

## **МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

по выполнению внеаудиторной работы при реализации программы дополнительного  
общеобразовательного образования

**«Подготовка к аттестации за 5 класс для дистанционного семейного  
обучения»**

## **ВВЕДЕНИЕ**

Данные методические рекомендации разработаны для обеспечения эффективного освоения обучающимися информатики в 5 классе. Документ направлен на формирование прочных теоретических знаний, развитие практических умений и навыков самостоятельной учебной деятельности в области цифровой грамотности, работы с информацией, программирования и кибербезопасности.

Рекомендации содержат алгоритмы выполнения различных видов учебных заданий, примеры их решения и критерии оценивания результатов работы. Материал представлен в доступной и систематизированной форме с учетом возрастных особенностей обучающихся.

### **Цель документа**

Цель настоящих рекомендаций — помочь обучающимся успешно освоить основные разделы программы информатики 5 класса, развить навыки самостоятельной работы и подготовиться к выполнению различных видов учебных заданий.

В рамках реализации данной цели предполагается решение следующих задач:

1. Формирование представления о требованиях к различным видам учебной деятельности по информатике.
2. Развитие навыков выполнения тестовых заданий, практических работ, проектной деятельности и решения алгоритмических задач.
3. Формирование умений самостоятельно анализировать результаты своей работы.
4. Повышение качества усвоения теоретического материала.
5. Развитие ответственности, аккуратности и познавательного интереса к изучению информатики.

### **Структура документа**

Рекомендации разделены на ключевые блоки, соответствующие основным видам учебной деятельности в 5 классе:

1. Контроль знаний (тестирование).
2. Практическая деятельность: выполнение практических работ и проектов.

### 3. Решение алгоритмических задач.

#### **Каждый раздел содержит:**

1. Рекомендации для обучающихся с конкретными алгоритмами действий.
2. Примеры заданий с образцами выполнения.
3. Критерии оценивания выполненных заданий.

#### **Методологические основы**

Рекомендации базируются на следующих принципах:

1. Научная обоснованность — использование современных представлений о информационных процессах и технологиях.
2. Доступность — изложение материала в понятной и логичной форме с учетом возрастных особенностей.
3. Последовательность — постепенное усложнение содержания от простых понятий к более сложным темам.
4. Практическая направленность — акцент на формирование умений применять знания в реальных и учебных ситуациях (работа с компьютером, интернетом, программным обеспечением).
5. Самостоятельность — развитие навыков планирования и организации собственной учебной деятельности.
6. Объективность контроля — четко сформулированные показатели оценивания.

#### **Целевая аудитория**

Методические рекомендации предназначены для обучающихся 5 класса и направлены на поддержку их учебной деятельности по информатике.

#### **Ожидаемые результаты реализации**

Использование данных рекомендаций позволит:

1. Лучше понимать изучаемый материал по информатике;
2. Успешно выполнять тестовые задания;

3. Грамотно оформлять практические работы и проекты;
4. Совершенствовать навыки решения алгоритмических задач и работы с программным обеспечением;
5. Развивать самостоятельность в обучении;
6. Формировать ответственное отношение к учебной деятельности;
7. Повышать уровень подготовки к текущему и итоговому контролю.

### **Особенности применения**

Рекомендации могут применяться:

1. При подготовке к практическим работам и проектам;
2. Во время выполнения домашних заданий;
3. При подготовке к диагностическим работам;
4. Для самостоятельного повторения и закрепления изученного материала.

### **Структура работы с документом**

Для наиболее эффективного использования рекомендаций рекомендуется придерживаться следующего порядка работы:

1. Ознакомиться с теоретическим материалом по изучаемой теме.
2. Изучить алгоритм выполнения соответствующего вида задания.
3. Рассмотреть предложенные примеры.
4. Выполнить задания самостоятельно.
5. Проверить результаты с использованием критериев оценивания.
6. Проанализировать допущенные ошибки и определить пути их устранения.

Методические рекомендации являются практическим помощником в изучении информатики. Их использование способствует более осознанному освоению учебного материала, развитию учебной самостоятельности и формированию устойчивого интереса к предмету.

# 1. ТЕСТИРОВАНИЕ

## Цели проведения тестирования

Тестирование используется как оценка способности ориентироваться в цифровой среде и направлено на:

- проверку понимания устройства ПК и принципов работы ОС;
- оценку знаний правил безопасного поведения в Интернете;
- контроль владения терминологией (биты, байты, расширения файлов, облачные сервисы);
- проверку умения анализировать информацию на достоверность;
- формирование у обучающихся навыков самоконтроля и ответственности за результат.

## Планируемые результаты

Обучающийся должен уметь:

- идентифицировать устройства ввода, вывода и хранения информации;
- определять типы файлов по их расширениям (.docx, .png, .sb3 и др.);
- классифицировать виды информации и её свойства;
- выбирать безопасные стратегии поведения при столкновении с киберугрозами;
- использовать инструменты работы с документами (презентации, текстовые документы, таблицы)
- использовать алгоритмы для написания инструкций компьютеру

## Методические рекомендации

Тесты по информатике часто содержат скриншоты интерфейсов или фрагменты кода. При выполнении следует:

- анализировать визуальные данные, если в вопросе есть изображение окна программы, внимательно изучите активные кнопки и вкладки;
- стремиться «убирать лишнее» для быстрого поиска ответа;

- стремиться «оптимизировать» ответ, облегчить содержание и конкретизировать ответ, особенно важно в заданиях с написанием кода;
- обращать особенное внимание к заданиям с написанием кода, тщательно анализировать задание перед выбором ответа или написанием кода.

Все задания основаны исключительно на изученном учебном материале, поэтому перед выполнением тестовой работы рекомендуется повторить основные понятия, определения, формулы и правила по соответствующей теме.

### **Временной регламент**

Стандартный тест из 10-15 вопросов рассчитан на 25–30 минут. Практические задания внутри теста могут занимать больше времени.

### **Пример структуры теста**

Общее количество заданий — 10, разделенных на четыре тематических блока. Ниже представлен детальный разбор каждого блока с примерами вопросов.

#### **Блок 1. Компьютерная грамотность и ОС (3 задания):**

- Задания на классификацию: определите, к какому типу относится устройство (например, «Выберите из списка только устройства ввода информации»).
- Задания на файловую систему: Работа с путями к файлам и расширениями (например, «В какой папке окажется файл, если его переместить из корня диска C: в папку Документы?»).
- Интерфейс ОС: Знание элементов Рабочего стола и Панели задач (например, «Какая кнопка позволяет свернуть все окна сразу?»).

#### **Блок 2. Интернет, облако и кибербезопасность (3 задания):**

- Поиск информации: Умение формулировать запросы и отличать факты от фейков (например, «Какой символ в поисковом запросе позволяет найти точную фразу?»).
- Облачные сервисы: Принципы работы Яндекс Документов и совместного доступа (например, «Что произойдет, если вы дадите другу доступ "Только просмотр"?»).

– Кибергигиена: Распознавание фишинга и вирусов (например, «Вам пришло письмо от "банка" с просьбой срочно ввести пароль. Ваши действия?»).

### **Блок 3. Алгоритмика (3 задания):**

– Линейные алгоритмы: Определение результата последовательных действий (например, «Где окажется котик после выполнения команд: Идти 50, Повернуть на 90, Идти 50?»).

– Циклы и условия: Понимание того, сколько раз выполнится действие (например, «Сколько раз промяукает спрайт, если блок "Мяукнуть" стоит внутри цикла "Повторить 10 раз"?»).

– Логические ошибки: Поиск «бага» в предложенном фрагменте кода (например, «Почему спрайт не двигается, хотя блок "Идти" есть в скрипте?» — ответ может быть связан с отсутствием блока «Когда флажок нажат»).

### **Блок 4. Программирование (1 задание):**

– Программирование и рефакторинг: Планирование и понимание построения структуры кода (например, «Напишите программу для движения кота с клавиатуры компьютера, улучшите движения добавив кнопки стрелок вверх и вниз»).

| <i><b>Уровень</b></i> | <i><b>Описание</b></i>                 | <i><b>Пример задания</b></i>                                                                                            |
|-----------------------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Базовый (60%)         | Знание терминов и простых действий     | Назовите устройство для вывода звука.                                                                                   |
| Повышенный (30%)      | Применение знаний в знакомой ситуации. | Установите соответствие между расширением файла (.jpg, .docx, .mp3) и типом информации.                                 |
| Творческий (10%)      | Анализ, синтез и решение проблем.      | Проанализируйте фрагмент кода в Scratch и определите, почему персонаж проходит сквозь стену, а не отталкивается от неё. |

## **2. ТЕХНОЛОГИЧЕСКИЙ ПРАКТИКУМ: РАБОТА В ОБЛАЧНЫХ СЕРВИСАХ И ГРАФИКЕ**

### **Цели практических работ**

Практическая деятельность — это основа блока информатики. Она направлена на:

- Освоение инструментов Яндекс Документов, Таблиц и Презентаций;
- Развитие навыков визуального дизайна в Adele Design;
- Формирование культуры совместной работы над документами в режиме реального времени.

### **Цели практических работ**

- Текст: создание рефератов с оглавлением, списками и вставленными объектами;
- Таблицы: автоматизация расчетов с помощью функций СУММ, СРЗНАЧ и визуализация через диаграммы;
- Презентации: создание интерактивных слайдов с анимацией и мультимедиа;
- Графика: создание рекламных баннеров и иконок с использованием слоев и векторов.

### **Рекомендации по выполнению работ**

- Прежде чем открывать редактор, набросайте структуру (какие будут заголовки, сколько слайдов, какие данные нужны для таблицы) например в блокноте или даже в физической тетради ручкой;
- Используйте стили заголовков вместо ручного изменения шрифта;
- В таблицах всегда используйте ссылки на ячейки в формулах, а не вводите числа вручную;
- В презентациях соблюдайте правило «белого пространства», не перегружайте слайд текстом.

***Критерии оценивания практических работ***

|                       |                                                                      |
|-----------------------|----------------------------------------------------------------------|
| Техническая точность  | Работают ли формулы? Правильно ли настроена анимация?                |
| Эстетика и дизайн     | Соблюден ли баланс цветов? Читаем ли шрифт?                          |
| Структурированность   | Есть ли логика в расположении элементов?                             |
| Соблюдение регламента | Сохранен ли файл в нужном формате (например, экспорт в PDF или PNG)? |

### 3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ (SCRATCH)

Программирование в 5 классе — это первый шаг к пониманию того, как работают сложные системы. В среде Scratch мы не просто «рисует мультики», а учимся управлять объектами с помощью логических конструкций.

#### Цели и задачи раздела:

- понимание того, что компьютер выполняет команды строго по порядку;
- освоение циклов (повторений), условий (выбора) и переменных (памяти программы);
- развитие навыка поиска и исправления логических ошибок в коде;
- проектное мышление: умение пройти путь от идеи игры до её полной реализации и тестирования.

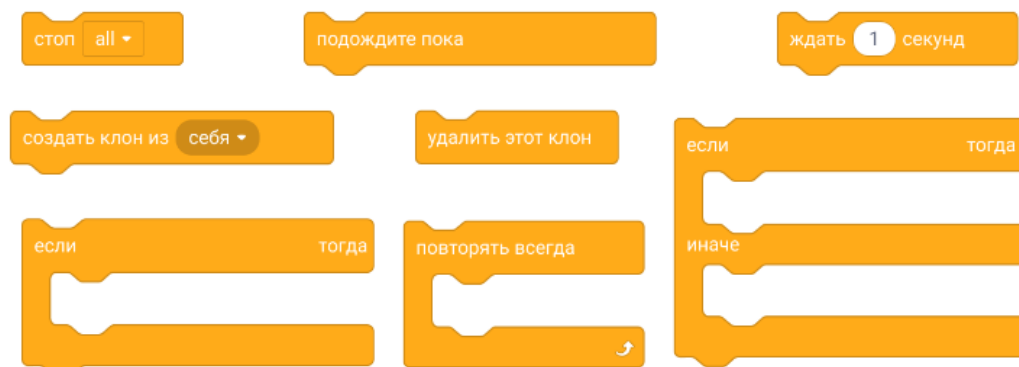
#### Основные инструменты Scratch (Чек-лист для обучающегося):

Для успешного создания проекта вы должны ориентироваться в следующих категориях блоков:

**События** (Желтые блоки): это «пусковые механизмы». Без них программа не начнет работу:



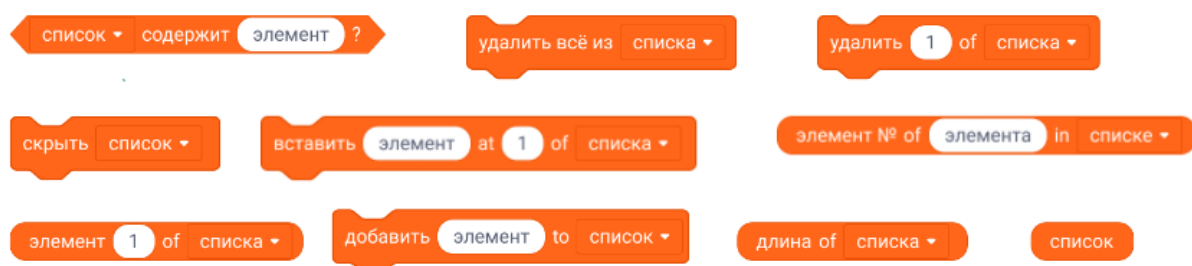
**Управление** (Оранжевые блоки): Цикл «Всегда»: используется для постоянных действий (например, проверка нажатия клавиш) Условие «Если... то...»: позволяет персонажу реагировать на события (например, касание края сцены или другого спрайта):



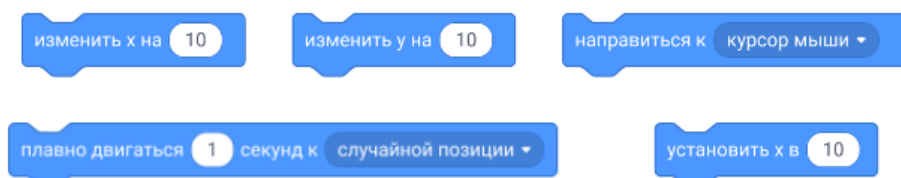
**Сенсоры** (Голубые блоки): это «органы чувств» вашего объекта. С их помощью он понимает, касается ли он цвета, другого объекта или нажата ли клавиша на клавиатуре:



**Переменные** (Темно-оранжевые блоки): это ячейки памяти. В них мы храним очки, жизни или время:



**Движение** (Темно-синие блоки): блоки движения и изменения координат объекта на сцене, именно эти блоки придают «жизни» происходящему на сцене:



## Методические рекомендации

Чтобы ваш проект (например, игра «Лабиринт» или «Охота за призом») работал корректно, следуйте алгоритму:

### 1. Подготовка сцены и героев.

Выберите фон и спрайты. Помните, что у каждого спрайта может быть несколько костюмов для создания анимации движения.

### 2. Метод «Маленьких шагов»

Не пытайтесь собрать весь код сразу.

- сначала настройте движение (клавиши стрелок). Проверьте;
- затем добавьте препятствия и условия касания. Проверьте;
- в последнюю очередь добавляйте звуки и подсчет очков.

### 3. Дебаггинг кода

Если программа работает не так, как вы задумали:

- «разберите» скрипт: отсоедините блоки и проверьте работу каждого по отдельности.;
- проверьте координаты: часто ошибка кроется в том, что спрайт пытается идти «в стену» или за пределы экрана;
- используйте «Говорить...»: заставьте спрайта говорить значения переменных, чтобы понять, что происходит внутри кода в данный момент.

### 4. Оптимизация и чистота кода

- Если у вас 10 одинаковых блоков «Идти», замените их одним циклом «Повторить 10 раз»;
- Удаляйте неиспользуемые блоки с рабочего поля, они «засоряют» проект и могут вызвать случайные срабатывания;

### *Критерии оценивания практических работ*

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| Работоспособность: | программа запускается по флажку и выполняет свою задачу без вылетов |
| Сложность          | использованы циклы, условия и хотя бы одна переменная               |
| Дизайн             | спрайты и фоны подобраны по смыслу, анимация плавная                |

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| Интерактивность | пользователь может управлять процессом (клавиатура/мышь)       |
| Оригинальность  | в проект внесены изменения, отличающие его от базового шаблона |

## ЗАКЛЮЧЕНИЕ

**1. Принцип «Сохраняйся часто»:** при работе в локальных программах приучите себя нажимать Ctrl+S каждые 5 минут. В облачных сервисах следите за статусом «Все изменения сохранены».

**2. Цифровая гигиена и безопасность:** никогда не используйте один пароль для всех сервисов. Относитесь критически к любым предложениям «скачать бесплатно» или «вы выиграли приз» в интернете.

**3. Исследовательский подход:** в Scratch или при вводе формул в таблицах один неверный блок или пропущенная скобка могут сломать всю работу. Будьте предельно внимательны к синтаксису.

**4. Сначала логика, потом дизайн:** сначала добейтесь того, чтобы программа или таблица работала правильно, и только потом приступайте к украшению, подбору шрифтов и картинок.

**5. Организация файлового хаоса:** заведите четкую систему папок по разделам (например: «01\_Устройства», «02\_Текст», «03\_Scratch»). Не называйте файлы «111», «фыв» или «презентация\_копия\_финал».

**6. Здоровье прежде всего:** Правильная осанка за компьютером — залог отсутствия усталости.

**7. Умение искать ответы:** если программа выдает ошибку или вы не знаете, как сделать действие, используйте поиск в интернете или встроенную справку. Умение находить решение — это главный навык айтишника.

**8. Творчество и инициатива:** выполняйте обязательный минимум, но всегда старайтесь добавить в проект что-то свое — уникальный спрайт, необычный переход или полезную функцию.

**9. Этика общения:** помните о правилах сетевого этикета (нетикета) при переписке в почте или работе в общих документах. Уважайте чужой труд и личное пространство.