

**Общество с ограниченной ответственностью «Онлайн-Гимназия Адель»**

(ООО «Онлайн-Гимназия Адель»)

ИНН 5022076651 ОГРН 1235000132344

140410, Московская область, г Коломна, ул. Зеленая, д. 31А

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОХОЖДЕНИЮ ТЕКУЩЕГО  
КОНТРОЛЯ ПО ИНФОРМАТИКЕ  
ПРИ РЕАЛИЗАЦИИ ПРОГРАММ ДОПОЛНИТЕЛЬНОГО  
ОБЩЕОБРАЗОВАТЕЛЬНОГО ОБЩЕРАЗВИВАЮЩЕГО ОБРАЗОВАНИЯ**

"Домашняя Гимназия. Подготовка к аттестации 5-9 класс"  
для дистанционного семейного обучения

Коломна

2025

## **ВВЕДЕНИЕ**

Настоящие методические рекомендации разработаны с целью систематизации работы по формированию и развитию ключевых компетенций в области информатики у обучающихся 7–9 модулей. Документ представляет собой комплексное руководство, объединяющее теоретические основы, практические рекомендации и конкретные примеры организации различных видов учебной деятельности.

### **Цель документа**

Основная цель данного документа заключается в создании единого методического пространства, обеспечивающего эффективное освоение обучающимися различных видов учебной деятельности по информатике.

Рекомендации направлены на достижение следующих задач:

1. Унификация подходов к организации и проведению основных видов учебной деятельности
2. Стандартизация критериев оценивания различных видов работ
3. Повышение качества образовательного процесса через чёткое определение целей и планируемых результатов
4. Создание условий для дифференцированного подхода в обучении
5. Обеспечение преемственности между различными этапами освоения учебного материала

### **Структура документа**

Методические рекомендации состоят из трёх основных разделов, каждый из которых посвящён конкретному виду учебной деятельности:

1. Тестирование — контроль знаний в формализованной форме
2. Практическая работа на компьютере: выполнение заданий в программных средах, создание информационных продуктов, работа с базами данных и электронными таблицами
3. Решение алгоритмических задач и программирование: разработка алгоритмов, написание и отладка программ, анализ кода

### **Каждый раздел содержит:**

1. Рекомендации для преподавателя с указанием целей и планируемых результатов
2. Рекомендации для обучающихся с конкретными алгоритмами действий
3. Примеры заданий с образцами выполнения
4. Критерии оценивания с чёткими параметрами и шкалами

### **Методологические основы**

Рекомендации построены на следующих принципах:

1. Принцип доступности — все материалы адаптированы к возрастным особенностям обучающихся 7–9 класса
2. Принцип системности — виды работ представлены в логической

последовательности от простого к сложному

3. Принцип практической направленности — акцент сделан на формирование конкретных умений и навыков работы с информационными технологиями
4. Принцип вариативности — предусмотрены различные варианты заданий и подходов к их выполнению
5. Принцип объективности — критерии оценивания носят прозрачный и измеримый характер

### **Целевая аудитория**

Данные рекомендации предназначены для:

1. Преподавателей информатики, работающих с обучающимися 7–9 класса
2. Обучающихся, стремящихся систематизировать свои знания и умения в области информационных технологий

### **Ожидаемые результаты реализации**

Внедрение данных рекомендаций в образовательную практику позволит:

#### **Для преподавателей:**

- Повысить эффективность учебного процесса
- Сократить время на подготовку к занятиям
- Обеспечить объективность оценивания
- Создать условия для индивидуального подхода

#### **Для обучающихся:**

- Сформировать чёткое понимание требований к различным видам работ
- Развить навыки самостоятельной работы с информационными системами
- Повысить мотивацию к изучению информатики и программирования
- Научиться объективно оценивать свои достижения

### **Особенности применения**

Рекомендации носят рекомендательный характер и могут быть адаптированы с учётом:

- Индивидуальных особенностей конкретного образовательного учреждения
- Уровня подготовки обучающихся
- Технических возможностей и программного обеспечения
- Регионального компонента в содержании образования

**Составитель:** Педагогическое объединение преподавателей информатики

**Дата составления:** 2025 год

**Уровень:** 7–9 модули

# 1.ТЕСТИРОВАНИЕ

## 1.1. Рекомендации для преподавателя

### Цели проведения тестирования:

6. Проверить усвоение ключевых понятий, терминов и определений по темам курса информатики
7. Оценить умение применять теоретические знания при работе с информационными системами
8. Выявить типичные ошибки в понимании алгоритмов, систем счисления, структур данных для последующей коррекции
9. Развивать навыки самостоятельной работы и логического мышления

### Планируемые результаты:

4. Обучающийся демонстрирует знание основных понятий информатики: алгоритм, система счисления, информация, файловая система, база данных
5. Умеет выбирать правильный ответ из предложенных вариантов, применяя логический анализ
6. Различает сходные понятия: аппаратное и программное обеспечение, оперативная и постоянная память
7. Применяет алгоритмы перевода чисел между системами счисления

### Методические рекомендации:

5. Разрабатывайте тесты с учётом возрастных особенностей и уровня цифровой грамотности обучающихся
6. Включайте разнообразные типы заданий:
  - Выбор одного правильного ответа
  - Выбор нескольких правильных ответов
  - Установление соответствия (например, устройство — функция)
  - Восстановление последовательности шагов алгоритма
7. Составляйте задания от простого к сложному
8. Используйте понятные формулировки, при необходимости сопровождайте задания схемами или таблицами
9. Предусматривайте оптимальное время выполнения (15–20 минут)
10. Организуйте предварительное повторение материала на предыдущем занятии

### Пример составления теста:

- 60% заданий базового уровня (определения, основные понятия)
- 30% заданий повышенной сложности (применение знаний, анализ)
- 10% заданий творческого характера (нестандартные ситуации, перевод чисел)

## 1.2. Рекомендации для обучающихся

### Как подготовиться к тесту:

6. Повторите основные понятия и термины темы по конспектам и учебнику
7. Разберите примеры перевода чисел в разные системы счисления
8. Обратите внимание на различия между похожими понятиями (например, бит и байт, RAM и ROM)
9. Составьте собственный словарь терминов с краткими определениями

### Как выполнять тест:

3. Внимательно прочитайте инструкцию к тесту
4. Просмотрите все задания перед началом работы
5. Начинайте с лёгких заданий, хорошо знакомых вам
6. Сложные задания оставьте на потом
7. Не задерживайтесь долго на одном задании
8. Если сомневаетесь, исключайте заведомо неправильные ответы
9. Обязательно проверьте работу перед сдачей

### Полезные советы:

- Действуйте по алгоритму: прочитал → подумал → выбрал → проверил
- Не меняйте ответы без веской причины — первый интуитивный выбор часто верен
- Следите за временем
- При работе на компьютере проверяйте, что ответ действительно выбран/отмечен

## 1.3. Пример задания и образец выполнения

### Тема: «Системы счисления и кодирование информации»

Тестовые задания:

**1. Какое минимальное количество бит необходимо для кодирования одного из 32 возможных символов?**

- а) 4
- б) 5
- в) 6
- г) 8

**2. Число  $1011_2$  в десятичной системе счисления равно...**

- а) 9
- б) 10
- в) 11
- г) 12

### 3. Алгоритм — это...

- а) язык программирования
- б) конечная последовательность точных и понятных инструкций для исполнителя
- в) программа на компьютере
- г) схема работы процессора

### 4. Какое устройство относится к устройствам вывода информации?

- а) Сканер
- б) Клавиатура
- в) Принтер
- г) Микрофон

### Образец выполнения:

10.б) 5

11.в) 11

12.б) конечная последовательность точных и понятных инструкций для исполнителя

13.в) Принтер

### 1.4. Критерии оценивания

Оценка	Количество правильных ответов	Характеристика выполнения
5	90–100%	Все задания выполнены правильно или допущена 1 негрубая ошибка
4	75–89%	Допущено 2–3 ошибки, но основные понятия усвоены
3	60–74%	Выполнено более половины заданий, но есть пробелы в знаниях
2	менее 60%	Большая часть заданий выполнена неправильно

### Примечания:

8. Орфографические ошибки в написании терминов не учитываются при автоматической проверке

9. За неверное оформление ответа (например, отмечено несколько вариантов вместо одного) снимается 0,5 балла
10. За исправления оценка не снижается, если они аккуратные и однозначные

## 2. ПРАКТИЧЕСКАЯ РАБОТА НА КОМПЬЮТЕРЕ

### 2.1. Рекомендации для преподавателя

Практическая работа на компьютере — это задание, выполняемое непосредственно в программной среде (текстовый редактор, электронные таблицы, графический редактор, среда программирования и др.), направленное на закрепление теоретических знаний и формирование практических умений работы с информационными технологиями.

#### Цели проведения практической работы:

- Развитие практических навыков работы с программным обеспечением
- Формирование умений создавать, редактировать и форматировать информационные продукты
- Воспитание аккуратности, внимательности и культуры оформления электронных документов
- Приобщение к профессиональным стандартам работы с информацией

#### Планируемые результаты:

- Умеет работать в основных программных средах (MS Word, Excel, Paint, среды программирования)
- Грамотно оформляет результаты работы: создаёт таблицы, диаграммы, применяет форматирование
- Следует структурированному плану выполнения работы: от постановки цели до проверки результата
- Сохраняет файлы в нужном формате и указанном месте

#### Методические рекомендации:

11. Перед каждой практической работой проводится краткий инструктаж о порядке выполнения и технике безопасности
12. Проверьте работоспособность компьютеров и наличие необходимого программного обеспечения до начала урока
13. Разрабатывайте систему заданий от простого к сложному:
  - Репродуктивные задания — воспроизведение по образцу
  - Реконструктивные задания — изменение по условию
  - Творческие задания — создание собственного продукта
14. Используйте пошаговые инструкции и скриншоты-памятки для самостоятельной работы
15. Организуйте работу как индивидуально, так и в парах при необходимости
16. Учите обучающихся сохранять работу по ходу выполнения (Ctrl+S)

## **2.2. Рекомендации для обучающихся**

### **Правила техники безопасности при работе за компьютером**

1. Перед началом работы убедитесь, что рабочее место в порядке, шнуры не свисают, клавиатура и мышь подключены.
2. Не включайте и не выключайте оборудование без разрешения преподавателя.
3. Не трогайте разъёмы и системный блок руками во время работы.
4. При возникновении неисправности (запах горелого, непривычный звук, мигание экрана) немедленно сообщите преподавателю.
5. Не пейте и не ешьте за рабочим местом.
6. Соблюдайте режим работы: делайте паузы для отдыха глаз каждые 20–25 минут.
7. После окончания работы закройте программы и приведите рабочее место в порядок.

### **Подготовка к практической работе**

Перед выполнением практической работы необходимо:

- ознакомиться с заданием и целью работы
- вспомнить или записать основные команды и приёмы работы в нужной программе
- создать папку для сохранения результатов (имя папки уточните у преподавателя)
- уточнить формат сохранения файла

### **Порядок выполнения работы**

Приступая к выполнению практической работы, следует придерживаться следующей последовательности действий:

10. Внимательно прочитайте задание целиком, прежде чем приступать к выполнению
11. Откройте нужную программу и создайте новый файл (если не указано иное)
12. Выполняйте задание по пунктам, не пропуская ни одного шага
13. Регулярно сохраняйте работу (Ctrl+S)
14. По завершении проверьте результат: соответствует ли он всем требованиям задания
15. Сохраните файл в указанную папку с нужным именем и форматом
16. Сообщите преподавателю о завершении работы

### **Оформление результатов**

Готовая практическая работа должна содержать:

- соответствующее имя файла (указывается преподавателем)
- все обязательные элементы задания (таблицы, диаграммы, рисунки и т.д.)
- аккуратное форматирование без лишних пробелов и переносов
- корректные данные, совпадающие с условием задачи

### 2.3. Пример задания и образец выполнения

**Тема: «Электронные таблицы. Создание таблицы расчёта среднего балла»**

**Цель работы:** Научиться создавать таблицу в MS Excel, вводить формулы для вычисления суммы и среднего значения, строить диаграмму.

**Задание:**

- 10.Создайте таблицу с оценками 5 учеников по 4 предметам (Математика, Русский язык, Информатика, История)
- 11.В столбце «Средний балл» введите формулу для вычисления среднего значения оценок каждого ученика (функция СРЗНАЧ)
- 12.В строке «Средний по классу» вычислите средний балл по каждому предмету
- 13.Постройте столбчатую диаграмму по данным таблицы
- 14.Сохраните файл в формате .xlsx с именем «Фамилия\_Практика2»

**Образец выполнения:**

Ученик	Математика	Рус. язык	Информатика	История	Средний балл
Иванов А.	5	4	5	4	=СРЗНАЧ(B2:E2)
Петрова М.	4	5	5	5	=СРЗНАЧ(B3:E3)
Сидоров К.	3	4	4	3	=СРЗНАЧ(B4:E4)
Козлова Е.	5	5	4	4	=СРЗНАЧ(B5:E5)
Новиков Д.	4	3	5	4	=СРЗНАЧ(B6:E6)
Ср. по классу	=СРЗНАЧ(B2:B6)	=СРЗНАЧ(C2:C6)	=СРЗНАЧ(D2:D6)	=СРЗНАЧ(E2:E6)	

### 2.4. Критерии оценивания

Оценка	Характеристика выполнения	Характеристика выполнения
5	Отлично	Все задания выполнены правильно, файл сохранён верно, форматирование аккуратное, формулы работают корректно
4	Хорошо	Допущены незначительные ошибки в оформлении

Оценка	Характеристика выполнения	Характеристика выполнения
		или 1–2 ошибки в формулах, не влияющие на общий результат
3	Удовл.	Выполнено более половины заданий, допущены ошибки в формулах или отсутствует диаграмма
2	Неудовл.	Большая часть заданий не выполнена, файл не сохранён в нужном формате, формулы отсутствуют

Защита практической работы может проводиться по контрольным вопросам, в ходе которых обучающийся объясняет применённые функции и форматирование.

## **3.РЕШЕНИЕ АЛГОРИТМИЧЕСКИХ ЗАДАЧ И ПРОГРАММИРОВАНИЕ**

### **3.1. Рекомендации для преподавателя**

Умение решать алгоритмические задачи и писать программы является ключевым навыком в курсе информатики. Алгоритмическая задача — это задача, требующая разработки пошаговой инструкции (алгоритма) для решения конкретной проблемы с использованием логических рассуждений и математических операций.

#### **Цели решения алгоритмических задач и программирования:**

- Закрепление теоретических знаний об алгоритмах и структурах данных
- Формирование навыков анализа задачи и разработки решения
- Развитие логического и структурного мышления
- Формирование умений записывать алгоритмы на языке программирования (Python, Pascal и др.)
- Обучение отладке программ и поиску ошибок

#### **Планируемые результаты:**

- Глубже понять алгоритмические конструкции: ветвление, циклы, подпрограммы
- Применять теоретические знания для решения реальных вычислительных задач
- Развить логическое мышление, навык пошагового анализа и структурирования решения
- Выработать навыки составления блок-схемы, записи алгоритма на псевдокоде и его реализации на языке программирования
- Научиться тестировать программы на различных входных данных

#### **Методические рекомендации:**

Методические рекомендации по решению алгоритмических задач включают структурированный подход:

- Анализ условия задачи и выделение входных и выходных данных
- Составление алгоритма на естественном языке или в виде блок-схемы
- Выбор алгоритмических конструкций (линейный, ветвление, цикл)
- Запись программы на языке программирования
- Тестирование и отладка

#### **Ключевые этапы работы с задачей:**

14. Анализ условия — что дано, что нужно найти
15. Разработка алгоритма — пошаговое описание решения
16. Кодирование — запись алгоритма на языке программирования
17. Тестирование — проверка на нескольких наборах данных

## 3.2. Рекомендации для обучающихся

В основу каждой алгоритмической задачи положена некоторая вычислительная или логическая проблема, которая должна быть решена точно и однозначно. Без твёрдого знания базовых конструкций алгоритмов нельзя рассчитывать на успешное решение даже сравнительно простых задач.

При этом правильный ответ при неверном оформлении алгоритма или кода может снизить оценку. Чтобы добиться наилучшего результата, придерживайтесь следующего алгоритма.

### 1. Внимательно прочитайте условие

Большая часть ошибок возникает на этапе чтения условия. Определите: что является входными данными, а что — результатом (выходными данными). Запишите это явно.

### 2. Определите входные и выходные данные

Запишите условие задачи в структурированном виде:

- Дано: перечислите все входные данные и их типы
- Найти: укажите, что нужно вычислить или вывести
- Ограничения: если есть — укажите диапазон значений

### 3. Составьте алгоритм

Перед написанием кода набросайте алгоритм в виде псевдокода или блок-схемы. Псевдокод — это запись алгоритма на понятном языке с использованием ключевых слов (начало, конец, если, то, иначе, пока, для).

### 4. Запишите программу

Реализуйте составленный алгоритм на языке программирования. При записи кода соблюдайте:

- правильные отступы (особенно в Python)
- осмысленные имена переменных
- комментарии к ключевым блокам кода
- корректный ввод и вывод данных

### 5. Протестируйте программу

Проверьте программу минимум на трёх наборах входных данных:

- типичный случай (стандартные значения)
- граничный случай (минимальное или максимальное значение)
- нестандартный случай (нулевые значения, отрицательные числа и т.д.)

### 6. Проверьте оформление

Убедитесь, что:

- программа запускается без ошибок
- вывод совпадает с ожидаемым результатом

- код аккуратно оформлен и снабжён комментариями

### 3.3. Пример задания и образец выполнения

#### Условие задачи:

Напишите программу на языке Python, которая вводит с клавиатуры два натуральных числа и выводит их наибольший общий делитель (НОД) с использованием алгоритма Евклида.

#### Образец выполнения:

#### Шаг 1. Входные и выходные данные:

- Дано: два натуральных числа  $a$  и  $b$
- Найти:  $\text{НОД}(a, b)$

#### Шаг 2. Алгоритм (псевдокод):

НАЧАЛО

Ввести  $a, b$

ПОКА  $b \neq 0$ :

$a, b := b, a \bmod b$

Вывести  $a$

КОНЕЦ

#### Шаг 3. Программа на Python:

```
# Нахождение НОД алгоритмом Евклида
```

```
a = int(input('Введите первое число: '))
```

```
b = int(input('Введите второе число: '))
```

```
while b != 0:
```

```
    a, b = b, a % b
```

```
print('НОД =', a)
```

#### Шаг 4. Тестирование:

Тест	Входные данные	Ожидаемый результат	Результат программы
Типичный	$a=12, b=8$	НОД = 4	НОД = 4
Граничный	$a=7, b=1$	НОД = 1	НОД = 1
Нестандартный	$a=100, b=0$	НОД = 100	НОД = 100

### 3.4. Критерии оценивания

**Итоговая оценка выводится следующим образом:**

**Отлично (5):**

- Входные и выходные данные определены верно
- Алгоритм или блок-схема составлены корректно
- Программа написана без синтаксических и логических ошибок
- Все тестовые случаи пройдены успешно

**Хорошо (4):**

- Дано/найти записаны не полностью, либо
- Алгоритм составлен с незначительной неточностью, либо
- В программе допущена 1 ошибка, исправленная после подсказки преподавателя, либо
- Тестирование проведено не полностью

**Удовлетворительно (3):**

Оценка «удовлетворительно» ставится, когда ошибки присутствуют в двух этапах решения задачи (например, некорректный алгоритм и ошибка в программе), либо программа работает только для части тестовых случаев.

**Неудовлетворительно (2):**

Оценка «неудовлетворительно» ставится, когда программа не запускается или выдаёт неверный результат для большинства тестов, задача фактически не решена.

## ЗАКЛЮЧЕНИЕ

### Общие рекомендации для преподавателя:

---

1. Дифференцированный подход: учитывайте индивидуальные особенности и уровень цифровой грамотности каждого обучающегося. Предлагайте задания разного уровня сложности.
2. Систематичность: регулярно проводите разные виды работ для формирования устойчивых навыков работы с информационными технологиями.
3. Поэтапность: от простых видов работ переходите к более сложным, обеспечивая преемственность от освоения интерфейса программы до самостоятельного программирования.
4. Мотивация: создавайте ситуацию успеха для каждого обучающегося, отмечайте даже небольшие достижения, связывайте задания с реальными жизненными ситуациями.
5. Обратная связь: давайте конкретные рекомендации по улучшению работы, разбирайте типичные ошибки на занятии.
6. Интеграция: связывайте разные виды работ между собой, используйте результаты практических работ как основу для создания проектов.

### Общие рекомендации для обучающихся:

---

1. Регулярность: выполняйте все задания систематически, не откладывайте на последний момент.
2. Внимательность: внимательно читайте задание, понимайте, что именно требуется сделать.
3. Планирование: перед написанием программы составляйте алгоритм — это экономит время.
4. Сохранение: регулярно сохраняйте работу в процессе выполнения.
5. Проверка: всегда тестируйте программы и проверяйте работу перед сдачей.
6. Аккуратность: соблюдайте отступы в коде, давайте переменным понятные имена.
7. Самостоятельность: старайтесь выполнять работу самостоятельно — это поможет лучше усвоить материал.
8. Любознательность: интересуйтесь информатикой, замечайте, как алгоритмы и программы используются в повседневной жизни.