

МИНИСТЕРСТВО НАУКИ ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.П.
АСТАФЬЕВА (КГПУ им. В.П. Астафьева)

Институт / факультет Институт Математики, физики и информатики
Кафедра информатики и информационных технологий в образовании

Гусейнова Гульнара Этибаровна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Развитие алгоритмического мышления обучающихся 7 - 8 классов в рамках
курса внеурочной деятельности по созданию игр в Construct 3**

Направление 44.03.05 Педагогическое образование (с двумя профилями
подготовки)

Направленность (профиль) образовательной программы Математика и
Информатика

ДОПУСКАЮ К ЗАЩИТЕ

Зав. кафедрой ИИТО

д-р пед. наук, профессор каф.

Пак Н.И.

(дата, подпись)

Руководитель:

канд. пед. наук, доцент каф.

ИИТО Яшина И.А.

(дата, подпись)

Обучающийся:

Гусейнова Г.Э.

(дата, подпись)

Дата защиты 21.06.2026

Оценка отлично

Красноярск 2026

ОГЛАВЛЕНИЕ

Введение.....	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ В СИСТЕМЕ ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ.....	6
1.1. Психолого-педагогические основы развития алгоритмического мышления у обучающихся 7-8 классов.....	6
1.2. Нормативно-правовое обеспечение и анализ существующих программ внеурочной деятельности по информатике для 7-8 классов.....	14
1.3. Сравнительный анализ средств визуального программирования для разработки игр в образовательном процессе.....	19
Выводы к главе 1.....	25
ГЛАВА 2. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И АПРОБАЦИЯ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ.....	26
2.1. Разработка курса внеурочной деятельности по созданию игр в Construct 3 на платформе КрЭШ.....	26
2.2. Аprobация курса и анализ результативности развития алгоритмического мышления обучающихся.....	31
Выводы к главе 2.....	36
ЗАКЛЮЧЕНИЕ.....	37
Библиографический список.....	39
ПРИЛОЖЕНИЕ А.....	45
ПРИЛОЖЕНИЕ Б.....	49
ПРИЛОЖЕНИЕ В.....	50
ПРИЛОЖЕНИЕ Г.....	52

Введение

Актуальность. В современном обществе необходимо уметь анализировать информацию и работать с ней, что требует развитого алгоритмического мышления. В условиях высокой степени автоматизации (беспилотные летательные аппараты, робототехнические комплексы, интеллектуальные технические средства) происходит стремительное насыщение технологической среды, однако подготовка специалистов, способных эффективно взаимодействовать с подобными системами, существенно отстаёт. Поскольку профессиональное переобучение взрослых сопряжено со значительными временными и ресурсными затратами, стратегическим приоритетом становится системная подготовка подрастающего поколения, в том числе через организованную внеурочную деятельность.

Исследование отвечает приоритетам государственной политики в области образования и технологического развития. Стратегия научно-технологического развития Российской Федерации, национальные проекты «Образование» и «Цифровая экономика», а также концепция развития дополнительного образования детей до 2030 года ориентируют систему образования на формирование кадрового потенциала для высокотехнологичных отраслей. В этих условиях алгоритмическое мышление трансформируется из узкопрофессиональной компетенции в базовую когнитивную основу, необходимую для адаптации к быстро меняющейся технологической среде.

Противоречие:

Между объективной потребностью общества и технологического уклада в специалистах с высоким уровнем алгоритмического мышления и отсутствием системно организованной внеурочной деятельности, направленной на его развитие у обучающихся 7-8 классов.

Проблема:

Как организовать внеурочную деятельность по созданию игр в Construct 3, чтобы целенаправленно развивать алгоритмическое мышление у школьников 7-8 классов в условиях быстрого технологического обновления?

Объект:

Процесс развития алгоритмического мышления обучающихся 7-8 классов.

Предмет:

Курс внеурочной деятельности по созданию игр в Construct 3 на платформе «КрЭШ», направленный на развитие алгоритмического мышления.

Цель:

Разработать и апробировать курс внеурочной деятельности по созданию игр в Construct 3 на платформе «КрЭШ», направленный на развитие алгоритмического мышления обучающихся 7-8 классов.

Задачи:

1. Изучить психолого-педагогическую и методическую литературу по проблеме формирования и развития алгоритмического мышления у обучающихся подросткового возраста;
2. Проанализировать нормативно-правовое обеспечение и существующие программы внеурочной деятельности по информатике для 7-8 классов;
3. Провести сравнительный анализ сред визуального программирования (Scratch, Construct 3, GDevelop и др.) с точки зрения их дидактического потенциала для развития алгоритмического мышления при разработке игр;
4. Разработать курс внеурочной деятельности по созданию игр в Construct 3 на платформе «КрЭШ»;
5. Апробировать разработанный курс и проанализировать результативность развития алгоритмического мышления обучающихся на основе валидных психолого-педагогических диагностических методик.

Структура работы состоит из введения, двух глав, выводов по главам, заключения, библиографического списка и приложений.

В первой главе рассматриваются теоретические основы исследования. В ней раскрываются психолого-педагогические особенности подростков 7-8 классов, анализируется нормативно-правовая база внеурочной деятельности и обосновывается выбор платформы Construct 3 на основе сравнения сред визуального программирования.

Во второй главе описывается практическая реализация исследования. В ней представлено методическое обеспечение курса внеурочной деятельности на платформе «КрЭШ», а также описан ход и результаты опытно-экспериментальной работы, подтверждающие его результативность.

В заключении сформулированы обобщающие выводы и намечены перспективы продолжения исследования и дальнейшей разработки темы

Библиографический список содержит источники информации, использованные при выполнении работы.

Работа содержит 3 рисунка, 10 таблиц. Приложения включают таблицу диагностики уровня сформированности алгоритмического мышления, интерактивное задание, карту рефлексии, образец игрового проекта обучающегося.

Результаты работы опубликованы:

Научное сообщество студентов: междисциплинарные науки: материалы III Всероссийской молодежной научно-практической конференции [Электронный ресурс] / редколлегия : Т. Н. Можарова, Н. А. Гончарова, И. И. Чернобровкина. Орёл: ОГУ имени И.С. Тургенева.2026. Название статьи: «От Scratch до Godot: обзор доступных платформ для создания игр школьниками»

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ В СИСТЕМЕ ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

1.1. Психолого-педагогические основы развития алгоритмического мышления у обучающихся 7-8 классов

В первом параграфе первой главы рассмотрены психолого-педагогические и нормативно-методические аспекты развития алгоритмического мышления у подростков 7-8 классов, уточнен понятийный аппарат («алгоритм», «алгоритмическое мышление») с учетом возрастных когнитивных особенностей (включая феномен клипового мышления), а также обоснована критериально-уровневая модель диагностики, позволяющая объективно оценивать динамику формирования алгоритмических умений обучающихся.

Для целенаправленного развития алгоритмического мышления необходимо четко определить сущность ключевых понятий исследования: «мышление», «алгоритм» и «алгоритмическое мышление». Проведем обзор трактовок данных понятий в научно-методической литературе, выделив общие и специфические характеристики.

Таблица 1

Сравнительный анализ определений понятия «мышление» в научно-методической литературе

№	Определения	Комментарий	Ключевые слова	Автор
1	Мышление – психический процесс обобщенного и опосредованного отражения устойчивых, закономерных свойств и отношений действительности, существенных для решения познавательных проблем,	Мышление – продуктивно, оно представляет собой способность ориентироваться в новых для нас данных, способность понимать их	Психический процесс, закономерность свойств и отношений, познавательная проблема	К.А. Абульханова – Славская, советский и российский психолог и философ. Академик РАО, доктор философских наук, кандидат педагогических наук (по

	схематической ориентации в конкретной ситуациях			психологии) [1]
2	Мышление – это психический процесс самостоятельного искания и открытия существенно нового, процесс отражения действительности в ходе ее анализа и синтеза, возникающий на основе практической деятельности	Процесс становления, формирования и развития, не завершённое в своем открытии новых свойств и отношений объекта	Психический процесс, формирование, развитие, открытие новых качеств	А.В. Брушлинский – советский и российский психолог [6]
3	Мышление – процесс отражения общих свойств предметов и явлений, нахождения закономерных связей и отношений между ними	Главное в данном определении, что мышление это процесс	Процесс отражения общих свойств предметов и явлений, закономерные связи, отношение между закономерными связями	К.К. Платонов – советский психолог, доктор психологических наук и доктор медицинских наук, профессор [32]
4	Мышление – это один из видов познания. Сущность мышления заключается в создании умственных моделей и действий с ними	Широта мышления – это способность охватить весь вопрос целиком, не упуская в то же время и необходимых для дела частных. Глубина мышления выражается в умении проникать в сущность сложных вопросов	Вид познания. Умственные модели. Действия с ними	Я.А. Пономарёв – советский ученый, психолог [33]

Проанализировав данные определения (таблица 1), можно сделать вывод, что мышление – это высший психический процесс познания закономерностей окружающего мира, основу которого составляет образование понятий, суждений и

умозаключений. Мышление включает в себя операции анализа, синтеза, сравнения и обобщения, направленные на решение познавательных задач.

Следующим ключевым понятием является «алгоритм». В таблице 2 приведены трактовки данного понятия в математике и информатике.

Таблица 2

Сравнительный анализ определений понятия «алгоритм» в научно-методической литературе

№	Определения	Комментарий	Ключевые слова	Автор
1	Алгоритм - это точное указание о том, какие действия и в каком порядке необходимо выполнить для решения любой задачи	Важным в данном определении является указание на то, какие действия и в каком порядке их необходимо выполнять	Действия, порядок выполнения	А.А. Столяра - советский педагог [23]
2	Алгоритм – это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату	Используется такое слово как «предписание». Предписание – это описание последовательности действий. Оно же определяет вычислительный процесс	Предписание, вычислительный процесс, варьируемые исходные данные	А.А. Марков – русский математик, академик [26]
3	Алгоритм – это всякая система вычислений, выполняемых по строго определенным правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи	В данном определении алгоритм понимается как система вычислений, выполняемые по строго определенным правилам	Система вычислений, определенные правила	А.Н. Колмогоров – один из крупнейших математиков XX века [18]
4	Алгоритм – набор предписаний, однозначно определяющий	Содержательно в данном определении то, что алгоритм не только	Набор предписаний, последовательность	В.Ф. Мелехин – кандидат технических наук, доцент,

	содержание и последовательность выполнения действий для систематического решения задачи	«предписание», а «набор предписаний», однозначно определяющий содержание и последовательность выполнения действий	выполнения действий, систематическое решение задачи	профессор [26]
5	Алгоритм – это описание конечной последовательности шагов в решении задачи, приводящей от исходных данных к требуемому результату	В данном определении алгоритм представлен конечной последовательностью шагов в решении задач	Конечная последовательность, исходные данные, результат	Л.Л. Босова – автор учебников по информатике [4]

На основе анализа таблицы 2 можно заключить, что алгоритм – это точное, однозначное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи за конечное число шагов.

Центральным понятием исследования является «алгоритмическое мышление». В таблице 3 представлены подходы к определению данного феномена в психолого-педагогической литературе.

Таблица 3

Подходы к определению понятия «алгоритмическое мышление» в психолого-педагогической литературе

№	Определения	Комментарий	Ключевые слова	Автор
1	Алгоритмический стиль мышления представляет собой специфический стиль мышления, предполагающий умение создать алгоритм, для чего необходимо наличие мыслительных схем, которые способствуют видению проблемы в целом, ее решению крупными	Из этого определения важным представляется специфический стиль мышления, предполагающий умение создать алгоритм	Алгоритм, мыслительные схемы, языковые формы	Г.Э. Белицкая, кандидат психологических наук [3]

	блоками с последующей детализацией и осознанным закреплением процесса получения конечного результата в языковых формах			
2	Алгоритмическое мышление – это искусство размышлять, умение планировать свои действия, способность предусматривать различные обстоятельства и поступать соответственно с ними	Главное в этом высказывании то, что алгоритмическое мышление можно понимать, как систему мыслительных приемов направленных на решение задач	Искусство размышлять, умение планировать свои действия, способность предусматривать различные обстоятельства	Ю.К. Корнилов, российский психолог, профессор, кандидат психологических наук [19]
3	Алгоритмический способ мышления позволяет принимать оптимальные решения в любой сфере человеческой деятельности	Используется такое понятие, как принятие оптимального решения	Оптимальное решение	И.В. Васюта, А.Л. Махотина учителя наставники [8]
4	Алгоритмическое мышление – познавательный процесс, характеризующийся наличием четкой, целесообразной (или рациональной) последовательности совершаемых мыслительных процессов с присущей детализацией, осознанным закреплением процесса получения конечного результата	В данном определении главным является наличие четкой, последовательности совершаемых мыслительных процессов с детализацией, наличием конечного результата	Познавательный, мыслительный процесс, последовательность процессов, детализация	Т. Н. Лебедева – кандидат педагогических наук, доцент [22]

На основе анализа таблицы 3 можно заключить, что под алгоритмическим мышлением понимается стиль мышления, включающий систему мыслительных операций (анализ, синтез, абстрагирование, моделирование), направленных на поиск эффективного решения задач путем построения четкого, детерминированного плана действий (алгоритма).

Однако целенаправленное формирование данного стиля мышления невозможно в отрыве от реального субъекта обучения. Построение четкого плана действий требует от ученика высокой степени концентрации, умения выстраивать сложные причинно-следственные связи и удерживать в уме многошаговые структуры. В связи с этим возникает объективная необходимость соотнести теоретическую модель алгоритмического мышления с актуальными психофизиологическими и когнитивными характеристиками современных школьников.

При организации педагогического эксперимента и выборе диагностического инструментария необходимо учитывать возрастные и когнитивные особенности обучающихся 7-8 классов (13-15 лет), так как именно эти характеристики определяют специфику восприятия учебной информации и способы ее обработки.

Современные подростки развиваются в условиях цифровой культуры, характеризующейся высокой насыщенностью информационного потока, его фрагментарностью и визуализацией. Исследователи отмечают, что интернет стал главным источником знаний и повседневных развлечений для подрастающего поколения. По статистике ВЦИОМ, среди подростков только 2% не посещают социальные сети, что свидетельствует о глубокой интеграции цифровых технологий в их повседневную жизнь [27].

В связи с этим в психолого-педагогической литературе активно обсуждается феномен «клипового мышления» – особого способа переработки информации, характерного для современного поколения.

Анализ научной литературы позволяет выделить следующие характеристики клипового мышления, проявляющиеся у подростков:

1. быстрое переключение на новый незнакомый раздражитель;
2. высокая скорость восприятия информации при поверхностном изучении объекта;
3. снижение способности к анализу и установлению логических связей;
4. умение одновременно выполнять несколько операций (многозадачность) ;
5. неумение концентрироваться на одной задаче в течение длительного времени [21, 35];
6. восприятие мира через короткие яркие образы и сообщения;
7. дискретность восприятия и языковой минимализм.

Важно отметить, что клиповое мышление не является исключительно негативным феноменом. С одной стороны, оно представляет собой адаптационный механизм к ускорившемуся темпу обмена информацией, развивает динамизм познавательной деятельности и способность перерабатывать большие объемы информации. С другой стороны, наблюдаются такие негативные проявления, как мозаичность восприятия, снижение способности к глубокому анализу, поспешность выводов и фрагментарность знаний.

Выявленные особенности когнитивного развития современных подростков обусловили выбор специфического диагностического инструментария и методических подходов к развитию алгоритмического мышления.

Специфика «клипового» восприятия информации вступает в очевидное противоречие с самой сутью алгоритмического мышления, требующего системности, последовательности и глубокого логического анализа. Данная когнитивная коллизия диктует особые требования не только к методам обучения, но и к способам объективной оценки сформированности алгоритмических умений. Традиционные подходы к диагностике, не учитывающие

фрагментарность восприятия современных подростков и их потребность в визуальной опоре, оказываются недостаточно валидными [9].

В современной методике преподавания информатики существует противоречие между необходимостью оценивать уровень развития алгоритмического мышления и отсутствием единого унифицированного инструментария. Опираясь на пространственно-уровневую модель императивного алгоритмического мышления И.В. Гавриловой, разработанную на базе информационной модели памяти Н.И. Пака и Т.А. Степановой, а также учитывая когнитивные процессы, описанные в модифицированной таксономии Б. Блума (Л. Андерсона), нами была адаптирована критериально-уровневая модель диагностики для обучающихся 7-8 классов [10,31,37-38].

Таблица 4

Критериально-уровневая модель диагностики алгоритмического мышления обучающихся

Область памяти / Критерий	Начальный уровень	Достаточный уровень	Оптимальный уровень
Представления (Чувственные образы)	Имеет общее, фрагментарное представление об алгоритме как о последовательности и действий в быту	Понимает алгоритмическую сущность повседневных действий, может привести корректные примеры из жизни	Видит алгоритмические процессы в окружающих природных и социальных явлениях, свободно оперирует примерами
Понятия (Знания)	Знает определение алгоритма и его базовые свойства (дискретность, результативность) на уровне узнавания	Владеет понятиями «исполнитель», «система команд», знает и различает базовые алгоритмические конструкции	Свободно оперирует понятиями «вложенный алгоритм», «оптимизация», понимает ограничения исполнителя

Умения (Модельные действия)	Умеет исполнять готовый простой линейный алгоритм по образцу или блок-схеме	Умеет составлять простые алгоритмы (линейные, с ветвлением), классифицировать задачи по типу алгоритма	Умеет разбивать задачу на подзадачи, генерировать сложные (циклические, комбинированные) алгоритмы, прогнозировать результат
Абстрагирование (Формальные связи)	Испытывает трудности при переходе от словесного описания к формальной записи или блок-схеме	Умеет устанавливать соответствие между задачей и типом алгоритма, строить простые информационные модели	Умеет строить математические/информационные модели для решения нестандартных задач, оптимизировать алгоритмы

Предложенная критериально-уровневая модель в таблице 4 служит надежным инструментом для педагога и исследователя. Она позволяет не просто констатировать факт наличия или отсутствия алгоритмических умений, но и выстроить индивидуальную траекторию развития подростка.

1.2. Нормативно-правовое обеспечение и анализ существующих программ внеурочной деятельности по информатике для 7-8 классов

Во втором параграфе первой главы рассмотрены нормативно-правовые основы организации внеурочной деятельности по информатике в основной школе, опирающиеся на требования федеральных государственных образовательных стандартов. Проведен сравнительный анализ существующих примерных рабочих программ курсов внеурочной деятельности с выстраиванием логической преемственности от начальной школы к подростковому возрасту.

На начальном этапе формирования алгоритмической культуры, охватывающем первую-четвертую ступени обучения, реализуется программа «Основы логики и алгоритмики» [28]. Данный пропедевтический курс успешно

закладывает базовый тезаурус, знакомя младших школьников с понятиями исполнителя, команды и линейного алгоритма. Дидактическим ядром здесь выступает визуальная блочная среда Scratch, которая полностью исключает синтаксические ошибки и позволяет детям в игровой форме постигать причинно-следственные связи. К моменту завершения начальной школы обучающиеся уже обладают устойчивым опытом создания простых скриптов и анимаций, однако когнитивные запросы подростков по мере их взросления неизбежно меняются.

Этот возрастной сдвиг учитывает программа «Основы программирования», предназначенная для пятого и шестого классов [29]. Она сохраняет визуальную наглядность блочных сред, но существенно усложняет содержание, переходя от простой анимации к созданию полноценных интерактивных игр, информационных моделей и работе с событиями. Несмотря на очевидный дидактический потенциал, к седьмому классу многие обучающиеся начинают воспринимать подобные среды как устаревший или слишком детский инструмент, что критически снижает их внутреннюю мотивацию к изучению алгоритмики. Возникает закономерная потребность в более взрослом, профессионально ориентированном инструментарии, который сохраняет визуальную поддержку, но предлагает качественно новый уровень сложности.

Существующая система образования пытается решить эту задачу путем резкого перехода в седьмом-девятом классах к текстовому программированию на языке Python в рамках курса «Основы программирования на Python» [30]. Безусловно, данная программа формирует строгую формализованную базу и учит переводить алгоритм на язык машины. Однако для подростков 13-14 лет, находящихся на хрупком этапе перехода от наглядно-образного к абстрактному мышлению, высокий уровень абстракции и строгие синтаксические требования текстового языка становятся непреодолимым барьером. Фокус внимания школьника неизбежно смещается с поиска алгоритмического решения на

преодоление синтаксических ошибок, что подменяет развитие алгоритмического мышления навыками слепого набора кода и демотивирует обучающихся [7].

В качестве альтернативы или дополнения в школах часто реализуется междисциплинарный курс «Функциональная грамотность» для 5-9 классов [42]. Эта программа направлена на развитие читательской, математической и критической грамотности, формируя умение работать с информацией и решать практико-ориентированные жизненные задачи. Хотя данный курс создает важную базу для алгоритмизации, алгоритмическое мышление и событийное моделирование не являются его ядром. Здесь алгоритмы рассматриваются скорее как бытовые инструкции, без углубления в кибернетические модели, циклические процессы и событийную логику, которые необходимы для создания сложных цифровых продуктов.

В связи с этим, рассмотрим, что именно изучают обучающиеся в области алгоритмизации в рамках обязательного курса информатики. Конкретизация содержания обучения информатике на уровне основного общего образования представлена в Федеральных рабочих программах (ФРП) по учебному предмету «Информатика» для 7-9 классов, утвержденных приказом Министерства просвещения РФ от 18.05.2023 № 370 и актуализированных в 2025 году [39, 40]. Документы разработаны для двух уровней изучения предмета: базового и углубленного.

На базовом уровне общее количество часов, рекомендованных для изучения информатики, составляет 102 часа (по 34 часа в 7, 8 и 9 классах, 1 час в неделю). Содержательная линия «Алгоритмы и программирование» на базовом уровне начинается с 8 класса и включает следующие темы: понятие алгоритма, исполнители алгоритмов, алгоритмические конструкции («следование», «ветвление», «повторение»), разработка несложных алгоритмов для управления формальными исполнителями (Робот, Черепашка, Чертежник), основы программирования на одном из языков высокого уровня (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык).

На углубленном уровне общее число часов составляет 204 часа (по 68 часов в 7, 8 и 9 классах, 2 часа в неделю). Изучение раздела «Алгоритмы и программирование» на углубленном уровне начинается уже в 7 классе и предполагает более глубокое освоение алгоритмических конструкций, работу с вспомогательными алгоритмами, рекурсией, массивами, а также решение задач повышенной сложности [3].

В таблице 5 представлено сравнение содержания раздела «Алгоритмы и программирование» в Федеральных рабочих программах базового и углубленного уровней для 7-8 классов.

Таблица 5

Сравнительная характеристика раздела «Алгоритмы и программирование» в ФРП по информатике (базовый и углублённый уровни)

Критерий сравнения	Базовый уровень	Углублённый уровень
Начало изучения раздела	8 класс	7 класс
Объем часов (7-8 классы)	34 часа (только 8 класс)	136 часов (68 часов в 7 классе + 68 часов в 8 классе)
Алгоритмические конструкции	Следование, ветвление, цикл с условием, цикл с переменной	Те же + вспомогательные алгоритмы, рекурсия, параметры подпрограмм
Исполнители	Робот, Черепашка, Чертежник	Те же + работа с графикой и анимацией
Язык программирования	Один из: Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык	Один из: Python, C++, Java, C#
Обработка данных	Одномерные массивы, обработка потока данных	Одномерные и двумерные массивы, динамическое программирование
Практическая направленность	Решение учебных задач, отладка программ	Проектная деятельность, моделирование, решение задач повышенной сложности

Анализ таблиц тематического планирования показывает, что на углубленном уровне уже в 7 классе обучающиеся знакомятся с понятием

алгоритма, исполнителями, алгоритмическими конструкциями и начинают практическую работу по составлению алгоритмов для управления исполнителями в визуальной или блочной среде [2, 5]. На базовом уровне аналогичный материал изучается в 8 классе, что создает определенные трудности для обучающихся, не имеющих предварительной подготовки по алгоритмизации.

При этом важно отметить, что к моменту начала изучения раздела «Алгоритмы и программирование» у обучающихся 7-8 классов уже сформированы определенные предпосылки алгоритмического мышления, а именно, на уроках математики они решают задачи, требующие пошагового планирования, на уроках технологии выполняют работы по инструкциям [34].

Федеральные рабочие программы предусматривают возможность реализации внеурочной деятельности как формы расширения и углубления предметного содержания. В пояснительных записках к ФРП указано, что образовательная организация вправе использовать часы, формируемые участниками образовательных отношений, для организации курсов внеурочной деятельности, факультативов, проектной работы. Это создает нормативные основания для разработки и реализации авторских курсов, направленных на развитие алгоритмического мышления средствами современных сред визуального программирования, таких как Construct 3.

Таким образом, нормативно-правовое обеспечение обучения информатике в школе, представленное ФГОС ООО и Федеральными рабочими программами 2025 года, определяет требования к развитию алгоритмического мышления обучающихся, устанавливает содержание обучения на базовом и углубленном уровнях и предоставляет возможности для организации внеурочной деятельности [41]. Данные документы служат методологической основой для проектирования курса внеурочной деятельности по созданию игр в среде Construct 3 для обучающихся 7-8 классов, направленного на систематическое развитие алгоритмического мышления в увлекательной и практико-ориентированной форме.

1.3. Сравнительный анализ средств визуального программирования для разработки игр в образовательном процессе

В третьем параграфе первой главы проведен сравнительный анализ сред визуального программирования для выявления оптимальной среды программирования.

Современным школьникам зачастую не хватает мотивации при изучении программирования, а традиционные языки изучаемые в школе (Pascal, Python и др.), могут вызывать у них трудности в освоении. В связи с этим педагоги постоянно находятся в поиске удобных и эффективных платформ, способных сделать обучение более увлекательным. В данной статье рассматриваются именно такие инструменты, которые превращают процесс освоения программирования в захватывающее занятие через создание собственных игр.

Scratch и его российский аналог Robbo Scratch являются специализированными средами, созданными для обучения программированию детей, начиная с 7 лет. Их ключевое преимущество – понятный интерфейс и визуальное представление кода в виде собираемых блоков-команд. Платформа Scratch предоставляет как веб-версию, так и офлайн-приложение, что позволяет удобно организовывать работу с учениками в классе без постоянного доступа к интернету. Важным мотивирующим фактором является возможность быстро получить готовый игровой проект уже в конце урока. Кроме того, глобальное сообщество Scratch позволяет пользователям делиться своими работами, изучать проекты со всего мира, а участие в профильных олимпиадах становится для детей дополнительным стимулом к развитию.

Благодаря своей гибкости и возможности изучения сложных алгоритмических конструкций, Scratch может использоваться не только в начальной школе, но и для обучения программированию обучающихся более старших классов. Как показано в исследовании Гурьевских М. и Рожиной И. В., специально разработанная система занятий позволяет эффективно применять эту среду для обучения школьников, осваивающих более сложные темы в 8 классе

[11]. Универсальность инструмента также подтверждается широким возрастным охватом участников олимпиад, в которых могут принимать участие не только школьники, но и взрослые.

Для школьников, освоивших базовые принципы, следующим шагом могут стать инструменты, позволяющие создавать более сложные и производительные 2D и 3D игры без написания кода при помощи списков событий. Одной из таких платформ является редактор Construct 3. Караванский А. Н. в своем исследовании указывает на применение Construct 3 для разработки обучающих игр, отмечая его интуитивный интерфейс, основанный на событийной системе, и возможность быстрой публикации проектов в веб-формате [17]. Этот инструмент хорошо подходит для реализации первых полноценных игровых проектов с разнообразной механикой.

Параллельно или в качестве альтернативы для создания мобильных приложений, в том числе игровых, может использоваться среда MIT App Inventor. Кукушкина Н. А. и Павлущенко И. А. рассматривают проектирование в этой среде как средство формирования познавательного интереса у школьников, поскольку результат работы можно сразу протестировать на собственном смартфоне [20]. Малова А. И. также подтверждает эффективность визуальной среды App Inventor для обучения программированию, акцентируя внимание на возможности перехода от визуального проектирования к текстовому программированию на основе сгенерированного кода [24].

Для обучающихся старших классов, проявляющих серьезный интерес к разработке игр, оптимальным выбором становится профессиональный, но при этом доступный игровой движок. В обзоре игровых движков для образовательных целей Степушкина П. К. выделяет Godot как один из наиболее перспективных вариантов [38]. Godot предоставляет возможность глубокого погружения в процессы разработки игр, от дизайна уровней и работы с физикой до создания сложных скриптов, что делает его отличным мостом к профессиональным инструментам.

Для наглядного представления возможностей рассмотренных инструментов ниже приведена их сравнительная таблица.

Таблица 6

Сравнительный анализ образовательных платформ

Критерий / Платформа	Scratch / Robbo Scratch	Construct 3	MIT App Inventor	Godot
1. Возрастная адресность	Начальная школа (1-6 кл.) Идеальный старт для детей 7-12 лет	Основная школа (5-9 кл.) Подходит для обучающихся, освоивших основы алгоритмизации	Основная школа (6-9 кл.). Ориентирован на создание мобильных приложений, что актуально для подростков	Старшая школа (9-11 кл.) и углубленное изучение. Для обучающихся, готовых к работе с профессиональным и инструментами
2. Парадигма программирования	Визуальное блочное. Программы собираются из готовых блоков-команд, что исключает синтаксические ошибки	Визуальное событийное. Логика строится через систему условий и реакций («если-то»), что ближе к текстовому коду	Визуальное блочное, с возможностью просмотра сгенерированного текстового кода (Java), обеспечивая плавный переход к его изучению	Текстовое программирование (GDScript, C#); визуальный скриптинг через плагины (Orchestretor). Полноценное профессиональное программирование
3. Кривая обучения	Очень пологая. Первый игровой проект создаётся на первом же занятии, что мгновенно даёт положительное подкрепление	Умеренная. Позволяет быстро создавать сложные 2D и 3D игры, но для реализации нестандартной логики требуется глубокое понимание событийной модели	Умеренная. Перенос логики на мобильное устройство интуитивно понятен, однако проектирование интерфейса требует дополнительного внимания	Крутая. Требуется изучение синтаксиса языка, архитектуры движка и принципов работы с ресурсами, что окупается безграничными возможностями

4.Целевая платформа/ Результат	Веб-браузер, офлайн-приложение. Интерактивные истории, анимации, простые 2D-игры	Веб-браузер (HTML5), десктоп. Качественные 2D-игры, готовые к публикации на игровых порталах	Мобильные устройства (Android). Работающие мобильные приложения, устанавливаемые на смартфон, что является мощным мотиватором	Кроссплатформенность (ПК, мобильные, веб). Полноценные 2D/3D-игры профессионального уровня
5. Поддержка образовательного процесса	Максимальная. Огромное глобальное и русскоязычное сообщество, учебные курсы, встроенная галерея проектов, система олимпиад (например, Robbo Olympiad)	Умеренная. Много видео-уроков, в том числе на русском, активное сообщество разработчиков	Высокая. Официальные учебные материалы, проекты для повторения, ориентированность на учебные задачи. Прямая связь с технологиями MIT	Растущая. Официальная документация, активное сообщество. Требует от педагога высокой квалификации или самообучения

Анализ таблицы 6 показывает, что каждая из рассмотренных платформ имеет свою целевую аудиторию и образовательные возможности. Scratch и его российский аналог Robbo Scratch являются специализированными средами, созданными для обучения программированию детей, начиная с 7 лет [13,14]. Их ключевое преимущество – понятный интерфейс и визуальное представление кода в виде собираемых блоков-команд. Однако для обучающихся 7-8 классов, уже имеющих базовые представления об алгоритмических конструкциях, данная среда может быть недостаточно сложной для дальнейшего развития алгоритмического мышления.

MIT App Inventor параллельно или в качестве альтернативы может использоваться для создания мобильных приложений, в том числе игровых. Проектирование в этой среде рассматривается как средство формирования познавательного интереса у школьников, поскольку результат работы можно сразу

протестировать на собственном смартфоне. Однако данная среда ориентирована преимущественно на создание мобильных приложений, а не игр, что не полностью соответствует целям разработанного курса.

Для обучающихся старших классов, проявляющих серьезный интерес к разработке игр, оптимальным выбором становится профессиональный, но при этом доступный игровой движок Godot. Он предоставляет возможность глубокого погружения в процессы разработки игр, от дизайна уровней и работы с физикой до создания сложных скриптов. Однако крутая кривая обучения и необходимость изучения синтаксиса языка программирования делают данную среду недостаточно подходящей для обучающихся 7-8 классов в рамках внеурочной деятельности.

Construct 3 занимает промежуточное положение между блочными средами программирования и профессиональными игровыми движками [44]. Данная среда позволяет учащимся создавать игры и интерактивные приложения, используя событийную модель программирования. Это снижает порог входа за счет отсутствия необходимости написания текстового кода, но при этом сохраняет требования к логике построения алгоритмов, использованию условий, циклов и переменных.

В таблице 7 представлены преимущества и недостатки среды Construct 3 для использования в школьном образовании.

Таблица 7

Преимущества и недостатки среды Construct 3 для школьного образования

Преимущества	Недостатки
Событийная модель программирования, близкая к текстовому коду	Требует понимания событийной модели для реализации нестандартной логики
Возможность быстрой публикации проектов в веб-формате (HTML5)	Умеренная поддержка образовательного процесса по сравнению со Scratch
Интуитивный интерфейс, снижающий порог входа	Необходимость доступа к интернету для веб-версии (хотя есть офлайн-версия)
Создание качественных 2D-игр с разнообразной механикой	Ограниченные возможности для создания 3D-игр по сравнению с профессиональными движками

Возможность экспорта проектов на различные платформы (веб, десктоп, мобильные)	Платная лицензия для полноценного использования (есть бесплатная версия с ограничениями)
Сохранение требований к логике построения алгоритмов (условия, циклы, переменные)	

Выбор среды Construct 3 для разработки курса внеурочной деятельности по созданию игр для обучающихся 7-8 классов обоснован следующими факторами:

1. Современные подростки не могут долго концентрироваться на монотонном тексте. Construct 3 обеспечивает мгновенный визуальный отклик и любое изменение в листе событий сразу же отражается в окне предпросмотра игры. Это удерживает внимание и превращает процесс отладки алгоритма в увлекательный эксперимент.
2. Событийная модель программирования в Construct 3 требует от обучающихся выстраивания логических цепочек «если-то», использования условий, циклов и переменных. Это сохраняет требования к логике построения алгоритмов, что способствует развитию алгоритмического мышления, являющемуся целью данного исследования.
3. Возможность быстро создать полноценный игровой проект и опубликовать его в веб-формате обеспечивает положительное подкрепление и поддерживает интерес обучающихся на протяжении всего курса. Создание лично значимого цифрового продукта повышает мотивацию к обучению.
4. Construct 3 позволяет структурировать учебный материал и размещать проекты в веб-формате, что обеспечивает возможность интеграции курса с онлайн-платформой «КрЭШ» для методической поддержки педагогов и организации очных занятий.
5. Работа в Construct 3 создает фундамент для последующего изучения более сложных языков программирования, так как событийная модель близка к

текстовому программированию, обеспечивая плавный переход от визуального к текстовому коду.

Для целей данного исследования оптимальным выбором является среда визуального программирования Construct 3, которая позволяет последовательно развивать компетенции ученика 7-8 классов – от визуального решения задач к абстрактно-логическому программированию, поддерживая устойчивую мотивацию через создание лично значимых цифровых продуктов в рамках курса внеурочной деятельности [12].

Выводы к главе 1

В первой главе были рассмотрены психолого-педагогические аспекты развития алгоритмического мышления у подростков 7-8 классов, уточнен понятийный аппарат, а также обоснована критериально-уровневая модель диагностики. Рассмотрены нормативно-правовые основы организации внеурочной деятельности по информатике в основной школе, опирающиеся на требования федеральных государственных образовательных стандартов. Проведен сравнительный анализ сред визуального программирования для выявления оптимальной среды программирования.

Подводя итоги первой главы, можно констатировать, что развитие алгоритмического мышления у подростков требует учета их когнитивных особенностей, в частности феномена клипового мышления, что обосновывает применение критериально-уровневой модели диагностики.

Анализ нормативной базы и существующих программ внеурочной деятельности выявил потребность в инструменте, обеспечивающем методическую преемственность между начальным и средним звеном.

В результате сравнительного анализа сред визуального программирования было доказано, что Construct 3 является наиболее оптимальным инструментом для обучающихся 7-8 классов. Эта среда позволяет целенаправленно развивать алгоритмическое мышление школьников через создание качественных 2D-игр.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И АПРОБАЦИЯ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

2.1. Разработка курса внеурочной деятельности по созданию игр в Construct 3 на платформе КрЭШ

В первом параграфе второй главы представлено содержательное и методическое обеспечение разработанного курса внеурочной деятельности. Курс рассчитан на обучающихся 7-8 классов, его объем составляет 12 занятий, что соответствует одной учебной четверти. Главной целью курса является не просто обучение навыкам программирования, а целенаправленное развитие алгоритмического мышления через создание игровых проектов в среде визуального программирования Construct 3 с электронным сопровождением на платформе «КрЭШ».

Структура курса выстроена по принципу «от простого к сложному» и включает три логических блока: вводно-диагностический, основной (обучающий) и проектно-итоговый.

Вводно-диагностический блок проводится до начала основных занятий. С целью фиксации исходного уровня сформированности алгоритмического мышления обучающиеся проходят входное диагностическое тестирование. Тест направлен на выявление умений декомпозировать задачи, работать с блок-схемами, выделять алгоритмические конструкции и выстраивать логические цепочки, что позволяет избежать подмены диагностики алгоритмического мышления проверкой предметных знаний по информатике (Приложение А).

Основной блок направлен на изучение среды Construct 3 и освоение базовых алгоритмических конструкций [15-16, 43].

Занятие 1 посвящено знакомству с интерфейсом среды Construct 3, изучению базовых инструментов и созданию первого простейшего проекта.

Занятия 2-9 включают изучение ключевых игровых механик и соответствующих им алгоритмических конструкций: объектов и событий, линейных, условных алгоритмов, циклов, работы с переменными и отладки кода.

Проектно-итоговый блок реализует проектный метод обучения.

Занятие 10 отводится под самостоятельное проектирование игры. На этом этапе обучающиеся не работают за компьютером. Они выполняют декомпозицию своей идеи на бумаге, составляют ментальную карту игры, прописывают логику событий и переменных, создавая тем самым формализованный план (алгоритм) будущего проекта.

Занятие 11 отводится под самостоятельную подготовку и разработку итогового игрового проекта обучающимися.

Занятие 12 включает защиту проектов, а после проводится итоговая диагностика.

Для наглядного представления структуры курса разработано детальное тематическое планирование (таблица 8).

Таблица 8

Тематическое планирование курса внеурочной деятельности

№ п/п	Тема занятия	Содержание деятельности	Формируемые компоненты алгоритмического мышления (АМ)	Форма контроля / Результат
1	Знакомство с Construct 3	Интерфейс среды Construct 3, объекты, макеты, базовые действия	-	Простейший интерактивный макет
2	Объекты и события. Создание движущегося персонажа	Понятие спрайта, покадровая анимация, привязка управления к клавишам	Выделение объектов и операций, построение линейной последовательности	Анимированный управляемый персонаж
3	Алгоритмическая конструкция «Следование»	Понятие линейного алгоритма. Цепочки событий	Выполнение алгоритма, понимание причинно-следственных связей	Мини-проект с линейной логикой взаимодействия
4	Конструкция «Ветвление». Реакция на столкновения	Условные операторы	Распознавание и реализация условных алгоритмов,	Механика взаимодействия с препятствиями

			декомпозиция условий	
5	Переменные и данные. Система очков и жизней	Глобальные и локальные переменные. Операции присваивания и изменения	Работа с абстракциями, отслеживание состояния системы	Работающая система подсчета и условий проигрыша
6	Цикл в Construct 3. Автоматизация действий	Циклические события, генерация объектов, фоновые процессы	Понимание цикличности, оптимизация повторяющихся действий	Механика бесконечного фона или генерации врагов
7	Декомпозиция задачи. Создание многоуровневой игры	Переходы между макетами, сохранение прогресса	Умение разбивать сложную задачу на подзадачи (уровни), модульность	Прототип игры с двумя и более уровнями
8	Отладка и поиск ошибок. Тестирование игры	Методы трассировки, использование отладочной консоли Construct 3	Самоконтроль, анализ и исправление логических ошибок	Стабильно работающий прототип без критических багов
9	Оптимизация алгоритма. Улучшение кода	Упрощение листов событий, удаление избыточных условий	Оценка эффективности алгоритма, выбор оптимального способа решения	Оптимизированный лист событий
10	Самостоятельный проект. Проектирование игры	Создание ментальной карты игры, словесное описание алгоритмов	Синтез знаний, абстрактное моделирование, формализация идеи до начала кодирования	Заполненный проектный лист будущей игры
11	Реализация проекта. Создание игры	Практическая сборка игры по составленному плану, тестирование механик	Применение всего комплекса алгоритмических умений в новой ситуации	Готовый игровой проект
12	Презентация проектов	Публичная защита, демонстрация игры	Рефлексия, аргументация выбранных алгоритмических решений, самооценка	Защита проекта

Платформа «КрЭШ» выступает в качестве средства электронного сопровождения очных занятий. На ней размещены краткий план урока для педагога, презентация с теоретическим и практическим материалом, а также видеоматериалы, а также интерактивные HTML-формы, созданные с применением инструментов искусственного интеллекта для проведения рефлексии и повторения материала.

Важной методической особенностью курса является категорический отказ от механического копирования готовых игр по инструкции. Обучающиеся не просто повторяют действия за учителем. На каждом уроке им демонстрируются и разбираются базовые алгоритмические конструкции и игровые механики. Видеоматериалы, входящие в комплект, предназначены в первую очередь для педагога. Они содержат подробные объяснения и разборы этих конструкций, выступая в роли методической шпаргалки и сценария занятия.

Оценка результативности курса базируется на компонентном подходе к диагностике алгоритмического мышления. Входная и итоговая диагностики проводятся с использованием идентичного тестового инструмента. Сравнение результатов тестирования позволяет количественно оценить прирост уровня развития алгоритмического мышления у обучающихся экспериментальной группы.

Таблица 9

План к уроку 3 Алгоритмическая конструкция «Следование»

Этап	Время	Деятельность учителя	Деятельность обучающихся	Формируемые компоненты АМ
1. Организационный момент	2 мин	Приветствие, проверка присутствующих	Настраиваются на работу	-
2. Актуализация	10 мин	Предлагает задание на повторение предыдущего материала в формате html (Приложение Б) Даёт задание на самостоятельное создание покадровой анимации для	Выполняют задание и самостоятельно создают покадровую анимацию в редакторе	Актуализация опыта, выполнение алгоритма

		спрайта (3-5 кадров). Напоминает: «На прошлом уроке мы работали с анимациями. Сегодня вы сами нарисуете анимацию для своего персонажа»		
3. Демонстрация	3 мин	Разбирает вместе с обучающимися понятие линейного алгоритма. Показывает примеры линейных алгоритмов	Наблюдают, задают вопросы	-
4. Практика	20 мин	Помогает индивидуально	Создают мини проект	Выполнение алгоритма
5. Рефлексия	5 мин	Обсуждает с обучающимися, что получилось, а что нет	Отвечают на вопросы. Заполняют html форму рефлексии (Приложение В)	Рефлексия

Представленный план (таблица 9) урока является прямым методическим ответом на вызовы, описанные в первом параграфе данной работы. В частности, этап актуализации и практической работы опирается на чувственную и модельную области памяти, позволяя подросткам с преобладающим визуальным восприятием через наглядную среду Construct 3 усвоить понятие линейного алгоритма [45, 46].

Логическим завершением курса является проектно-итоговый блок. В качестве примера того, какой результат могут получить обучающиеся к концу четверти, в Приложении Г представлен разработанный игровой проект в жанре «платформер».

Данная игра представляет собой не просто набор разрозненных скриптов, а целостную алгоритмическую систему, созданную самими обучающимися. Проект включает в себя несколько макетов для реализации системы уровней, стартового меню и экрана завершения игры, а также сложные, многоуровневые листы событий. В основе механики игры лежит алгоритм обработки столкновений. При успешном перепрыгивании персонажа через платформу срабатывает событие,

которое инициирует цепочку действий по начислению очков, обновлению интерфейса и сохранению прогресса.

Успешная реализация подобного платформера требует от школьника умения декомпозировать сложную задачу на подзадачи, работать с глобальными и локальными переменными и прогнозировать результат исполнения алгоритма, что полностью соответствует переходу от достаточного к оптимальному уровню сформированности алгоритмического мышления.

Разработанный курс представляет собой целостную методическую систему, в которой содержание (создание игр), формы организации (сочетание фронтальной работы и проектной деятельности) и диагностический инструментарий методически согласованы и направлены на достижение главной цели исследования.

2.2. Апробация курса и анализ результативности развития алгоритмического мышления обучающихся

Во втором параграфе второй главы представлено организационное и содержательное обеспечение опытно-экспериментальной работы, направленной на проверку результативности разработанного курса внеурочной деятельности.

Опытно-экспериментальная работа проводилась на базе Муниципального автономного общеобразовательного учреждения «Средняя школа № 152» г. Красноярска. В эксперименте приняли участие обучающиеся 8 класса. Класс был разделен на две подгруппы. В экспериментальной группа 16 обучающихся и контрольной группа 16 обучающихся.

Целью данного эксперимента являлась проверка результативности разработанного курса внеурочной деятельности по созданию игр в среде Construct 3 в развитии алгоритмического мышления обучающихся.

Экспериментальная работа проводилась в три этапа:

1. Определение исходного уровня сформированности алгоритмического мышления, у обучающихся экспериментальной и контрольной групп;

2. Частичная реализация разработанного курса внеурочной деятельности с обучающимися экспериментальной группы;

3. Повторная диагностика уровня сформированности алгоритмического мышления, сравнительный анализ полученных результатов и оценка динамики.

В рамках педагогической практики была проведена частичная апробация курса. Из 12 запланированных занятий с обучающимися экспериментальной группы было проведено 4 урока, направленных на знакомство со средой визуального программирования и базовыми алгоритмическими конструкциями.

Для оценки уровня сформированности алгоритмического мышления была использована диагностическая методика. С учетом максимального балла за диагностическую работу (28 баллов), выделены три уровня, характеристики которых представлены в таблице 10.

Таблица 10

Критерии оценки алгоритмического мышления

Уровень сформированности АМ	Диапазон баллов	Характеристика уровня
Начальный	0 - 14	Недостаточное представление об алгоритмической деятельности; низкий уровень абстракции; отсутствие опыта создания сложных алгоритмических конструкций; присутствие шаблонного характера деятельности. Обучающийся распознает простейшие алгоритмы, умеет исполнять линейные алгоритмы по образцу, но испытывает затруднения при декомпозиции задач и переходе от словесного описания к формальной записи.
Достаточный	15 - 23	Понимание значимости алгоритмической деятельности; теоретические знания алгоритмических конструкций; умение строить простые алгоритмы по образцу. Обучающийся владеет понятиями «исполнитель», «система команд», различает базовые алгоритмические конструкции, умеет классифицировать задачи по типу алгоритма и устанавливать соответствие между задачей и типом алгоритма.

Оптимальный	24 - 28	Высокий уровень абстракции; умение строить алгоритмы сложной конструкции; знание теоретических основ алгоритмизации. Обучающийся умеет разбивать задачу на подзадачи, генерировать сложные (циклические, комбинированные) алгоритмы,
-------------	---------	--

На первом этапе было проведено входное тестирование. Результаты показали, что исходный уровень сформированности алгоритмического мышления (АМ) в обеих группах практически идентичен, что подтверждает сопоставимость экспериментальной группы (ЭГ) и контрольной группы (КГ). Распределение обучающихся по уровням сформированности АМ на начало эксперимента представлен на рисунке 1.

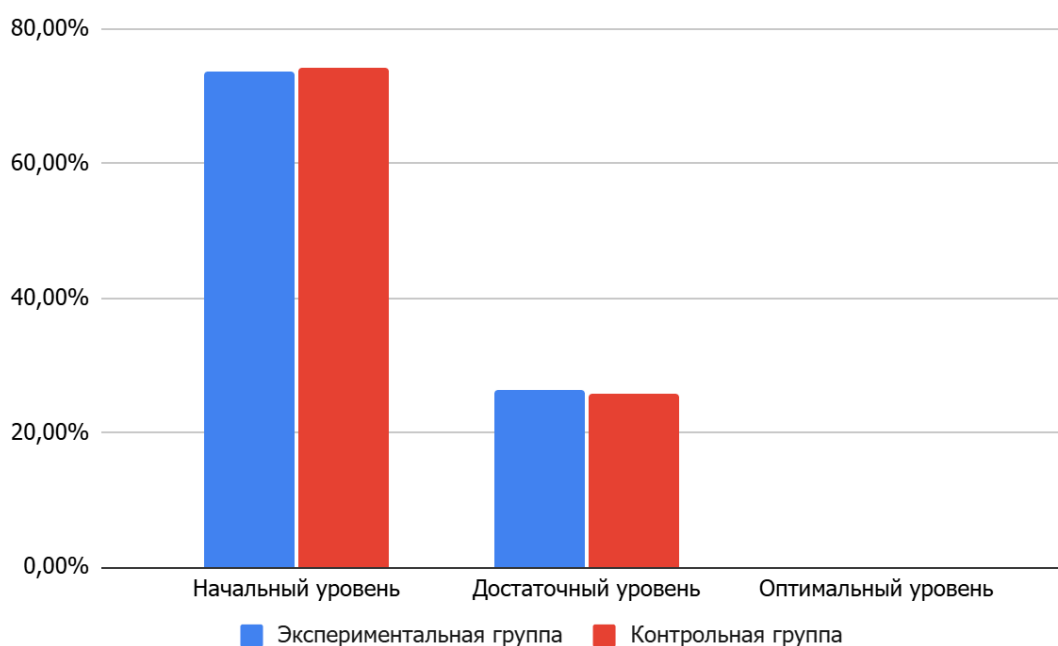


Рисунок 1 – Распределение обучающихся по уровням сформированности алгоритмического мышления (входная диагностика)

Анализ данных показывает, что в обеих группах (экспериментальной и контрольной) преобладает начальный уровень сформированности алгоритмического мышления. В экспериментальной группе – 73,6% обучающихся, в контрольной группе – 74,2%.

Достаточный уровень демонстрируют 26,4% школьников в экспериментальной группе и 25,8% – в контрольной группе. Оптимальный уровень сформированности алгоритмического мышления не зафиксирован ни в одной из групп (0%).

Средний балл по результатам диагностики составил 12,65, что свидетельствует о сопоставимости групп на начало эксперимента.

После проведения 4 занятий курса внеурочной деятельности в экспериментальной группе было проведено итоговое тестирование с использованием аналогичных диагностических материалов. Результаты контрольного этапа продемонстрировали следующую динамику (рис.2).

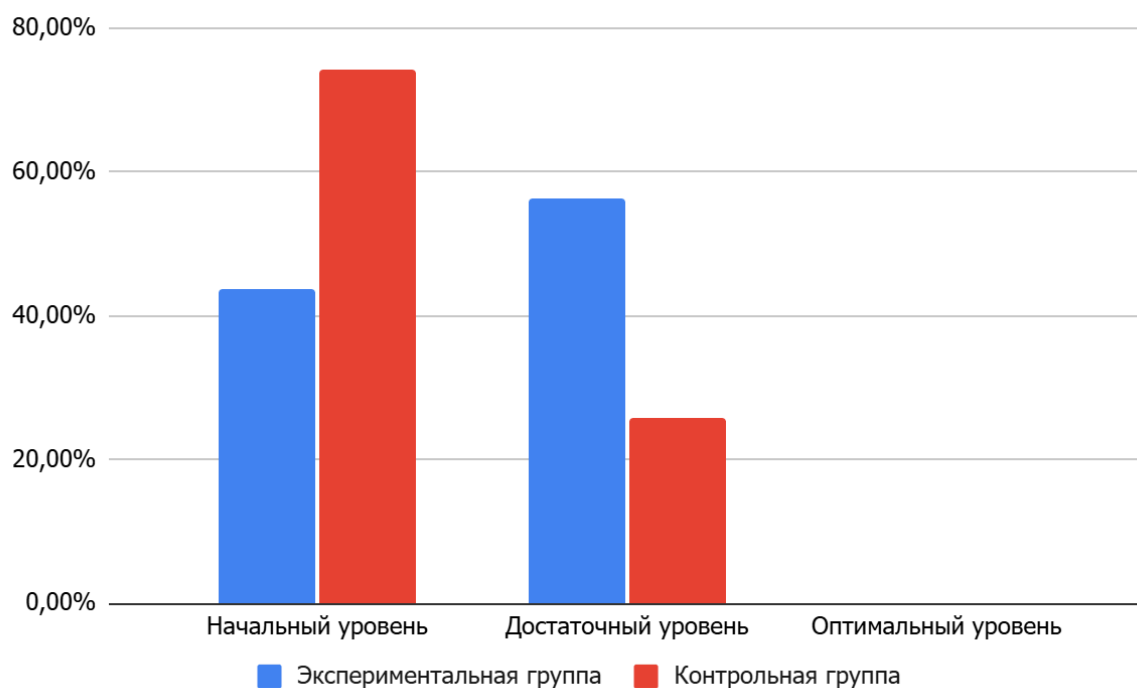


Рисунок 2 – Сравнительная динамика уровней сформированности АМ после частичной апробации

Средний балл по экспериментальной группе вырос с 12,66 до 15,17 баллов, что составляет прирост в 19,8%. Средний балл по контрольной группе изменился незначительно: с 12,66 до 12,77 баллов.

Полученные данные в ходе систематического анализа показывают наличие положительной, статистически значимой динамики в экспериментальной группе.

Переход 5 обучающихся с низкого уровня на средний свидетельствует о том, что даже краткосрочное погружение в среду визуального программирования (Construct 3) способствует структурированию мышления.

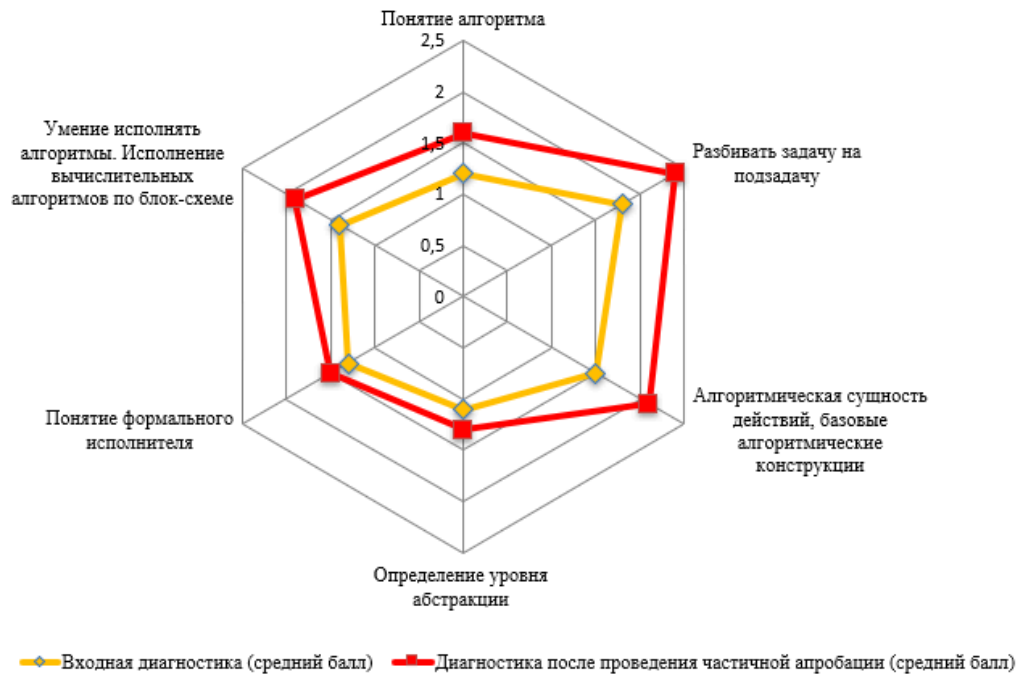


Рисунок 3 – Динамика развития алгоритмического мышления обучающихся экспериментальной группы по индикаторам диагностики (до и после частичной апробации курса)

Анализ результатов повторной диагностики в экспериментальной группе после проведения четырех занятий курса выявил выраженную положительную динамику по ключевым индикаторам сформированности алгоритмического мышления (рис.3). Наиболее значимый прирост зафиксирован в индикаторах «Разбивать задачу на подзадачу» и «Алгоритмическая сущность действий, базовые алгоритмические конструкции». Это напрямую обусловлено дидактической спецификой первых четырех уроков, в ходе которых обучающиеся осваивали визуальную среду Construct 3, работали с объектами и событиями, а также конструировали линейные алгоритмы (конструкция «Следование») и алгоритмы с ветвлением. Среда визуального программирования, лишенная

синтаксического барьера, позволила школьникам сосредоточиться именно на логике событийного программирования, что эффективно стимулировало развитие навыков декомпозиции игровых механик и понимание базовых алгоритмических структур.

Умеренная, но стабильная положительная динамика наблюдается также в индикаторах «Понятие алгоритма» и «Умение исполнять алгоритмы», тогда как показатели «Уровень абстракции» и «Понятие формального исполнителя» выросли незначительно, что вполне закономерно, так как для их глубокого формирования требуется изучение циклических структур, переменных и многоуровневых макетов на последующих этапах курса.

Отсутствие существенной динамики в контрольной группе (прирост менее 1%) подтверждает, что развитие алгоритмического мышления не происходит стихийно или исключительно за счет базовых уроков информатики, и требует целенаправленной педагогической поддержки через специально организованную внеурочную деятельность.

Выводы к главе 2

В первом параграфе второй главы было осуществлено проектирование и разработка курса внеурочной деятельности по созданию игр в среде визуального программирования Construct 3 с электронным сопровождением на платформе «КрЭШ». Обоснован дидактический потенциал выбранной среды для наглядной визуализации базовых алгоритмических конструкций (следование, ветвление, цикл) и их интеграции с образовательными задачами.

Описаны организация и результаты частичной апробации разработанного курса в условиях Муниципального автономного общеобразовательного учреждения «Средняя школа № 152» г. Красноярска. Сравнительный анализ результатов входной и итоговой диагностики продемонстрировал статистически значимую положительную динамику в экспериментальной группе.

ЗАКЛЮЧЕНИЕ

На основе теоретического анализа психолого-педагогической и методической литературы было охарактеризовано понятие алгоритмического мышления как ключевого когнитивного навыка. Выявлены особенности его формирования у подростков 7-8 классов и обоснована необходимость использования качественных, компонентных подходов к его диагностике, избегающих распространенной ошибки отождествления алгоритмического мышления с предметными знаниями по программированию.

Проведенный анализ нормативно-правового обеспечения и существующих программ внеурочной деятельности позволил выявить дефицит системных методических решений, направленных на целенаправленное развитие алгоритмического мышления. Сравнительный анализ сред визуального программирования (Scratch, Construct 3, GDevelop) обосновал выбор платформы Construct 3 ввиду ее событийной модели, наиболее точно визуализирующей базовые алгоритмические конструкции.

Результативность разработанного курса была проверена в ходе опытно-экспериментальной работы. Экспериментальной базой являлось Муниципальное автономное общеобразовательное учреждение «Средняя школа № 152» г. Красноярска. В исследовании приняли участие обучающиеся 8 класса, разделенные на экспериментальную и контрольную группы (по 16 человек).

Таким образом, все поставленные задачи решены, цель исследования достигнута.

Перспективой дальнейшего исследования может стать полная апробация всех 12 занятий разработанного курса, а также адаптация его содержательных и диагностических компонентов для обучающихся старшей ступени школы и в рамках других предметных областей.

Практическая ценность данной работы состоит в том, что разработанный курс могут быть использованы педагогами в реальном процессе организации

внеурочной деятельности и дополнительного образования школьников 7-8 классов.

Библиографический список

1. Абульханова-Славская К. А. Стратегия жизни Москва : Жизнь, 1991. 299 с.
2. Андреева Д. Д. Методические подходы для развития алгоритмического стиля мышления школьников на уроках информатики // Современное профессиональное образование. 2025. №2. С. 172-176
3. Белицкая Г. Э. Типология проблемности социального мышления личности / Ин-т психологии. Москва, 1991. 18 с.
4. Босова Л. Л. Информатика: учебник для 5 класса Москва: БИНОМ. Лаборатория знаний, 2013. 213 с.
5. Бражникова А. Н. Особенности процесса обучения программированию в основной школе // Молодежь в мире науки: Материалы XII открытой региональной студенческой научно-практической конференции, Сургут, 29 ноября 2024 года. Сургут: РИО БУ "Сургутский государственный педагогический университет", 2025. С. 98-99.
6. Брушлинский А. В. Субъект: мышление, учение, воображение / Москва: Институт практической психологии ; Воронеж : НПО «Модэк», 1996. 392 с.
7. Быкова К. И. Влияние изучения языка программирования python на развитие алгоритмического мышления // Проблемы, тенденции и последствия цифровизации общества и личности: междисциплинарные исследования: материалы конференции, Белгород, 18–19 мая 2023 года. Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, 2023.
8. Васюта И. В. Использование приемов развития критического мышления на уроках информатики /Москва: Издательский центр «Академия», 2007. 309 с.
9. Волченкова А. А., Новикова А.В. Клиповое мышление подростков // Человеческий фактор: Социальный психолог. 2024. № 1(49). С. 177-182.

10. Гаврилова И. В. Критерии диагностики алгоритмического мышления // Информатизация непрерывного образования: материалы Международной научной конференции, Москва, 2018. С. 29-33.
11. Гурьевских М. Е. Разработка системы занятий для обучения программированию школьников с использованием среды разработки Scratch // Наука и перспективы. 2019. № 4. С. 31-41.
12. Гусейнова Г.Э. От Scratch до Godot: обзор доступных платформ для создания игр школьниками // Научное сообщество студентов: междисциплинарные науки: материалы III Всероссийской молодежной научно-практической конференции. Орёл. 2026. С. 100-104.
13. Дарски Н. П., Лысенко А. П. Развитие алгоритмического мышления обучающихся 5-7 классов средствами визуального программирования // Путь к новому пониманию: интеграция междисциплинарных исследований. 2026. С. 74.
14. Иванова С. Г. Развитие алгоритмического мышления с помощью среды программирования trik studio junior для учащихся 5 классов // Аммосов С. 325-330.
15. Игровой движок "Construct 3" на уроках Информатики [Электронный ресурс] URL: <https://stepik.org/course/191008/syllabus> (дата обращения: 25.05.2026)
16. Жемчужников Д. Г. Методика обучения программированию, основанная на создании школьниками динамических компьютерных игр: автореф. дис. канд. пед. наук. Москва, 2013. 24 с.
17. Караванский А. Н. Применение редактора Construct в процессе разработки обучающих игр // XXIV Вишняковские чтения «Вузовская наука: условия эффективности социально-экономического и культурного развития региона»: материалы Междунар. науч. конф., Бокситогорск. 2021. С. 256-260.
18. Колмогоров А. Н. Математическая логика / КомКнига, 2006. 240 с.

19. Корнилов Ю. К. Особенности педагогического мышления как вида мышления практического // Психология профессионального педагогического мышления. Москва: Просвещение. 2003. 257 с.
20. Кукушкина Н. А. Проектирование мобильных приложений в среде MIT App Inventor как средство формирования познавательного интереса у школьников при обучении программированию // Сборник статей по итогам Всероссийской научно-практической конференции «Инфо-Стратегия 2016: Общество. Государство. Образование». Самара, 2016. С. 48-54.
21. Кузнецова Е.В., Резер Т.М. Клиповое мышление как фактор изменения образовательных технологий в высшей школе // Акмеология профессионального образования: Мат-лы 14-й Всероссийской научно-практической конференции, 14–15 марта 2018 г., Екатеринбург. Екатеринбург: Изд-во РГППУ, 2018. С. 367–370.
22. Лебедева Т. Н. Формирование алгоритмического мышления школьников в процессе обучения рекурсивным алгоритмам в профильных классах средней общеобразовательной школы : автореф. Москва, 2005. 24 с.
23. Макаренко Ю. А., Столяр А.А. Что такое алгоритм?: беседы со старшеклассником / Минск : Нар. асвета, 1989. 127 с.
24. Малова А. И. Использование визуальной среды App Inventor при обучении программированию школьников // XIX Всероссийская научно-практическая конференция молодых исследователей образования.. Томск, 2020. С. 414-417.
25. Марков А. А. Теория алгоритмов / Москва: Наука, 1984. 432 с.
26. Мелехин В. Ф. Вычислительные машины: учебник для студ. учреждений высш. проф. образования / Москва: Издательский центр «Академия». 2013. 384 с.
27. Опрос всероссийского центра изучения общественного мнения. [Электронный ресурс]. URL: <http://wciom.ru/index.php?id=236&uid=115294> (дата обращения: 10.06.2026)

28. Основы логики и алгоритмики: рабочая программа курса внеурочной деятельности (для 1-4 классов образовательных организаций). [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
29. Основы программирования: примерная рабочая программа курса внеурочной деятельности (для 5-6 классов образовательных организаций). [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
30. Основы программирования на Python: примерная рабочая программа курса внеурочной деятельности (для 7-9 классов образовательных организаций). [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
31. Пак Н. И. О модели мышления и ментальных схемах // Практико-ориентированное обучение в профессиональном образовании: проблемы и пути развития: материалы научно-практ. конф. «Решетневские чтения». Красноярск: СибГАУ, 2014. С. 306-310.
32. Платонов К. К. Психология: [учеб. пособие для повышения квалификации инж.-пед. работников] / Москва: Высш. школа, 1977. 247 с.
33. Пономарев Я. А. Знания, мышление и умственное развитие / Москва: Педагогика, 1967.
34. Попова А. В., Тюменева Ю. А. Новый метод оценки алгоритмического мышления на непрограммистских задачах: разведывательное исследование // Мир психологии. 2024. № 2 (117). С. 268–285.
35. Старицына О.А. Клиповое мышление vs образование. Кто виноват и что делать? // Азимут научных исследований: педагогика и психология. 2018. Т. 7. № 2(23). С. 270–274.
36. Степанова Т. А. Развитие алгоритмического стиля мышления при обучении программированию в вузе // Теоретические и практические аспекты

- психологии и педагогики: коллективная монография. Уфа: Аэтерна, 2015. С. 101-118.
37. Степанова Т. А. Теория алгоритмического мышления: учебное пособие для магистрантов, учителей общеобразоват. учреждений, преподавателей вузов / Красноярск : Краснояр. гос. пед. ун-т им. В.П. Астафьева, 2014. 72 с.
38. Стёпушкина П. К. Игровые движки для образовательных целей // Вестник науки. 2024. С. 851-854.
39. Федеральная рабочая программа по учебному предмету «Информатика» базовый уровень. [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
40. Федеральная рабочая программа по учебному предмету «Информатика» углубленный уровень. [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
41. Федеральный государственный образовательный стандарт основного общего образования : утвержден приказом Министерства образования и науки Российской Федерации от 17.12.2010 г. № 1897 (ред. от 08.12.2022)
42. Функциональная грамотность: учимся для жизни: программа курса внеурочной деятельности (основное общее образование). [Электронный ресурс]. URL: <https://edsoo.ru/rabochie-programmy> (дата обращения: 10.06.2026)
43. Шелевер Л. В. Construct 2 создание приложений с нуля// International scientific review. 2020. № LXVI. С. 92-94.
44. Construct 3 [Электронный ресурс] URL: <https://www.construct.net/> (дата обращения: 10.06.2026).
45. de Groot T. Learning in and about a filtered universe: young people's awareness and control of algorithms in social media // Learning, Media and Technology. 2023. Vol. 48. No. 4. P. 701-713.
46. Zega D. D. I. N., Ginting W. W., Damanik R. R. Interactive Educational Game for Cognitive and Language Development of Early Childhood Based on Android

Using Construct 3 // International Multidisciplinary Journal. 2025. Vol. 1. No. 01. P. 90-96.

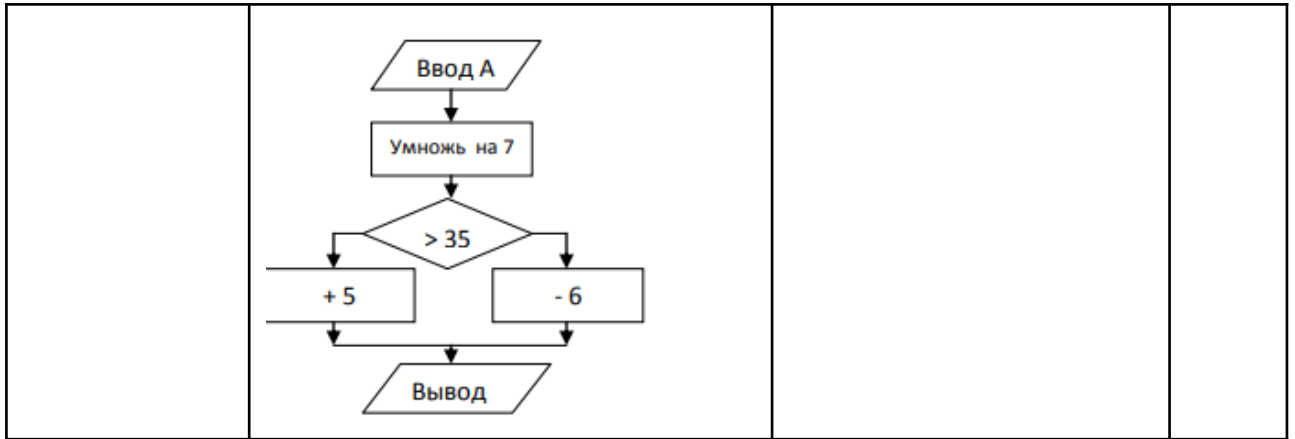
ПРИЛОЖЕНИЕ А

Таблица диагностики уровня сформированности алгоритмического мышления


Индикатор	Задания	Правильные ответы	Баллы
Понятие алгоритма	№1 Из предложенного перечня выбрать алгоритмы: 1. Рецепт пирога 2. Инструктаж учителя о том, как правильно переходить дорогу 3. Советы бабушки по уборке квартиры 4. Правило умножения дробей 5. Сценарий кинофильма	1, 2, 4	2
Разбивать задачу на подзадачу, определение количества линейных алгоритмов, обеспечивающих их решение задачи	№2 Прочитайте задачу. Из скольких этапов, по-вашему, будет состоять ее решение? а) Найти периметр нарисованного прямоугольника б) В поезде вам необходимо запарить лапшу быстрого приготовления	а) 4 этапа 1. Измерить длину 2. Измерить ширину 3. Сложить длину и ширину 4. Умножить на 2 б) 5-6 этапов 1. Открыть упаковку 2. Вынуть приправу 3. Залить кипятком 4. Накрыть крышкой 5. Подождать 3-5 мин 6. Перемешать	4
	№3 В систему команд исполнителя Водoley входит 6 команд: 1- Наполнить сосуд А 2- Наполнить сосуд В 3- Перелить из А в В 4- Перелить из В в А 5- Вылить из А 6- Вылить из В Объем сосуда А=4 литра, сосуда В=3 литра. Составьте инструкцию для Водoley, с помощью которой в одном из сосудов окажется ровно 2 литра. В ответе запишите номера команд. Найдите самый быстрый способ	1-3-6-4-1-3 Пошаговое решение: Наполнить А (А=4, В=0) Перелить из А в В (А=1, В=3) Вылить из В (А=1, В=0) Перелить из А в В (А=0, В=1) Наполнить А (А=4, В=1) Перелить из А в В (А=2, В=3)	6

<p>Алгоритмическая сущность действий, базовые алгоритмические конструкции, установление соответствия между задачей и типом используемого алгоритма</p>	<p>№4 Распределите задачи на группы, используя таблицу: Покраска забора, решение линейного уравнения, определение стоимости покупки, определение необходимости оплатить услуги интернета, выбор одежды в зависимости от погоды, приготовление овощного салата, пришивание пуговиц</p> <table border="1" data-bbox="464 602 967 994"> <tr> <td data-bbox="464 602 632 927">Последовательное выполнение шагов</td> <td data-bbox="632 602 799 927">Выполнение шагов, предполагает проверку условия</td> <td data-bbox="799 602 967 927">Решение задачи предполагает повторение некоторых шагов</td> </tr> <tr> <td data-bbox="464 927 632 994"></td> <td data-bbox="632 927 799 994"></td> <td data-bbox="799 927 967 994"></td> </tr> </table>	Последовательное выполнение шагов	Выполнение шагов, предполагает проверку условия	Решение задачи предполагает повторение некоторых шагов				<p>Последовательное выполнение шагов: - Покраска забора - Решение линейного уравнения - Определение стоимости покупки - Пришивание пуговицы</p> <p>Выполнение шагов предполагает проверку условия: - Определение необходимости оплатить интернет - Выбор одежды в зависимости от погоды</p> <p>Решение предполагает повторение некоторых шагов: - Приготовление овощного салата (нарезка овощей)</p>	2
Последовательное выполнение шагов	Выполнение шагов, предполагает проверку условия	Решение задачи предполагает повторение некоторых шагов							
<p>Определение уровня абстракции через: решение задачи в общем виде, применение алгоритма для решения простых жизненных задач, установление причинно следственных связей</p>	<p>№5 Прочитайте предлагаемые задачи. В некоторых из них используются искусственные слова, бессмысленные буквосочетания они заменяют наши обычные слова. Вы можете в уме подставлять вместо них понятные вам обычные слова. В ответе надо указать одно слово.</p> <ol style="list-style-type: none"> 1. Толя веселее, чем Коля. Коля веселее, чем Алик. Кто веселее всех? 2. Миша темнее, чем Коля. Миша светлее, чем Вова. Кто темнее всех? 3. Всиг слабее, чем Рпти. Всиг сильнее, чем Гшде. Кто слабее всех? 4. Вшфп клмн, чем Дтсв. Дтсв клмн, чем Пнчб. Кто клмн всех? 5. Лошадь ниже, чем муха. Лошадь выше, чем жираф. Кто выше всех? 	<ol style="list-style-type: none"> 1. Толя (Толя > Коля > Алик) 2. Вова (Коля < Миша < Вова) 3. Гшде (Гшде < Всиг < Рпти) 4. Вшфп (Пнчб < Дтсв < Вшфп) 5. Жираф (Жираф < Лошадь < Муха) 	6						

<p>Понятие формального исполнителя</p>	<p>№6 Какой из следующих исполнителей не является формальным: 1) автомат на конвейере, наполняющий бутылки лимонадом; 2) компьютер, выполняющий проверку правописания 3) мама, готовящая салат по рецепту из книги 4) фармацевт, готовящий лекарство по рецепту</p>	3,4	1																																																																
<p>Умение исполнять алгоритмы. Исполнение вычислительных алгоритмов по блок-схеме</p>	<p>№7 Отгадайте пословицу, обойдя поле ходом шахматного коня. Начальная клетка – верхняя левая клетка с буквой «Н»</p> <table border="1" data-bbox="475 824 852 992"> <tr><td>Н</td><td></td><td></td><td></td><td>А</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td>Е</td><td></td><td></td><td></td><td></td><td>Ш</td></tr> <tr><td>И</td><td></td><td></td><td></td><td>Л</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td>Г</td><td></td><td></td><td></td><td>Ь</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>Е</td><td></td><td></td><td>И</td></tr> <tr><td></td><td></td><td>Т</td><td></td><td></td><td>К</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>А</td><td></td><td></td><td></td><td>У</td></tr> <tr><td></td><td></td><td></td><td></td><td>Р</td><td></td><td></td><td></td></tr> </table>	Н				А						Е					Ш	И				Л						Г				Ь						Е			И			Т			К						А				У					Р				Не игла шьет, а руки	2
	Н				А																																																														
			Е					Ш																																																											
И				Л																																																															
		Г				Ь																																																													
				Е			И																																																												
		Т			К																																																														
			А				У																																																												
				Р																																																															
<p>№8 Исполнитель умеет выполнять следующие команды: 1-заменять каждую букву в слове на следующую за ней по алфавиту, 2-записывать слово наоборот. Что получится из слова «МИР», если к нему применить следующую инструкцию – 12112</p>	<p>ПИЙУ</p> <p>Пошаговое решение: Исходное: МИР Шаг 1 (замена): НЙС Шаг 2 (наоборот): СЙН Шаг 1 (замена): ТЙО Шаг 1 (замена): УЙП Шаг 2 (наоборот): ПИЙУ</p>	2																																																																	
<p>№9 Найти значение переменной x, используя блок схему. Заполните таблицу</p> <table border="1" data-bbox="464 1538 956 1671"> <tr><td>А</td><td>0</td><td>3</td><td>6</td></tr> <tr><td>Х</td><td></td><td></td><td></td></tr> </table>	А	0	3	6	Х				<p>А = 0: $0 \times 7 = 0$ $0 > 35?$ НЕТ $0 - 6 = -6$</p> <p>А = 3: $3 \times 7 = 21$ $21 > 35?$ НЕТ $21 - 6 = 15$</p> <p>А = 6: $6 \times 7 = 42$ $42 > 35?$ ДА $42 + 5 = 47$</p> <p>Ответ: X = -6, 15, 47</p>	3																																																									
А	0	3	6																																																																
Х																																																																			




Интерактивное задание





Анимация спрайтов


Интерактивные задания


 **Уровень 1: Алгоритм создания анимации**


Задание: Расставьте шаги в правильном порядке. Перетащите элементы мышью или используйте кнопки ↑ ↓


?
Создать спрайт


?
Открыть редактор спрайта



?
Добавить кадры (нарисовать или загрузить)


?
Установить скорость анимации


?
Включить за цикливание



?
Проверить в режиме просмотра


✓ Проверить порядок
🔄 Сбросить
➔ Следующий уровень →



Анимация спрайтов

Интерактивные задания

 **Уровень 2: Объект-Событие-Действие**


Задание: Заполните таблицу, выбрав правильные примеры из Construct 3

№	Объект	Событие	Действие
1	Выберите объект... ▾	Выберите событие... ▾	Выберите действие... ▾
2	Выберите объект... ▾	Выберите событие... ▾	Выберите действие... ▾
3	Выберите объект... ▾	Выберите событие... ▾	Выберите действие... ▾
4	Выберите объект... ▾	Выберите событие... ▾	Выберите действие... ▾



✓ Проверить таблицу
🔄 Сбросить
💡 Показать ответы

🏠 На главную








Карта рефлексии «Три кадра моего урока»

 **Три кадра моего урока**




1 Кадр 1: Начало урока (моё настроение)

 Скучал	 Обычное настроение	Было интересно
---	---	----------------

2 Кадр 2: Что я делал на уроке (отметь всё, что делал)

 Рисовал кадры анимации	Настраивал скорость анимации	 Включал за цикливание
 Программировал клавиши управления	 Создавал анимацию «Стоп»	 Создавал анимацию «Бег»
 Создавал анимацию «Прыжок»	 Создавал мини-игру	

3 Кадр 3: Мой результат

 Готовая мини-игра!	 Почти готово, но есть ошибки	 Не успел закончить
---	---	---

Вопросы для размышления

1. Что такое линейный алгоритм? (выбери правильный ответ)

Алгоритм с условиями «если/иначе»

Последовательность действий, выполняемых строго по порядку

Алгоритм, который повторяется бесконечно

2. Приведи пример линейного алгоритма из своей игры в Construct 3

Например: «Когда игрок касается монетки:»

1. Добавить 10 к счёту

2. Воспроизвести звук «Дзынь»

3. Уничтожить монетку

Напиши свой пример линейного алгоритма...

3. Какое событие в Construct 3 ты запомнил лучше всего?

При нажатии клавиши →

При нажатии клавиши ←

При нажатии клавиши ↑

При остановке (Платформер)

При столкновении с монеткой

4. Что было самым интересным на уроке?

Напиши свой ответ...

5. Чему я научился сегодня? (закончи предложения)

Я научился создавать...

Я понял, как работает...

Я смог запрограммировать...

6. Оцени свою работу на уроке:



Показать мою рефлексию

Начать заново

Образец игрового проекта обучающегося



Список событий 2 x

1	→ Тач	При тапу по объекту Кнопка	Система	Перейти к макету "Game"
			Система	Установить значения глобальных переменных на начальные (включая статических: false)
				Добавить действие
2	Результ...	Результат < Очки	Результ...	Установить текст на Очки
				Добавить действие

Список событий 1 x

	Глобальная число	Прокрутка = 1000		
	Глобальная число	НоваяПлатформа = 0		
	Глобальная число	СиняяПлатформа = 0		
	Глобальная число	КраснаяПлатформа = 0		
	Глобальная число	Очки = 0		
1	Монстр	Платформер на полу	Монстр	Имитировать Платформер нажатие Прыжок
				Добавить действие
2	Монстр	Y < Прокрутка	Система	Установить Прокрутка значение Монстр.Y
				Добавить действие
3	Система	Каждый тик	Система	Прокрутить до (160, Прокрутка)
				Добавить действие
4	Система	НоваяПлатформа ≥ ВерхОбластиПросмотра("Игра")	Система	Создать объект Платформа на слое "Игра" в (Случайный(20, ШиринаМакета - 40), ВерхОбластиПросмотра("Игра")), создать иерархию: false, шаблон: ""
			Система	Установить НоваяПлатформа значение ВерхОбластиПросмотра("Игра") - Случайный(10,100)
			Система	Добавить 1 в Очки
				Добавить действие
5	Система	КраснаяПлатформа ≥ ВерхОбластиПросмотра("Игра")	Система	Создать объект КраснаяПлатформа на слое "Игра" в (Случайный(20, ШиринаМакета - 40), ВерхОбластиПросмотра("Игра")), создать иерархию: false, шаблон: ""
			Система	Установить КраснаяПлатформа значение ВерхОбластиПросмотра("Игра") - Случайный(50,500)
				Добавить действие
6	Система	СиняяПлатформа ≥ ВерхОбластиПросмотра("Игра")	Система	Создать объект СиняяПлатформа на слое "Игра" в (Случайный(20, ШиринаМакета - 40), ВерхОбластиПросмотра("Игра")), создать иерархию: false, шаблон: ""
			Система	Установить СиняяПлатформа значение ВерхОбластиПросмотра("Игра") - Случайный(30,1000)
			Система	Добавить 1 в Очки
				Добавить действие
7	Платформа	На экране	Платформа	Уничтожить
	Платформа	Y > НижняяОбластиПросмотра("Игра")		Добавить действие
8	КраснаяПл...	На экране	КраснаяПл...	Уничтожить
	КраснаяПл...	Y > НижняяОбластиПросмотра("Игра")		Добавить действие
9	СиняяПлат...	На экране	СиняяПлат...	Уничтожить
	СиняяПлат...	Y > НижняяОбластиПросмотра("Игра")		Добавить действие
10	Монстр	При столкновении с КраснаяПлатформа	КраснаяПл...	Уничтожить
				Добавить действие
11	Монстр	На экране	Система	Перейти к макету "Меню"
	Монстр	Y > НижняяОбластиПросмотра("Игра")		Добавить действие
12	Очки	Очки = 0	Очки	Установить текст на Очки
				Добавить действие