

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

по выполнению внеаудиторной работы при реализации программы дополнительного
общеобразовательного образования

**«Подготовка к аттестации за 6 класс для дистанционного семейного
обучения»**

ВВЕДЕНИЕ

Настоящие методические рекомендации подготовлены для поддержки обучающихся 6-х классов в процессе освоения информатики. В 6 классе информатика углубляется в теоретические основы, знакомит с системами счисления, логическими операциями, расширяет навыки программирования и работы с мультимедиа. Данный документ призван не только успешно пройти аттестацию, но и научиться осознанно использовать компьютерные технологии для решения повседневных и учебных задач, а также развить критическое мышление в отношении цифрового контента.

Материал структурирован таким образом, чтобы обучающийся мог самостоятельно пройти путь от понимания двоичного кода до создания собственных минимальных программных продуктов, работы с графикой и мультимедиа, а также освоения основ кибербезопасности.

Цель документа

Основная цель рекомендаций систематизировать процесс самостоятельного изучения информатики, предоставить четкие алгоритмы работы с программным обеспечением и сформировать культуру безопасного и этичного поведения в цифровой среде, включая критическую оценку контента, созданного ИИ.

В рамках реализации данной цели предполагается решение следующих задач:

1. Разъяснить требования к выполнению различных типов цифровых заданий, включая работу с теоретическими основами информатики.
2. Сформировать навыки работы с современными облачными сервисами (Яндекс Документы, Таблицы, Презентации) на более продвинутом уровне.
3. Обучить базовым принципам алгоритмизации и программирования в визуальной среде Scratch, а также основам текстового программирования.
4. Развить критическое мышление при работе с информацией в сети Интернет, включая распознавание контента, созданного ИИ.
5. Обеспечить понимание основ кибербезопасности, защиты персональных данных и сетевого этикета.

6. Повысить уровень самостоятельности при выполнении практических работ, включая работу с мультимедиа и 3D-моделированием.

Структура документа

Рекомендации разделены на ключевые блоки, соответствующие основным видам учебной деятельности в 6 классе:

1. Контроль знаний (тестирование).
2. Инструментально-технологический блок: работа с офисными приложениями, графикой, мультимедиа и 3D-моделированием.
3. Решение алгоритмических задач.

Каждый раздел содержит:

1. Пошаговые алгоритмы действий («чек-листы»).
2. Типичные сценарии выполнения заданий с разбором нюансов.
3. Систему критериев для самопроверки.

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 6 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. Уверенно ориентироваться в интерфейсе современных операционных систем и понимать принципы работы компьютера;
3. Создавать профессионально оформленные текстовые документы, презентации с мультимедиа и использовать электронные таблицы для сложных расчетов;
4. Разрабатывать интерактивные проекты в Scratch и осваивать основы текстового программирования;
5. Распознавать фишинг, вирусы, социальную инженерию и контент, созданный ИИ;
6. Эффективно организовывать свое рабочее цифровое пространство и соблюдать сетевой этикет
7. Получить базовые навыки работы с графикой, мультимедиа и 3D-моделированием
8. Развивать самостоятельность в обучении;
9. Формировать ответственное отношение к учебной деятельности;
10. Повышать уровень подготовки к текущему и итоговому контролю.

Особенности применения

Рекомендации могут применяться:

1. Как руководство при выполнении практических работ и проектов;
2. Во время выполнения домашних заданий;
3. При подготовке к диагностическим работам;
4. Для устранения пробелов в знаниях при самостоятельном изучении пропущенных тем.

Структура работы с документом

Для наиболее эффективного использования рекомендаций рекомендуется придерживаться следующего порядка работы:

1. Изучить теоретическую справку по теме занятия в КТП и дополнительные материалы по системам счисления и логике.
2. Открыть соответствующий раздел данных рекомендаций.
3. Выполнить подготовительные действия (проверка интернет-соединения, запуск нужного ПО, подготовка медиафайлов).
4. Следовать алгоритму выполнения задания, обращая внимание на советы по оформлению и технические нюансы.
5. Провести самоаудит работы по критериям оценивания.
6. В случае возникновения ошибок — использовать методы отладки (для Scratch) или правки (для документов, графики).

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 в информатике 6 класса направлено на оценку не только базовых знаний, но и более глубокого понимания цифровых процессов:

- Проверка понимания устройства ПК, принципов работы ОС и компьютерных сетей;
- Оценка знаний правил безопасного поведения в Интернете, включая распознавание социальной инженерии и кибербуллинга;
- Контроль владения терминологией (биты, байты, двоичный код, алгоритмический язык, ИИ, робототехника);
- Проверка умения анализировать информацию на достоверность, включая контент, созданный искусственным интеллектом;
- Формирование у обучающихся навыков самоконтроля и ответственности за результат.

Планируемые результаты

Обучающийся должен уметь:

- Идентифицировать устройства ввода, вывода, хранения информации и понимать их взаимодействие;
- Определять типы файлов по их расширениям и единицы измерения информации;
- Классифицировать виды информации, её свойства и формы представления (текст, графика, звук, видео);
- Выбирать безопасные стратегии поведения при столкновении с киберугрозами, распознавать фишинг, вирусы и социальную инженерию;
- Понимать основы двоичного кодирования информации и принципы работы алгоритмов;
- Различать контент, созданный человеком, от контента, сгенерированного ИИ (изображения, видео).

Методические рекомендации

Тесты по информатике 6 класса могут содержать более сложные вопросы, требующие не только знания определений, но и понимания принципов работы. При выполнении следует:

- анализировать визуальные данные, если в вопросе есть изображение окна программы, внимательно изучите активные кнопки и вкладки;
- стремиться «убирать лишнее» в вопросах про устройства или понятия часто можно отсеять заведомо неверные варианты, вспомнив основную функцию или определение.;
- стремиться «оптимизировать» ответ, облегчить содержание и конкретизировать ответ, особенно важно в заданиях с написанием кода;
- обращать особенное внимание к деталям, особенно это касается вопросов по системам счисления и алгоритмам, где важна каждая цифра или порядок действий.

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 три тематических модуля. Ниже представлен детальный разбор каждого блока с примерами вопросов.

Блок 1. Устройство компьютера и Теоретические основы информатики (3-4 задания):

- Задания на классификацию: определите, к какому типу относится устройство (например, «Выберите из списка только устройства вывода информации»).

- Задания про системы счисления: Перевод чисел из одной системы счисления в другую (например, «Переведите число 101101_2 в десятичную систему счисления»).
- Задания про единицы измерения информации: Расчет объема информации (например, «Сколько байт в 2 килобайтах?»).
- Задания про виды информации: Определение формы представления информации (например, «К какой форме представления информации относится звуковой файл?»).
- Задания про ПО и ВПО: Различие между системным, прикладным ПО и вредоносными программами (например, «Что такое антивирусная программа?»).

Блок 2. Интернет, облака и кибербезопасность (2-3 задания):

- Поиск информации: Умение формулировать запросы и отличать факты от фейков (например, «Какой оператор поиска позволяет исключить слово из результатов?»).
- Облачные сервисы: Принципы работы Яндекс Документов, Таблиц, Презентаций и совместного доступа (например, «Какое преимущество дает совместный доступ к документу в облаке?»).
- Кибергигиена и безопасность: Распознавание фишинга, вирусов, социальной инженерии и кибербуллинга (например, «Что такое социальная инженерия и как от неё защититься?»).
- ИИ-контент: Умение отличать контент, созданный человеком, от контента, сгенерированного ИИ (например, «По каким признакам можно определить, что изображение создано искусственным интеллектом?»).

Блок 3. Алгоритмика и Программирование (3 задания):

- Алгоритмические конструкции: Определение типа алгоритма (линейный, разветвляющийся, циклический) (например, «К какому типу относится алгоритм приготовления чая?»).
- Блочное программирование (Scratch): Чтение и анализ простых скриптов, понимание работы циклов, условий, переменных (например, «Что будет делать спрайт,

если он коснется края сцены и в скрипте есть блок "Если касается края, оттолкнуться"?»).

– Текстовое программирование: Базовое понимание команд и синтаксиса (например, «Что выведет на экран программа: `print("Привет, мир!")`?»).

– Логические ошибки (Дебаггинг): Поиск «бага» в предложенном фрагменте кода или алгоритме (например, «Почему программа не считает сумму чисел правильно?»).

Уровень	Описание	Пример задания
Базовый (50%)	Знание терминов, простых действий и определений.	Назовите единицу измерения информации, равную 1024 байтам.
Повышенный (35%)	Применение знаний в знакомой ситуации, решение простых задач.	Переведите число 1010_2 в десятичную систему счисления.
Творческий (15%)	Анализ, синтез, решение проблем, критическая оценка.	Предложите способ защиты от фишинговых писем и объясните, почему он эффективен

2. ТЕХНОЛОГИЧЕСКИЙ ПРАКТИКУМ: РАБОТА В ОБЛАЧНЫХ СЕРВИСАХ, ГРАФИКЕ И МУЛЬТИМЕДИА

Цели практических работ

Практическая деятельность в 6 классе расширяется за счет более сложных задач в облачных сервисах, работы с мультимедиа и введения в 3D-моделирование. Она направлена на:

- Освоение продвинутых инструментов Яндекс Документов, Таблиц и Презентаций (интерактивные элементы, мультимедиа). Развитие навыков визуального дизайна в Adele Design;
- Развитие навыков визуального дизайна в Adele Design, включая работу с растровой и векторной графикой, а также генерацию изображений с помощью ИИ;
- Формирование культуры совместной работы над документами в режиме реального времени и соблюдение сетевого этикета;
- Получение базовых навыков 3D-моделирования, в том числе с использованием ИИ.

Планируемые результаты

- Текст: создание структурированных документов с использованием стилей, оглавлений, сносок и встроенных объектов (таблицы, изображения, диаграммы);
- Таблицы: автоматизация расчетов с помощью функций (СУММ, СРЗНАЧ, МИН, МАКС), использование условного форматирования, создание сложных диаграмм и графиков;
- Презентации: создание интерактивных слайдов с анимацией, переходами, встроенными аудио- и видеофайлами, гиперссылками;
- Графика: создание рекламных баннеров, обложек книг, логотипов с использованием слоев, фильтров, эффектов, а также генерация изображений с помощью ИИ;
- Мультимедиа: понимание принципов работы с аудио- и видеофайлами, их интеграция в презентации;

– 3D-моделирование: создание простых 3D-объектов, понимание базовых принципов 3D-графики, знакомство с ИИ для 3D-моделирования.

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;

– Активно используйте стили для заголовков и подзаголовков, это упростит создание оглавления. Вставляйте объекты (таблицы, изображения) с учетом обтекания текстом;

– Всегда используйте ссылки на ячейки в формулах, а не вводите числа вручную.

– Соблюдайте правило «белого пространства» и не перегружайте слайд текстом.

– Используйте качественные изображения и видео. Анимацию и переходы применяйте умеренно, чтобы не отвлекать от содержания. Изучите возможности интерактивных элементов и гиперссылок;

– начните с простых 3D форм, постепенно усложняя их. Изучите основные инструменты для создания и редактирования объектов.

<i>Критерии оценивания практических работ</i>	
Техническая точность	работают ли формулы? Правильно ли настроена анимация и интерактивные элементы? Корректно ли отображаются мультимедиа?
Эстетика и дизайн	соблюден ли баланс цветов и шрифтов? Читаем ли текст? Гармонично ли расположены элементы? Качество сгенерированных ИИ изображений.
Структурированность	есть ли логика в расположении элементов? Соответствует ли структура документа его содержанию?

Соблюдение регламента	сохранен ли файл в нужном формате (например, экспорт в PDF, PNG, SVG)? Соблюдены ли требования к именованию файлов?
Функциональность	является ли модель целостной и соответствует ли заданию?

3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ (SCRATCH И ТЕКСТОВОЕ ПРОГРАММИРОВАНИЕ)

Программирование в 6 классе — это развитие логики через создание более сложных игр и анимаций, а также знакомство с основами текстового программирования. Мы учимся не просто управлять объектами, но и создавать интерактивные сценарии, используя переменные, события и более сложные алгоритмические конструкции.

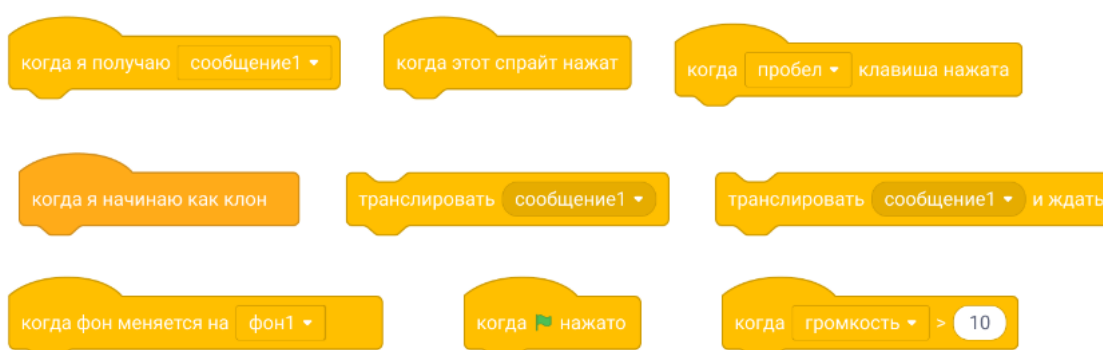
Цели и задачи раздела:

- углубленное понимание последовательности выполнения команд, ветвлений и циклов;
- освоение сложных циклов, вложенных условий, работы с переменными для создания счетчиков и таймеров;
- развитие навыка поиска и исправления логических ошибок в коде, понимание причин их возникновения;
- проектное мышление: умение пройти путь от идеи игры до её полной реализации и тестирования;
- знакомство с базовыми командами и синтаксисом простого языка программирования.

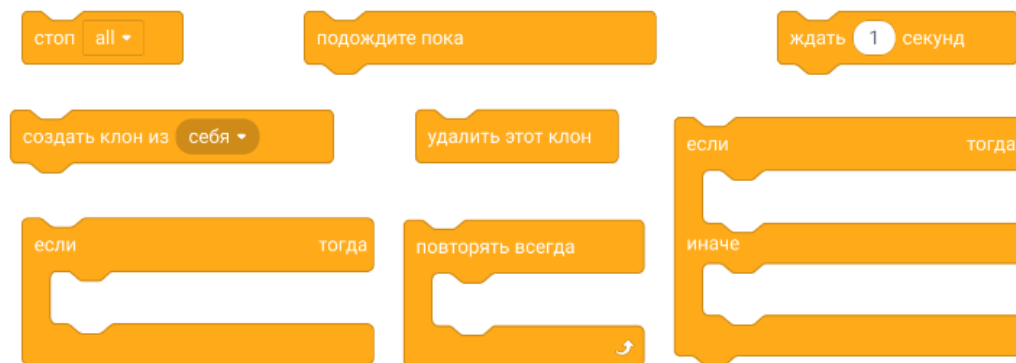
Основные инструменты Scratch (Чек-лист для ученика):

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

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



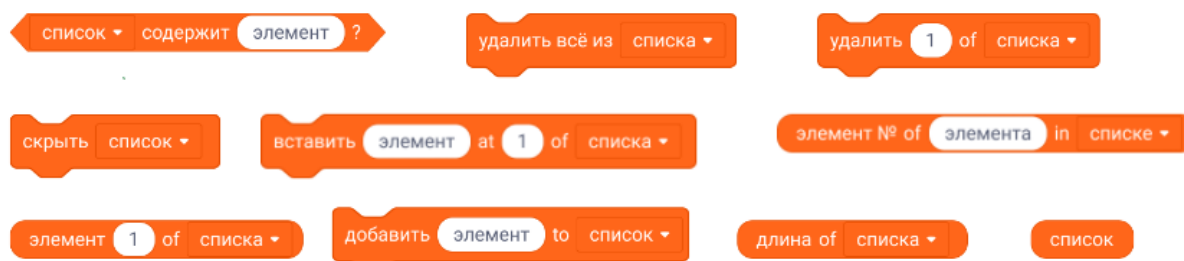
Управление (Оранжевые блоки): Цикл «Всегда»: используется для постоянных действий (например, проверка нажатия клавиш) Условие «Если... то...»: позволяет персонажу реагировать на события (например, касание края сцены или другого спрайта):



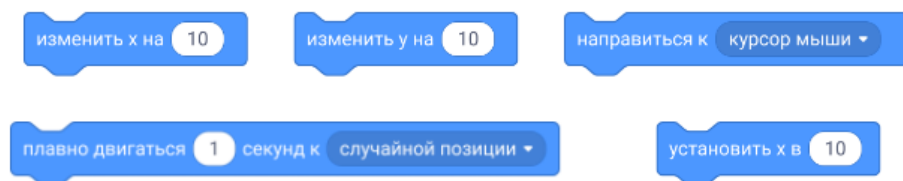
Сенсоры (Голубые блоки): это «органы чувств» вашего объекта. С их помощью он понимает, касается ли он цвета, другого объекта или нажата ли клавиша на клавиатуре:



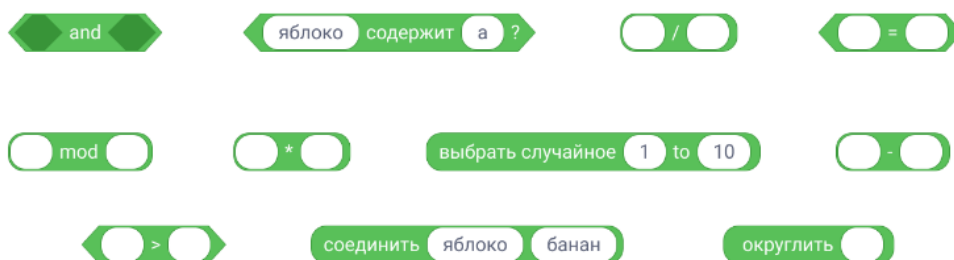
Переменные (Темно-оранжевые блоки): это ячейки памяти. В них мы храним очки, жизни или время:



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



Операторы (Зеленые блоки): Математические операции, сравнения и логические операторы (И, ИЛИ, НЕ). Они позволяют создавать сложные условия:



Методические рекомендации

Чтобы ваш проект (например, игра «Платформер» или «Квиз») 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. Набросайте сюжет игры, определите главных героев, их роли, цели и правила. Продумайте, какие объекты будут взаимодействовать, какие переменные нужны (очки, жизни, время).

2. Метод «Маленьких шагов»

Не пытайтесь собрать весь код сразу.

- сначала настройте движение (клавиши стрелок). Проверьте;
- затем добавьте препятствия и условия касания. Проверьте;
- реализуйте подсчет очков, таймеры, условия проигрыша/выигрыша;
- 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;
- используйте блоки «Мои блоки» для создания собственных функций, это сделает код более читаемым и компактным.

Основы текстового программирования:

В 6 классе ученики знакомятся с базовыми принципами текстового программирования. Это может быть простой язык, такой как Python (или его упрощенная версия), где вы узнаете:

- Выводить текст на экран (например, `print("Hello, World!")`).
- Работать с переменными (например, `x = 10`).
- Выполнять простые математические операции.

<i>Критерии оценивания практических работ</i>	
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
Чистота кода	код организован, используются комментарии, нет лишних блоков

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: используйте разные сложные пароли для разных сервисов. Всегда проверяйте источник информации, особенно если она кажется слишком «сенсационной». Учитесь распознавать фейки и контент, созданный ИИ.

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: поддерживайте четкую систему папок по разделам и проектам. Используйте осмысленные имена файлов, включающие дату или версию (например, Проект_Игра_v2_2026-07-14.sb3).

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.