

**Основные понятия:** табличный процессор, ячейка, диапазон ячеек, ссылка, принцип относительной адресации, формула, формат ячеек, форматирование ячеек, форматирование электронной таблицы, встроенная функция, математические функции, статистические функции, логические функции, текстовые функции, финансовые функции, диаграмма, сортировка данных, фильтрация данных, условное форматирование, подбор параметра.

### Электронное приложение к учебнику:

- интерактивный тест 1.

### Дополнительные печатные материалы:

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

### Особенности изложения содержания темы урока

Повторение и обобщение всего изученного материала можно организовать с помощью интерактивного теста 1 к первой главе учебника. Вопросы теста можно выводить на большой экран и обсуждать их во фронтальном режиме. Работу с тестом можно организовать по группам или индивидуально. Вопросы теста и ответы на них представлены ниже.

#### Интерактивный тест 1 к главе 1 «Обработка информации в электронных таблицах»

1. Табличный процессор – это:
  - двумерный массив данных, состоящий из строк и столбцов;
  - ◎ **прикладная программа для организации табличных расчётов;**
  - автоматизированный аналог обычной таблицы;
  - инструмент работы с таблицами, встроенный в текстовый редактор.
2. Основным преимуществом электронных таблиц по сравнению с обычными, созданными в текстовом редакторе, является:
  - возможность настраиваемой сортировки данных по различным признакам;
  - более наглядное представление данных;
  - дружественный удобный интерфейс;
  - ◎ **встроенная возможность осуществления автоматических расчётов.**
3. Лист электронной таблицы – это:
  - файл, содержащий только одну электронную таблицу;
  - ◎ **одна страница рабочей книги, разделённая на строки и столбцы;**
  - группа смежных ячеек;
  - область, определяемая именами верхней левой и правой нижней ячеек.
4. Установите соответствие между объектами табличного процессора и его свойствами.

1) рабочая книга;	А) общее количество строк и столбцов;
2) электронная таблица;	Б) тип, вид, название, размер области диаграммы, цветовая гамма;
3) строка;	В) имя, количество листов;
4) диаграмма.	Г) номер, высота, количество заполненных данными ячеек.

Ответ:

1	2	3	4
В	А	Г	Б

5. Какое число будет записано в ячейку В1 после ввода формулы?

	A	B
1	1	=A1+2*A2
2	2	

Ответ: 5.

6. Запишите адрес выделенного диапазона.

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

Ответ: C3:E11.

7. Редактирование книги состоит в:

- добавлении в неё новых листов;
- добавлении в неё новых строк или столбцов;
- удалении из неё имеющихся листов;
- удалении из неё имеющихся строк или столбцов;
- перемещении существующих листов;
- копировании существующих листов.

8. Основными операциями форматирования объектов электронной таблицы являются:

- форматирование данных;
- форматирование формул;
- форматирование ячеек;
- изменение ширины столбцов;
- изменение высоты строк.

9. Установите соответствие между видом числа в ячейке и заданным для этой ячейки форматом.

- |                      |                |
|----------------------|----------------|
| 1) общий;            | А) 19.01.1900; |
| 2) денежный;         | Б) 1,97E+01;   |
| 3) дата;             | В) 19,7;       |
| 4) экспоненциальный; | Г) 19 2/3;     |
| 5) дробный.          | Д) 19,70р.     |

Ответ:

1	2	3	4	5
В	Д	А	Б	Г

10. При форматировании ячеек электронной таблицы можно устанавливать:
- границы ячейки, их цвет, тип линий и др.;**
  - ширину столбца, в который входит ячейка;
  - цвет фона ячейки, цвет и стиль узора, способы заливки и др.;**
  - высоту строки, в которую входит ячейка;
  - значения свойств символов в ячейке: шрифт, стиль шрифта, размер, подчёркивание, горизонтальное и вертикальное выравнивание, ориентация, расположение и др.**
11. В ячейке A1 электронной таблицы записана формула =D1-\$D2. Какой вид приобретёт формула после того, как ячейку A1 скопируют в ячейку B1?
- =E1-\$E2;
  - =E1-\$D2;**
  - =E2-\$D2;
  - =D1-\$E2.
12. Заполните пропуски в предложении.  
В любом ... процессоре используются встроенные ... – заранее написанные процедуры преобразования ... .
- табличном, формулы, чисел;
  - электронном, функции, чисел;
  - табличном, функции, данных;**
  - электронном, формулы, данных.
13. Что из перечисленного может быть аргументом функции?
- число;**
  - текст;**
  - выражение;**
  - ссылка на ячейку или диапазон ячеек;**
  - имя листа рабочей книги;
  - результат другой функции.**
14. Среди встроенных в табличные процессоры функций принято выделять:
- тригонометрические;
  - статистические;**
  - профилактические;
  - текстовые;**
  - динамические.
15. В электронной таблице значение формулы =СУММ(C3:E3) равно 15. Чему равно значение формулы =СРЗНАЧ(C3:F3), если значение ячейки F3 равно 5?
- 20;
  - 10;
  - 5;**
  - 4.

16. На рисунке приведён фрагмент электронной таблицы. Чему будет равно значение ячейки В4, в которую записали формулу =СУММ(А1:В2;С3)?

	А	В	С	Д
1	1	2	3	
2	4	5	6	
3	7	8	8	
4				

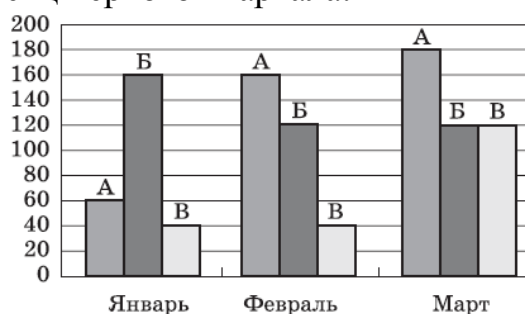
- 14;                       15;                       17;                       20.
17. Какие типы диаграмм позволяют строить табличные процессоры?

- гистограммы;  
 кардиограммы;  
 круговые диаграммы;  
 графики;  
 линейчатые диаграммы;  
 трубчатые диаграммы;  
 лепестковые диаграммы.

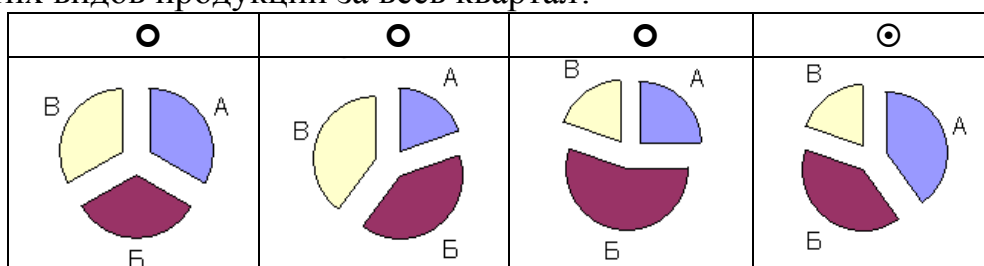
18. Какие объекты можно выделить в диаграмме любого типа?

- таблица данных;  
 область диаграммы;  
 ось построения;  
 подписи данных;  
 поле для заметок;  
 легенда.

19. На диаграмме показаны объёмы выпуска продукции трёх видов (А, Б и В) за каждый месяц первого квартала:



Какая из диаграмм правильно отражает соотношение объёмов выпуска этих видов продукции за весь квартал?



20. Все ученики старших классов (с 9-го по 11-й) участвовали в школьной спартакиаде. По результатам соревнований каждый из них получил от 0 до 3 баллов. На диаграмме 1 показано количество учеников по классам, а на диаграмме 2 – количество учеников, набравших баллы от 0 до 3.

Диаграмма 1

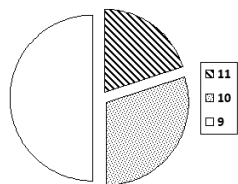
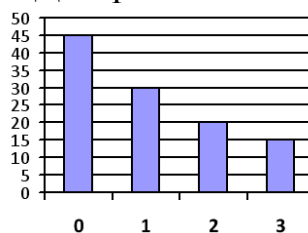


Диаграмма 2



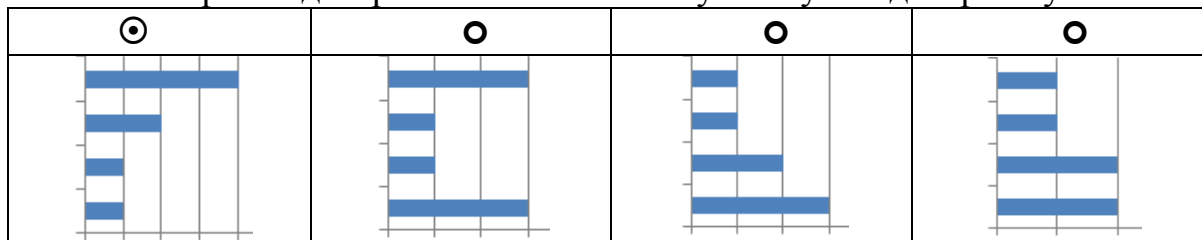
Какое из этих утверждений следует из анализа обеих диаграмм?

- среди учеников 9 класса есть хотя бы один, набравший 2 или 3 балла;
- все ученики, набравшие 0 баллов, могут быть 9-классниками;**
- все 10-классники могли набрать ровно по 2 балла;
- среди набравших 3 балла нет ни одного 10-классника.

21. Дан фрагмент электронной таблицы:

	A	B	C	D
1		3	4	
2	=C1-B1	=B1-A2*2	=C1/2	=B1+B2

После выполнения вычислений по значениям диапазона ячеек A2:D2 была построена диаграмма. Укажите получившуюся диаграмму.



Можно провести письменную контрольную работу № 1 «Обработка информации в электронных таблицах», имеющую два варианта одинакового уровня сложности.

Ответы к заданиям контрольной работы № 1:

№	Вариант 1	Вариант 2
2	5	68
4	=E1-\$D2	14
5	5	-3
6	20	900
7	ЛОЖЬ    ИСТИНА ИСТИНА    ЛОЖЬ ЛОЖЬ    ИСТИНА	ЛОЖЬ    ЛОЖЬ ИСТИНА    ИСТИНА ИСТИНА    ИСТИНА
8	Обычно	Иногда



## Урок 7. Основные сведения об алгоритмах

### Планируемые образовательные результаты:

- *предметные*: определять результат выполнения алгоритма при заданных исходных данных; выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; понимать и использовать основные понятия, связанные со сложностью вычислений (время работы, размер используемой памяти);
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью.
- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) актуализация основных сведений об исполнителях, алгоритмах, их свойствах, способах записи алгоритма;
- 2) решение задач, предполагающих выполнение имеющегося алгоритма при заданных исходных данных, анализ имеющегося алгоритма, составление линейного алгоритма;
- 3) знакомство с понятием сложности алгоритма.

**Основные понятия:** алгоритм, исполнитель алгоритма, свойства алгоритма, дискретность, детерминированность, понятность, результативность, конечность, массовость, вычислительный процесс, сложность алгоритма.

### Электронное приложение к учебнику:

- презентация «Основные сведения об алгоритмах»;
- онлайн-тест № 5 «Основные сведения об алгоритмах».

### Дополнительные интернет-ресурсы:

- РЭШ (<https://resh.edu.ru>). Урок 1. Основные сведения об алгоритмах.

### Особенности изложения содержания темы урока

Актуализацию опорных знаний по теме «Алгоритмизация и программирование» рекомендуется организовать на основе презентации.

Обратите внимание, что в данной презентации большое количество интерактивных элементов, которые позволяют учителю построить урок на своё усмотрение, по индивидуальному плану.

На слайде «Исполнитель алгоритма» (слайд 3) представлено два вида исполнителей, выполняющих одну и ту же работу – вышивание. Слева — неформальный исполнитель золотшвейка с одноимённой картины художника Василия Тропинина (1826), справа — автоматизированные машины для вышивания. С обучающимися сначала можно обсудить, в чём принципиальные различия между этими исполнителями, а затем проверить свои рассуждения, нажав на интерактивные элементы — прямоугольники.

На слайде «Понятие алгоритма» (слайд 4) предусмотрены переходы на скрытые слайды с алгоритмами, возврат с которых осуществляется на слайд их вызова. Выбор рассматриваемых алгоритмов зависит от учителя. Можно рассмотреть какой-нибудь один пример алгоритма или все. На слайдах с алгоритмами кнопка «Выполнить» обеспечивает наглядное пошаговое выполнение алгоритма.

На слайде «Свойства алгоритма» (слайд 10) свойства перечислены слева на интерактивных кнопках, нажатие на которые приводит к подробному описанию того или иного свойства алгоритма. Учитель может либо дать своё более краткое определение, либо озвучить описание, размещённое на слайде. Также при наличии времени можно сначала обсудить с учащимися свойство («Как вы думаете, что означает свойство «дискретность»?»), а затем проверить их ответы.

На слайде «Способы записи алгоритмов» (слайд 12) интерактивные элементы – фигурные прямоугольники – вызывают появление примера записи алгоритма одним из четырёх представленных на слайде способов (словесная запись на естественном языке; запись алгоритма с помощью формул, рисунков, таблиц; запись алгоритма в виде блок-схемы; запись алгоритма на языке программирования). Рассматривать примеры подробно нет необходимости, они являются визуальным образом того или иного способа. Повторный щелчок мышью по объекту восстанавливает экран. Переход на следующий слайд — щелчок вне изображений или пробел.

Презентация также знакомит обучающихся с новым для них понятием «сложность алгоритма» (слайд 14). Данное понятие закрепляется в процессе решения практических задач.

*Ответы и решения на вопросы и задания в презентации.*

*Вопрос* (слайд 11). Можно ли кулинарный рецепт считать алгоритмом? Ответ обоснуйте с точки зрения свойств алгоритма.

*Ответ.* Нет, так как не выполняются свойства детерминированности (результат при исходном наборе данных может быть разным), понятности (соль по вкусу, выпекать до готовности, жарить до золотистой корочки и т. п.).

*Задание* (слайд 15). Оцените сложность алгоритмов.

*Примечание.* В данных задачах за одно действие принимается открытие книги. Действиями по снятию книг с полок можно пренебречь.

а) «Найти книгу с секретом». В старинной библиотеке в одном из 1000 томов, посвященных кладам и тайникам, спрятана книга-сейф. Надо найти её.

*Решение.* При линейном поиске – последовательной проверки всех книг подряд – сложность в худшем случае будет равна количеству книг, т. е.  $O(n) = 1000$ .

*Ответ:* 1000.

б) «Поиск в телефонной книге». В сейфе оказался клочок страницы с фамилией и первой цифрой номера телефона. Надо найти страницу с нужной фамилией в телефонном справочнике, в котором 1000 страниц.

*Решение.* Сложность алгоритма будет  $O(\log_2 n)$ .

Таким образом, в книге объёмом в 1000 страниц страница с нужной фамилией находится не больше, чем за  $O(\log_2 1000) \approx 10$  раз.

*Ответ:* 10.

*Пример 4* (слайд 16). Найти  $X^{40}$  с помощью рассмотренного в презентации алгоритма «Возведение числа в натуральную степень ( $x^n$ )».

*Решение:*

1.  $40 = 101000_2$ .

2. Строим последовательность: КХ К КХ К К К.

3. Вычёркиваем крайнюю левую пару КХ: ККХККК.

4. Вычисляем искомое значение:

К: возвести  $x$  в квадрат ( $x^2$ );

К: возвести результат в квадрат ( $x^4$ );

Х: умножить результат на  $x$  ( $x^5$ );

К: возвести результат в квадрат ( $x^{10}$ );

К: возвести результат в квадрат ( $x^{20}$ );

К: возвести результат в квадрат ( $x^{40}$ ).

*Ответ.* Итак, мы можем вычислить  $X^{40}$  за 6 умножений.

*Задание 3* (слайд 21) (*задание № 4 к § 5*). Есть двое песочных часов: на 3 и на 8 минут. Для приготовления эликсира бессмертия его надо варить ровно 7 минут. Как это сделать? Придумайте систему команд исполнителя Колдун. Запишите с их помощью план действий исполнителя по приготовлению эликсира.

*Примечание для учителя.* На слайде представлен графический способ решения задачи с элементами анимации (которая появляется по пробелу или щелчку мыши). В тетрадах можно выполнить вторую часть задания – разработать СКИ (систему команд исполнителя) и записать алгоритм.

*Возможное решение.*

СКИ:

- Часы (K), где K — количество минут, — песочные часы ставятся на указанное время; запись «Часы (K) = 0» означает, что весь песок оказался в нижней части часов;
- Варить — эликсир ставится на огонь;
- Стоп — снять или не снимать эликсир с плиты; запись «Стоп = да» означает, что эликсир надо снять с плиты;
- + — одновременное действие;
- нц пока ... кц — цикл с предусловием;
- если ... то... иначе — условный оператор.


Алгоритм:

1. Стоп = нет
2. Часы (8) + Часы (3)
3. нц пока Стоп = нет  
     Часы (3)  
     если Часы (8) = 0 то Варить иначе Стоп = да  
     кц

При наличии времени в конце урока можно предложить ученикам выполнить онлайн-тест № 5.

### Онлайн-тест № 5. Основные сведения об алгоритмах

1. Заполните пропуски в предложении.  
     Алгоритм – ... и точное предписание исполнителю выполнить ... последовательность действий, приводящих от исходных данных к конечному ...
  - понятное, конечную, состоянию;
  - конечное, понятную, состоянию;
  - ⊙ **понятное, конечную, результату;**
  - конечное, понятную, результату.
2. Понятность – это свойство алгоритма, заключающееся в том, что:
  - каждая команда алгоритма понятна разработчику алгоритма;
  - компьютер понимает результат выполнения каждой команды;
  - ⊙ **каждая команда алгоритма понятна исполнителю алгоритма;**
  - человек понимает язык, на котором записан алгоритм.
3. Какую смысловую нагрузку несёт блок?
 



  - ввод-вывод;
  - ⊙ **модификация;**
  - процесс;
  - решение.
4. Саша и Женя играют в такую игру. Саша пишет слово русского языка. Женя заменяет в нём каждую букву на другую букву так, чтобы были выполнены следующие правила.

- Гласная буква меняется на согласную, согласная – на гласную.
- В получившемся слове буквы следуют в алфавитном порядке.

*Пример.* Саша написала: ЖЕНЯ. Женя может написать, например, ЕНОТ или АБУЧ. Но не может написать МАМА или ИВАН.

*Для справки.* В алфавите буквы идут в таком порядке:  
АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

Саша написала: КОТ. Укажите, какое из следующих слов может написать Женя.

- ЭЛЬ;
- ЕНОТ;
- ◎ АНЯ;
- ЭЛЯ.

5. Исполнитель КАЛЬКУЛЯТОР имеет только две команды, которым присвоены номера:

1. умножь на 3;
2. вычти 2.

Выполняя команду номер 1, КАЛЬКУЛЯТОР умножает число на экране на 3, а выполняя команду номер 2, вычитает из числа на экране 2. Напишите программу, содержащую не более 5 команд, которая преобразует число 1 в число 23. Укажите лишь номера команд.

Ответ: **11122**.

**Домашнее задание:** § 5, вопросы и задания № 1–3, 5–9, 11–15.

*Дополнительное задание:* № 10 — это задание может быть предложено ученикам «гуманитарного» склада, не планирующим сдавать ЕГЭ по информатике.

### Указания, комментарии, ответы и решения

№ 5. Всего можно составить  $2^5 = 32$  разные программы.

Из них одна будет приводить к результату  $x + 25$ ; 5 будут приводить к результату  $x + 18$ ; 10 — к результату  $x + 11$ ; 10 — к результату  $x + 4$ ; 5 — к результату  $x - 3$ ; одна — к результату  $x - 10$ .

№ 6. Приведём вариант возможного обоснования. Предположим, такой исполнитель существует. Тогда среди его допустимых действий могут быть следующие: 1 — «поднять любой камень»; 2 — «создать камень, который невозможно поднять». Противоречие, которое при этом возникает, доказывает невозможность существования исполнителя, для которого любое действие является допустимым.

№ 9. СКИ:

- цифра( $n$ ) – нахождение  $n$ -й цифры заданного числа;
- сумма( $n, m$ ) – нахождение суммы  $n$ -й и  $m$ -й цифр заданного числа;
- поиск в последовательности из трёх чисел числа, не превышающего двух других чисел;

- удаление в последовательности из трёх чисел числа, не превышающего двух других чисел;
- запись двух чисел в порядке убывания без разделителей.

Результатом работы исполнителя число:

- 1) 1610 быть не может, так как «16» и «10» записаны по убыванию;
- 2) 1010 может быть, например, если в качестве исходного рассмотреть число 5528 (10, 7, 10);
- 3) 1019 быть не может, так как наибольшая сумма двух цифр в десятичной системе счисления не превышает 18.

Минимальное значение результата работы исполнителя: 10.

Максимальное значение результата работы исполнителя: 1818.

9959 – наибольшее значение  $x$ , при котором алгоритм выдаст результат 1418; 1599 – минимальное значение  $x$ , при котором алгоритм выдаст результат 1418.

№ 13. Умножение двух чисел столбиком в случае, если одно из них состоит из  $n$ , а другое — из  $m$  цифр, требует  $n \times m$  элементарных умножений и заведомо не больше по порядку  $n \times m$  элементарных сложений. То есть данный алгоритм имеет сложность порядка  $O(n \times m)$ . Выражение показывает только порядок величины.

№ 15.  $152 = 128 + 16 + 8 = 10011000_2 \rightarrow \text{KXKKKXKXKKK} \rightarrow X^2 \rightarrow X^4 \rightarrow X^8 \rightarrow X^9 \rightarrow X^{18} \rightarrow X^{19} \rightarrow X^{38} \rightarrow X^{76} \rightarrow X^{152}$ . Можно обратить внимание учащихся на сходство этого алгоритма со схемой Горнера.

## Урок 8. Алгоритмические структуры

### Планируемые образовательные результаты:

- *предметные*: определять результат выполнения алгоритма при заданных исходных данных; выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью.

- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

#### **Решаемые учебные задачи:**

- 1) актуализация основных сведений о базовых алгоритмических конструкциях (структурах);
- 2) решение задач, предполагающих выполнение имеющегося алгоритма при заданных исходных данных, анализ имеющегося алгоритма, составление алгоритма.

**Основные понятия:** алгоритм, основные алгоритмические конструкции, последовательная структура, ветвящаяся структура, циклическая структура.

#### **Электронное приложение к учебнику:**

- презентация «Алгоритмические структуры»;
- онлайн-тест № 6 «Алгоритмические структуры».

#### **Дополнительные интернет-ресурсы:**

- РЭШ (<https://resh.edu.ru>). Урок 2. Базовые алгоритмические структуры.

#### **Дополнительные печатные материалы:**

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

#### **Особенности изложения содержания темы урока**

Актуализацию опорных знаний по теме урока рекомендуется организовать на основе презентации. Основное внимание в презентации уделено решению задач, в том числе типовых заданий для подготовки к ЕГЭ по информатике. Все задачи, рассмотренные в презентации, имеют решение и ответ, которые появляются при выборе соответствующих кнопок на слайде.

*Ответы и решения на вопросы и задания в презентации.*

*Пример 1 (слайд 4).*

*Ответ.* Данный алгоритм возводит заданное число в 9-ю степень.

*Пример 2 (слайд 5).*

*Ответ.* Данный алгоритм находит значение переменной  $y$ , заданной системой:

$$y = \begin{cases} -x, & x < 1 \\ -1, & 1 \leq x < 4 \\ x - 5, & x \geq 4 \end{cases}$$

*Вопрос (слайд 8).* Какие ассоциации, связанные с основными алгоритмическими конструкциями, вызывают данные объекты. Объясните свой выбор.

*Примерные ответы:*

- последовательные: эволюция человечества, жизнь человека, песочные часы;
- ветвящиеся: развилка на дороге, выбор, шекспировский Гамлет («Быть или не быть...»), сферы специализации левого и правого полушарий головного мозга;
- циклические: часы (со стрелками), метаморфозы бабочки;
- рекурсивные: дерево, лист папоротника, снежинка.

*Примеры обоснования:* как в линейном алгоритме все команды выполняются строго друг за другом, так и эволюция идёт в одном направлении без возможности возврата назад; движение стрелок по циферблату похоже на циклический алгоритм: стрелки часов проходят один и тот же путь по кругу (тело цикла) до тех пор, пока не кончится завод или батарейка (условие завершения цикла).

*Задание 1-А* (слайд 9). У исполнителя Вычислитель три команды:

- **прибавь 1** – увеличивает число на экране на 1;
- **умножь на 2** – удваивает число;
- **умножь на 3** – утраивает число.

Сколько существует различных программ, которые число 1 преобразуют в число 12?

*Примечание для учителя.* Задание 22 из демонстрационного варианта ЕГЭ 2017 г. (Федеральная служба по надзору в сфере образования и науки Российской Федерации). На уроке желательно рассмотреть актуальное задание данного типа. Решение может быть оформлено разными способами. В учебнике и презентации рассматриваются два разных способа оформления. Учитель может выбрать любой на своё усмотрение. Если способ, предложенный в презентации не подходит, то можно проигнорировать кнопку «Решение» и выбрать сразу кнопку «Ответ».

*Решение:*

	1	2	3	4	5	6	7	8	9	10	11	12
<b>+1</b>		1	2	3	5	5	10	10	15	18	23	23
<b>× 2</b>		1		2		3		5		5		10
<b>× 3</b>			1			2			3			5
<b>Всего</b>	1	2	3	5	5	10	10	15	18	23	23	38

*Ответ:* 38.

*Задание 1-Б* (слайд 11). Сколько существует различных программ, которые число 1 преобразуют в число 12 и при этом траектория вычислений содержит число 6?

*Примечание для учителя.* Задача на основе задачи 1-А. При решении можно ориентироваться на ранее построенную таблицу.

*Решение:*

	1	2	3	4	5	6!	7	8	9	10	11	12
<b>+1</b>		1	2	3	5	5	10	10	10	10	10	10
<b>× 2</b>		1		2		3		–		–		10



× 3			1			2			–			–
<b>Всего</b>	1	2	3	5	5	10	10	10	10	10	10	20

Ответ: 20.

**Задание 1-В** (слайд 12). Сколько существует различных программ, которые число 1 преобразуют в число 12 и при этом траектория вычислений не содержит число 4?

*Примечание для учителя.* Задача на основе задачи 1-А.

*Решение:*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<b>+1</b>		1	2		0	0	5	5	5	8	8	8
<b>× 2</b>		1				3		0		0		5
<b>× 3</b>			1			2			3			0
<b>Всего</b>	1	2	3	<b>0</b>	0	5	5	5	8	8	8	13

Ответ: 13.

**Задание 2** (слайд 13). Решение задачи детально рассмотрено в презентации.

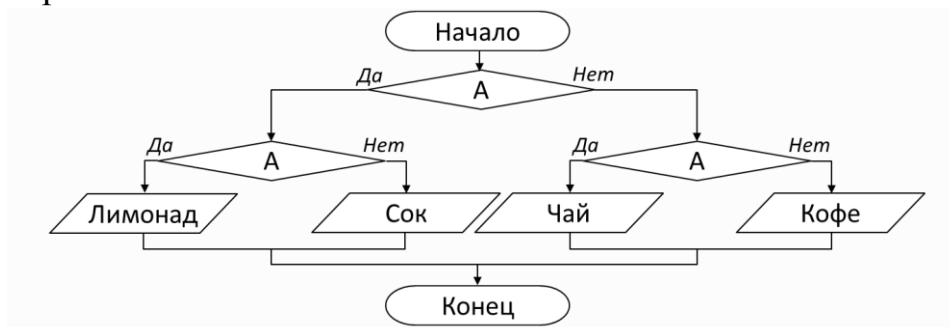
**Задание 3** (слайд 16) (аналогично заданию № 5 к § 6).

*Примечание для учителя.* Данная задача в 11-м классе носит скорее не содержательный, а развлекательный характер.

*Решение.*

Слайд с интерактивными элементами – кнопками А и В. Нажатие кнопок приводит к появлению напитков: АА – лимонад, АВ – сок, ВА – чай, ВВ – кофе. Щелчок по появившемуся напитку удаляет его с экрана. Также он исчезает при начале новой серии команд.

Пример блок-схемы:



При наличии времени в конце урока можно предложить ученикам выполнить онлайн-тест № 6.

### Онлайн-тест № 6. Алгоритмические структуры

1. Какое определение можно использовать для циклического алгоритма?
  - способ представления алгоритма с помощью геометрических фигур;
  - ◎ **алгоритм, содержащий многократные повторения некоторых операторов;**
  - алгоритм, содержащий условия;
  - представление алгоритма в форме таблиц и расчётных формул.

2. Какой тип алгоритма должен быть выбран при решении квадратного уравнения?
- линейный;
  - разветвляющийся;**
  - циклический;
  - любой.

3. Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

**Вперед  $n$** , где  $n$  – целое число, — команда, вызывающая передвижение черепашки на  $n$  шагов в направлении движения;

**Направо  $m$** , где  $m$  – целое число, — команда, вызывающая изменение направления движения на  $m$  градусов по часовой стрелке.

Запись **Повтори 5 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 5 раз.

Черепашке был дан для исполнения следующий алгоритм:

**Повтори 5 [Вперед 10 Направо 72]**

Какая фигура появится на экране?

- незамкнутая ломаная линия;
  - правильный треугольник;
  - квадрат;
  - правильный пятиугольник.**
4. Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив следующую программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка A1)?

НАЧАЛО

ПОКА < слева свободно ИЛИ сверху свободно >

  ЕСЛИ < слева свободно >

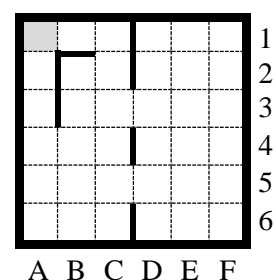
    ТО влево

    ИНАЧЕ вверх

  КОНЕЦ ЕСЛИ

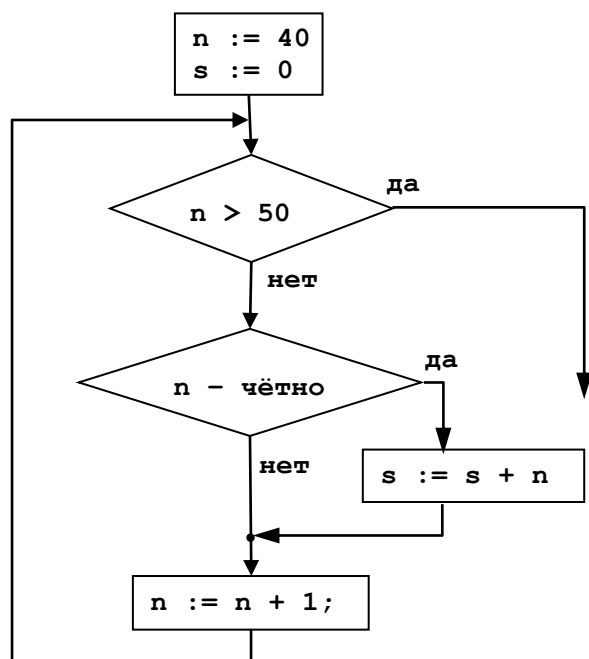
КОНЕЦ ПОКА

КОНЕЦ



- 14;
- 20;**
- 26;
- 28.

5. Определите значение переменной  $s$  после выполнения фрагмента алгоритма.



Ответ: 270.

В сборнике самостоятельных и контрольных работ представлена самостоятельная работа № 1 «Алгоритмы и исполнители», содержащая три разноуровневых варианта.

№	Вариант 1	Вариант 2	Вариант 3
1	1) Всего 8 программ: 111, 112, 121, 122, 211, 212, 221, 222. 2) 64. 3) 121211	1) 2212. 2) 22211	4
2	АНЯ	10139; 101413 (867 и 646)	1) 8352. 2) 2358. 3) 8532
3	$N = 5, a = 6, b = 4$	20	6

**Домашнее задание:** § 6, вопросы и задания № 1–4, 6–9.

### Указания, комментарии, ответы и решения

*Ответы на вопросы в тексте параграфа*

Вопросы к примеру 4 на стр. 82.

При  $N = 22$  получается строка 33; при  $N = 23$  – строка из одной цифры 2; при  $N = 24$  – строка 23.

Вопросы к примеру 4 на стр. 83.

При  $N = 2017$  получается строка 33; при  $N = 12\ 345$  – строка 233.

При  $N = 2015$  – строка 332; при  $N = 12\ 347$  – строка 3.

Вопросы к примеру 5 на стр. 84.

Значение переменной  $S$  в первой блок-схеме не определено, так как переменная  $B$  не определена и её значение по ходу выполнения алгоритма неизвестно, если допустить, что неопределённые переменные равны 0 по умолчанию (в некоторых языках это так), то алгоритм находит первую

значащую цифру числа  $X$ . Рекомендуется дать учащимся задание: «Измените блок-схему так, чтобы вычислялась сумма цифр десятичной записи числа  $X$ ».

Вторая блок-схема позволяет найти количество значащих цифр десятичной записи числа  $X$ .  $X \in \{102, 111, 120, 201, 210, 300\}$ . Ответ на второй вопрос представлен к уже изменённому первому алгоритму, т. е. к алгоритму нахождения суммы цифр ( $S$ ) целого числа ( $X$ ).

*Ответы на вопросы к параграфу*

№ 2. Из условия задачи следует, что в каждом подъезде 64 квартиры.

1. Определим нужный подъезд:  $362 \div 64 = 5$ , следовательно, искомая квартира находится в 6-м подъезде.
2. Определим нужный этаж:  $362 \bmod 64 \div 4 = 10$ , следовательно, искомая квартира находится на 11-м этаже.

№ 9. 1) 2; 2) 3; 3) 3.

## **Уроки 9–11. Запись алгоритмов на языке программирования Паскаль. Анализ программ с помощью трассировочных таблиц. Функциональный подход к анализу программ**

**Планируемые образовательные результаты:**

- *предметные*: выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; определять результат выполнения алгоритма при заданных исходных данных; узнавать изученные алгоритмы обработки чисел и числовых последовательностей; создавать на их основе несложные программы анализа данных; читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; создавать на алгоритмическом языке программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций;
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью.

- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

#### **Решаемые учебные задачи:**

- 1) систематизация представлений о структурной организации данных;
- 2) актуализация основных сведений о языке программирования Pascal (Паскаль);
- 3) рассмотрение разных способов анализа программ.

**Основные понятия:** языки программирования, данные, структура данных, идентификаторы, операторы, трассировочные таблицы.

#### **Электронное приложение к учебнику:**

- презентация «Запись алгоритмов на языках программирования»;
- онлайн-тест № 7 «Запись алгоритмов на языках программирования».

#### **Дополнительные интернет-ресурсы:**

- РЭШ (<https://resh.edu.ru>). Урок 3. Запись алгоритмов на языках программирования. Язык программирования Паскаль (Питон).

#### **Дополнительные печатные материалы:**

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

#### **Особенности изложения содержания темы урока**

**На первом уроке** рекомендуется обсудить с учащимися вопросы, касающиеся структурной организации данных, актуализировать полученные учащимися в основной школе знания и умения по программированию. В презентации соответствующая информация представлена на слайдах 3–10.

В школах используют разные версии языка Pascal (Паскаль), которые могут различаться группой используемых типов данных, их характеристиками; набором стандартных функций и процедур; возможностями и т. д. Все решения проверялись на базе среды программирования PascalABC.NET, однако многие дополнительные возможности данной версии Pascal намеренно не использовались, чтобы решение по возможности было корректным и для других, более ранних, версий языка.

Далее можно рассмотреть примеры программ<sup>1</sup>, приведённые на стр. 91, 92 учебника (примеры 1 и 2).

После этого можно предложить учащимся восстановить навыки программирования в процессе выполнения заданий № 11–13, 15 и 16 к § 7 (по усмотрению учителя).

---

<sup>1</sup> В учебнике все программы написаны на языке Pascal. Учитывая, что сегодня очень популярен язык программирования Python, при наличии времени можно познакомить учеников с этим языком программирования и предложить им далее работать уже на Python.

В качестве **домашнего задания к первому уроку** можно предложить вопросы и задания № 1–6 к § 7, а также один из вариантов разноуровневой самостоятельной работы № 2 «Запись алгоритмов на языке программирования» (ответы даны ниже в разделе «Возможные варианты программ разноуровневой самостоятельной работы № 2»).

**На втором уроке** рекомендуется продолжить работу с презентацией (слайды 11–13), рассмотреть примеры 3–6 на стр. 92–96 учебника, предложить ученикам начать выполнение разноуровневой самостоятельной работы № 3 «Анализ алгоритмов».

№	Вариант 1	Вариант 2	Вариант 3
1	$a = 32, c = 5$	$a = 10, b = -18$	$n = 101, s = 2550$
2	$x = 2, y = 5, t = 5$	$a = 4682, b = 4680$	$P = 24, i = 10$
3	$c = 12$	1024	126
4	1234	19	14
5	29	56	Корректно для $F, G$
6	Корректно для $B$	Корректно для $F$	1) 333, 2) 117, 3) 900, 4) 1001, 5) 11000

В качестве **домашнего задания ко второму уроку** можно предложить вопросы и задания № 7, 8 и 14 к § 7.

**На третьем уроке** продолжается работа с презентацией (слайды 14–19), рассматриваются примеры 7 и 8 на стр. 96–98 учебника, завершается выполнение разноуровневой самостоятельной работы № 3 «Анализ алгоритмов».

В качестве **домашнего задания к третьему уроку** можно предложить вопросы и задания № 9, 10 к § 7.

При наличии времени в конце урока можно предложить ученикам выполнить онлайн-тест № 7.

### Онлайн-тест № 7. Запись алгоритмов на языках программирования

1. Определите значение целочисленных переменных  $a$  и  $b$  после выполнения фрагмента программы:

```
a := 2468;
b := (a mod 1000) * 10;
a := a div 1000 + b;
```

- $a = 22, b = 20$ ;
  - $a = 4682, b = 4680$ ;
  - $a = 8246, b = 246$ ;
  - $a = 470, b = 468$ .
2. Определите значение переменной  $c$  после выполнения следующего фрагмента программы, в котором  $a, b$  и  $c$  — переменные вещественного (действительного) типа:

```
a := 120;
b := 100;
```

```

a := a + b / 2;
if b < a / 2
  then c := b + a
  else c := b + a / 2

```

- $c = 155$ ;
- $c = 160$ ;
- $c = 185$ ;
- $c = 270$ .

3. Определите значение переменной  $P$  после выполнения следующего фрагмента программы:

```

P:=1; i:=3;
while i <= 9 do
  begin
    P := P * (i div 3);
    i := i + 1;
  end

```

- 1;
- 3;
- 9;
- 24.

4. Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```

var n, y: integer;
begin
  y := 0;
  for n := 6 downto 3 do
    begin
      y := y + 1;
      y :=y * 2;
    end
  end.

```

Ответ: **30**.

5. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```

var k, s: integer;
begin
  s:=0;
  k:=1;
  while s < 66 do
    begin
      k:=k+3;
      s:=s+k;
    end;
  write(s);
end.

```

Ответ: **69**.

### Указания, комментарии, ответы и решения

*Ответы на вопросы в тексте параграфа*

Вопрос к примеру 1 на стр. 91.

В усовершенствованной программе строка заголовка цикла может иметь вид:

```
for i:=2 to round(sqrt(n)) do
```

Вопрос к примеру 4 на стр. 94.

При  $d = 31$   $s = 186$ .

При  $d = 35$   $s = 210$ ; при  $d = 36$   $s = 216$ . Такого значения  $d$ , при котором  $s = 212$ , не существует.

Вопрос к примеру 6 на стр. 96.

При  $n = 25$  получим строку: 11001. Находится двоичная запись натурального числа.

Вопрос к примеру 7 на стр. 97.

При условии  $s < 2990$   $n = 432$ ; при  $s \leq 2992$   $n = 434$ ; при  $s \leq 300$   $n = 0$ .

Вопрос к примеру 8 на стр. 98.

Речь идёт о формуле, которая позволяет найти число сочетаний из  $n$

объектов по  $k$ :  $C_n^k = \frac{n!}{(n-k)! \cdot k!}$ . Можно воспользоваться и

формулой для числа перестановок с повторениями:

$$P_n(n_1, n_2) = \frac{n!}{n_1! \cdot n_2!} \cdot P_n(n_1, n_2) = \frac{n!}{n_1! \cdot n_2!}$$

Наибольшее число: 55111; наименьшее число: 11155; остальные числа: 51511, 51151, 51115, 15511, 15151, 15115, 11551, 11515.

*Ответы на задания к параграфу*

№ 6. Алгоритм решает задачу поиска наибольшего из трёх чисел.

Программа поиска наибольшего из четырёх чисел:

```
var x, y, z, w, m : real;
begin
  writeln ('Введите x, y, z, w');
  readln (x, y, z, w);
  m:=x;
  if m<y then m := y;
  if m<z then m := z;
  if m<w then m := w;
  writeln ('m=', m);
end.
```

№ 7. Построим трассировочную таблицу второго вида. Будем считать, что контрольная точка поставлена в строке  $s := s + 2$ .

Результат в КТ	$n$	$s$
Начальные значения	1	0
1	2	2
2	4	4
3	8	6
4	16	8
5	32	10

Ответ:  $n = 32$ .



№ 8. Построим трассировочную таблицу второго вида. Будем считать, что контрольная точка поставлена в строке  $s := s + 2 * i - j$ .

Результат в КТ	$i$	$j$	$s$
Начальные значения	–	–	0
1	1	1	1
2		2	1
3		3	0
4		4	-2
5	2	2	0
6		3	1
7		4	1
8	3	3	4
9		4	6

Ответ:  $s = 6$ .

№ 9. При каждом выполнении тела цикла значение переменной  $s$  уменьшается на 2. Для выхода из цикла оно должно уменьшиться на 902 (до 98); это будет сделано за  $n = 451$  раз.

№ 10. 8888, 1008. Всего таких чисел 2248. Посчитать их можно так: всего четырёхзначных чисел без 9 (8 – самая большая цифра):  $8 \times 9 \times 9 \times 9 = 5832$ . Из них без 8:  $7 \times 8 \times 8 \times 8 = 3584$ . Следовательно, четырёхзначных чисел, в записи которых есть цифра 8, но нет цифры 9 будет 2248.

Более сильным ученикам можно предложить организовать подсчёт таких чисел с помощью программы:

```

var d1, d2, d3, d4, k: integer;
begin
k:=0;
for d1:=1 to 8 do
  for d2:=0 to 8 do
    for d3:=0 to 8 do
      for d4:=0 to 8 do
        if (d1=8) or (d2=8) or (d3=8) or (d4=8)
          then k:=k+1;
      write(k)
    end.

```

Кроме того, можно предложить ученикам прокомментировать следующий способ решения этой задачи:

	A	B	C	D	E	F
1		К-во вариантов				
2	Число вида	1-я цифра	2-я цифра	3-я цифра	4-я цифра	
3		1..7	0..8	0..8	0..8	
4	8 x x x	1	9	9	9	729
5	x 8 x x	7	1	9	9	567
6	x x 8 x	7	8	1	9	504
7	x x x 8	7	8	8	1	448
8						2248
9						

№ 11.

```
Program N_11;  
var i: integer;  
begin  
  for i:=100 to 998 do  
    if i mod 2=0 then write (i:5)  
  end.
```

Рекомендуется обсудить с учащимися, как можно ускорить работу этой программы.

Самое маленькое чётное трёхзначное число — 100, самое большое — 998, значит можно организовать цикл от 50 до 499 и сразу получать те числа, которые являются чётными, при этом избежав проверки на чётность в теле цикла. Быстродействие программы увеличивается в два раза.

```
Program N_11;  
var i: integer;  
begin  
  for i:=50 to 499 do  
    write (i*2:5)  
  end.
```

№ 12.

```
Program N_12;  
var i, n, s: integer;  
begin  
  s := 0;  
  writeln ('Введите n');  
  readln (n);  
  for i := 1 to n do  
    s:=s+i*i;  
  write (s:5)  
end.
```

№ 13.

```
Program N_13;  
var n, x: integer; flag: boolean;  
begin  
  writeln ('Введите цифру n');  
  readln (n);  
  writeln ('Введите число x');  
  readln (x);  
  flag := false;  
  while x>0 do  
    begin  
      if x mod 10 = n  
        then flag := true;  
      x:=x div 10  
    end;  
  if flag  
    then write ('Да')  
    else write ('Нет')  
end.
```

№ 14. Приводим достаточно громоздкое, но доступное большинству учащихся решение. С оператором CASE решение было бы изящней. Можно также ввести дополнительную переменную.

```
Program N_14;  
var n: integer; nt: string;  
begin  
  writeln ('Введите число n');  
  readln (n);  
  nt:='';  
  while n>0 do  
    begin  
      if n mod 3 = 0  
        then nt := '0' + nt;  
      if n mod 3 = 1  
        then nt := '1' + nt;  
      if n mod 3 = 2  
        then nt := '2' + nt;  
      n:=n div 3  
    end;  
  write (nt)  
end.
```

Вариант решения с процедурой str:

```
Program N_14;  
var n: integer; nt, c: string;  
begin  
  writeln('Введите число n');  
  readln (n);  
  nt:='';  
  while n>0 do  
    begin  
      str(n mod 3, c);  
      nt:=c+nt;  
      n:=n div 3  
    end;  
  write (nt)  
end.
```

№ 15.

```
Program N_15_1;  
var x, y: real;  
begin  
  writeln('Введите x, y');  
  readln(x, y);  
  if (x*x+y*y>=9) (abs(x)<=3) and (abs(y)<=3)  
    then writeln('Да')  
    else writeln('Нет');  
end.
```

```
Program N_15_2;  
var x, y: real;  
begin  
  writeln('Введите x, y');
```

```

readln (x, y);
if ((x*x+y*y<=9) and (x*x+y*y>=4) and
    (y>=0) and (x<=0)) or ((x*x+y*y<=4)
    and (y<=0) and (x>=0))
    then writeln('Да')
    else writeln('Нет');
end.

```

№ 16.

```

Program N_16;
var a, b: integer;
begin
    for a:=1 to 9 do
        for b:=0 to 9 do
            if a+b+a*b=a*10+b
                then write (a*10+b:3)
end.

```

*Возможные варианты программ разноуровневой самостоятельной работы  
№ 2 «Запись алгоритмов на языке программирования»*

Вариант 1.

№ 1.

```

program r2_v1_1;
var a, b, c: integer; sa: real;
begin
    a := random(10);
    b := random(10);
    c := random(10);
    sa := (a+b+c)/3;
    writeln ('a=', a);
    writeln ('b=', b);
    writeln ('c=', c);
    writeln ('sa=', sa);
end.

```

№ 2.

```

program r2_v1_2;
var x: integer; t: string;
begin
    writeln ('Сколько Вам полных лет?');
    readln (x);
    if x<6 then t:='Рано';
    if x=6 then t:='1-й отряд';
    if (x>=7) and (x<9) then t:='2-й отряд';
    if (x>=9) and (x<11) then t:='3-й отряд';
    if (x>=11) and (x<13) then t:='4-й отряд';
    if (x>=13) and (x<=15) then t:='5-й отряд';
    if x>15 then t := 'Поздно';
    writeln (t)
end.

```

№ 3.

```

program r2_v1_3;

```

```

var i: integer; N: real;
begin
  N := 40000;
  for i:=1 to 5 do
    N := N + 0.03*N;
  writeln (int(N))
end.

```

## Вариант 2.

### № 1.

```

program r2_v2_1;
var a, b, c: integer; sg: real;
begin
  a := random(20)+1;
  b := random(20)+1;
  c := random(20)+1;
  sg := Exp(Ln(a*b*c)/3);
  writeln ('a=', a);
  writeln ('b=', b);
  writeln ('c=', c);
  writeln ('sg=', sg);
end.

```

*Примечание.* Для вычисления кубического корня в Паскале нет стандартной функции, однако, его можно вычислить, используя функции экспоненты и логарифма. Так как  $a^x = e^{x \cdot \ln a}$ , а извлечение кубического корня — это возведение в степень  $1/3$ , то для положительных чисел можно воспользоваться формулой:

$$x^{1/3} = \exp\left(\frac{1}{3} \ln x\right).$$

### № 2.

```

Program r2_V2_2;
var a, b, c, best: real;
begin
  writeln ('Введите a, b, c');
  readln (a,b,c);
  best:=a;
  if best>b then best := b;
  if best>c then best := c;
  writeln ('Лучший результат - ', best:4:1);
end.

```

*Примечание.* Эта задача на поиск минимального из трёх чисел. Ученики зачастую «лучший» понимают как «максимальный».

### № 3.

```

Program r2_V2_3;
var V, VP, VL: real; i: integer;
begin
  writeln ('Введите значение V');
  readln (V);

```

```

VP := V;
VL := V;
for i:=1 to 10 do
  begin
    VP := VP*1.1;
    VL := VL*1.07;
  end;
writeln ('VP>VL в ',VP/VL:4:2,' раз(a)')
end.

```

### Вариант 3.

№ 1.

```

Program r2_V3_1;
var x, y:real;
begin
  writeln('Введите значение x');
  readln (x);
  y := x + 2.5;
  y := y * x;
  y := y - 1;
  y := y * x;
  y := y + 1;
  writeln('y=',y:6:3)
end.

```

№ 2.

```

Program r2_V3_2;
var k, d1, d2: integer; c: string;
begin
  writeln('Введите k');
  read(k);
  d1 := k mod 10;
  d2 := k mod 100 div 10;
  if k<0 then
    writeln('Ошибка')
  else
    begin
      write('Мы нашли в лесу ',k,' гриб');
      if (d1=1) and (d2<>1) then c := ' '
      else if (d2<>1) and (d1<5) and
        (d1<>1) and (d1<>0)
        then c:='a'
        else c:='об'
    end;
    write(c)
  end.

```

№ 3.

```

Program r2_V3_3;
var x, d: longint;
begin
  readln(x);
  while x>0 do

```

```
begin
  d := x mod 2;
  write(d, ' ');
  x := x div 2
end;
end.
```

## Уроки 12–14. Структурированные типы данных. Массивы. Задачи обработки массивов. Сортировка массивов

### Планируемые образовательные результаты:

- *предметные*: создавать на алгоритмическом языке программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций; использовать знания о постановках задач поиска и сортировки, их роли при решении задач анализа данных; использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; выполнять созданные программы;
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью;
- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) актуализация основных сведений о массивах;
- 2) рассмотрение типовых задач обработки массивов;
- 3) рассмотрение алгоритмов сортировки массивов.

**Основные понятия:** массив, размерность массива, описание массива, типовые задачи обработки одномерных массивов за один просмотр, сортировка массива, метод «пузырька», сортировка выбором.

### **Электронное приложение к учебнику:**

- презентация «Структурированные типы данных. Массивы»;
- онлайн-тест № 8 «Структурированные типы данных. Массивы».

### **Дополнительные интернет-ресурсы:**

- РЭШ (<https://resh.edu.ru>). Урок 5. Массивы.

### **Дополнительные печатные материалы:**

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

### **Особенности изложения содержания темы урока**

При рассмотрении содержания темы рекомендуется использовать презентацию, которую можно считать электронным сборником задач. В ней разобраны девять типовых задач обработки одномерных массивов:

- поиск элементов с заданными свойствами;
- поиск максимумов и минимумов;
- подсчёт элементов, удовлетворяющих условию;
- проверка массива на упорядоченность;
- удаление из массива элемента с индексом  $k$ ;
- вставка в массив элемента на место с индексом  $k$ ;
- перестановка элементов в обратном порядке;
- сортировка массива методом «пузырька»;
- сортировка выбором.

Для учителя предусмотрена удобная система навигации по слайдам презентации. На слайде «Типовые задачи обработки одномерных массивов» (слайд 5) расположены кнопки перехода на слайды с алгоритмами решения задач. Дополнительный вызов решения пошагово демонстрирует программу на языке Паскаль. Для удобства номера примеров совпадают с номерами в учебнике (в презентации разобраны не все программы). Все слайды скрытые (за исключением алгоритмов сортировки), поэтому возврат осуществляется по специальным кнопкам.

Данный способ организации информации в презентации позволяет использовать её не на одном уроке, а на серии уроков, осуществляя выбор необходимого примера.

Особое внимание в презентации уделено сортировке элементов массива (слайды 20–30). Сортировка методом «пузырька» является наиболее простым способом сортировки. В презентации данный метод представлен наглядно с использованием примитивной анимации (слайды 22–27). На слайдах предусмотрена возможность прекратить демонстрацию (кнопка «Стоп» в правом нижнем углу).

На **первом уроке** рекомендуется рассмотреть общие сведения об одномерных массивах; алгоритм последовательного поиска в неупорядоченном массиве; алгоритм поиска максимума (минимума). Соответствующие алгоритмы подробно рассмотрены в примерах 1–4 в § 8 учебника и представлены на слайдах презентации. Можно начать



выполнение разноуровневой самостоятельной работы № 4 «Способы заполнения и типовые приёмы обработки одномерных массивов».

В качестве **домашнего задания к первому уроку** можно предложить вопросы и задания № 1–5 к § 8 учебника.

На **втором уроке** можно рассмотреть алгоритм подсчёта количества элементов, удовлетворяющих некоторому условию, алгоритм проверки массива на упорядоченность, алгоритмы удаления и вставки элементов массива, алгоритм перестановки всех элементов в обратном порядке. Соответствующие алгоритмы подробно рассмотрены в примерах 5–10 в § 8 учебника и представлены на слайдах презентации. Можно продолжить выполнение разноуровневой самостоятельной работы № 4 «Способы заполнения и типовые приёмы обработки одномерных массивов».

В качестве **домашнего задания ко второму уроку** можно предложить вопросы и задания № 6–10 к § 8 учебника.

На **третьем уроке** рекомендуется рассмотреть алгоритмы сортировки массива. Можно завершить выполнение разноуровневой самостоятельной работы № 4 «Способы заполнения и типовые приёмы обработки одномерных массивов».

В качестве **домашнего задания к третьему уроку** можно предложить задание № 11 к § 8 учебника, а также один из вариантов (часть заданий) самостоятельной работы № 5 «Решение задач по обработке одномерных массивов».

При наличии времени в конце третьего урока можно предложить ученикам выполнить онлайн-тест № 8. В тест включено несколько заданий на работу с двумерными массивами, хотя в основном учебном материале речь о них не шла.

### **Онлайн-тест № 8. Структурированные типы данных. Массивы**

1. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведённом ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do A[i]:=i;
for i:=0 to 10 do
begin
A[10-i]:=A[i];
A[i]:=A[10-i];
end;
```

Чему будут равны элементы этого массива?

- 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0;
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10;
  - 10, 9, 8, 7, 6, 5, 6, 7, 8, 9, 10;
  - 0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0.
2. Элементы двумерного массива A размером  $9 \times 9$  задаются с помощью следующего фрагмента программы:

```
for n:=1 to 9 do
```

```

for k:=1 to 9 do
  A[n, k] := n+k+1;

```

Сколько элементов массива  $A$  будут принимать чётные значения?

- 36;     
 40;     
 41;     
 45.

3. Значения элементов двумерного массива  $A[1..100, 1..100]$  задаются с помощью следующего фрагмента программы:

```

for i:=1 to 100 do
  for k:=1 to 100 do
    if i=k
      then A[i, k] := 1
      else A[i, k] := -1;

```

Чему равна сумма элементов массива после выполнения этого фрагмента программы?

- 0;     
 -9800;     
 -9900;     
 -10000.

4. Дан фрагмент программы, обрабатывающий двумерный массив  $A$  размером  $n \times n$ .

```

for i:=1 to n-1 do
  for j:=i to n do
    if A[i, 1] < A[j, 1] then
      begin
        k:=A[i, 1];
        A[i, 1] := A[j, 1];
        A[j, 1] := k;
      end;

```

В этом фрагменте:

- упорядочивается первая строка массива по убыванию;  
 упорядочивается первый столбец массива по убыванию;  
 упорядочивается первая строка массива по возрастанию;  
 упорядочивается первый столбец массива по возрастанию.
5. Ниже представлен фрагмент программы, обрабатывающей одномерный целочисленный массив с индексами от 0 до 10. Известно, что в начале выполнения этого фрагмента в массиве находилась возрастающая последовательность чисел, т. е.  $A[0] < A[1] < \dots < A[10]$ . Какое наибольшее значение может иметь переменная  $s$  после выполнения данной программы?

```

s:=27;
n:=10;
for i:=0 to n-1 do
  begin
    s:=s+A[i]-A[i+1]+2
  end;

```

Ответ: 37.

### Указания, комментарии, ответы и решения

Ответы на вопросы в тексте параграфа

Вопрос к примеру 2 на стр. 105.

Возможный вариант модификации программы:

```
program slova;  
var simbol: array [1..20] of char; i, n: integer;  
begin  
  writeln('Введите n:');  
  readln(n);  
  writeln('Введите слово - цепочку из n символов');  
  i:=0;  
  repeat  
    i := i + 1;  
    read(simbol[i]);  
  until i>n;  
  n := i-1;  
  writeln('Перевернутое слово:');  
  for i:=n downto 1 do  
    write(simbol[i]);  
end.
```

Следует отметить, что работа со строками может быть организована более эффективно:

```
var  
  s1,s2:string; i:integer;  
begin  
  Write('Введите слово: ');  
  Readln(s1);  
  s2:='';  
  for i:=length(s1) downto 1 do  
    s2:=s2+s1[i];  
  Writeln('Перевернутое слово: ',s2)  
end.
```

Вопрос к примеру 3 на стр. 107.

Соответствующий алгоритм рассматривался в учебнике 9 класса.

Вопрос к примеру 4 на стр. 108.

Нахождение минимального элемента массива с помощью цикла for:

```
const n=10;  
var a: array [1..n] of integer;  
    i, min: integer;  
begin  
  writeln('Ввод значений элементов массива:');  
  for i:=1 to n do  
    read(a[i]);  
  min:=a[1];  
  for i:=2 to n do  
    if a[i]<min then min:=a[i];  
  writeln('Min=', min)  
end.
```

Алгоритм поиска наибольшего элемента в неупорядоченном массиве обладает линейной сложностью  $O(n)$ : необходимо просмотреть все  $n$  элементов массива, чтобы понять, какой из них максимальный.

Вопрос к примеру 6 на стр. 109.

```
const n=7;  
var a: array [1..n] of integer; i, k: integer;
```

```

begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n do
    read(a[i]);
  k:=0;
  for i:=1 to n do;
    if (a[i]>9) and (a[i]<100) then k:=k+1;
  if k=n then writeln('Да') else writeln('Нет')
end.

```

Вопрос к примеру 7 на стр. 109.

```

const n=7;
var a: array [1..n] of integer;
    i: integer; flag: boolean;
begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n do
    read(a[i]);
  flag:=true;
  for i:=1 to n-1 do
    if a[i]>a[i+1] then flag:=false;
  if flag then writeln('Да') else writeln('Нет')
end.

```

Эта задача может быть решена и путём подсчёта пар, таких, что  $a[i] > a[i+1]$ :

```

const n=7;
var a: array [1..n] of integer;
    i, k: integer;
begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n do
    read(a[i]);
  k:=0;
  for i:=1 to n-1 do;
    if a[i]>a[i+1] then k:=k+1;
  if k=0 then writeln('Да') else writeln('Нет')
end.

```

Вопрос к примеру 8 на стр. 110.

```

const n=7;
var a: array [1..n] of integer; i, k: integer;
begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n do
    read(a[i]);
  writeln('Введите номер удаляемого элемента:');
  read(k);
  for i:=k to n-1 do
    a[i]:=a[i+1];
  for i:=1 to n-1 do
    writeln(a[i])
end.

```

Вопрос к примеру 9 на стр. 111.

```

const n=8;

```

```

var a: array [1..n] of integer; i, k: integer;
begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n-1 do
    read(a[i]);
  writeln('Введите номер места вставляемого элемента:');
  read(k);
  for i:=n downto k+1 do
    a[i]:=a[i-1];
  writeln('Введите значение вставляемого элемента:');
  read(a[k]);
  for i:=1 to n do
    writeln(a[i])
end.

```

Вопрос к примеру 10 на стр. 112.

```

const n=7;
var a: array [1..n] of integer; i, r: integer;
begin
  writeln('Ввод значений элементов массива:');
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n div 2 do
    begin
      r:=a[i];
      a[i]:=a[n-i+1];
      a[n-i+1]:= r
    end;
  for i:=1 to n do
    writeln(a[i])
end.

```

Вопрос к примеру 11 на стр. 114.

```

const n=7;
var a: array[1..n] of integer;
    i, j, r: integer;
begin
  writeln('Ввод значений элементов исходного массива:');
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n-1 do
    for j:=1 to n-i do
      if a[j]>a[j+1] then
        begin
          r:=a[j];
          a[j]:=a[j+1];
          a[j+1]:=r
        end;
  writeln('Отсортированный массив: ');
  for i:=1 to n do
    write(a[i]:6)
end.

```

Вопрос к примеру 12 на стр. 115.

```

const n=7;
var a: array[1..n] of integer;

```

```

    i, j, imin, per: integer;
begin
    writeln('Ввод значений элементов исходного массива:');
    for i:=1 to n do
        read(a[i]);
    for i:=1 to n-1 do
        begin
            imin:=i;
            for j:=i+1 to n do
                if a[j]<a[imin] then imin:=j;
            per:=a[i];
            a[i]:=a[imin];
            a[imin]:=per;
        end;
    writeln('Отсортированный массив: ');
    for i:=1 to n do write(a[i]:6)
end.

```

### *Ответы на задания к параграфу*

№ 3. 1) Будет найдена сумма индексов  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$ .

2) Вместо суммы элементов программа находит сумму индексов – сумму чисел от 1 до 10. Поэтому, чтобы программа работала правильно, можно, например, ввести числа 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 в любом порядке. В целом же, так как ответ не зависит от значения элементов массива, то программа выдаёт верный ответ во всех случаях, когда сумма элементов равна 55. Например, если  $A[1]=55$ , а все остальные элементы равны 0.

3) Вместо  $s:=s+i$  надо  $s:=s+a[i]$ .

№ 4. 1) 0.

2) Любой массив из десяти элементов, имеющий хотя бы одно нулевое значение.

3) Вместо  $p:=0$  надо  $p:=1$ .

№ 5.

```

const n=7;
var a: array[1..n] of integer;
    i, j, min, max : integer;
begin
    writeln('Ввод значений элементов исходного массива:');
    for i:=1 to n do
        read(a[i]);
    max:=a[1];
    min:=a[1];
    for i:=2 to n do
        if a[i]<min
            then min:=a[i]
            else if a[i]>max
                then max:=a[i];
    writeln('min= ', min, ' max=', max);
end.

```

№ 6. Пусть  $k = 4$ . Преобразованный массив:

<i>i</i>	1	2	3	4	5	6	7
<i>a[i]</i>	10	12	5	4	15	20	20

№ 8. Преобразованный массив:

<i>i</i>	1	2	3	4	5	6	7
<i>a[i]</i>	20	15	4	8	5	12	10

№ 9. Результат выполнения программы:  $\max 1=6$ ,  $\max 2=6$ .

Решается задача нахождения двух наибольших элементов одномерного массива, которые могут быть как равны между собой, так и различаться;  $\max 1$  – наибольший,  $\max 2$  – следующий по величине элемент массива.

№ 10.

```
const t=5;
var a: array[1..t] of integer;
    i, n, max, min, s, p, k: integer;
begin
  write('n=');
  read(n);
  k:=0;
  while n>0 do
    begin
      k:=k+1;
      a[k]:=n mod 10;
      n:=n div 10
    end;
  max:=a[1]; min:=a[1];
  s:=a[1]; p:=a[1];
  for i:=2 to k do
    begin
      if a[i]>max then max:=a[i];
      if a[i]<min then min:=a[i];
      s:=s+a[i];
      p:=p*a[i];
    end;
  writeln('max=', max, ', min=', min);
  writeln('s=', s, ', p=', p);
end.
```

*Самостоятельная работа № 4*

Вариант 1.

№ 1.

```
program r4_v1_1;
const t=4;
var a: array[1..t] of integer; i: integer;
begin
  write ('Введите элементы массива ');
  for i:=1 to t do
    read(a[i]);
  for i:=1 to t do
    writeln('a[' , i, ']=', a[i])
```

end.

№ 2.

```
program r4_v1_2;
const t=20;
var a: array[1..t] of integer; i: integer;
begin
  for i:=1 to t do
    a[i]:=i*i;
  for i:=1 to t do
    writeln('a[' , i, ']=' , a[i])
end.
```

№ 3.

```
program r4_v1_3;
const t=15;
var a: array[1..t] of integer; i: integer;
begin
  for i:=1 to t do
    a[i]:=random(100);
  for i:=1 to t do
    writeln('a[' , i, ']=' , a[i])
end.
```

№ 4.

```
program r4_v1_4;
const t=11;
var a: array[1..t] of real; i: integer; s: real;
begin
  s:=0;
  write ('Введите элементы массива');
  for i:=1 to t do
    begin
      read(a[i]);
      s:=s+a[i];
    end;
  writeln('s=' , s)
end.
s = 123.5.
```

№ 5.

```
program r4_v1_5;
const t=11;
var a: array[1..t] of real; i, k1, k2, k3: integer;
begin
  k1:=0; k2:=0; k3:=0;
  write ('Введите элементы массива');
  for i:=1 to t do
    begin
      read(a[i]);
      if a[i]<0 then k1:=k1+1;
      if a[i]=0 then k2:=k2+1;
```



```

        if a[i]>0 then k3:=k3+1;
    end;
    writeln('отрицательных - ', k1);
    writeln('нулевых - ', k2);
    writeln('положительных - ', k3);
end.

```

№ 6.

```

program r4_v1_6;
const t=11;
var a: array[1..t] of real; i: integer; p: real;
begin
    p:=1;
    write('Введите элементы массива');
    for i:=1 to t do
        begin
            read(a[i]);
            if a[i]<>0 then p:=p*a[i];
        end;
    writeln('p=', p);
end.
p = 718641000.

```

№ 7.

```

program r4_v1_7;
const t=11;
var a: array[1..t] of integer;
    i: integer; s1, s2: real;
begin
    s1:=0; s2:=0;
    write ('Введите элементы массива');
    for i:=1 to t do
        begin
            read(a[i]);
            if a[i] mod 2 = 0
                then s1:=s1+a[i]
                else s2:=s2+a[i];
        end;
    if s1>s2
        then writeln('сумма чётных значений
            больше суммы нечётных на ', s1-s2)
        else if s1=s2
            then writeln('суммы чётных и нечётных
                значений равны ')
            else writeln('сумма нечётных значений
                больше суммы чётных на ', s2-s1)
    end.

```

Сумма нечётных значений больше суммы чётных на 37.

№ 8.

```

program r4_v1_8;
const t=11;

```

```

var a: array[1..t] of real;
    i, k: integer; s, sa: real;
begin
    k:=0; s:=0;
    write ('Введите элементы массива');
    for i:=1 to t do
        begin
            read(a[i]);
            if a[i]>0 then
                begin
                    s:=s+a[i];
                    k:=k+1
                end
            end;
        sa:=s/k;
        writeln('sa=', sa:8:4)
    end.
sa = 19.6429.

```

№ 9.

```

program r4_v1_9;
const t=11;
var a: array[1..t] of real;
    i, k: integer; z: real;
begin
    write('Введите z=');
    read(z);
    k:=0;
    write('Введите элементы массива');
    for i:=1 to t do
        begin
            read(a[i]);
            if a[i]>z then k:=k+1;
        end;
    writeln('k=', k)
end.

```

№ 10.

```

program r4_v1_10;
const t=12;
var a: array[1..t] of integer; i : integer;
begin
    for i:=1 to t do
        begin
            a[i]:=random(10)+1;
            write(a[i]:3);
        end;
    writeln;
    for i:=1 to t do
        if a[i]=i
            then writeln('a[' , i, ']=' , a[i])
    end.

```

## Вариант 2.

№ 1.

```
program r4_v2_1;
const t=12;
var b: array[1..t] of integer;
    i : integer;
begin
  b[1]:=1;
  for i:=2 to t do
    b[i]:=b[i-1]*i;
  for i := 1 to t do
    writeln('B[' , i, ']=' , b[i])
end.
```

B[1]=1	B[4]=24	B[7]=5040	B[10]=3628800
B[2]=2	B[5]=120	B[8]=40320	B[11]=39916800
B[3]=6	B[6]=720	B[9]=362880	B[12]=479001600

№ 2.

```
program r4_v2_2;
const t=20;
var b: array[1..t] of integer;
    i : integer;
begin
  for i:=1 to t do
    begin
      b[i]:=random(81);
      write (b[i]:3);
    end;
end.
```

№ 3.

```
program r4_v2_3;
const t=20;
var a, b: array[1..t] of integer;
    i : integer;
begin
  for i:=1 to t do
    begin
      b[i]:=random(81);
      write (b[i]:3);
    end;
  writeln;
  for i:=1 to t do
    begin
      a[i]:=b[20-i+1];
      write (a[i]:3);
    end;
end.
```

№ 4.

```
program r4_v2_4;
```

```

const t=100;
var a, b: array[1..t] of integer;
    i, k1, k2, k3 : integer;
begin
    for i:=1 to t do
        begin
            a[i]:=random(100);
            b[i]:=random(100);
        end;
    k1:=0; k2:=0; k3:=0;
    for i:=1 to t do
        begin
            if a[i]<b[i] then k1:=k1+1;
            if a[i]=b[i] then k2:=k2+1;
            if a[i]>b[i] then k3:=k3+1;
        end;
        writeln ('k1=',k1);
        writeln ('k2=',k2);
        writeln ('k3=',k3)
    end.

```

№ 5.

```

program r4_v2_5;
const t=11;
var b: array[1..t] of real;
    i, k: integer; s, sa: real;
begin
    k:=0; s:=0;
    write ('Введите элементы массива');
    for i:=1 to t do
        begin
            read(b[i]);
            if b[i] < 0 then
                begin
                    s := s + abs(b[i]);
                    k := k + 1
                end
            end;
        sa:=s/k;
        writeln('sa=', sa:8:4)
    end.
    sa = 84.6250.

```

№ 6.

```

program r4_v2_6;
const t=11;
var b: array[1..t] of real;
    i, k: integer; p, s: real;
begin
    p:=1; s:=0; k:=0;
    write ('Введите элементы массива');
    for i:=1 to t do
        begin

```

```

    read(b[i]);
    if b[i]<0 then s := s + b[i];
    if b[i]=0 then k := k + 1;
    if b[i]>0 then p := p * b[i];
  end;
  writeln('произведение положительных - ', p);
  writeln('сумма отрицательных: ', s);
  writeln('количество нулевых - ', k)
end.

```

№ 7.

```

program r4_v2_7;
const t=11;
var b: array[1..t] of real;
    i: integer; p: real;
begin
  p:=1;
  write ('Введите элементы массива');
  for i:=1 to t do
    begin
      read(b[i]);
      if (abs(b[i]<=20)) and (b[i]<>0)
        then p:=p*b[i];
    end;
  writeln('p=', p)
end.

```

№ 8.

```

program r4_v2_8;
const t=11;
var b: array[1..t] of real;
    i, k: integer; s, sa: real;
begin
  s:=0;
  write ('Введите элементы массива');
  for i:=1 to t do
    begin
      read(b[i]);
      s:=s+b[i];
    end;
  sa:=s/t;
  k:=0;
  for i:=1 to t do
    if b[i]>sa then k:=k+1;
  writeln('k=', k)
end.

```

№ 9.

```

program r4_v2_9;
const t=11;
var b: array[1..t] of real;
    i: integer; max: real;
begin

```

```

write ('Введите элементы массива');
for i:=1 to t do
  read(b[i]);
max:=b[1];
for i:=2 to t do
  if b[i]>max then max:=b[i];
writeln('max=', max)
end.

```

№ 10.

```

program r4_v2_10;
const t=12;
var b: array[1..t] of integer;
    i, maxotr : integer;
begin
  for i:=1 to t do
    begin
      b[i]:=random(20)-10;
      write (b[i]:4)
    end;
  maxotr:= -10;
  for i:=1 to t do
    if (b[i]<0) and (b[i]>maxotr)
      then maxotr:=b[i];
  writeln;
  write ('maxotr=', maxotr)
end.

```

Вариант 3.

№ 1.

```

program r4_v3_1;
const t=20;
var c: array[1..t] of integer; i : integer;
begin
  for i:=1 to t do
    begin
      if i mod 2 = 0
        then c[i] := i div 2
        else c[i] := 0;
      writeln ('c[' , i, ']=', c[i])
    end
  end.

```

№ 2.

```

program r4_v3_2;
const t=20;
var c: array[1..t] of integer; i : integer;
begin
  c[1]:=1;
  c[2]:=2;
  for i:=3 to t do
    begin

```

```

        c[i]:=c[i-1]+c[i-2];
        writeln ('C[' ,i, ']=' ,c[i])
    end
end.

```

C[1]=1	C[6]=13	C[11]=144	C[16]=1597
C[2]=2	C[7]=21	C[12]=233	C[17]=2584
C[3]=3	C[8]=34	C[13]=377	C[18]=4181
C[4]=5	C[9]=55	C[14]=610	C[19]=6765
C[5]=8	C[10]=89	C[15]=987	C[20]=10946

№ 3.

```

program r4_v3_3;
const t=20;
var c: array[1..t] of integer;
    i, s : integer;
begin
    c[1]:=1;
    c[2]:=2;
    s:=c[1];
    for i:=3 to t do
        begin
            s:=s+c[i-1];
            c[i]:=s;
            writeln ('c[' ,i, ']=' ,c[i])
        end
    end.

```

№ 4.

```

program r4_v3_4;
const t=20;
var c: array[1..t] of integer; i : integer;
begin
    for i:=1 to t do
        begin
            c[i]:=random(81)+20;
            write (c[i]:3);
        end;
    end.

```

№ 5.

```

program r4_v3_5;
const t=100;
var c: array[1..t] of integer;
    i, max, k : integer;
begin
    for i:=1 to t do
        begin
            c[i]:=random(81)+20;
            write (c[i]:4);
        end;
    max:=c[1];
    for i:=2 to t do
        if c[i]> max then max:=c[i];

```

```

writeln;
writeln ('max=', max);
k := 0;
for i:=1 to t do
    if c[i]=max then k:=k+1;
writeln('k=', k)
end.

```

№ 6.

```

program r4_v3_6;
const t=100;
var c: array[1..t] of integer;
    i, max, min, raz : integer;
begin
    for i:=1 to t do
        begin
            c[i]:=random(81)+20;
            write (c[i]:4);
        end;
    max:=c[1];
    min:=c[1];
    for i:=2 to t do
        begin
            if c[i]>max then max:=c[i];
            if c[i]<min then min:=c[i];
        end;
    writeln;
    raz:=max-min;
    writeln ('max=', max);
    writeln ('min=', min);
    writeln ('raz=', raz)
end.

```

№ 7.

```

program r4_v3_7;
const t=11;
var c: array[1..t] of real;
    i, imax, imin: integer; d: real;
begin
    write ('Введите элементы массива');
    for i:=1 to t do
        read(c[i]);
    imax:=1;
    imin:=1;
    for i:=2 to t do
        begin
            if c[i]>c[imax] then imax:=i;
            if c[i]<c[imin] then imin:=i;
        end;
    writeln ('max=', c[imax], ' imax=', imax);
    writeln ('min=', c[imin], ' imin=', imin);
    d:=c[imin];
    c[imin]:=c[imax];

```



```

    c[imax]:=d;
    for i:=1 to t do
        writeln (c[i]:5);
    end.

```

№ 8.

```

program r4_v3_8;
const t = 11;
var c: array[1..t] of real;
    i, j : integer; k: real;
begin
write ('Введите элементы массива');
    for i:=1 to t do
        read(c[i]);
    for i := 1 to t-1 do
        for j := 1 to t-i do
            if c[j]>c[j+1] then
                begin
                    k := c[j];
                    c[j] := c[j+1];
                    c[j+1] := k
                end;
        write ('Отсортированный массив: ');
    for i:=1 to t do
        write (c[i]:5);
end.

```

№ 9.

```

program r4_v3_9;
const t=10;
var c: array[1..t] of integer;
    i, j : integer; flag : boolean;
begin
    for i:=1 to t do
        begin
            c[i]:=random(20);
            write (c[i]:5);
        end;
    flag:=false;
    for i:=1 to t-1 do
        for j:=i+1 to t do
            if c[i]=c[j] then flag := true;
    writeln;
    if flag then write('Да') else write('Нет');
end.

```

№ 10.

```

program r4_v3_10;
const t=100;
var c: array[1..t] of integer;
    i, d, s : integer;
begin
    for i := 1 to t do

```

```

    begin
        c[i]:=random(2);
        write (c[i]);
    end;
d:=0;
s:=0;
for i:=1 to t do
    if c[i]=0
        then s:=s+1
        else begin
            if s>d then d:=s;
            s:=0;
        end;
    if s>d then d:=s;
    writeln;
    write('Длина самой большой цепочки подряд
        идущих 0 равна ', d)
end.

```

## *Самостоятельная работа № 5*

### Вариант 1.

#### № 1.

```

program r5_v1_1;
const t=100;
var c: array[1..t] of integer;
    i, k : integer;
begin
    writeln ('Исходный массив');
    for i := 1 to t do
        begin
            c[i] := random(2);
            write (c[i]);
        end;
    k := 0;
    for i := 1 to t do
        if c[i] = 0 then k := k + 1;
        writeln;
        writeln ('Упорядоченный массив');
        for i := 1 to k do
            write (0);
        for i := k + 1 to t do
            write (1);
    end.

```

#### № 2.

```

program r5_v1_2;
const t=10;
var a, b: array[1..t] of integer;
    i : integer;
begin
    for i := 1 to t do
        begin

```

```

        a[i] := random(100)+80;
        b[i] := a[i]+random(10)-5;
    end;
for i := 1 to t do
begin
    write('Было: ', a[i]:4, ' Стало: ', b[i]:4);
    if b[i]<a[i] then write(' Поздравлю!');
    writeln;
end
end.

```

№ 3.

```

program r5_v1_3;
const n=10;
var a: array[1..n] of integer;
    i, s: integer; sa: real;
begin
    s:=0;
    for i:=1 to n do
        begin
            a[i]:=random(12000)+20000;
            write(a[i]:8);
            s:=s+a[i]
        end;
    writeln;
    sa := s/n;
    writeln('sa=', sa:10:4)
end.

```

№ 4.

```

program r5_v1_4;
const n=20;
var a: array[1..n] of integer;
    i, s, k: integer; sa: real;
begin
    s:=0; k:=0; i:=1;
    while i<=n do
        begin
            a[i] := random(40)+140;
            a[i+1] := random(40)-200;
            write(a[i]:6); write(a[i+1]:6);
            i:=i+2
        end;
    for i:=1 to n do
        if a[i]<0 then
            begin
                s := s+abs(a[i]);
                k := k+1;
            end;
    writeln;
    sa := s/k;
    writeln('sa=', sa:10:4)
end.

```

№ 5.

```
program r5_v1_5;
const n=20;
var a: array[1..n] of integer;
    i, s, k: integer; sa, p: real;
begin
    s:=0; k:=0;
    for i:=1 to n do
        begin
            a[i]:=random (4)+2;
            write(a[i]:3);
            s := s+a[i];
            if a[i]>2 then k:=k+1;
        end;
    writeln;
    sa := s/n;
    writeln('sa=', sa:4:2);
    p := k*100/n;
    writeln('p=', p:4:1)
end.
```

Вариант 2.

№ 1.

```
program r5_v2_1;
const n=10;
var a: array[1..n] of integer;
    i, x, k: integer;
begin
    a[1] := random(10);
    write(a[1]);
    for i:=2 to n do
        begin
            a[i]:=a[i-1]+random (10)+1;
            write(a[i]:3);
        end;
    writeln;
    writeln('x=');
    readln(x);
    for i:=1 to n-1 do
        if (x>=a[i]) and (x<=a[i+1]) then
            begin
                k := i;
                writeln ('[' ,a[i],';',a[i+1],']')
            end;
        if k=0 then writeln ('x не принадлежит
                               ни одному из отрезков')
    end.
```

№ 2.

```
program r5_v2_2;
const t=50;
var a: array[1..t] of integer;
    i, j : integer;
```

```

begin
  writeln ('Исходный массив');
  for i := 1 to t do
    begin
      a[i]:=random(5);
      write (a[i]);
    end;
  writeln;
  j:=0;
  for i := 1 to t do
    if a[i]<>0 then
      begin
        j := j+1;
        a[j]:=a[i]
      end;
    for i := j+1 to t do
      a[i]:=0;
    writeln ('Упорядоченный массив');
    for i := 1 to t do
      write (a[i])
    end.

```

№ 3.

```

program r5_v2_3;
const t=30;
var a, b: array[1..t] of integer;
    i, k1, k2 : integer;
begin
  writeln ('Исходный массив - температура');
  for i := 1 to t do
    begin
      a[i]:=random(20)-10;
      write (a[i]:3);
    end;
  writeln;
  writeln ('Исходный массив - осадки');
  for i := 1 to t do
    begin
      b[i]:=random(5);
      write (b[i]:3);
    end;
  writeln;
  k1 := 0;
  k2 := 0;
  for i := 1 to t do
    begin
      if (a[i]<0) and (b[i]<>0) then k1 := k1+1;
      if (a[i]>0) and (b[i]<>0) then k2 := k2+1
    end;
  writeln ('Дней со снегом k1=',k1);
  writeln ('Дней с дождем k2=',k2);
end.

```

№ 4.

```
program r5_v2_4;
const t=30;
var r, v: array[1..t] of integer;
    i, sr, sv : integer; flag: boolean;
begin
  writeln ('Исходные данные');
  for i := 1 to t do
    begin
      r[i]:=random(40)+140;
      write (r[i]:5);
    end;
  for i := 1 to t do
    begin
      v[i]:=random(30)+40;
      write (v[i]:5);
    end;
  writeln;
  sr := 0;
  sv := 0;
  for i := 1 to t do
    begin
      sr:=sr+r[i];
      sv:=sv+v[i]
    end;
  sr := sr div t;
  sv := sv div t;
  writeln ('sr=',sr);
  writeln ('sv=',sv);
  flag := false;
  for i := 1 to t do
    if (r[i]=sr) and (v[i]=sv) then flag:=true;
  if flag then writeln ('Есть')
    else writeln ('Нет')
end.
```

№ 5.

```
program r5_v2_5;
var a: array[1..30] of integer;
    b,c: array[1..31] of integer;
    i,k1,k2,k3, max : integer;
begin
  k1:=0; k2:=0; k3:=0;
  writeln ('Исходные данные');
  for i := 1 to 30 do
    begin
      a[i]:=random(2)+1;
      write (a[i]);
      if a[i]=1 then k1 := k1+1
    end;
  writeln;
  for i := 1 to 31 do
    begin
```

```

        b[i]:=random(2)+1;
        write (b[i]);
        if b[i]=1 then k2 := k2+1
    end;
writeln;
for i := 1 to 31 do
    begin
        c[i]:=random(2)+1;
        write (c[i]);
        if c[i]=1 then k3 := k3+1
    end;
writeln;
writeln(k1:5, k2:5, k3:5);
max := k1;
if k2>max then max := k2;
if k3>max then max := k3;
if max=k1 then writeln ('Июнь');
if max=k2 then writeln ('Июль');
if max=k3 then writeln ('Август');
end.

```

### Вариант 3.

№ 1.

```

program r5_v3_1;
const n=10;
var a: array[1..n] of string;
    b: array[1..n] of integer;
    i, j, k : integer; f: string;
begin
    writeln ('Исходные данные');
    for i := 1 to n do
        begin
            write ('Введите фамилию ');
            readln (a[i]);
            write ('Введите результат ');
            readln (b[i]);
        end;
    for i := 1 to n-1 do
        for j := 1 to n-i do
            if b[j] < b[j+1] then
                begin
                    k := b[j];
                    b[j] := b[j+1];
                    b[j+1] := k;
                    f := a[j];
                    a[j] := a[j+1];
                    a[j+1] := f;
                end;
        writeln ('Отсортированные массивы: ');
        for i := 1 to n do
            begin
                write (a[i], ' ', b[i]);
                writeln

```

```
end  
end.
```

№ 2.

```
program r5_v3_2;  
var a: array[1..92] of integer;  
    i,d,s,indt, ind : integer;  
begin  
    writeln ('Исходные данные');  
    for i := 1 to 92 do  
        begin  
            a[i]:=random(2);  
            write (a[i]);  
        end;  
    d := 0; s := 0; ind:=0;  
    for i := 1 to 92 do  
        if a[i] = 0  
            then begin  
                s := s + 1;  
                indt := i  
            end  
            else begin  
                if s > d then  
                    begin  
                        d := s;  
                        ind := indt-d+1  
                    end;  
                s := 0;  
            end;  
        if s > d then  
            begin  
                d := s;  
                ind := indt-d+1;  
            end;  
        writeln;  
        writeln ('Длина самой большой цепочки  
                подряд идущих 0 равна ', d);  
        writeln ('Начало цепочки: ', ind);  
        writeln ('Конец цепочки: ', ind+d-1)  
    end.
```

Здесь предложена основа решения. Чтобы получить ответы на вопросы, поставленные в задаче, можно дополнить программу, например, следующими строками:

```
if ind<=31 then writeln ('Начало - в июне');  
if (ind>31) and (ind<=61) then  
    writeln ('Начало - в июле');  
if ind>61 then writeln ('Начало - в августе')
```

№ 3. Алгоритм тот же, что и в задании № 9 из варианта 3 предыдущей самостоятельной работы № 4.

№ 4.

```
program r5_v3_4;
```



```

const a: array[1..10] of string = ('Пушкин',
  'Толстой', 'Толстой', 'Тургенев',
  'Достоевский', 'Пушкин', 'Достоевский',
  'Лесков', 'Достоевский', 'Гоголь');
var b: array[1..10] of string;
    c: array[1..100] of integer;
    i, j, k, f: integer;
begin
  k := 1;
  b[k] := a[1];
  for i:=1 to 10 do
    begin
      f := 0;
      for j:=1 to k do
        if a[i]=b[j] then f := 1;
      if f=0 then
        begin
          k:=k+1;
          b[k] := a[i]
        end
      end;
    for i := 1 to k do
      begin
        c[i]:=0;
        for j:=1 to 10 do
          if b[i]=a[j] then c[i] := c[i]+1;
        end;
      for i := 1 to k do
        begin
          write (b[i], '-', c[i]);
          writeln
        end;
      end.

```

№ 5.

```

program r5_v3_5;
const a: array[1..5] of integer =
  (1, 3, 5, 7, 9);
    b: array[1..7] of integer =
  (2, 3, 4, 5, 6, 7, 8);
    c: array[1..10] of integer =
  (0, 1, 2, 4, 6, 8, 9, 10, 11, 12);
var i, j, k, l : integer;
begin
  l:=0;
  for i:=1 to 5 do
    for j:=1 to 7 do
      for k:=1 to 10 do
        if b[j]*b[j]-4*a[i]*c[k]>0
          then l:=l+1;
      write ('L=', l);
    end.
  L = 89.

```

## Уроки 15–16. Структурное программирование. Рекурсивные алгоритмы

### Планируемые образовательные результаты:

- *предметные*: выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; создавать на алгоритмическом языке программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций; использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; выполнять созданные программы;
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью;
- *личностные*: мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

### Решаемые учебные задачи:

- 1) познакомить учащихся с идеологией структурного программирования;
- 2) актуализировать и углубить представления о вспомогательных алгоритмах;
- 3) познакомить учащихся с рекурсивными алгоритмами и способами решения соответствующих задач;
- 4) познакомить учащихся с записью вспомогательных алгоритмов на языке Pascal.

**Основные понятия:** структурное программирование, вспомогательный алгоритм, рекурсия, подпрограммы, процедуры, функции, фактические и формальные параметры, параметры-значения и параметры-переменные.

### Электронное приложение к учебнику:

- презентация «Структурное программирование»;
- онлайн-тест № 9 «Структурное программирование».

### Дополнительные интернет-ресурсы:

- РЭШ (<https://resh.edu.ru>). Урок 4. Вспомогательные алгоритмы.

### Дополнительные печатные материалы:

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

### Особенности изложения содержания темы урока

При рассмотрении содержания темы рекомендуется использовать презентацию, которая знакомит учеников с принципами структурного программирования и понятиями «подпрограмма», «процедура», «функция», «формальные параметры», «фактические параметры», «параметры-значения», «параметры-переменные».

При описании функций пользователя в среде программирования PascalABC.NET можно вместо *<имя\_функции>:=<результат>* использовать *result:=<выражение-результат>*.

В презентации много примеров и задач. Большинство решений не привязаны к конкретной версии языка Паскаль. Исключение составляет задание 2 (слайд 18) из раздела «Вопросы и задания». Использование особенностей языка PascalABC.NET делает решение задачи более «красивым».

*Ответы и решения на вопросы и задания в презентации.*

*Вопрос* (слайд 6). Каким будет результат работы алгоритма из примера 1 при следующих исходных данных:  $X_A = 1$ ,  $Y_A = 1$ ,  $X_B = 1$ ,  $Y_B = 5$ ,  $X_C = 4$ ,  $Y_C = 1$ ?

*Решение:*

$$|AB| = \sqrt{(1-1)^2 + (5-1)^2} = 4.$$

$$|BC| = \sqrt{(1-4)^2 + (5-1)^2} = 5.$$

$$|AC| = \sqrt{(1-4)^2 + (1-1)^2} = 3.$$

$$P = 4 + 5 + 3 = 12.$$

*Ответ:* периметр = 12.

*Вопрос* (слайд 8). Приведите примеры рекурсии, встречающиеся в жизни, природе или литературных произведениях.

*Примерные ответы обучающихся* (следует понимать, что многие примеры являются достаточно спорными, если следовать точному определению рекурсии, но являются допустимыми на первом этапе знакомства с рекурсией): дерево, лист папоротника, цветная капуста, снежинка, круговорот воды в природе, жизнь человечества, матрёшка. Из литературы: стихотворение С. Маршака «Дом, который построил Джек», стихотворение М. Лермонтова «Сон»; стихотворения из фольклора: «У попа

была собака», «Из Нью-Йорка в Лиссабон...», «Вот море, а на море суша...». Песня М. Леонидова «Я оглянулся посмотреть, не оглянулась ли она, чтоб посмотреть, не оглянулся ли я...».

*Задание 2* (слайд 18).

*Примечание для учителя.*

**Важно!** Программа составлена на языке PascalABC.NET. Процедура ввода `read(x)` позволяет вводить последовательность чисел через пробел, что даёт более наглядный эффект.

**На первом уроке** можно рассмотреть вопросы, касающиеся концепции структурного программирования и вспомогательных алгоритмов, в том числе их записи на языке программирования.

**Домашнее задание к первому уроку:** вопросы № 1–6 к § 9.

**На втором уроке** рекомендуется уделить внимание рекурсивным алгоритмам и предложить учащимся выполнить разноуровневую самостоятельную работу № 6.

**Домашнее задание ко второму уроку:** задания № 7–13 к § 9 (по усмотрению учителя).

При наличии времени в конце второго урока можно предложить ученикам выполнить онлайн-тест № 9.

### **Онлайн-тест № 9. Структурное программирование**

1. Алгоритм вычисления значений функций  $F(w)$  и  $Q(w)$ , где  $w$  — натуральное число, задан следующими соотношениями:

$$F(1) = 1; Q(1) = 1;$$

$$F(w) = F(w-1) + 2 \cdot Q(w-1) \text{ при } w > 1;$$

$$Q(w) = Q(w-1) - 2 \cdot F(w-1) \text{ при } w > 1.$$

Чему равно значение функции  $F(5) + Q(5)$ ?

- 62;       -14;       -12;       31.

2. Дан рекурсивный алгоритм:

```
procedure F(n: integer);
begin
  writeln('*');
  if n>0 then begin
    F(n-2);
    F(n div 2);
    F(n div 2);
  end
end;
```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова  $F(5)$ ?

- 12;       23;       34;       45.

3. Дан рекурсивный алгоритм:

```
procedure F(n: integer);
```

```

begin
  writeln(n);
  if n<5 then begin
    F(n+1);
    F(n+2);
    F(n*3)
  end
end;

```

Найдите сумму чисел, которые будут выведены при вызове  $F(2)$ .

Ответ: **79**.

4. Дан рекурсивный алгоритм:

```

function F(n: integer): integer;
begin
  if n>2 then
    F:=F(n-1)+F(n-2)
  else
    F:=n;
end;

```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова  $F(5)$ ?

Ответ: **8**.

5. Ниже записаны две рекурсивные функции  $F$  и  $G$ :

```

function F(n: integer): integer;
begin
  if n>2 then
    F:=F(n-1)+G(n-2)
  else
    F:=n;
end;
function G(n: integer): integer;
begin
  if n>2 then
    G:=G(n-1)+F(n-2)
  else
    G:=n+1;
end;

```

Чему будет равно значение, вычисленное при выполнении вызова  $F(6)$ ?

Ответ: **17**.

## Указания, комментарии, ответы и решения

### Самостоятельная работа № 6

#### Вариант 1.

№ 1. 315.

№ 2. 8.

Команда «прибавь 2» обозначена сверху маленькой стрелкой; команда «умножь на 3» – снизу большой.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	-	2	-	2	-	2	-	4	-	4	-	4	-	6	-	6	-	6	-	8	-	8	-	8

№ 3. 15.

<b><i>n</i></b>	$\leq 0$	1	2	3	4	5	6	7
<b><i>F(n)</i></b>	1	1	1	1	1	1	1	1
		1	1	1	3	5	5	9
		<u>1</u>	<u>3</u>	<u>3</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>
		3	5	5	9	11	11	15

Вариант 2.

№ 1. –11.

№ 2. 26.

№ 3. 42. Строим таблицу и заполняем её справа налево:

<b><i>n</i></b>	1	2	3	4	5	6
<b><i>F(n)</i></b>	1	2	3	4	5	6
	23	5	6	7		
	<u>18</u>	<u>6</u>	<u>9</u>	<u>12</u>		
	42	13	18	23		

Вариант 3.

№ 1. –14.

№ 2. 23900.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	3	4	7	12	19	31	53	84	137	225	362	587	956

16	17	18	19	20	21	22	23	24	25
956	1912	2868	4780	–	4780	4780	9560	14340	23900

№ 3. 34.

№ 4. 79.

№ 5. 17.

*Задания в учебнике*

№ 7.

<b><i>n</i></b>	$\leq 0$	1	2	3	4	5	6	7	8	9	10
<b><i>F(n)</i></b>	2	6	14	26	54	94	174	294	422	770	1286

№ 8. 1) 60; 2) 26; 3) 14.

№ 11. 97.

```

function F(n: integer): integer;
begin
    if n>2 then F := F(n-1) + 3*F(n-2)
    else F := 1
end;
begin

```

```
    write (F(7))  
end.
```

№12.

```
function fact(l: longint): longint;  
begin  
    if l=1 then fact:=1  
    else if l>1 then fact:=l*fact(l-1);  
end;  
var c, n, k : integer;  
begin  
    write ('n='); read (n);  
    write ('k='); read (k);  
    c:=fact(n)div(fact(n-k)*fact(k));  
    writeln ('c=', c)  
end.
```

№ 13. Выводится столбик чисел: 2, 6, 2, 10, 3, 1.

## Урок 17. Обобщение и систематизация изученного материала по теме «Алгоритмы и элементы программирования»

### Планируемые образовательные результаты:

- *предметные*: определять результат выполнения алгоритма при заданных исходных данных; узнавать изученные алгоритмы обработки чисел и числовых последовательностей; создавать на их основе несложные программы анализа данных; читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; создавать на алгоритмическом языке программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций; использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; выполнять созданные программы;
- *метапредметные*: самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута; оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели; выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты; организовывать эффективный поиск ресурсов, необходимых для

достижения поставленной цели; сопоставлять полученный результат деятельности с поставленной заранее целью;

- **личностные:** мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества.

#### **Решаемые учебные задачи:**

- 1) обобщить представления учащихся об алгоритмах, алгоритмических структурах, структурном программировании;
- 2) проверить умение учащихся решать задачи, связанные с использованием базовых алгоритмических структур и записью алгоритмов на языках программирования.

**Основные понятия:** алгоритм, исполнитель алгоритма, основные алгоритмические конструкции, языки программирования, данные, структура данных, идентификаторы, операторы, трассировочные таблицы, массив, размерность массива, вспомогательный алгоритм, рекурсия, подпрограммы, процедуры, функции.

#### **Электронное приложение к учебнику:**

- интерактивный тест 2.

#### **Дополнительные печатные материалы:**

- Информатика. 11 класс. Базовый уровень. Самостоятельные и контрольные работы / Л. Л. Босова, А. Ю. Босова, Н. А. Аквилянов. – М. : БИНОМ. Лаборатория знаний, 2019.

#### **Особенности изложения содержания темы урока**

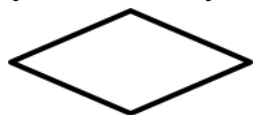
Повторение и обобщение всего изученного материала можно организовать с помощью интерактивного теста 2 ко второй главе учебника. Вопросы теста можно выводить на большой экран и обсуждать их во фронтальном режиме. Работу с тестом можно организовать по группам или индивидуально. Вопросы теста и ответы на них представлены ниже.

#### **Интерактивный тест 2 к главе 2 «Алгоритмы и элементы программирования»**

1. Какой из документов можно считать алгоритмом?
  - правила техники безопасности;
  - ◎ **инструкция по приготовлению пищи;**
  - список книг в школьной библиотеке.
2. Массовость – это свойство алгоритма, заключающееся в том, что:
  - алгоритм предназначен для множества исполнителей;
  - ◎ **алгоритм может использоваться на множестве однотипных задач;**
  - алгоритм состоит из множества конечных команд;
  - в результате работы алгоритма может получаться множество различных результатов.



3. Какую смысловую нагрузку несёт блок?



- блок начала-конца алгоритма;
- блок ввода-вывода;
- блок обработки;
- логический блок.**

4. Предлагается некоторая операция над двумя произвольными трёхзначными десятичными числами:

- 1) записывается результат сложения старших разрядов этих чисел;
- 2) к нему дописывается результат сложения средних разрядов по такому правилу: если он меньше первой суммы, то полученное число приписывается к первому слева, иначе – справа;
- 3) итоговое число получают приписыванием справа к числу, полученному после второго шага, суммы значений младших разрядов исходных чисел.

Какое из перечисленных чисел могло быть построено по этому правилу?

- 141310;
- 102113;
- 101421;
- 101413.**

5. У исполнителя Калькулятор две команды, которым присвоены номера:  
1. прибавь 2;  
2. умножь на 3.

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, утраивает его. Запишите порядок команд в программе получения из 0 числа 28, содержащей не более 6 команд, указывая лишь номера команд. (Например, программа 21211 – это программа:

умножь на 3  
прибавь 2  
умножь на 3  
прибавь 2  
прибавь 2,

которая преобразует число 1 в 19).

Ответ: **121211.**

6. Какое определение можно использовать для разветвляющегося алгоритма?

- алгоритм, который может быть записан с помощью набора геометрических фигур;
- алгоритм, в котором команды выполняются последовательно друг за другом;

- алгоритм, в котором одни и те же действия исполняются многократно;
  - алгоритм, в котором есть хотя бы одно условие.**
7. Какой тип алгоритма используется для вычисления площади треугольника по известным длинам его трёх сторон?
- линейный;**
  - разветвляющийся;
  - циклический;
  - любой.

8. Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперёд  $n$ , где  $n$  – целое число, вызывающая передвижение черепашки на  $n$  шагов в направлении движения;

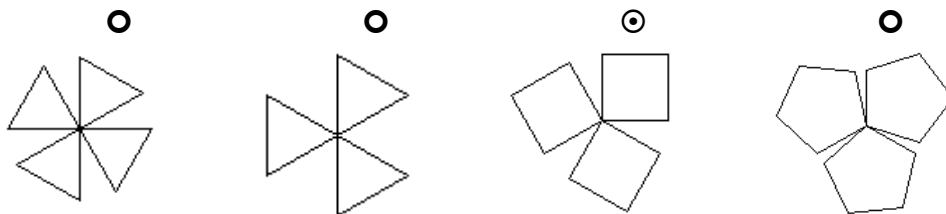
Направо  $m$ , где  $m$  – целое число, вызывающая изменение направления движения на  $m$  градусов по часовой стрелке.

Запись Повтори 5 [Команда1 Команда2] означает, что последовательность команд в скобках повторится 5 раз.

Черепашке был дан для исполнения следующий алгоритм:

Повтори 5 [Повтори 4 [Вперёд 40 Направо 90] Направо 120]

Какая фигура появится на экране?

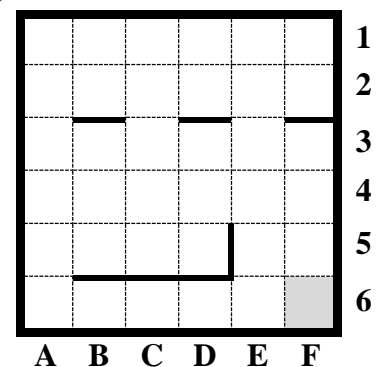


9. Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив следующую программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

```

НАЧАЛО
ПОКА <справа свободно ИЛИ снизу свободно>
  ПОКА <снизу свободно>
    вниз
  КОНЕЦ ПОКА
ПОКА <справа свободно>
  вправо
КОНЕЦ ПОКА
КОНЕЦ ПОКА
КОНЕЦ

```



- 22;
- 17;
- 19;**
- 21.

10. Определите значение целочисленных переменных  $x$ ,  $y$  и  $t$  после выполнения фрагмента программы:

```
x:=5;  
y:=7;  
t:=x;  
x:=y mod x;  
y:=t;
```

- $x = 2, y = 5, t = 5$ ;  
  $x = 7, y = 5, t = 5$ ;  
  $x = 2, y = 2, t = 2$ ;  
  $x = 5, y = 5, t = 5$ .

11. Определите значение переменной  $c$  после выполнения следующего фрагмента программы:

```
a:=6;  
b:=15;  
a:=b-a*2;  
if a>b  
  then c:=a+b  
  else c:=b-a;
```

- 3;             33;             18;             12.

12. Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```
var i, y: integer;  
begin  
  y:=0;  
  for i:=1 to 4 do  
    begin  
      y:=y*10;  
      y:=y+i;  
    end  
  end.
```

Ответ: **1234**.

13. Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```
var y : real; i : integer;  
begin  
  y:=0;  
  i:=5;  
  while i>2 do  
    begin  
      i:=i-1;  
      y:=y+i*i  
    end;
```

end.

Ответ: **29**.

14. Определите значение переменной  $y$ , которое будет получено в результате выполнения следующей программы:

```
var y : real; i : integer;
begin
  y:=0;
  i:=1;
  repeat
    i:=2*i;
    y:=y+i
  until i>5;
end.
```

Ответ: **14**.

15. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведённом ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do
  A[i]:=i+1;
for i:=0 to 10 do
  A[i]:=A[10-i];
```

Чему будут равны элементы этого массива?

- 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0;
  - 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1;
  - 11, 10, 9, 8, 7, 6, 7, 8, 9, 10, 11;**
  - 10, 9, 8, 7, 6, 5, 6, 7, 8, 9, 10.
16. Все элементы двумерного массива  $A$  размером  $5 \times 5$  равны 0. Сколько элементов массива после выполнения фрагмента программы будут равны 1?

```
for n:=1 to 5 do
  for m:=1 to 5 do
    A[n,m] := (m-n) * (m-n) ;
```

- 2;                     5;                     **8;**                     14.

17. Дан фрагмент программы, обрабатывающей линейный массив  $A$  из 6 элементов.

```
for i:=1 to 3 do
  if A[i]>A[i+3] then
  begin
    c:=A[i];
    A[i]:=A[i+3];
    A[i+3]:=c;
  end;
```

Определите, какой из данных массивов станет упорядоченным по возрастанию после обработки алгоритмом.

- 6, 3, 7, 35, 24, 13;
- 13, 6, 35, 3, 24, 7;
- 3, 7, 13, 24, 6, 35;
- 35, 3, 13, 24, 6, 7.

18. Ниже представлен фрагмент программы, в которой описан одномерный целочисленный массив  $A$  и обрабатываются элементы массива с индексами от 1 до 10.

```
n:=10;
for i:=1 to n do begin
  A[n+1-i]:=2*A[i];
end;
```

Перед началом выполнения фрагмента элементы массива имеют значения соответственно 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, т. е.  $A[1] = 1$ ;  $A[2] = 2$  и т. д.

Укажите значение, которое после выполнения указанного фрагмента программы имеют два или более рассмотренных в этом фрагменте элемента массива. Если таких чисел несколько, укажите наибольшее из них.

- такого значения нет;
- 10;
- 8;
- 4.

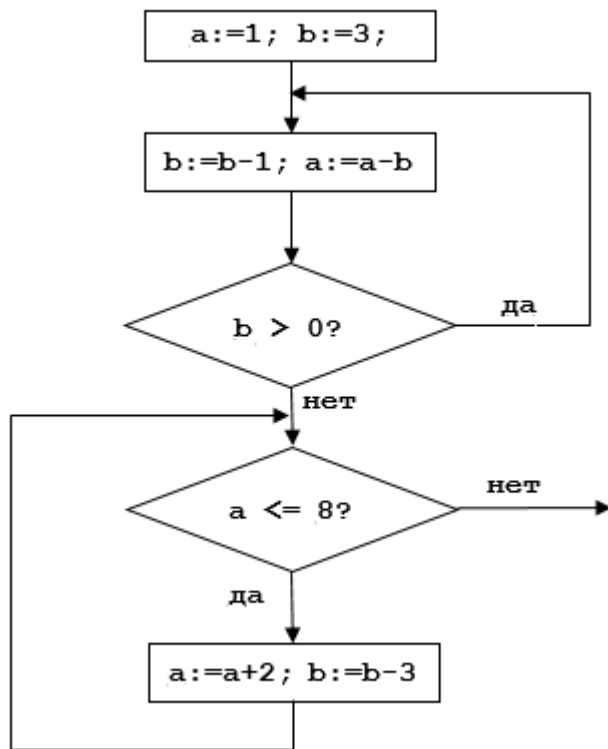
19. В программе описан одномерный целочисленный массив  $A$  с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

```
for i:=0 to 10 do
  A[i]:=i-1;
for i:=1 to 10 do
  A[i-1]:=A[i];
A[10]:=10;
```

Как изменятся элементы этого массива после выполнения фрагмента программы?

- все элементы, кроме последнего, окажутся равны между собой;
- все элементы окажутся равны своим индексам;
- все элементы, кроме последнего, будут сдвинуты на один элемент вправо;
- все элементы, кроме последнего, уменьшатся на единицу.

20. Определите значение переменной  $b$  после выполнения фрагмента алгоритма.



Ответ: **-18**

Кроме того, можно предложить завершить выполнение самостоятельной работы № 5 «Решение задач по обработке одномерных массивов» (см. ответы в конце уроков 12–14).