МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ПЕДАГОГИКИ, ПСИХОЛОГИИ
И СОЦИАЛЬНЫХ ТЕХНОЛОГИЙ

КАФЕДРА ТЕОРИИ И МЕТОДИКИ ТЕХНОЛОГИЧЕСКОГО
И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ВКР УдГУ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(ДИПЛОМНЫЙ ПРОЕКТ)

Тема: **Развитие алгоритмического мышления у учащихся восьмых классов на уроках информатики**

Выполнил:

студент группы ОБ-44.03.05.01-55 Свинин Тимофей Алексеевич

_____
подпись

Руководитель:

старший преподаватель каф. ТМТПО Вахрушев Андрей Владимирович

_____
подпись

Заведующий кафедрой ТМТПО:

кандидат педагогических наук, доцент Причинин Алексей Евгеньевич

_____
подпись

Ижевск 2026

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ПЕДАГОГИКИ, ПСИХОЛОГИИ
И СОЦИАЛЬНЫХ ТЕХНОЛОГИЙ

КАФЕДРА ТЕОРИИ И МЕТОДИКИ ТЕХНОЛОГИЧЕСКОГО
И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ВКР УдГУ

ЗАДАНИЕ

студенту Свинину Тимофею Алексеевичу

на выполнение выпускной квалификационной работы (дипломного проекта) по направлению 44.03.05 «Педагогическое образование (с двумя профилями подготовки)», профиль Технология и Информатика

Тема выпускной
квалификационной работы
(дипломного проекта)
рассмотрена и утверждена
на заседании кафедры ТМТПО
«14» мая 2026 г.
Протокол № 9

Зав. кафедрой_____/ Причинин А.Е.

Ижевск 2026

1. Тема выпускной квалификационной работы (дипломного проекта):

Развитие алгоритмического мышления у учащихся восьмых классов на уроках информатики

2. Состав работы (проекта):
   - 2.1. Пояснительная записка (96 стр.)
   - 2.2. Материальное средство обучения
   - 2.3. Электронное средство обучения
   - 2.4. Презентация.
3. Содержание работы (проекта)
   - 3.1. Информационно-аналитическое исследование.
   - 3.2. Педагогическая технология, способствующую развитию алгоритмического мышления у учащихся восьмых классов на уроках информатики.
   - 3.3. Комплексное средство обучения, способствующее развитию алгоритмического мышления.
   - 3.4. Опытно-экспериментальная работа.
4. Сроки выполнения и защиты:
   - 4.1. Предварительная защита 9 июня 2026 года
   - 4.2. Защита на ГЭК 29 июня 2026 года
5. Руководитель проекта:

Вахрушев Андрей Владимирович, старший преподаватель каф. ТМТПО

_____

ВКР УдГУ

РЕФЕРАТ

Выпускная квалификационная работа посвящена проблеме развития алгоритмического мышления у учащихся 8-х классов на уроках информатики при изучении модуля «Алгоритмы и программирование».

Пояснительная записка содержит 95 страниц, введение, три главы, источники информации состоит из 52 источников, заключения и 4 приложений.

Во введении выявлена актуальность выбранной темы, обусловленная необходимостью формирования у школьников алгоритмического мышления как важного метапредметного результата обучения, закреплённого требованиями Федерального государственного образовательного стандарта основного общего образования. Определены проблема, субъект, объект и предмет исследования. Сформированы цель, гипотеза и задачи.

В первой главе проведен анализ психолого-педагогической и методической литературы по проблеме алгоритмического мышления, рассмотрены современные педагогические инструменты и средства его развития, изучены возрастные особенности учащихся среднего подросткового возраста.

Во второй главе выполнена модернизация содержания модуля «Алгоритмы и программирование» по программе «Информатике» 8 класса, разработана педагогическая технология, основанная на интеграции игровой иммерсивной среды Minecraft и программирования на языке Python. Разработаны программное средство обучения – образовательная карта «Алгоритмический мир» в среде Minecraft[48], дидактические шашки, а также комплект дидактических заданий.

В третьей главе приведены результаты педагогического эксперимента и описано проведение опытно-экспериментальной работы. Проведен анализ результатов исследования.

В заключении подведены итоги всего проекта.

ВКР УлГУ

КЛЮЧЕВЫЕ СЛОВА

- Алгоритмическое мышление
- Алгоритмизация
- Программирование
- Педагогическая технология
- Деятельностный подход
- Игровая форма обучения
- Иммерсивная образовательная среда
- Визуальное программирование
- Язык программирования Python
- Эксперимент

ВКР УдГУ

- **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ.....7

1. ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЕ ИССЛЕДОВАНИЕ ПО ПРОБЛЕМЕ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ У ШКОЛЬНИКОВ НА УРОКАХ ИНФОРМАТИКИ.....13

1.1 Сущность и структура понятия «алгоритмическое мышление» в психолого-педагогической литературе.....13

1.2 Педагогические инструменты и средства развития алгоритмического мышления.....17

1.3 Психолого-педагогическая характеристика учащихся 8-х классов как основа развития алгоритмического мышления.....22

2. РАЗРАБОТКА ПЕДАГОГИЧЕСКОЙ ТЕХНОЛОГОГИИ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ У УЧАЩИХСЯ 8-Х КЛАССОВ НА УРОКАХ ИНФОРМАТИКИ.....30

2.1 Модернизация учебной программы по модулю «Алгоритмы и программирование» для 8-го класса.....30

2.2 Разработка педагогической технологии.....35

2.3 Разработка средств обучения.....41

Выводы по главе 2.....50

3. ОПЫТНО-ЭКСПЕРИМЕНТАЛЬНАЯ РАБОТА.....52

3.1 Методика проведения опытно-экспериментальной работы.....52

3.2 Проведение опытно-экспериментальной работы.....59

3.3 Анализ результатов опытно-экспериментальной работы.....64

Выводы по главе 3.....71

ЗАКЛЮЧЕНИЕ.....73

ИСТОЧНИКИ ИНФОРМАЦИИ.....76

ПРИЛОЖЕНИЕ А Фрагмент учебной программы по предмету «Информатика», модуль «Алгоритмы и программирование» (8 класс)..82

ПРИЛОЖЕНИЕ Б Сборник алгоритмических задач.....85

ПРИЛОЖЕНИЕ В Диагностический материал для определения уровня алгоритмического мышления учащихся 8-х классов.....89

ПРИЛОЖЕНИЕ Г Фотоматериалы с занятий.....93

ВКР УлГУ

ВВЕДЕНИЕ

В современном мире, характеризующемся стремительной цифровизацией всех сфер жизни, способность к алгоритмическому мышлению становится одной из ключевых компетенций, не менее важной, чем традиционная грамотность или умение считать. Цифровая трансформация экономики и общества, обозначенная в национальном проекте «Цифровая экономика Российской Федерации»[20], формирует устойчивый запрос на специалистов, способных не просто использовать готовые программные продукты, но и мыслить логически, структурировать задачи, разрабатывать эффективные последовательности действий для их решения. Аналитики ведущих рекрутинговых платформ, таких как HeadHunter, регулярно отмечают в своих обзорах (например, «Тренды на рынке труда 2024»)[47], что среди soft skills работодатели все чаще выделяют системное и логическое мышление, умение решать сложные задачи и когнитивную гибкость, что является прямой проекцией сформированного алгоритмического мышления. По данным Stack Overflow Developer Survey 2024[50], знание алгоритмов и структур данных входит в топ-3 навыков, которые работодатели проверяют на технических собеседованиях. В материалах Яндекса[51] отмечается, что алгоритмические секции на собеседованиях позволяют оценить фундаментальные знания (умение применять стандартные алгоритмы, писать чистый код, выбирать подходящие структуры данных) и инженерное мышление (аналитический подход к декомпозиции проблем, внимательность к деталям, способность чётко объяснять ход мыслей).

Это свидетельствует о том, что формирование данного качества перестает быть узкопредметной задачей курса информатики и становится общеобразовательным приоритетом.

Требования к метапредметным результатам, закрепленные в Федеральном государственном образовательном стандарте основного общего образования (ФГОС ООО), напрямую коррелируют с развитием

ВКР УдГУ

алгоритмического мышления. В частности, пункт 10 ФГОС ООО [15] гласит, что учащийся должен «овладеть основами логического и алгоритмического мышления… при решении учебных, познавательных и учебно-практических задач». Более конкретно эти требования раскрываются в Федеральной рабочей программе основного общего образования по информатике[16], где в рамках модуля «Алгоритмы и программирование» для 7-9 классов прямо указана необходимость формирования «умений составлять и отлаживать программы для исполнителей, анализировать условия задач, осуществлять декомпозицию задачи на подзадачи». Таким образом, развитие алгоритмического мышления является не просто педагогическим пожеланием, а законодательно закреплённой необходимостью, ответом на вызовы времени.

Как подчеркивает в своих работах А.В. Хуторской[31], в условиях реализации ФГОС на первый план выходит формирование именно метапредметных результатов, к которым в полной мере относится и алгоритмическое мышление, лежащее в основе системного подхода к решению задач любого типа. Оно становится тем фундаментом, который позволяет учащимся успешно осваивать не только информатику, но и математику, физику, технологию, а в перспективе – эффективно действовать в профессиональной и бытовой сферах.

Однако, как отмечает М.А. Тихонова[27], существующая практика обучения программированию в школе зачастую сводится к изучению синтаксиса конкретных языков программирования и решению узконаправленных, типовых задач, что не способствует целостному и осознанному формированию данного качества. Преобладание знаниево-ориентированного подхода над деятельностным приводит к тому, что учащиеся, даже зная основы языка, испытывают значительные трудности в самостоятельной разработке и реализации алгоритмов для решения новых, нестандартных проблем. Они успешно воспроизводят заученные конструкции, но теряются при столкновении с задачей, не имеющей готового шаблона решения.

ВКР УлГУ

Это подтверждается статистикой сдачи единого государственного экзамена.[53]

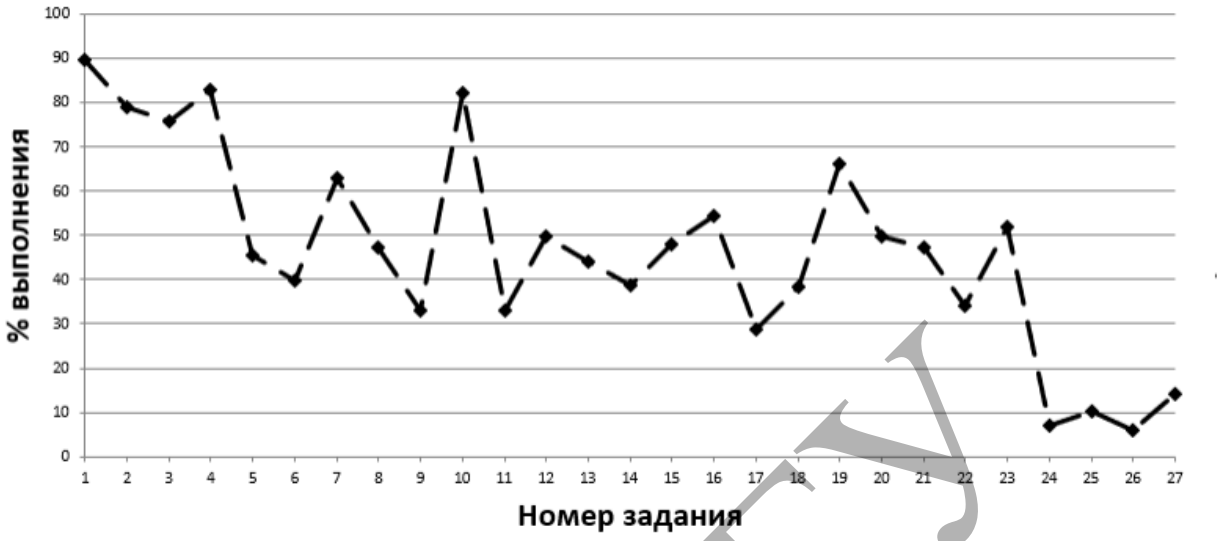


Рис. 1 Средний процент выполнения заданий
по результатам ЕГЭ по информатике 2025 года

Анализ результатов показывает, что:

1) Только **20%** выпускников демонстрируют высокий уровень алгоритмического мышления

2) Большинство участников испытывают серьезные трудности с задачами на разработку алгоритмов

3) Наблюдается значительный разрыв между теоретическими знаниями и практическими навыками программирования

4) Требуется усиление практической подготовки в области алгоритмизации

Этот разрыв между требованием стандартов и реальной практикой усугубляется психологическими особенностями современных учащихся 8-х классов. Для них характерны клиповость мышления, высокая скорость восприятия визуальной информации и устойчивая потребность в немедленной обратной связи. Традиционные, зачастую абстрактные методы обучения, основанные на записи кода в тетради или среде разработки без визуализации

ВКР УлГУ

результата, не соответствуют их когнитивным паттернам и приводят к потере учебной мотивации.

Значительный потенциал игровых и визуальных образовательных сред для развития алгоритмического мышления подтверждается исследованиями отечественных и зарубежных учёных. Так, С.А. Бешенков и Е.А. Ракитина отмечают[12], что наглядное представление алгоритмических конструкций способствует более глубокому пониманию учащимися принципов алгоритмизации и программирования. По мнению М. Резника, создателя среды Scratch,[42] визуальное программирование позволяет школьникам осваивать сложные алгоритмические понятия через конструирование и экспериментирование, снижая когнитивные трудности, связанные с изучением синтаксиса языков программирования. Вместе с тем анализ школьной практики показывает, что возможности игровых и визуальных сред обучения используются недостаточно широко, вследствие чего их развивающий потенциал для формирования алгоритмического мышления учащихся реализуется не в полной мере.

Следовательно, существует насущная потребность в поиске и внедрении таких педагогических инструментов и технологий, которые позволили бы преодолеть формализм в обучении, соответствовали бы требованиям ФГОС и психологическим особенностям современных школьников, делая процесс развития алгоритмического мышления наглядным, увлекательным, практико-ориентированным и, что наиболее важно, эффективным. Разработка такой технологии, интегрирующей в себе прогрессивные педагогические подходы и адекватные образовательные среды, и составляет ядро актуальности данного исследования.

Противоречия:

1. Противоречие между требованием ФГОС к формированию алгоритмического мышления и преобладанием знание-ориентированного подхода в преподавании модуля «Алгоритмы и программирование», при

ВКР УлГУ

котором упор делается на запоминание правил синтаксиса, а не на развитие мыслительных процессов.

2. Противоречие между высоким развивающим потенциалом игровых и визуальных сред для формирования алгоритмических структур мышления и их недостаточным использованием в реальной образовательной практике на уроках информатики в 8-х классах.

3. Противоречие между естественным для учащихся подросткового возраста интересом к интерактивным, игровым формам деятельности и традиционными, зачастую абстрактными методами обучения программированию, что снижает учебную мотивацию и познавательную активность.

Проблема: какова педагогическая технология развития алгоритмического мышления у учащихся 8-ых классов.

Цель работы: Разработать педагогическую технологию развития алгоритмического мышления у учащихся 8-х классов на уроках информатики при изучении модуля «Алгоритмы и программирование».

Объект исследования: Процесс обучения информатике учащихся 8-х классов при изучении модуля «Алгоритмы и программирование» в общеобразовательной школе.

Предмет исследования: Развитие алгоритмического мышления у учащихся 8-х классов на уроках информатики.

Гипотеза исследования: Развитие алгоритмического мышления у учащихся 8-х классов будет более эффективным, если:

1. Будет осуществлен отбор и структуризация содержания учебного материала по программированию на основе системы прогрессивно сложных алгоритмических задач, направленных на формирование умений анализировать условие, декомпозировать задачу, планировать последовательность действий и рефлексировать результат. (когнитивный, процедурно-деятельностный и рефлексивный компоненты)

ВКР УлГУ

2. Педагогическая технология будет основана на интеграции иммерсивной игровой среды, последовательном переходе от визуального программирования к текстовому на языке высокого уровня и включении этапов экспериментального моделирования и рефлексивного анализа результатов.

3. Средством обучения будет являться учебный мир, включающий в себя набор заданий в игровой платформе, предусматривающей возможность программирования, а также набор для логической игры, стилизованный под платформу.

Задачи исследования:

1. Провести теоретический анализ психолого-педагогической и методической литературы по проблеме развития алгоритмического мышления у учащихся основной школы.

2. Разработать учебную программу по модулю «Алгоритмы и программирование», содержащую задачи, направленные на формирование умений анализировать условие, декомпозировать задачу, планировать последовательность действий и рефлексировать результат.

3. Разработать педагогическую технологию развития алгоритмического мышления, основанную на интеграции иммерсивной игровой среды, последовательном переходе от визуального программирования к текстовому на языке высокого уровня и включении этапов экспериментального моделирования и рефлексивного анализа результатов.

4. Разработать электронное средство обучения, представляющее учебный мир на игровой платформе с возможностью программирования, а также материальное средство обучения в виде стилизованной логической игры.

5. Экспериментально проверить эффективность разработанной педагогической технологии в процессе обучения учащихся 8-х классов.

ВКР УдГУ

1. ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЕ ИССЛЕДОВАНИЕ ПО ПРОБЛЕМЕ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ У ШКОЛЬНИКОВ НА УРОКАХ ИНФОРМАТИКИ

1.1 Сущность и структура понятия «алгоритмическое мышление» в психолого-педагогической литературе

В контексте нашего исследования центральным является понятие «алгоритмическое мышление». Несмотря на широкое использование этого термина в современной педагогике и методике преподавания информатики, в научной литературе не существует единого, универсально принятого его определения. Это обусловлено междисциплинарным характером феномена, который находится на стыке кибернетики, психологии мышления, педагогики и частных методик[3]. Для построения эффективной педагогической технологии необходимо провести глубокий теоретический анализ данного понятия через призму различных авторских позиций, выявив их сильные стороны и ограничения.

Рассматривая проблему в широком образовательном контексте, А.В. Хуторской связывает алгоритмическое мышление с достижением метапредметных результатов. Он определяет его как «способность субъекта к построению последовательности действий, ведущих к достижению цели, а также к выделению и описанию алгоритмов в окружающей действительности»[31] Автор акцентирует внимание на том, что алгоритмическое мышление не сводится к написанию кода. Это, прежде всего, «алгоритмическая культура» личности, которая проявляется в умении видеть структуру и порядок в хаотичных процессах, прогнозировать результаты своих действий и выстраивать эффективные стратегии достижения целей в любой сфере деятельности, будь то учебная задача или бытовая ситуация. С точки зрения развития, это означает формирование у ученика универсального умения самостоятельно конструировать пути решения проблем, выходящих за рамки школьных предметов.

ВКР УдГУ

В противовес широкому философскому подходу, М.А. Тихонова предлагает более узкое, практико-ориентированное и операциональное определение. Она описывает алгоритмическое мышление как «совокупность умений по анализу условия задачи, построению и реализации плана её решения, включая разбиение на подзадачи (декомпозицию), и последующему анализу полученного результата» [27]В данной трактовке акцент смещается с абстрактной «способности» на конкретные, измеряемые действия. Развитие в этом ключе подразумевает целенаправленное формирование и отработку конкретных операционных навыков: выявления исходных данных и требуемого результата, поиска инвариантов (повторяющихся действий), выбора управляющих конструкций (ветвление, цикл) и, что особенно важно, рефлексии (анализа причин успеха или неудачи алгоритма).

Е.К. Хеннер подчеркивает связь алгоритмического мышления с теоретическим мышлением и информационным моделированием. По его мнению, оно включает «умение планировать структуру действий, направленных на решение задачи; умение строить информационные модели процессов, протекающих в природе и обществе; умение выделять в них элементарные операции и определять их последовательность» [30]Хеннер указывает на то, что алгоритм не возникает из ниоткуда: ему всегда предшествует этап построения модели объекта или процесса. Следовательно, алгоритмическое мышление, по Хеннеру, ведет к формированию системного подхода, при котором учащийся учится абстрагироваться от несущественных деталей и работать с формализованным представлением реальности.

Проведенный анализ позволяет сгруппировать рассмотренные подходы по трем ключевым аспектам:

1) Когнитивно-философский аспект (Хуторской, Ершов): рассматривает алгоритмическое мышление как универсальную метапредметную компетенцию и элемент общей культуры, направленный на упорядочивание деятельности и прогнозирование. [8]

ВКР УлГУ

2) Операционально-деятельностный аспект (Тихонова, Лапчик): фокусируется на конкретных технологических приемах (декомпозиция, формализация, выбор структур, отладка), необходимых для решения задач с помощью формального исполнителя.[13]

3) Модельно-системный аспект (Хеннер): связывает алгоритмизацию с процессами информационного моделирования и системного анализа.[30]

Все эти определения взаимодополняют друг друга, однако ни одно из них в отдельности не покрывает в полной мере специфику современного обучения подростков[29] в условиях цифровой трансформации. Подход Хуторского может быть слишком абстрактным для непосредственного планирования урока. Подход Тихоновой, будучи методически точным, иногда недооценивает мотивационный и творческий компоненты, сводя процесс к механическому исполнению инструкций.

На основе проведенного синтеза, а также с учетом цели данной работы, в рамках нашего исследования мы будем рассматривать **алгоритмическое мышление** как интегративное качество личности, проявляющееся в способности к системному анализу условия задачи, декомпозиции сложной проблемы на формализуемые подзадачи, построению, реализации и отладке последовательности действий (алгоритма) с учетом возможностей конкретного исполнителя, а также к рефлексивной оценке и оптимизации полученного результата.

Мотивированное обоснование выбора данного рабочего определения:

- Соответствие требованиям ФГОС ООО. Определение включает в себя не только предметные действия (построение алгоритма), но и метапредметные (анализ, декомпозиция, рефлексия), что напрямую коррелирует с требованиями к результатам освоения модуля «Алгоритмы и программирование».
- Учет возрастных особенностей учащихся 8-х классов. Подростки 14-15 лет находятся на этапе развития теоретического, гипотетико-

ВКР УдГУ

дедуктивного мышления[5]. Включение в определение этапов «декомпозиции» и «рефлексивной оценки» отвечает их потребности в самостоятельном, осмысленном действии, а не в простом копировании кода.

- Обоснованность выбранного инструментария. Формулировка «с учетом возможностей конкретного исполнителя» и «реализации» легитимизирует использование в исследовании двухступенчатой системы: сначала визуальное, наглядное конструирование алгоритма в иммерсивной среде Minecraft (где результат виден мгновенно, что снижает когнитивную нагрузку), а затем его формализация и оптимизация на текстовом языке Python.
- Преодоление знаниевого подхода. Акцент на «интегративном качестве» и «оптимизации результата» смещает фокус с заучивания синтаксиса языка программирования на развитие собственно мыслительных процессов, что решает одно из ключевых противоречий, выявленных во введении.

Развитие алгоритмического мышления, исходя из принятого определения, не является стихийным процессом. Оно предполагает целенаправленное педагогическое управление, обеспечивающее перевод ученика через систему последовательных уровней:

1) Репродуктивный уровень: умение понимать, читать и выполнять готовые алгоритмы, предложенные учителем или средой.

2) Эвристический (конструктивный) уровень: способность самостоятельно разбивать новую задачу на подзадачи, выбирать известные алгоритмические конструкции (линейные, ветвящиеся, циклические) и комбинировать их для решения нестандартной проблемы.

3) Творческий (оптимизирующий) уровень: умение не только решить задачу, но и предложить несколько вариантов алгоритмов, сравнить их по критериям эффективности (например, количество шагов, читаемость кода) и выбрать оптимальный, а также перенести найденный алгоритмический паттерн в совершенно новую предметную область.

ВКР УлГУ

Таким образом, **алгоритмическое мышление** определяется как интегративное качество личности, проявляющееся в способности к системному анализу условия задачи, декомпозиции сложной проблемы на формализуемые подзадачи, построению, реализации и отладке последовательности действий (алгоритма) с учетом возможностей конкретного исполнителя, а также к рефлексивной оценке и оптимизации полученного результата, а владение алгоритмическим мышлением при этом может быть определено на репродуктивном, эвристическом или творческом уровне.

Можно выделить следующие компоненты алгоритмического мышления:

- когнитивный компонент (анализ, синтез, выявление закономерностей);
- процедурно-деятельностный компонент (декомпозиция, выстраивание логики, прогнозирование результата);
- рефлексивный компонент (самооценка, поиск ошибок, выбор оптимального решения).

1.2 Педагогические инструменты и средства развития алгоритмического мышления

Эффективное развитие алгоритмического мышления у учащихся 8-х классов невозможно в рамках традиционной лекционно-семинарской системы, ориентированной на пассивное восприятие информации. Современному подростку свойственны клиповость мышления и потребность в немедленной визуальной обратной связи. В связи с этим, ключевым условием развития алгоритмического мышления становится выбор адекватных педагогических инструментов, способных трансформировать абстрактные логические конструкции в наглядную, личностно значимую деятельность. Центральное место среди таких инструментов в современной методике занимает игровая форма обучения, органично дополняемая цифровыми иммерсивными средами и средствами программирования.

ВКР УлГУ

В педагогической литературе часто наблюдается терминологическая путаница между понятиями «игровая форма обучения», «геймификация» и «дидактическая игра». Для построения корректной технологии необходимо четко разграничить эти феномены.

Геймификация (gamification) представляет собой применение отдельных игровых элементов (баллы, бейджи, рейтинговые таблицы) в неигровом контексте для повышения внешней мотивации. Она не меняет сути учебной деятельности, лишь «упаковывая» её, и часто не способствует глубокому пониманию алгоритмических структур.

Дидактическая игра – это кратковременное, фрагментарное упражнение с четко заданными правилами, направленное на закрепление конкретного навыка (например, карточки с задачами на упорядочивание действий). Её потенциал ограничен рамками одного учебного элемента.

В отличие от них, *игровая форма обучения* – это целостная, системно организованная форма учебной деятельности, в которой содержание учебного материала, познавательные задачи и методы их решения органично вплетены в игровой сюжет, роли и правила.

Согласно классификации Г.К. Селевко[22], игровая форма обучения включает в себя несколько обязательных структурных компонентов: наличие игрового замысла (сюжета), распределение ролей, наличие правил, регулирующих взаимодействие, и, что наиболее важно для алгоритмизации, наличие цели, достижение которой требует выполнения определенной последовательности действий. Определим игровую форму обучения как целенаправленно организованный инструмент педагогического взаимодействия, в котором учебное содержание воплощается через моделирование проблемных ситуаций, что обеспечивает высокий уровень личностной вовлеченности и создает безопасную среду для экспериментирования.

Связь между игровой формой обучения и развитием алгоритмического мышления носит фундаментальный, а не случайный характер. Любая игра, по

ВКР УлГУ

своей сути, является алгоритмической системой: она имеет исходные данные (игровое поле, ресурсы), целевое состояние (победа, выполнение квеста) и набор правил (алгоритм), определяющих допустимые преобразования данных. Участвуя в игре, ученик неосознанно выполняет алгоритмический анализ: он декомпозирует глобальную цель на подзадачи, выстраивает последовательность действий, проверяет условия (ветвление) и, столкнувшись с неудачей, осуществляет отладку своего «внутреннего алгоритма». Игровая форма легитимизирует ошибку не как неудачу, а как необходимый этап итеративного процесса поиска решения, что критически важно для формирования алгоритмической культуры.

Для реализации игровой формы обучения в цифровую эпоху наиболее эффективным средством становится иммерсивная игровая среда. Под иммерсивной игровой средой понимается цифровое пространство, которое за счет трехмерной графики, интерактивности и свободы действий создает у пользователя эффект глубокого погружения (присутствия) и эмоциональной вовлеченности, стирая грань между условностью игры и реальностью действия.[19]

Анализ педагогической практики и научных исследований (М.С. Цветкова, А.В. Горячев, зарубежные исследования по Computational Thinking) показывает наличие нескольких классов инструментов для развития алгоритмического мышления:[33] [7] [45]

1) Двумерные визуальные конструкторы (Scratch, Blockly). Эффективны на начальном этапе, но для учащихся 8-х классов (14-15 лет) часто воспринимаются как «детские», что снижает внутреннюю мотивацию.

2) Образовательная робототехника (Lego Mindstorms, Arduino). Обеспечивает отличный тактильный отклик, но требует значительных материальных затрат, сложна в масштабировании на весь класс и часто отвлекает внимание от чистой логики алгоритма на технические проблемы сборки.

ВКР УлГУ

3) Иммерсивные песочницы (Minecraft). Данный инструмент представляет собой оптимальный баланс. Иммерсивная среда напрямую поддерживает игровую форму обучения, предоставляя бесконечное «полотно» для применения алгоритмов. Отношение иммерсивной среды к игровой форме обучения здесь инструментальное: среда выступает тем самым «контекстом»[44], который наполняет абстрактные алгоритмические конструкции (циклы, условия) реальным, осязаемым смыслом. Построение моста с помощью цикла в трехмерном мире дает мгновенный, визуально мощный результат, что идеально соответствует потребности учащихся в прагматичности и быстрой обратной связи, одновременно развивая пространственное и алгоритмическое мышление в их единстве.

Неотъемлемым элементом технологии развития алгоритмического мышления является использование методов визуального программирования. Визуальное программирование – это метод создания программ путем манипулирования графическими объектами (блоками), которые представляют собой синтаксически правильные конструкции языка программирования[37].

Главная дидактическая ценность визуального программирования для формирования основ алгоритмического мышления заключается в снижении когнитивной нагрузки. Когда учащийся освобожден от необходимости запоминать правила синтаксиса, расстановки скобок или точек с запятой, он может сконцентрировать все когнитивные ресурсы на семантике – на логике построения алгоритма. Перетаскивание блоков «Если-То» или «Повторять N раз» позволяет учащемуся физически ощутить структуру алгоритма, понять принцип вложенности и последовательности. Визуальная среда выступает в роли «строительных лесов», позволяя ученику быстро достичь работающего результата, что поддерживает мотивацию и формирует базовую ментальную модель алгоритмического процесса.

Однако опора исключительно на визуальное программирование имеет существенные ограничения. Визуальные среды искусственно ограничивают сложность создаваемых проектов, не позволяют в полной мере освоить работу

ВКР УлГУ

с текстовыми данными, сложными структурами или внешними библиотеками. Кроме того, они не готовят учащихся к реальной профессиональной деятельности, где взаимодействие с компьютером осуществляется через текстовый код.

Поэтому важнейшим методом развития алгоритмического мышления в 8-м классе является управляемый переход от визуального конструирования к программированию на языке высокого уровня[28]. Этот переход не должен быть резким разрывом; он должен опираться на принцип преемственности: учащийся сначала отлаживает логику алгоритма в визуальной или иммерсивной среде, а затем транслирует эту уже понятную ему логику в текстовый код.

В качестве целевого языка высокого уровня для данной технологии обоснованно выбран язык Python. Этот выбор мотивирован рядом педагогических и технических факторов, подтвержденных современной методикой преподавания информатики:

1) Читаемость и простота синтаксиса. Синтаксис Python максимально приближен к естественному языку (в частности, к псевдокоду, который учащиеся изучают при составлении блок-схем). Отсутствие избыточных служебных символов (например, фигурных скобок) снижает порог входа и позволяет сохранить фокус на алгоритмической сути задачи, а не на формальных правилах записи.

2) Строгая структурированность через отступы. Требование Python использовать отступы для обозначения блоков кода (тело цикла, условие) дисциплинирует мышление учащегося, визуально и структурно закрепляя понятие вложенности алгоритмических конструкций, что напрямую способствует развитию алгоритмической культуры.

3) Мощный потенциал для интеграции. Python обладает богатыми библиотеками (например, mcpi для Minecraft), которые позволяют напрямую связать текстовый код с действиями в иммерсивной игровой среде. Это создает

ВКР УлГУ

идеальный мост: ученик пишет текстовый код на Python, но мгновенно видит результат своих действий в привычном игровом мире.

4) Профессиональная востребованность. Python является одним из самых популярных языков программирования в мире, используемым в анализе данных, искусственном интеллекте и веб-разработке. Осознание того, что они изучают «настоящий», востребованный инструмент, удовлетворяет потребность старших подростков в самоопределении и придает учебной деятельности высокий личностный смысл[49].

Таким образом, развитие алгоритмического мышления у учащихся 8-х классов требует комплексного применения педагогических инструментов[17]. Игровая форма обучения задает мотивационный и смысловой контекст, превращая решение задач в увлекательный процесс. Иммерсивная игровая среда (Minecraft) выступает пространством для реализации этой формы, обеспечивая наглядность и немедленную обратную связь. Визуальное программирование служит безопасным фундаментом для формирования первоначальных алгоритмических компетенций без страха синтаксических ошибок. Наконец, планомерный переход к языку Python обеспечивает необходимую формализацию знаний, развитие абстрактного мышления и подготовку учащихся к решению задач профессионального уровня. Синтез этих инструментов образует целостную, методически обоснованную среду, максимально благоприятную для развития алгоритмического мышления в подростковом возрасте.

ВКР УдГУ

1.3 Психолого-педагогическая характеристика учащихся 8-х классов как основа развития алгоритмического мышления

Развитие алгоритмического мышления у учащихся 8-х классов требует учета не только предметного содержания обучения, но и возрастных, когнитивных и социально-психологических особенностей подростков. Эффективность педагогической технологии напрямую зависит от того, насколько она соответствует уровню интеллектуального развития,

мотивационной сфере и особенностям восприятия информации современными школьниками. Именно поэтому при разработке технологии развития алгоритмического мышления необходимо опираться на современные возрастно-психологические классификации и исследования когнитивного развития подростков.

В рамках данного исследования особое значение имеет возраст учащихся 8-го класса - 14–15 лет.

В современной психологии все чаще используются возрастные классификации, основанные на когнитивном и социально-эмоциональном развитии личности, а не только на смене ведущих видов деятельности. Одной из наиболее актуальных является классификация Всемирной организации здравоохранения (ВОЗ), а также исследования современных отечественных и зарубежных психологов в области подросткового развития.

Согласно классификации ВОЗ, подростковый возраст подразделяется на три этапа:

1. Ранний подростковый возраст - 10–13 лет.
2. Средний подростковый возраст - 14–16 лет.
3. Поздний подростковый возраст - 17–19 лет.

Учащиеся 8-х классов относятся именно к периоду среднего подросткового возраста, который характеризуется активным развитием высших когнитивных функций, формированием устойчивого абстрактного мышления и переходом к более сложным формам интеллектуальной деятельности.

Современные исследования когнитивного развития подростков [43] [36] [40] показывают, что в возрасте 14–15 лет происходит качественное усложнение мыслительных процессов. Подростки начинают:

- оперировать гипотетическими ситуациями;
- выстраивать причинно-следственные связи;
- прогнозировать результаты действий;
- анализировать альтернативные варианты решений;

ВКР УлГУ

- осуществлять рефлексию собственной деятельности;
- строить логические модели сложных процессов.

Именно эти качества лежат в основе алгоритмического мышления.

Современные психологи рассматривают возраст 14–15 лет как период активного формирования исполнительных функций мозга, обеспечивающих способность к сложной интеллектуальной деятельности. Исследования A. Diamond, P. Zelazo и других ученых показывают, что в подростковом возрасте интенсивно развиваются:

- рабочая память;
- когнитивная гибкость;
- способность к планированию;
- контроль внимания;
- логический анализ;
- прогнозирование результата действий.

Эти функции являются базовыми компонентами алгоритмического мышления[38] [46].

Важнейшей особенностью подростков 8-го класса становится переход от конкретно-образного мышления к абстрактно-логическому. Учащийся начинает не только воспринимать готовые схемы действий, но и самостоятельно конструировать модели решения задач. Он способен удерживать в сознании несколько условий одновременно, сравнивать варианты решений и выбирать наиболее эффективный алгоритм.

Для развития алгоритмического мышления данная особенность имеет принципиальное значение, поскольку процесс алгоритмизации требует:

- анализа исходных данных;
- выделения существенных признаков;
- построения последовательности действий;
- прогнозирования результата;
- проверки корректности решения;
- поиска ошибок и их исправления.

ВКР УдГУ

В более раннем возрасте подобные операции выполняются преимущественно на наглядном уровне и требуют постоянной поддержки со стороны учителя. В 8-м классе учащиеся уже способны осуществлять данные действия самостоятельно, что делает возможным переход к полноценному программированию и проектной деятельности.

Одной из ключевых особенностей среднего подросткового возраста является развитие рефлексии и метакогнитивных навыков. Современные исследования J. Flavell, D. Kuhn и A. Ефклидес показывают, что подростки 14–15 лет начинают осознавать собственные мыслительные процессы и способны анализировать эффективность используемых стратегий[39].

Для алгоритмического мышления данная особенность имеет особое значение, поскольку создание алгоритма невозможно без:

- оценки правильности решения;
- анализа допущенных ошибок;
- сравнения различных способов решения;
- оптимизации алгоритма;
- поиска более эффективных стратегий.

Подросток начинает понимать, что одна и та же задача может иметь несколько способов решения, отличающихся по эффективности. Это создает основу для формирования не только исполнительского, но и творческого уровня алгоритмического мышления.

Именно в этом возрасте учащиеся способны переходить от простого воспроизведения готовых алгоритмов к самостоятельному проектированию собственных способов решения задач.

Современные подростки относятся к поколению цифровой среды, которое в научной литературе часто обозначается как поколение Z или поколение Alpha-transition. Исследователи M. Prensky, J. Twenge, Н. Солдатова и Е.О. Смирнова отмечают, что для современных школьников характерны следующие особенности:

- постоянное взаимодействие с цифровыми технологиями;

ВКР УдГУ

- высокая скорость восприятия информации;
- потребность в визуализации;
- ориентация на интерактивность;
- необходимость немедленной обратной связи;
- предпочтение практической деятельности теоретическому изложению;
- высокая мотивация к игровым форматам взаимодействия.

Эти особенности существенно влияют на процесс обучения информатике[41] [44] [24] [25].

Традиционные формы преподавания программирования, основанные исключительно на объяснении синтаксиса языка и решении абстрактных задач, часто оказываются недостаточно эффективными для современных восьмиклассников. Подростки быстро теряют интерес к деятельности, если не видят практического результата или эмоциональной вовлеченности.

В то же время именно алгоритмическое мышление может эффективно развиваться через деятельность, соответствующую особенностям цифрового поколения:

- игровые образовательные среды;
- визуальное программирование;
- проектную деятельность;
- интерактивные задачи;
- моделирование реальных ситуаций;
- командную работу.

Использование иммерсивных игровых сред позволяет соединить возрастные интересы подростков с образовательными целями. Визуальный и мгновенно наблюдаемый результат выполнения алгоритма снижает когнитивную нагрузку и поддерживает устойчивую учебную мотивацию.

Средний подростковый возраст характеризуется усилением потребности в самостоятельности, признании и социальной значимости собственной деятельности. Подростки стремятся к:

ВКР УлГУ

- демонстрации компетентности;
- признанию со стороны сверстников;
- самостоятельному принятию решений;
- проявлению инициативы;
- созданию значимого продукта деятельности.

Алгоритмическая деятельность в информатике хорошо соответствует данным потребностям. Создание программ, игровых объектов, автоматизированных систем или цифровых проектов позволяет подростку видеть конкретный результат собственной интеллектуальной деятельности.

Кроме того, современное программирование все чаще носит коллективный характер. Поэтому развитие алгоритмического мышления эффективно осуществляется через:

- парное программирование;
- командные проекты;
- совместный анализ задач;
- коллективную отладку алгоритмов;
- обсуждение способов решения.

Подобные формы работы соответствуют возрастной потребности подростков в коммуникации и социальном взаимодействии.

Проведенный анализ современных психологических исследований позволяет сделать вывод о том, что возраст 14–15 лет является наиболее благоприятным периодом для целенаправленного развития алгоритмического мышления.

- Во-первых, именно в этом возрасте формируется устойчивое абстрактно-логическое мышление, необходимое для понимания алгоритмических структур, циклов, условий и моделей программирования.
- Во-вторых, подростки приобретают способность к рефлексии и анализу собственных интеллектуальных действий, что делает возможным осознанное проектирование и оптимизацию алгоритмов.

ВКР УдГУ

- В-третьих, высокий уровень цифровой вовлеченности современных восьмиклассников создает благоприятную основу для использования игровых и программных сред обучения.
- В-четвертых, потребность подростков в самостоятельности, творчестве и социальном признании позволяет эффективно организовывать проектную и исследовательскую деятельность в области программирования.
- В-пятых, содержание школьного курса информатики в 8-м классе впервые предполагает систематическое изучение программирования, что совпадает с психологической готовностью учащихся к освоению алгоритмических конструкций.

Таким образом, 8-й класс представляет собой оптимальный возрастной этап для формирования алгоритмического мышления, поскольку именно в этот период когнитивное, мотивационное и социально-психологическое развитие подростков достигает уровня, необходимого для осознанного освоения алгоритмизации и программирования. Учет данных особенностей является обязательным условием построения эффективной педагогической технологии развития алгоритмического мышления у учащихся основной школы.

Выводы по главе 1:

Алгоритмическое мышление определяется как интегративное качество личности, проявляющееся в способности к системному анализу условия задачи, декомпозиции сложной проблемы на формализуемые подзадачи, построению, реализации и отладке последовательности действий (алгоритма) с учетом возможностей конкретного исполнителя, а также к рефлексивной оценке и оптимизации полученного результата (когнитивный, процедурно-деятельностный и рефлексивный компоненты), а владение алгоритмическим мышлением при этом может быть определено на репродуктивном, эвристическом или творческом уровне.

ВКР УлГУ

Анализ современных педагогических инструментов и средств развития алгоритмического мышления показал, что наиболее эффективными для учащихся 8-х классов являются игровая форма обучения, иммерсивные образовательные среды и программирование. Обоснована целесообразность использования среды Minecraft как средства создания наглядного и мотивирующего образовательного пространства. Показано, что последовательный переход от визуального программирования к текстовому программированию на языке Python позволяет снизить когнитивную нагрузку и сосредоточить внимание учащихся на логике построения алгоритмов.

Изучение психолого-педагогических особенностей учащихся 8-х классов позволило выявить, что возраст 14–15 лет является благоприятным периодом для развития алгоритмического мышления благодаря формированию абстрактно-логического мышления, способности к рефлексии, развитию исполнительных функций и высокой цифровой вовлечённости подростков. Установлено, что современные восьмиклассники нуждаются в интерактивных, наглядных и практико-ориентированных формах обучения, обеспечивающих возможность самостоятельной деятельности и получения быстрого результата.

Таким образом, теоретический анализ позволил определить сущность алгоритмического мышления, выявить эффективные педагогические средства его развития и установить психолого-педагогические условия организации обучения учащихся 8-х классов. Полученные результаты стали основой для разработки педагогической технологии развития алгоритмического мышления, представленной во второй главе исследования.

ВКР УлГУ

2. РАЗРАБОТКА ПЕДАГОГИЧЕСКОЙ ТЕХНОЛОГОГИИ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ У УЧАЩИХСЯ 8-Х КЛАССОВ НА УРОКАХ ИНФОРМАТИКИ

2.1 Модернизация учебной программы по модулю «Алгоритмы и программирование» для 8-го класса

В соответствии с поставленной целью исследования – разработкой педагогической технологии развития алгоритмического мышления у учащихся 8-х классов – необходимо не только проанализировать существующую учебную программу, но и модернизировать её содержание, структуру учебно-тематического планирования и систему учебных задач. В качестве основы используется Федеральная рабочая программа основного общего образования по информатике (ФРП ООО), раздел «Алгоритмы и программирование» для 8-го класса[16].

Проведённый анализ показал, что типовая программа в достаточной степени отражает предметные результаты обучения, однако не обеспечивает целостного развития алгоритмического мышления как метапредметного качества личности. Основной акцент в существующей практике делается на изучение синтаксиса языка программирования и решение типовых задач репродуктивного характера, тогда как развитие навыков анализа, декомпозиции, проектирования алгоритмов и рефлексии требует иной организации содержания обучения.

В связи с этим предлагается модернизация программы по следующим направлениям:

1. Изменение структуры учебно-тематического планирования.
2. Выбор в качестве основного языка программирования Python.
3. Включение системы прогрессивно усложняющихся алгоритмических задач.
4. Интеграция игровых и иммерсивных образовательных сред.

ВКР УдГУ

5. Организация последовательного перехода от визуального программирования к текстовому.

Фрагмент модернизированной рабочей программы по информатике приведен в приложении А.

№	Тема занятия	Основное содержание	Кол-во часов
1	Введение в Minecraft: Python и диалоговая отладка	Знакомство с игровой средой, выполнение линейных алгоритмов, работа с маршрутом агента	1
2	Цикл с условием while	Построение моста заданной длины, анализ повторяющихся действий	1
3	Цикл с параметром for	Автоматическая расстановка объектов через равные интервалы	1
4	Вложенные циклы	Построение прямоугольных структур (стены, площадки)	1
5	Работа со строками	Создание простых игровых команд, обработка текстовых данных	1
6	Анализ и оптимизация алгоритмов	Оптимизация готовых алгоритмов, уменьшение избыточности кода	1

Таблица 1

Учебно-тематический план модуля «Алгоритм и программирование»

В отличие от традиционного изучения программирования, основанного преимущественно на последовательном освоении синтаксических конструкций языка, модернизированный модуль «Алгоритмы и программирование» строится на принципе поэтапного развития алгоритмического мышления через систему прогрессивно усложняющихся алгоритмических задач.

Содержание обучения организуется таким образом, чтобы учащиеся проходили последовательный путь от понимания и выполнения готовых линейных алгоритмов к самостоятельному построению циклических и вложенных конструкций, а далее – к анализу, оптимизации и проектированию собственных алгоритмических решений.

ВКР УлГУ

В основу обновлённого содержания была положена серия практических занятий в среде Minecraft с использованием языка Python и библиотеки mcpi.

Ключевым элементом модернизации программы является система прогрессивно сложных алгоритмических задач, представленных в приложении Б. Под прогрессивным усложнением понимается последовательное увеличение:

- сложности алгоритмических конструкций;
- уровня самостоятельности учащихся;
- степени абстракции;
- количества условий и взаимосвязей;
- необходимости анализа и оптимизации решения.

Система задач строится поэтапно. Первый этап – линейные алгоритмы и исполнитель. На начальном этапе учащиеся осваивают базовые принципы алгоритмизации:

- последовательность действий;
- понятие исполнителя;
- пошаговое выполнение команд;
- контроль результата.

Так, на первом занятии учащиеся должны провести игрового агента по заранее подготовленному маршруту, используя команды библиотеки mcpi и средства диалоговой отладки. Основной задачей является понимание логики выполнения программы и поиск ошибок через анализ промежуточных результатов.

Данный этап соответствует репродуктивному уровню развития алгоритмического мышления.

Второй этап – циклические конструкции. Этот уровень связан с переходом от простого выполнения последовательности команд к выявлению повторяющихся действий и их автоматизации. Уже на этом этапе проявляется текстовое программирование, в отличии от прошлого. На занятиях учащиеся:

- строят мост с использованием цикла while;

ВКР УдГУ

- автоматизируют установку факелов через цикл for;
- самостоятельно определяют условие завершения цикла;
- выбирают шаг изменения параметра;
- анализируют корректность работы алгоритма.

Задачи данного этапа требуют уже не воспроизведения готового решения, а самостоятельного проектирования структуры алгоритма. На этом уровне формируются:

- навыки декомпозиции;
- прогнозирование результата;
- умение выявлять закономерности;
- понимание принципов оптимизации.

Третий этап – вложенные конструкции и структурирование алгоритма. На третьем занятии учащиеся переходят к решению более сложных пространственных задач, требующих использования вложенных циклов.

Например, при построении прямоугольной стены внешний цикл отвечает за высоту, а внутренний – за длину конструкции.

Учащиеся анализируют структуру задачи и разбивают её на уровни. Использование вложенных циклов позволяет значительно сократить объём кода и повысить эффективность алгоритма.

На данном этапе развивается способность к:

- структурированию сложной задачи;
- многоуровневому планированию;
- построению иерархических алгоритмов.

Четвёртый этап – абстрагирование и работа со строками. Пятое занятие предполагает переход к более абстрактному уровню программирования через обработку строк и создание собственных игровых команд. Учащиеся используют операции со строками, создают команды для управления игровыми объектами, связывают текстовый ввод с алгоритмическими действиями в игровой среде. Данный этап обеспечивает переход от визуально-конкретного действия к абстрактному представлению алгоритма.

ВКР УлГУ

Пятый этап – анализ и оптимизация алгоритмов. На завершающем этапе учащиеся получают готовые громоздкие алгоритмы строительства объектов, содержащие избыточные конструкции. Основной задачей становится оптимизация кода: сокращение количества команд, замена повторяющихся действий циклами, повышение эффективности алгоритма. Таким образом, учащиеся переходят от простого создания алгоритмов к их критическому анализу и сравнению различных способов решения задачи. Именно данный этап соответствует творческому уровню алгоритмического мышления, предполагающему рефлексию, оценку и оптимизацию результата.

В модернизированной программе изменяется не только тематика занятий, но и сама логика учебной деятельности.

Структура каждого занятия включает:

- постановку проблемной ситуации в игровом мире Minecraft;
- анализ условий задачи;
- самостоятельное или парное проектирование алгоритма;
- практическую реализацию решения;
- отладку и поиск ошибок;
- рефлексию и обсуждение эффективности алгоритма.

Подобная организация обеспечивает переход от знание-ориентированного подхода к деятельностному, что соответствует требованиям ФГОС ООО и задачам формирования метапредметных результатов.

Кроме того, использование среды Minecraft позволяет реализовать принцип наглядности и немедленной обратной связи, соответствующий возрастным особенностям современных учащихся 8-х классов.

Таким образом, модернизация модуля «Алгоритмы и программирование» заключается не только в обновлении планируемых результатов, но и в перестройке содержания обучения, учебно-тематического планирования и характера алгоритмических задач. Разработанная система прогрессивно сложных задач обеспечивает поэтапное развитие

ВКР УлГУ

алгоритмического мышления. Интеграция игровой среды Minecraft, языка Python и практико-ориентированных задач позволяет сделать процесс обучения более наглядным, мотивирующим и соответствующим требованиям ФГОС ООО, а также возрастным особенностям учащихся 8-х классов.

2.2 Разработка педагогической технологии

Разработка педагогической технологии развития алгоритмического мышления у учащихся 8-х классов основывается на следующих педагогических инструментах и средствах развития алгоритмического мышления: игровая форма обучения, иммерсивные среды и программирование. Анализ, проведённый в 1 главе, показал, что эффективное развитие алгоритмического мышления современных подростков невозможно в рамках исключительно традиционного объяснительно-иллюстративного подхода. Для учащихся 8-х классов необходима образовательная среда, сочетающая наглядность, интерактивность, практическую направленность, возможность экспериментирования и немедленную визуальную обратную связь.

В связи с этим разработанная педагогическая технология основывается на синтезе системно-деятельностного, метапредметного, проблемного и игрового подходов[32]. Алгоритмическое мышление рассматривается не только как совокупность предметных умений по программированию, но и как интегративное качество личности, включающее способность анализировать задачу, декомпозировать её, строить алгоритм решения, осуществлять отладку и рефлексию результата.

Цель технологии – развитие алгоритмического мышления учащихся 8-х классов посредством интеграции игровой иммерсивной среды Minecraft и программирования на языке Python при изучении модуля «Алгоритмы и программирование».

Методологическую основу технологии составляют:

• системно-деятельностный подход;

ВКР УлГУ

• метапредметный подход;

• контекстное и проблемное обучение;

• игровые технологии;

• учёт возрастных особенностей подростков.

Разработанная технология опирается на следующие принципы:

• принцип иммерсивности и проблемности;

• принцип наглядности и постепенного абстрагирования;

• принцип практической направленности;

• принцип рефлексивности.

ВКР УдГУ

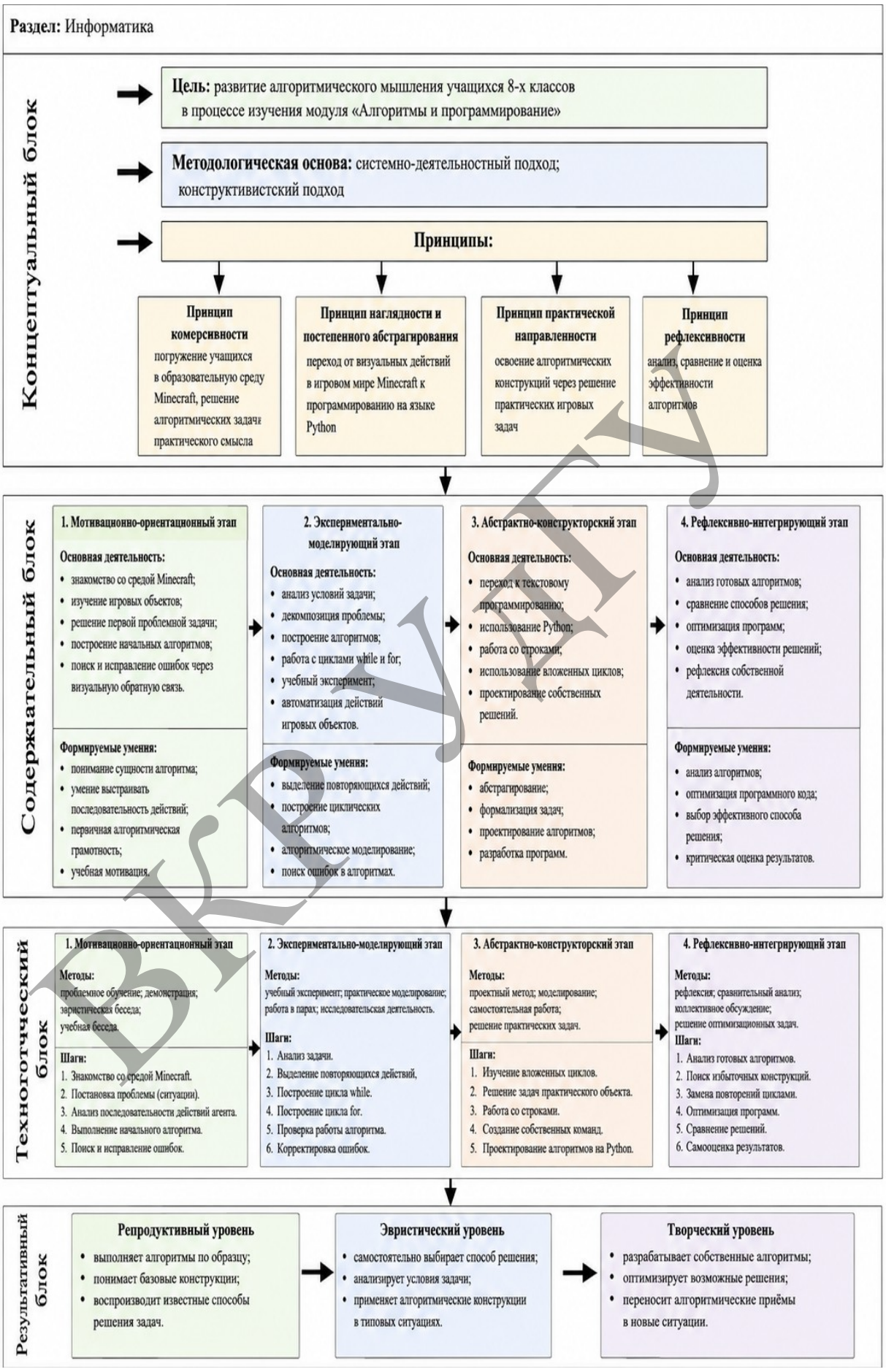


Рис. 2 Модель педагогической технологии

ВКРУПГУ

Принцип иммерсивности предполагает погружение учащихся в игровую образовательную среду Minecraft, где алгоритмические задачи получают практический и личностно значимый смысл.

Принцип наглядности и постепенного абстрагирования реализуется через последовательный переход от визуальных действий в игровом мире к текстовому программированию на языке Python.

Принцип практической направленности обеспечивает освоение алгоритмических конструкций через решение игровых и проектных задач.

Принцип рефлексивности предполагает обязательный анализ, сравнение и оценку эффективности разработанных алгоритмов.

Разработанная педагогическая технология включает четыре взаимосвязанных этапа:

1. Иммерсивно-мотивационный этап.

На данном этапе учащиеся погружаются в образовательную среду Minecraft, знакомятся с игровыми объектами и получают первую проблемную задачу. Основная цель этапа – формирование устойчивой мотивации к изучению алгоритмизации и программирования.

2. Экспериментально-моделирующий этап.

На данном этапе учащиеся учатся анализировать условия задачи, декомпозировать проблему и строить работающие алгоритмы в игровой среде. Основными видами деятельности становятся учебный эксперимент, работа с линейными алгоритмами и циклическими конструкциями.

3.Абстрактно-конструкторский этап.

Учащиеся осваивают циклы, условия, работу со строками и проектируют собственные алгоритмические решения.

4.Рефлексивно-интегрирующий этап.

На заключительном этапе учащиеся анализируют эффективность алгоритмов, сравнивают различные способы решения задач, оптимизируют программы и осуществляют рефлексию собственной деятельности.

ВКР УдГУ

Содержание технологии реализуется через систему последовательных практических занятий.

Тема 1. Введение в Minecraft: Python и диалоговая отладка.

На первом занятии учитель создаёт проблемную игровую ситуацию, в рамках которой учащимся необходимо провести игрового агента по заданному маршруту. Учитель демонстрирует возможности среды Minecraft и объясняет связь между действиями агента и алгоритмом.

Учащиеся анализируют последовательность действий, экспериментируют с линейными алгоритмами и наблюдают результат выполнения программы непосредственно в игровом мире. Основное внимание уделяется пониманию логики алгоритма и поиску ошибок через визуальную обратную связь.

Тема 2. Цикл с условием while.

Учитель предлагает учащимся задачу автоматизации построения моста заданной длины. В ходе обсуждения учащиеся приходят к необходимости использования повторяющихся действий.

Под руководством учителя учащиеся формулируют условие завершения цикла и реализуют алгоритм с использованием конструкции while. В процессе выполнения программы осуществляется анализ ошибок и корректировка алгоритма.

Тема 3. Цикл с параметром for.

На данном занятии учащиеся решают задачу автоматической расстановки факелов через одинаковые промежутки. Учитель организует обсуждение повторяющихся действий и подводит учащихся к использованию цикла for.

Учащиеся самостоятельно определяют количество повторений и шаг изменения параметра, после чего реализуют алгоритм в Minecraft и Python.

Тема 4. Вложенные циклы.

ВКР УдГУ

Учащимся предлагается задача построения прямоугольной стены или пола. В процессе анализа учащиеся приходят к необходимости разделения задачи на несколько уровней.

Учитель организует совместное проектирование алгоритма, в рамках которого учащиеся выделяют внешний и внутренний циклы, после чего реализуют программу на Python.

Тема 5. Работа со строками.

На данном этапе учащиеся переходят к более абстрактному уровню программирования. Учитель демонстрирует, каким образом текстовые команды могут управлять игровыми объектами в Minecraft.

Учащиеся создают собственные игровые команды, используют операции со строками и связывают текстовый ввод с действиями игровых объектов.

Тема 6. Анализ и оптимизация алгоритмов.

На завершающем этапе учащимся предлагаются готовые алгоритмы строительства объектов, содержащие избыточные конструкции.

Основной задачей становится оптимизация кода: сокращение количества команд, замена повторяющихся действий циклами и повышение эффективности алгоритма.

Таким образом, учащиеся переходят от простого выполнения алгоритмов к их анализу, сравнению и оптимизации.

Таблица 2.

Сводная таблица педагогической технологии

Тема занятия	Методы обучения	Формы организации	Средства обучения	Формируемые результаты
1. Введение в Minecraft: Python и диалоговая отладка	проблемное обучение, демонстрация	фронтальная работа, индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Понимание линейного алгоритма
2. Цикл с условием while	учебный эксперимент	парная работа, индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Умение строить циклические алгоритмы

ВКР УдГУ

Тема занятия	Методы обучения	Формы организации	Средства обучения	Формируемые результаты
3. Цикл с параметром for	практическое моделирование	групповая работа, индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Умение выявлять повторяющиеся действия
4. Вложенные циклы	декомпозиция задачи	работа в группах, индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Умение структурировать алгоритмы
5. Работа со строками	проблемно-поисковый	индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Умения абстрагирования и формализации
6. Анализ и оптимизация алгоритмов	рефлексия, сравнительный анализ	коллективное обсуждение, индивидуальная работа	Карта «Алгоритмический мир», python, дидактические шашки	Умения анализа и оптимизации решений

Ожидаемым результатом внедрения технологии является повышение уровня сформированности алгоритмического мышления учащихся, проявляющееся в способности самостоятельно анализировать задачи, проектировать алгоритмы, осуществлять их отладку и применять алгоритмические принципы в новых учебных и практических ситуациях.

2.3 Разработка средств обучения

В рамках реализации разработанной педагогической технологии необходимым условием является создание специальных средств обучения, обеспечивающих развитие алгоритмического мышления учащихся в процессе практической деятельности. В соответствии с задачами исследования были разработаны два взаимосвязанных средства обучения: программное средство – образовательная карта «Алгоритмический мир» в среде Minecraft, а также комплект стилизованной под Minecraft логической игры для работы на уроках информатики.

Использование данных средств позволяет организовать обучение таким образом, чтобы учащиеся не только изучали алгоритмические конструкции

ВКР УлГУ

теоретически, но и применяли их в практической деятельности, связанной с программированием игровых объектов и решением проблемных задач.

Программное средство обучения: Образовательная карта «Алгоритмический мир» в среде Minecraft.

Основными аналогами для обучения программированию и алгоритмике являются среды визуального программирования (Scratch, Blockly), игровые платформы (Code.org, «Час кода») и текстовые среды (IDLE для Python, онлайн-компиляторы).

Scratch / Blockly

Контекст и иммерсивность: Умеренная. Работа ведётся на абстрактной сцене со спрайтами, контекст часто искусственный.

Мотивация: Средняя. Игровой элемент присутствует, но может восприниматься как «детский» подростками 14-15 лет.

Наглядность и обратная связь: Хорошая. Анимация и движение спрайтов наглядно демонстрируют результат.

Потенциал для проектной деятельности: Средний. Позволяет создавать игры и мультфильмы, но в ограниченной двумерной среде.

Переход к текстовому программированию: Плавный, но абстрактный. Блоки похожи на конструкции языка, но переход к чисто текстовой среде всё равно является скачком.

Традиционные текстовые среды (Python)

Контекст и иммерсивность: Низкая. Абстрактная работа с кодом и консолью, слабая связь с визуальным, осязаемым результатом.

Мотивация: Низкая. Требует высокой степени абстракции и самодисциплины, часто противоречит клиповому мышлению.

Наглядность и обратная связь: Отложенная и абстрактная. Обратная связь – это текст в консоли или изменение значений переменных, что требует дополнительного мысленного преобразования.

ВКР УлГУ

Потенциал для проектной деятельности: Высокий, но с барьером входа. Позволяет создавать что угодно, но для получения визуально значимого результата требуются дополнительные сложные библиотеки.

Переход к текстовому программированию: Отсутствует как необходимость. Является целевой, а не переходной средой.

Платформы типа Code.org

Контекст и иммерсивность: Умеренная. Задачи облечены в короткие игровые сюжеты, но мир фрагментарен и ограничен рамками задания.

Мотивация: Высокая на начальном этапе, но может быстро приедаться из-за однообразия формата.

Наглядность и обратная связь: Мгновенная, но упрощённая. Результат часто сводится к прохождению уровня, без создания сложного, сохраняемого продукта.

Потенциал для проектной деятельности: Низкий. Деятельность строго ограничена рамками предложенных головоломок.

Переход к текстовому программированию: Слабый. Платформы часто самодостаточны и не ориентированы на последующий выход в «большое» программирование[52].

Образовательная карта в Minecraft

Контекст и иммерсивность: Высокая. Полное погружение в целостный, знакомый и мотивирующий мир. Алгоритмы создаются для решения задач, органично встроенных в среду.

Мотивация: Максимальная. Minecraft – часть цифровой культуры учащихся. Потребность в исследовании, созидании и социальном взаимодействии внутри мира является мощным внутренним стимулом.

Наглядность и обратная связь: Мгновенная и многослойная. Результат выполнения алгоритма виден непосредственно в трёхмерном мире (построенный объект, пройденный путь). Ошибка приводит к наглядному, но безопасному последствию (робот упал в лаву).

ВКР УдГУ

Высокий потенциал для проектной деятельности позволяет создавать сложные, масштабные и визуально впечатляющие проекты (автоматизированные фермы, логические схемы, архитектурные сооружения), которыми можно гордиться и делиться[14].

Таким образом, образовательная карта в Minecraft обладает уникальным сочетанием иммерсивности, мотивационного потенциала и наглядности, что делает её предпочтительным и выигрышным средом для первоначального формирования алгоритмических концепций у подростков, полностью соответствуя их психологическим особенностям и требованиям деятельностного подхода ФГОС.

Основным средством обучения стала специально разработанная образовательная карта в среде Minecraft. Карта представляет собой игровое пространство, разделённое на зоны, каждая из которых предназначена для изучения определённой алгоритмической конструкции.

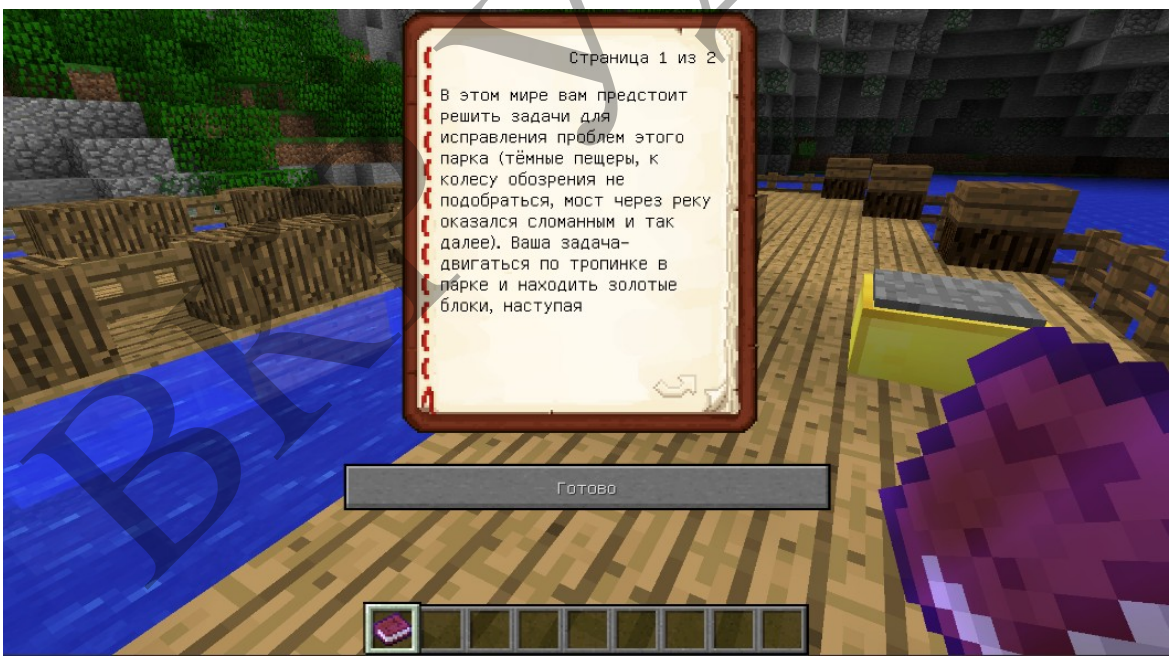


Рис. 3. Карта «Алгоритмический мир» в minecraft

Каждая игровая зона включает практические задания, связанные со строительством объектов, автоматизацией действий игровых агентов и программированием различных игровых процессов.

ВГУ

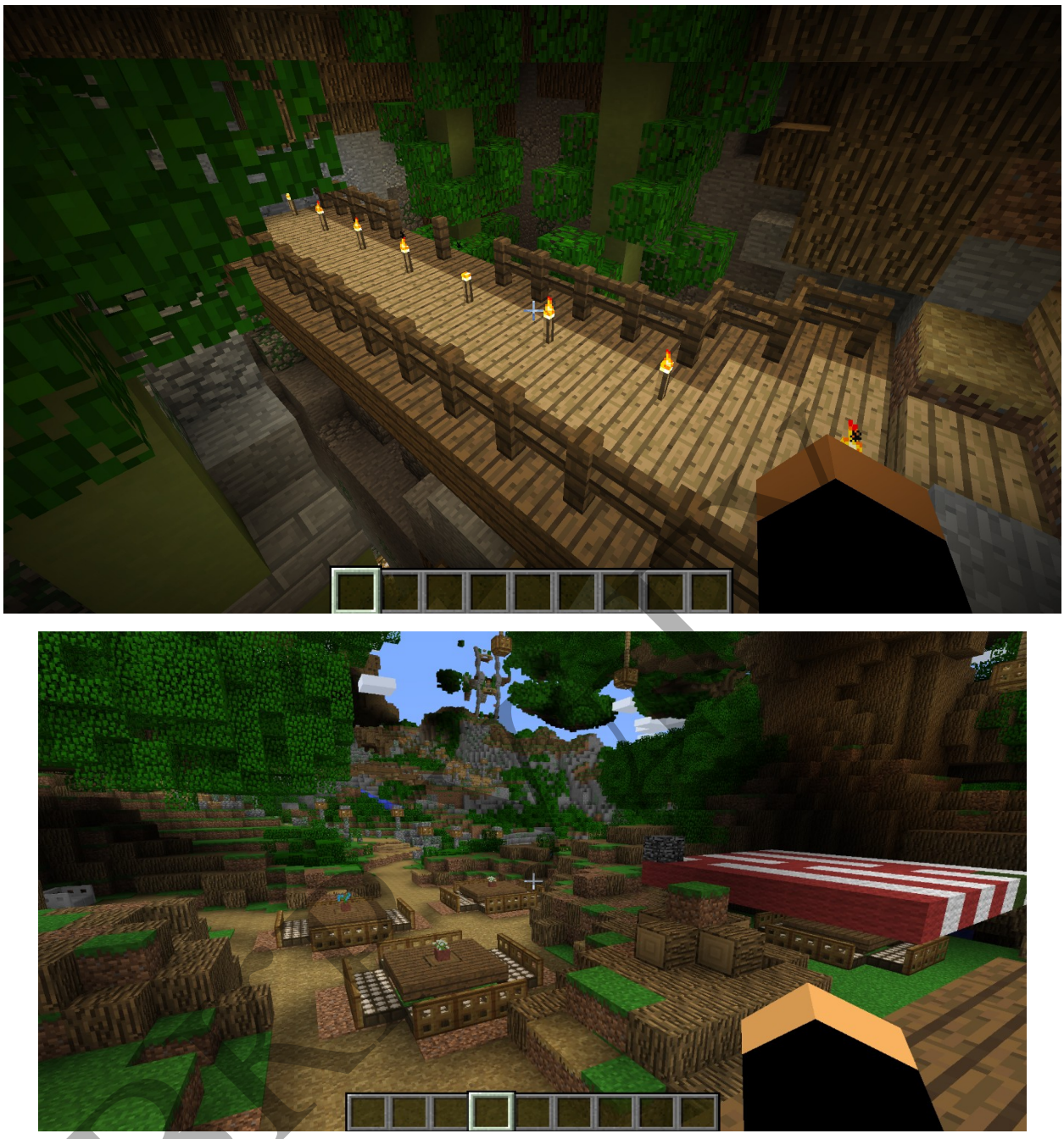


Рис.4-5. Карта «Алгоритмический мир» в minecraft

Разработка карты осуществлялась с учётом возрастных особенностей учащихся 8-х классов. Игровая среда обеспечивает высокий уровень вовлечённости, наглядность результата[34] и возможность немедленной проверки правильности алгоритма.

В процессе разработки карты были созданы:

• игровые маршруты для изучения линейных алгоритмов;

• задания на автоматизированное строительство объектов;

• игровые кейсы для изучения циклических конструкций;

• проблемные ситуации, требующие оптимизации алгоритмов.

Использования образовательной карты предполагает более активное включение учащиеся в процесс решения алгоритмических задач по сравнению с традиционными формами работы. Наличие визуального результата и игровой цели способствует повышению мотивации и более устойчивому усвоению алгоритмических конструкций.

Особое внимание уделяется постепенному усложнению задач. Сначала учащиеся выполняют готовые линейные алгоритмы, затем переходят к самостоятельному использованию циклов и, в дальнейшем, к проектированию собственных решений.

Материальное средство обучения: Дидактические шашки «Алгоритмический паттерн» в стилистике Minecraft

Классическими материальными аналогами для развития логического мышления являются шахматы, шашки, го, различные логические настольные игры.

Абстрактные логические игры (Го, Коридор)

Связь с основным средством и темой: Отсутствует. Контекст игр абстрактен.

Акцент на алгоритмическое планирование: Имплицитный. Требуют глубокого расчёта, но также без акцента на вербализацию последовательности простых шагов.

Доступность и скорость вовлечения: Низкая. Правила Го или «Коридора» требуют отдельного изучения, что является барьером для спонтанной игры на перемене.

Потенциал для неформального обучения: Средний. Может заинтересовать любителей головоломок, но массового привлечения не вызывает.

Классические шашки/шахматы

Связь с основным средством и темой: Отсутствует. Игра существует в собственном, отдельном контексте, не связанном с IT-тематикой.

ВКР УлГУ

Акцент на алгоритмическое планирование: Имплицитный. Планирование является неотъемлемой частью игры, но не акцентируется и не вербализуется как алгоритм. Фокус на тактике и стратегии, а не на структурировании последовательности.

Доступность и скорость вовлечения: Средняя/Низкая. Шахматы имеют высокий порог входа и длительность партии. Классические шашки проще, но менее вариативны без модификаций.

Потенциал для неформального обучения: Средний. Игра может быть интересна сама по себе, но не вызывает прямых ассоциаций с уроками информатики.

Шашки «Алгоритмический паттерн»

Связь с основным средством и темой: Прямая и явная. Визуальная стилистика Minecraft создаёт смысловой мост между уроком и внеурочной деятельностью, поддерживая интерес и погружение в тематику.

Акцент на алгоритмическое планирование: Явный и усиленный. Модифицированные правила и карточки-команды целенаправленно требуют формулировки плана действий до его исполнения, что тренирует именно прогностическую и плановую составляющую мышления.

Доступность и скорость вовлечения: Высокая. Правила базовой игры знакомы большинству. Модификации интуитивно понятны и вводятся постепенно. Партия может быть короткой.

Потенциал для неформального обучения: Максимальный. Привлекательный дизайн и связь с популярной игрой провоцируют самостоятельную активность во внеурочное время, превращая развитие алгоритмического мышления в увлечение.

Таким образом, материальное средство, будучи стилистически и концептуально связанным с основным программным средством, выполняет критически важную функцию «переноса» алгоритмических принципов из цифровой в физическую реальность. Это способствует их лучшему осознанию

ВКР УдГУ

и закреплению, а также создаёт условия для развития мышления в непринуждённой, игровой обстановке за пределами урока.

Дополнительным средством обучения выбрана настольная игра – шашки, стилизованные под вселенную Minecraft. Комплект включает:

1. Игровое поле, выполненное в виде сетки (как блоки в игре), с выделенными зонами.

2. Два набора фишек: «Игроков» (стилизованы под персонажей-исполнителей, например, «Стив») и «Мобов» (стилизованы под мобов, например, «Крипер» или блоки препятствий).

3. Карточки с дополнительными условиями, вводящими базовые алгоритмические команды («Сделай ход на IF (если) поле», «Повтори ход N раз (цикл)», «Выполни последовательность из 2-х ходов»).

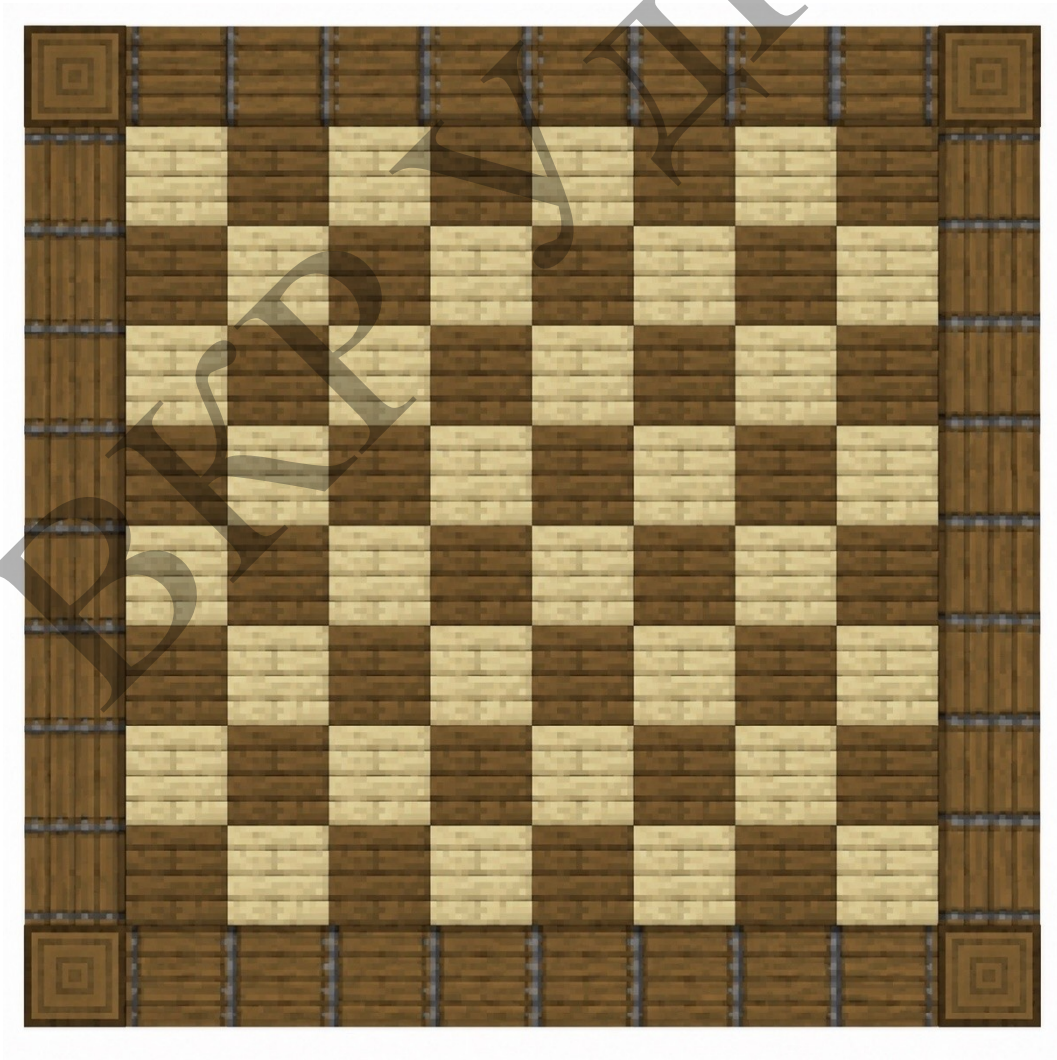


Рис. 6. Поле для игры в шашки

РЕПОЗИТОРИЙ

Механика игры модифицирована для акцента на планирование: перед тем как сделать ход, игрок (или команда) должен не просто выбрать фигуру, но и озвучить простой алгоритм своего хода и, возможно, нескольких последующих, с учётом карточек-условий. Это превращает классическую игру в упражнение на предварительное планирование последовательности действий и прогнозирование ответа соперника.

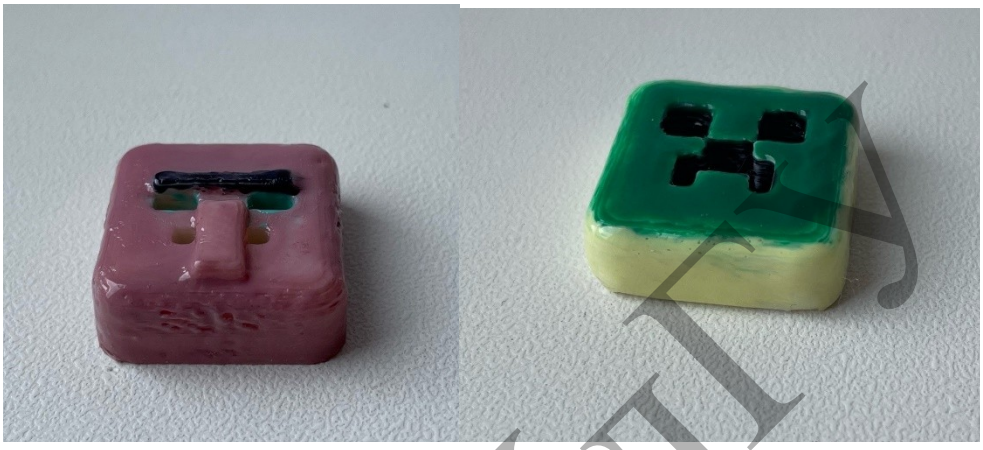


Рис. 7 Фигуры для игры в шашки

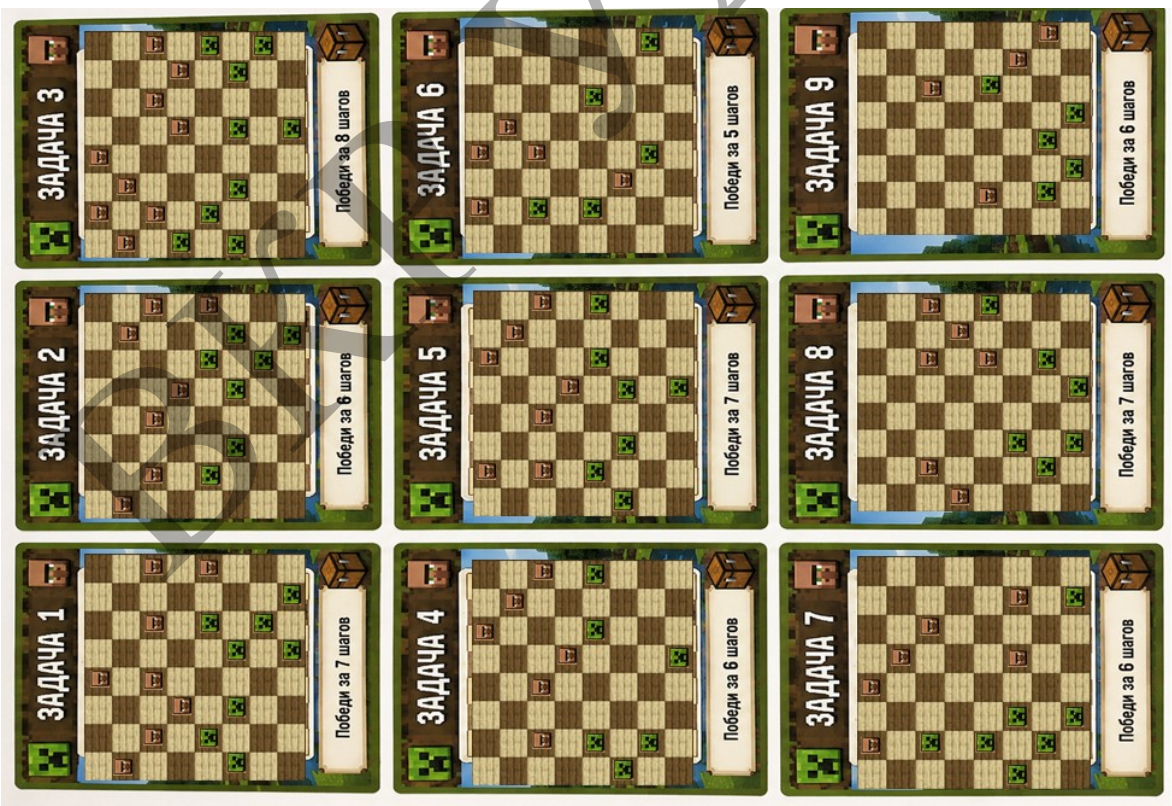


Рис. 8. Карточки с задачами

Разработанные программное и материальное средства образуют единый дидактический комплекс, пространственно и содержательно расширяющий

ВГСУДА

среду для развития алгоритмического мышления. Основное средство (карта в Minecraft) обеспечивает глубокое, мотивированное освоение алгоритмических концепций в цифровой среде. Дополнительное средство (шашки) способствует их закреплению, абстрагированию и переносу в ситуацию живого социального взаимодействия, тренируя ключевой компонент АМ-планирование.

Выводы по главе 2

Осуществлена модернизация рабочей программы модуля «Алгоритмы и программирование» для 8-го класса. Разработано обновлённое учебно-тематическое планирование, ориентированное не только на освоение программирования, но и на целенаправленное развитие алгоритмического мышления. В основу содержания положена система прогрессивно усложняющихся задач, обеспечивающая переход учащихся от выполнения готовых алгоритмов к самостоятельному проектированию, анализу и оптимизации решений.

Разработана педагогическая технология развития алгоритмического мышления учащихся 8-х классов. Технология основана на сочетании системно-деятельностного, метапредметного, проблемного и игрового подходов. Определены её цель, принципы, этапы и содержание. В структуру технологии включены иммерсивно-мотивационный, экспериментально-моделирующий, абстрактно-конструкторский и рефлексивно-интегрирующий этапы, обеспечивающие последовательное формирование компонентов алгоритмического мышления.

Разработаны средства обучения, обеспечивающие практическую реализацию педагогической технологии. В качестве основного средства создана образовательная карта «Алгоритмический мир» в среде Minecraft, содержащая систему тематических локаций и заданий для изучения алгоритмических конструкций. Дополнительно разработано материальное средство обучения – дидактические шашки «Алгоритмический паттерн»,

ВКР УлГУ

направленные на развитие логического и алгоритмического мышления во внеурочной деятельности. Анализ разработанных средств показал их соответствие поставленным целям и возрастным особенностям учащихся.

Полученные результаты стали основой для проведения опытно-экспериментальной работы по проверке эффективности разработанной педагогической технологии.

ВКР УдГУ

3. ОПЫТНО-ЭКСПЕРИМЕНТАЛЬНАЯ РАБОТА

3.1 Методика проведения опытно-экспериментальной работы

Проверка эффективности разработанной педагогической технологии возможна с помощью анализа учебного процесса, а также оценивания уровня развития алгоритмического мышления до и после внедрения[10] технологии с помощью подобранных диагностических методик.

Цель проведения экспериментальной работы – проверка эффективности формирования алгоритмического мышления у учащихся 8-х классов на уроках информатики в модуле «Алгоритмы и программирование».

Для достижения цели были поставлены следующие задачи:

1. Определить начальный уровень развития алгоритмического мышления у учащихся до внедрения разработанной педагогической технологии;
2. Реализовать разработанную технологию, основанную на использовании проблемных ситуаций, проектной деятельности и поэтапном переходе от визуального программирования к текстовому в игровой среде;
3. Зафиксировать результаты внедрения технологии на итоговом этапе эксперимента;
4. Проанализировать динамику показателей по выделенным критериям и сделать вывод об эффективности предложенного подхода.
5. Сравнить результаты диагностик контрольной и экспериментальной групп.

Диагностическим материалом для определения уровня алгоритмического мышления у учащихся 8-х классов были выбраны три методики, каждая из которых диагностирует конкретный компонент рабочего определения алгоритмического мышления и проверяет положения выдвинутой гипотезы:

ВКР УлГУ

1. Субтесты теста структуры интеллекта Р. Амтхауэра [1](адаптация Л.А. Ясюковой) – диагностика *когнитивного компонента* (анализ, синтез, выявление закономерностей);

2. Методика «Планирование последовательности действий» (адаптированная на основе материалов А.В. Горячева[7] и современных стандартов ФГОС[15]) – диагностика *процедурно-деятельностного компонента* (декомпозиция, выстраивание логики, прогнозирование результата);

3. Методика «Рефлексивный анализ и оптимизация алгоритма» (на основе материалов по формированию УУД[15]) – диагностика *рефлексивного компонента* (самооценка, поиск ошибок, выбор оптимального решения).

Рассмотрим каждую методику подробнее:

1. Субтесты теста структуры интеллекта Р. Амтхауэра (адаптация Л.А. Ясюковой)

Цель: Выявить уровень развития понятийного логического мышления, способности к установлению закономерностей и умения осуществлять мыслительные операции, лежащие в основе алгоритмической деятельности. Для диагностики были отобраны три субтеста, наиболее релевантных когнитивному компоненту алгоритмического мышления:

- **Субтест 2 – «Исключение лишнего» (ИЛ).** Диагностирует аналитико-синтетические способности, умение выделять существенные признаки. Учащимся предлагаются ряды из пяти слов, где необходимо найти элемент, не входящий в общую смысловую группу.
- **Субтест 3 – «Поиск аналогий» (ПА).** Диагностирует способность к установлению логических связей, умение рассуждать по аналогии, что критически важно для переноса алгоритмических схем на новые задачи.
- **Субтест 6 – «Определение закономерностей» (ОЗ).** Диагностирует способность выявлять скрытые правила в числовых и логических рядах, основу для понимания циклов и условий в программировании.

ВКР УлГУ

Процедура проведения: Каждый субтест выполняется в ограниченное время (2 субтест – 6 мин, 3 субтест – 7 мин, 6 субтест – 7 мин). Ответы фиксируются в бланке. Каждое правильное совпадение с ключом оценивается в 1 балл.

Обработка и интерпретация результатов:

1. **Высокий уровень (14–20 баллов):** Учащийся демонстрирует устойчивое владение аналитико-синтетическими операциями, гибкость мышления, способность к абстрагированию и переносу правил.

2. **Средний уровень (8–13 баллов):** Интеллектуальные операции находятся в стадии формирования. Учащийся справляется с типовыми задачами, но испытывает затруднения при многошаговом анализе или нестандартных условиях.

3. **Низкий уровень (0–7 баллов):** Учащийся испытывает трудности при выделении существенных признаков и выявлении закономерностей, требуется коррекционная работа по развитию логического анализа.

2. Методика «Планирование последовательности действий» (адаптированная)

Цель: Выявить уровень сформированности умения декомпозировать задачу, выстраивать правильную последовательность шагов и прогнозировать результат.

Процедура проведения: Учащимся предлагаются задания двух типов.

- **Тип 1. Восстановление последовательности.** Набор действий расположен хаотично. Необходимо расположить их в логически верном порядке с учётом условий перехода. *Пример:* «Восстанови порядок команд для робота, чтобы он объехал препятствие и вернулся в исходную точку».

- **Тип 2. Составление последовательности по заданному результату.** Даны начальная и конечная ситуации; необходимо описать промежуточные шаги, включая проверки условий. *Пример:* «Опиши последовательность действий для автоматизации полива: если почва сухая → включить насос, если влажная → ждать 2 часа».

ВКР УлГУ

Критерии оценивания (шкала 0–3 балла):

- **3 балла:** Последовательность верна, шаги логически обоснованы, учтены условия, порядок оптимален; учащийся аргументирует выбор.
- **2 балла:** Логика в целом сохранена, но есть незначительные нарушения порядка или пропущены вспомогательные шаги.
- **1 балл:** Нарушена базовая логика достижения результата, требуются существенные пояснения или помощь.
- **0 баллов:** Задание не выполнено или последовательность хаотична.

Уровни сформированности умения (сумма баллов за 4 задания):

1. **Высокий (8-10 баллов):** Самостоятельно выстраивает многошаговые последовательности, учитывает ветвления, прогнозирует результат.
2. **Средний (5-7 баллов):** Справляется с типовыми последовательностями, допускает ошибки при многошаговых или условных конструкциях, требует незначительной поддержки.
3. **Низкий (0–4 баллов):** Испытывает трудности даже при простых линейных последовательностях, действует методом случайного подбора.

3. Методика «Рефлексивный анализ и оптимизация алгоритма»

Цель: Оценить способность учащегося переводить словесное описание в формализованный алгоритм, осуществлять переход от визуальной к текстовой форме, а также рефлексивно анализировать и оптимизировать решение.

Процедура проведения: Учащимся предлагаются практические задачи, требующие составления, отладки и оптимизации алгоритма. *Примеры заданий:*

1. Составь алгоритм вычисления суммы ряда чисел до тех пор, пока она не превысит заданное значение N. Представь в виде блок-схемы и запиши на языке программирования (Python/Pascal).

ВКР УдГУ

2. Дан алгоритм движения персонажа в лабиринте с ошибкой (зацикливание). Найди ошибку, объясни её причину и предложи исправленный вариант.

3. Сравни два алгоритма сортировки массива по количеству шагов и выбери оптимальный для условия задачи. Обоснуй выбор.

Критерии оценивания (шкала 0–3 балла):

- **3 балла:** Алгоритм корректен, учтены все условия и ветвления, форма представления соответствует требованиям, учащийся аргументированно выбирает оптимальный вариант.
- **2 балла:** Алгоритм в целом верен, но есть незначительные нарушения последовательности или формы записи, не влияющие на результат.
- **1 балл:** Присутствуют существенные логические ошибки, отсутствуют условия перехода или ветвления, форма записи не соответствует стандарту.
- **0 баллов:** Задание не выполнено или представленный текст не является алгоритмом.

Уровни сформированности умения (сумма баллов за 3 задания):

1. **Высокий (8–10 баллов):** Самостоятельно составляет, формализует и оптимизирует алгоритмы разной сложности, уверенно переходит между визуальной и текстовой средами, демонстрирует развитую рефлексию.

2. **Средний (5–7 баллов):** Составляет алгоритмы для типовых задач, испытывает трудности при отладке сложных условий или при переходе к текстовому коду, требует направляющей помощи.

3. **Низкий (0–4 баллов):** Не способен самостоятельно выстроить корректную последовательность, не видит ошибок в готовом алгоритме, отсутствует рефлексивный анализ.

На основе рабочего определения алгоритмического мышления, выдвинутой гипотезы и проанализированных методик, в следующей таблице

ВКР УлГУ

представлены итоговые критерии и характеристики уровней, которые будут использоваться для анализа динамики в ходе эксперимента.

Таблица 3

Критерии и характеристики уровней сформированности алгоритмического мышления

Показатель (компонент)	**Высокий уровень**	**Средний уровень**	**Низкий уровень**
Когнитивный (анализ, выявление закономерностей, абстрагирование)	Свободно оперирует аналитико-синтетическими операциями, выявляет скрытые закономерности, переносит алгоритмические принципы в новые контексты без помощи.	Владеет отдельными операциями, решает типовые задачи с частичной поддержкой; выявляет закономерности преимущественно в знакомом материале.	Затрудняется в выполнении базовых мыслительных операций (анализ, сравнение, обобщение) на всех этапах работы с алгоритмами.
Процедурно-деятельностный (декомпозиция, планирование, реализация, отладка, переход визуальный → текстовый)	Самостоятельно декомпозирует задачу, выстраивает оптимальную последовательность, реализует алгоритм в разных средах, находит и исправляет ошибки.	Осуществляет декомпозицию и планирование с незначительной помощью; реализует алгоритмы по образцу; испытывает трудности при отладке и смене среды программирования.	Отказывается от самостоятельного планирования, действует хаотично или строго по готовому шаблону; не способен к отладке и коррекции действий.
Рефлексивный (самооценка, оптимизация, аргументация выбора)	Устойчиво критически анализирует свои и чужие решения, сравнивает альтернативные алгоритмы, выбирает оптимальный, аргументирует решения.	Ситуативно оценивает результат, но не процесс; требуется направляющая помощь для выявления ошибок и поиска способов их устранения.	Отсутствие стремления к самооценке, неспособность выявить ошибки даже при указании на них, рефлексия результатов не сформирована.

Для обеспечения преемственности между теоретическими положениями исследования и системой диагностики необходимо соотнести выделенные в

ВКР УлГУ

параграфе 1.1 уровни алгоритмического мышления с диагностируемыми компонентами алгоритмического мышления. В рамках данного исследования уровень сформированности алгоритмического мышления определяется на основании степени развития когнитивного, процедурно-деятельностного и рефлексивного компонентов.

Репродуктивный уровень характеризуется преимущественным освоением отдельных алгоритмических действий по образцу. Учащийся способен понимать условие задачи и воспроизводить известные способы решения в типовых ситуациях. Когнитивный компонент проявляется на уровне распознавания алгоритмических конструкций и понимания их назначения, процедурно-деятельностный — на уровне выполнения готового алгоритма по инструкции, а рефлексивный компонент выражен слабо и ограничивается фиксацией правильности или неправильности результата.

Эвристический уровень предполагает более высокий уровень самостоятельности при решении задач. Учащийся способен анализировать условие задачи, выполнять её декомпозицию, выбирать подходящие алгоритмические конструкции и адаптировать известные способы решения к изменённым условиям. Когнитивный компонент проявляется в способности устанавливать связи между элементами задачи и выбирать рациональный способ решения, процедурно-деятельностный — в самостоятельном построении и реализации алгоритма, а рефлексивный — в анализе возникающих ошибок и корректировке собственных действий.

Творческий уровень характеризуется способностью самостоятельно проектировать новые алгоритмические решения в нестандартных ситуациях, осуществлять их оптимизацию и критическую оценку. На данном уровне когнитивный компонент проявляется в системном анализе задачи и построении её модели, процедурно-деятельностный — в разработке эффективных алгоритмов и программ, а рефлексивный — в осознанном сравнении различных способов решения, выявлении их преимуществ и

ВКР УлГУ

недостатков, а также в обоснованном выборе наиболее эффективного варианта.

Таким образом, репродуктивный, эвристический и творческий уровни отражают последовательные стадии развития алгоритмического мышления учащихся и определяются степенью сформированности когнитивного, процедурно-деятельностного и рефлексивного компонентов, рассматриваемых в качестве основных критериев диагностики в настоящем исследовании: творческий уровень алгоритмического мышления характеризуется владением всех трёх компонентов на высоком уровне, эвристический уровень алгоритмического мышления предполагает сформированность всех трёх компонентов на уровне не ниже среднего, а при отсутствии выполнения этих условий владение алгоритмическим мышлением оценивается на репродуктивном уровне.

3.2 Проведение опытно-экспериментальной работы

Целью формирующего этапа эксперимента явилась апробация разработанной педагогической технологии развития алгоритмического мышления у учащихся 8-х классов на уроках информатики в рамках модуля «Алгоритмы и программирование».

Экспериментальная работа проводилась на базе МБОУ «Средняя общеобразовательная школа № 19» города Ижевска в период март-май 2026 года. В эксперименте приняли участие учащиеся 8 «В» и 8 «Г». В обучение в группах был внедрён разработанный мной комплекс занятий на базе среды Minecraft с языком Python (библиотека mcpi).

Методика строилась с опорой на поэтапное формирование умственных действий (П.Я. Гальперин)[6] и включала работу по трём ключевым направлениям, соответствующим компонентам алгоритмического мышления:

- анализ условия и формализация задачи (перевод игровой ситуации в алгоритмическую модель);

ВКР УдГУ

- декомпозиция и планирование (разбиение сложного действия (например, постройки моста) на последовательность элементарных шагов);
- реализация, отладка и рефлексия (написание кода, диалоговая отладка, поиск ошибок, оптимизация).

В рамках модуля «Алгоритмы и программирование» была разработана серия из 6 занятий (длительность каждого урока – 35 минут). Занятия проводились 1 раз в неделю. Структура каждого занятия включала:

- организационный момент и актуализация знаний;
- постановка проблемной задачи в игровом мире Minecraft;
- самостоятельная или парная работа по написанию и отладке кода;
- рефлексия и обсуждение решений.

Краткое описание занятий:

- Занятие 1 («Диалоговая отладка»). Учащиеся осваивали библиотеку mcpi, с помощью заранее подготовленного кода программы. Им необходимо было дополнить код командой Print, для отладки каждого шага и с помощью команд в игровом чате провести агента по подготовленному маршруту.
- Занятия 2–3 («Циклы»). Учащиеся решали задачу строительства моста (цикл while) и расстановки факелов (цикл for). Требовалось не просто написать цикл, а правильно задать условие продолжения, выбрать шаг и предусмотреть остановку.
- Занятие 4 («Вложенные циклы»). Построение прямоугольной стены: внешний цикл – по высоте, внутренний – по длине. Учащиеся убеждались, что вложенные циклы позволяют сократить код в десятки раз.
- Занятие 5 («Обработка строк»). Учащиеся, используя наработки с прошлых уроков создавали собственные команды для активации кода.
- Занятие 6 («Анализ алгоритмов»). Учащимся предлагался массивный код для строительства дома или бассейна, их задачей было – оптимизировать данный код с помощью циклов.

ВКР УлГУ

В ходе занятий использовались следующие формы организации деятельности, соответствующие возрастным особенностям учащихся 8-х классов и положениям разработанной технологии:

- Парная работа «Диалоговая отладка». Один ученик пишет код, второй – визуально контролирует выполнение в Minecraft, ищет логические ошибки (например, бесконечный цикл, неправильное смещение блоков) и предлагает исправления. Роли меняются.
- Приём «Сломанный алгоритм». Учитель выдаёт заведомо неработающий код (например, цикл без изменения счётчика), учащиеся должны найти ошибку, объяснить её и исправить.
- Соревновательный элемент. Кто построит объект за минимальное количество строк кода или за наименьшее число итераций цикла.

Пример одного из заданий (Занятие 2, цикл while)

Тема: Цикл с условием.

Игровая задача в Minecraft: «Агент Стив должен построить мост через лавовую реку. Длина моста вводится игроком с клавиатуры (переменная length). Мост строится из каменных блоков (id=1) в направлении оси X от текущего положения игрока. После установки каждого блока в консоль выводится сообщение о текущем шаге. Необходимо, чтобы цикл остановился ровно после установки нужного количества блоков».

Исходный код (заготовка) на языке Python:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
pos = mc.player.getPos()
x = int(pos.x)
y = int(pos.y)
z = int(pos.z)
length = int(input("Введите длину моста: "))
block = 1
i = 0
```

ВКР УдГУ

```
while i < length:
    mc.setBlock(x + i, y, z, block)
    print(f"Поставлен блок номер {i+1}")
print("Мост готов!")
```

Наглядное представление:



Рис. 9 Исходная ситуация



Рис. 10 Результат выполнения программы

Ход работы:

ВКР УлГУ

1. Проанализируйте условие: что дано (координаты, длина), что нужно получить (мост из блоков).

2. Найдите логическую ошибку в коде (отсутствие инкремента переменной i → бесконечный цикл).

3. Исправьте ошибку, добавив i = i + 1 или i += 1.

4. Дополнительное задание:

- Модифицируйте программу так, чтобы мост построился на 1 блок выше игрока.
- Измените тип блока из которого построится мост.

Ожидаемый результат после исправления:

text

Поставлен блок номер 1

Поставлен блок номер 2

...

Мост готов!

В Minecraft игрок видит построенную линию из каменных блоков.

В начале каждого занятия ученикам давались теоретические знания, необходимые для выполнения конкретных заданий на уроке. После ученики спешили сесть за заранее подготовленные компьютеры (запущенный Minecraft и компилятор), чтобы приступить к выполнению заданий. Когда возникали сложности, учащиеся поднимали руку или просто звали педагога для помощи с решением задачи. Примеры выполнения заданий от детей приведены в приложении Г.

Таким образом, в ходе формирующего этапа эксперимента была реализована педагогическая технология, основанная на интеграции игровой среды Minecraft, диалоговой отладки, поэтапного усложнения алгоритмических конструкций (от линейных алгоритмов к циклам, вложениям и обработке строк) и рефлексивного анализа. Разработанный комплекс из 6 занятий позволил учащимся 8-х классов экспериментальных групп:

ВКР УдГУ

- освоить базовые конструкции языка Python в практически значимом, визуально привлекательном контексте;
- сформировать умение анализировать задачу, выделять повторяющиеся действия и заменять их циклами;
- овладеть приёмами диалоговой отладки, что способствовало развитию самоконтроля и критического мышления;
- научиться сравнивать разные алгоритмические решения и выбирать оптимальное.

Полученные в ходе занятий данные (результаты выполнения заданий, карты наблюдения) легли в основу последующего анализа эффективности разработанной технологии.

3.3 Анализ результатов опытно-экспериментальной работы

Целью данного этапа явился анализ внутригрупповой динамики показателей алгоритмического мышления учащихся 8-х классов по результатам констатирующего и контрольного срезов, а также оценка эффективности разработанной педагогической технологии.

Обработка данных осуществлялась на основе трёх диагностических методик, каждая из которых измеряла отдельный компонент алгоритмического мышления, выделенный в рабочем определении.

Таблица 4

Результаты учащихся экспериментальной группы до эксперимента

Ученик	Когнитивный компонент		Процедурно-деятельностный компонент		Рефлексивный компонент		Уровень алгоритмического мышления
	балл	уровень	балл	уровень	балл	уровень	
В-1	18	высокий	2	низкий	6	средний	репродуктивный
В-2	6	низкий	5	средний	5	средний	репродуктивный
В-3	6	низкий	2	низкий	4	низкий	репродуктивный
В-4	18	высокий	7	средний	2	низкий	репродуктивный
В-5	10	средний	7	средний	5	средний	эвристический
В-6	8	средний	4	низкий	2	низкий	репродуктивный
В-7	18	высокий	5	средний	6	средний	эвристический
В-8	7	низкий	5	средний	4	низкий	репродуктивный
В-9	16	высокий	3	низкий	4	низкий	репродуктивный

ВКР УлГУ

Ученик	Когнитивный компонент		Процедурно-деятельностный компонент		Рефлексивный компонент		Уровень алгоритмического мышления
	балл	уровень	балл	уровень	балл	уровень	
В-10	11	средний	2	низкий	5	средний	репродуктивный
В-11	11	средний	5	средний	7	средний	эвристический
В-12	13	средний	2	низкий	2	низкий	репродуктивный
В-13	10	средний	3	низкий	4	низкий	репродуктивный
Г-1	7	низкий	6	средний	6	средний	репродуктивный
Г-2	17	высокий	2	низкий	6	средний	репродуктивный
Г-3	14	высокий	4	низкий	6	средний	репродуктивный
Г-4	14	высокий	2	низкий	6	средний	репродуктивный
Г-5	7	низкий	5	средний	7	средний	репродуктивный
Г-6	9	средний	3	низкий	3	низкий	репродуктивный
Г-7	12	средний	7	средний	3	низкий	репродуктивный
Г-8	17	высокий	4	низкий	6	средний	репродуктивный
Г-9	8	средний	5	средний	4	низкий	репродуктивный
Г-10	8	средний	3	низкий	7	средний	репродуктивный
Г-11	6	низкий	5	средний	6	средний	репродуктивный
Г-12	6	низкий	4	низкий	4	низкий	репродуктивный
Г-13	13	средний	7	средний	3	низкий	репродуктивный

Таблица 5

Результаты учащихся экспериментальной группы после эксперимента

Ученик	Когнитивный компонент		Процедурно-деятельностный компонент		Рефлексивный компонент		Уровень алгоритмического мышления
	балл	уровень	балл	уровень	балл	уровень	
В-1	19	высокий	8	высокий	9	высокий	творческий
В-2	7	низкий	6	средний	6	средний	репродуктивный
В-3	9	средний	3	низкий	5	средний	репродуктивный
В-4	19	высокий	8	высокий	3	низкий	репродуктивный
В-5	12	средний	9	высокий	7	средний	эвристический
В-6	9	средний	6	средний	4	низкий	репродуктивный
В-7	20	высокий	9	высокий	9	высокий	творческий
В-8	10	средний	6	средний	6	средний	эвристический
В-9	19	высокий	4	низкий	6	средний	репродуктивный
В-10	12	средний	3	низкий	7	средний	репродуктивный
В-11	14	высокий	7	средний	9	высокий	эвристический
В-12	15	высокий	3	низкий	3	низкий	репродуктивный
В-13	12	средний	5	средний	5	средний	эвристический
Г-1	10	средний	8	высокий	8	высокий	эвристический
Г-2	18	высокий	8	высокий	9	высокий	творческий
Г-3	16	высокий	6	средний	8	высокий	эвристический
Г-4	15	высокий	3	низкий	8	высокий	репродуктивный
Г-5	9	средний	7	средний	8	высокий	эвристический
Г-6	11	средний	4	низкий	5	средний	репродуктивный
Г-7	13	средний	8	высокий	4	низкий	репродуктивный

ВКРУДГУ

Ученик	Когнитивный компонент		Процедурно-деятельностный компонент		Рефлексивный компонент		Уровень алгоритмического мышления
	балл	уровень	балл	уровень	балл	уровень	
Г-8	18	высокий	8	высокий	9	высокий	творческий
Г-9	11	средний	6	средний	5	средний	эвристический
Г-10	9	средний	5	средний	8	высокий	эвристический
Г-11	9	средний	6	средний	8	высокий	эвристический
Г-12	8	средний	6	средний	6	средний	эвристический
Г-13	14	высокий	9	высокий	5	средний	эвристический

Таблица 6

Результаты учащихся контрольной группы после эксперимента

Ученик	Когнитивный компонент		Процедурно-деятельностный компонент		Рефлексивный компонент		Уровень алгоритмического мышления
	балл	уровень	балл	уровень	балл	уровень	
К-1	6	низкий	5	средний	4	низкий	репродуктивный
К-2	13	средний	3	низкий	6	средний	репродуктивный
К-3	8	средний	7	средний	2	низкий	репродуктивный
К-4	12	высокий	6	средний	6	средний	эвристический
К-5	14	высокий	4	низкий	5	средний	репродуктивный
К-6	10	средний	7	средний	7	средний	эвристический
К-7	7	низкий	6	средний	5	средний	репродуктивный
К-8	11	средний	4	низкий	6	средний	репродуктивный
К-9	9	средний	5	средний	3	низкий	репродуктивный
К-10	15	высокий	3	низкий	6	средний	репродуктивный
К-11	8	средний	6	средний	4	низкий	репродуктивный
К-12	13	средний	5	средний	3	низкий	репродуктивный
К-13	6	низкий	7	средний	6	средний	репродуктивный
К-14	12	средний	4	низкий	5	средний	репродуктивный
К-15	10	средний	6	средний	2	низкий	репродуктивный

Таблица 7

Динамика когнитивного компонента алгоритмического мышления (%)

Уровень	**Констатирующий срез**	**Контрольный срез**	**Динамика**
Высокий	31%	42%	+11%
Средний	42%	54%	+12%
Низкий	27%	4%	–23%

Диагностика проводилась с помощью субтестов теста структуры интеллекта Р. Амтхауэра (адаптация Л.А. Ясюковой). Результаты отражают способность учащихся к анализу, выявлению закономерностей и

ВГУ ИТ

абстрагированию, что является базой для понимания алгоритмических структур.

Анализ: Наблюдается рост доли учащихся с высоким (+11%) и средним уровнем (+12%) и сокращение группы риска (–23%). Учащиеся стали увереннее выявлять скрытые закономерности в числовых и логических рядах, что напрямую коррелирует с успешным освоением циклических конструкций и условных переходов в программировании.

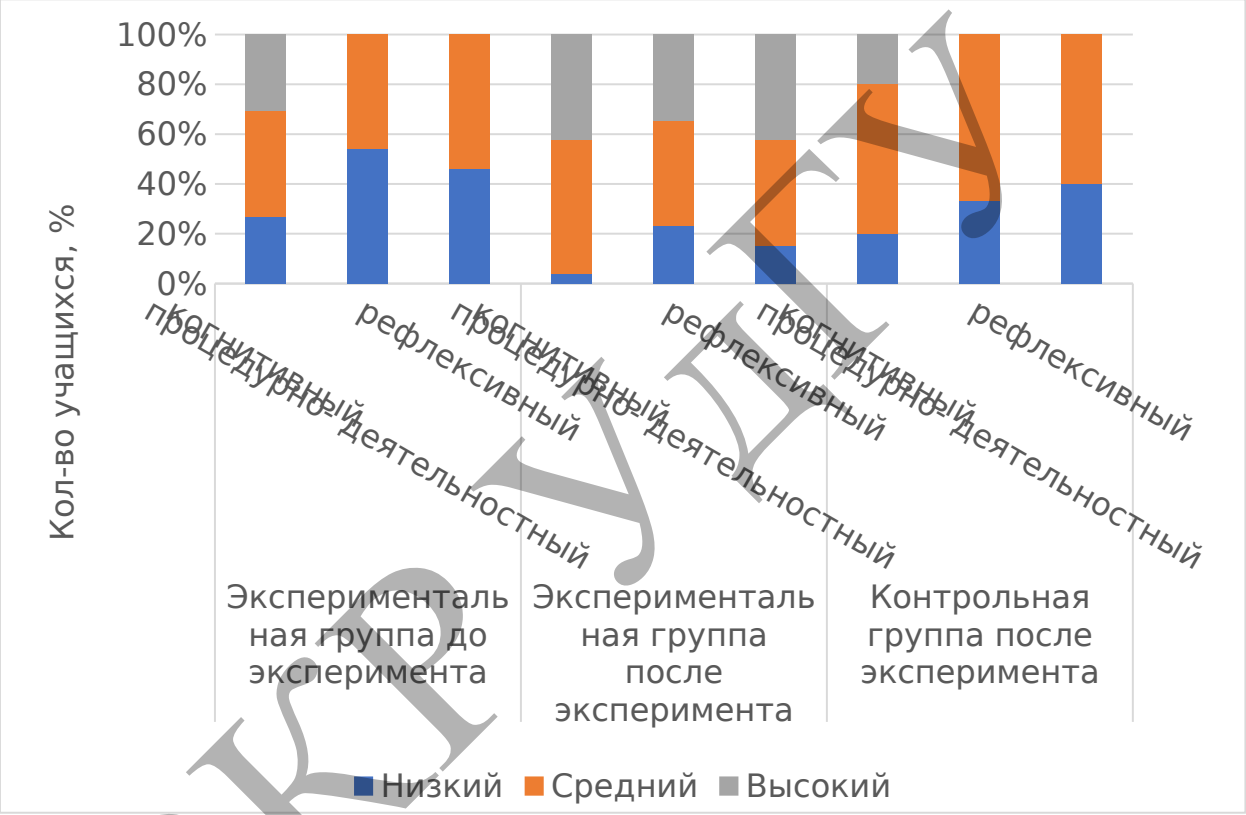


Рис. 11 Распределение уровней компонентов алгоритмического мышления для экспериментальной и контрольной групп

Таблица 8

Динамика процедурно-деятельностного компонента (%)

Уровень	**Констатирующий срез**	**Контрольный срез**	**Динамика**
Высокий	0%	35%	+35%
Средний	46%	42%	–4%
Низкий	54%	23%	–31%

Данный компонент оценивался методикой «Планирование последовательности действий». Фиксировались умения декомпозировать задачу, выстраивать логику, реализовывать и отлаживать алгоритм.

ВКР УлГУ

Анализ: Наиболее выраженный рост зафиксирован в умении самостоятельно декомпозировать игровые задачи и переводить их в последовательность команд. По наблюдению, на контрольном этапе учащиеся реже прибегали к методу случайного подбора, чаще применяли стратегию «план → реализация → отладка». Переход от визуального конструирования в Minecraft к текстовому коду на Python осваивался успешнее благодаря предварительной «физической» отработке логики в игровой среде.

Таблица 9

Динамика рефлексивного компонента (%)

Уровень	**Констатирующий срез**	**Контрольный срез**	**Динамика**
Высокий	0%	42%	+42%
Средний	54%	42%	–12%
Низкий	46%	16%	–30%

Оценка проводилась по методике «Рефлексивный анализ и оптимизация алгоритма». Фиксировалась способность критически оценивать собственное решение, находить логические ошибки, сравнивать альтернативные варианты и обосновывать выбор оптимального алгоритма.

Анализ: Качественный скачок отмечен в преодолении низкого уровня (–30%) и в росте доли учащихся с высоким уровнем (+42%). На констатирующем этапе рефлексия носила ситуативный характер («сработало/не сработало»), тогда как после реализации технологии учащиеся научились аргументированно сравнивать варианты решений, выявлять избыточные операции и оптимизировать код.

Значимость полученный изменений в уровне алгоритмического мышления требуется проверить с помощью статистического критерия. В качестве такого подходит Т-критерий Вилкоксона. Этот критерий применяется для сопоставления показателей, измеренных в двух разных условиях на одной и той же выборке испытуемых. Он позволяет установить не только направленность изменений, но и их выраженность. С его помощью определяется, является ли сдвиг показателей в каком-то одном направлении более интенсивным, чем в другом.

ВКР УлГУ

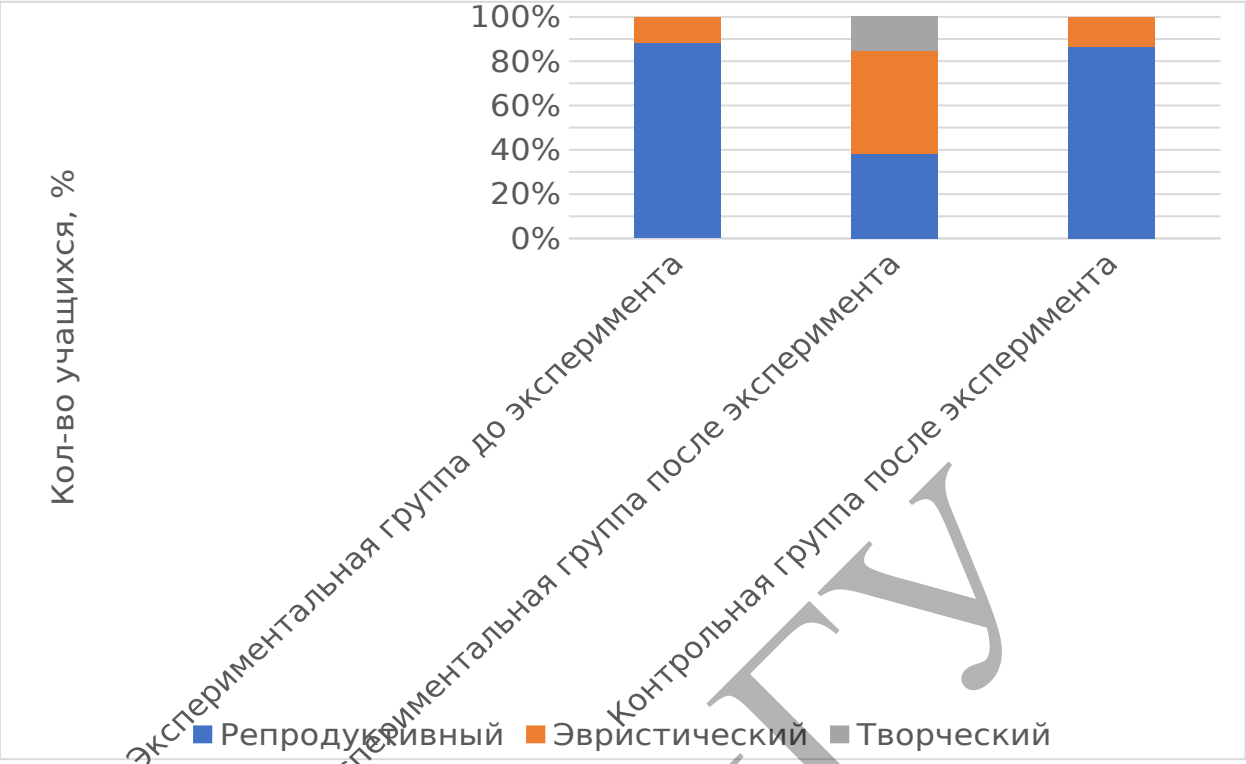


Рис. 12 Распределение уровней алгоритмического мышления для экспериментальной и контрольной групп

Статистические гипотезы могут быть сформулированы следующим образом:

H0: Интенсивность сдвигов уровня алгоритмического мышления в направлении от репродуктивного к эвристическому и далее к творческому не превосходит интенсивности сдвигов в обратном направлении, а также отсутствия этих сдвигов.

H1: Интенсивность сдвигов уровня алгоритмического мышления в направлении от репродуктивного к эвристическому и далее к творческому превышает интенсивность сдвигов в обратном направлении, а также отсутствие этих сдвигов.

При измерении эмпирическое значение критерия установлено в значении $T_{Эмп} = 78$. Критические значения Т-критерия при n=26 следующие: $T_{кр0.01} = 84$, $T_{кр0.05} = 110$.

ВКР УдГУ

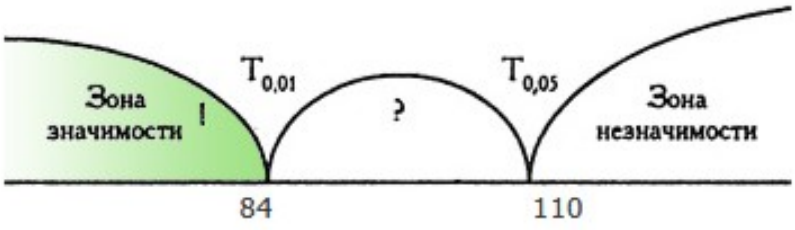


Рис. 13 Ось значимости

Полученное эмпирическое значение $T_{эмп}$ находится в зоне значимости, что дает основание принять гипотезу **H1**: Интенсивность сдвигов уровня алгоритмического мышления в направлении от репродуктивного к эвристическому и далее к творческому превышает интенсивность сдвигов в обратном направлении, а также отсутствие этих сдвигов. Полученные в ходе эксперимента изменения являются значимыми.

Вывод: наблюдается статистически значимый рост уровня алгоритмического мышления в экспериментальной группе после проведённого обучения.

Значимость отличия в уровне алгоритмического мышления между контрольной и экспериментальной группами после эксперимента также требуется проверить с помощью статистического критерия. В качестве такого подходит U-критерий Манна-Уитни. Этот критерий применяется для оценки различий между двумя выборками по уровню какого-либо признака.

Статистические гипотезы могут быть сформулированы следующим образом:

H0: Уровень алгоритмического мышления в контрольной группе не ниже уровня алгоритмического мышления в экспериментальной группе.

H1: Уровень алгоритмического мышления в контрольной группе ниже уровня алгоритмического мышления в экспериментальной группе.

При измерении эмпирическое значение критерия установлено в значении $U_{Эмп}$ = 97. Критические значения U-критерия при n_1=26, n_2=15 следующие: $U_{кр0.01}$ = 108, $U_{кр0.05}$ = 133.

ВКР УлГУ

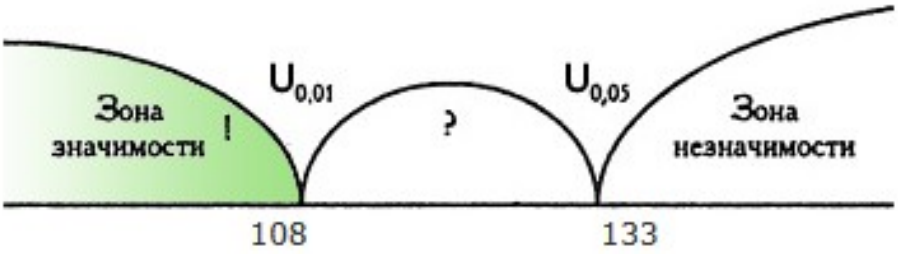


Рис. 14 Ось значимости

Полученное эмпирическое значение $T_{эмп}$ находится в зоне значимости, что дает основание принять гипотезу **H1**: Уровень алгоритмического мышления в контрольной группе ниже уровня алгоритмического мышления в экспериментальной группе.

Вывод: наблюдается статистически значимое превышение уровня алгоритмического мышления в экспериментальной группе над уровнем владения алгоритмического мышления в контрольной группе после проведённого обучения. Это подтверждает эффективность применённой в экспериментальной группе методики по сравнению с традиционным обучением, при этом контрольная группа сохранила результаты, близкие к исходному уровню экспериментальной группы.

Выводы по главе 3

Разработана система диагностики эффективности педагогической технологии развития алгоритмического мышления. Для оценки результатов исследования были подобраны методики, позволяющие диагностировать когнитивный, процедурно-деятельностный и рефлексивный компоненты алгоритмического мышления. Определены критерии и уровни сформированности исследуемого качества, что обеспечило возможность объективного анализа изменений, происходящих в ходе эксперимента.

Представлено описание реализации разработанной педагогической технологии в образовательном процессе. Формирующий этап эксперимента был проведён на базе МБОУ «СОШ № 19» города Ижевска с учащимися 8-х классов. В процессе обучения использовалась серия практических занятий в

ВКР УдГУ

среде Minecraft с применением языка Python. Реализация технологии обеспечила включение учащихся в деятельность по анализу задач, построению алгоритмов, программированию, отладке и рефлексии результатов.

Проведён анализ результатов опытно-экспериментальной работы. Полученные данные свидетельствуют о положительной динамике по всем диагностируемым компонентам алгоритмического мышления. У учащихся повысился уровень развития логического анализа, навыков декомпозиции и планирования, способности к реализации и оптимизации алгоритмов. Качественный анализ показал рост учебной мотивации, самостоятельности и осознанности при решении алгоритмических задач. Полученные результаты подтвердили эффективность разработанной педагогической технологии и выдвинутую гипотезу исследования.

Таким образом, опытно-экспериментальная работа показала, что использование игровой иммерсивной среды Minecraft в сочетании с программированием на языке Python создаёт благоприятные условия для развития алгоритмического мышления учащихся 8-х классов. Разработанная педагогическая технология способствует формированию ключевых компонентов алгоритмического мышления и может быть рекомендована для использования в процессе обучения информатике при изучении модуля «Алгоритмы и программирование».

ВКР УлГУ

ЗАКЛЮЧЕНИЕ

В условиях цифровой трансформации общества развитие алгоритмического мышления становится одной из важнейших задач современного образования. Требования Федерального государственного образовательного стандарта основного общего образования ориентируют процесс обучения информатике не только на формирование предметных знаний и умений, но и на развитие метапредметных результатов, связанных со способностью учащихся анализировать информацию, планировать деятельность, осуществлять поиск эффективных способов решения задач и оценивать полученные результаты. В связи с этим проблема развития алгоритмического мышления у учащихся 8-х классов приобретает особую актуальность.

Целью данной выпускной квалификационной работы являлась разработка педагогической технологии развития алгоритмического мышления у учащихся 8-х классов на уроках информатики при изучении модуля «Алгоритмы и программирование».

В ходе исследования был проведён анализ психолого-педагогической и методической литературы по проблеме алгоритмического мышления. Установлено, что алгоритмическое мышление представляет собой интегративное качество личности, включающее способность к анализу условий задачи, декомпозиции сложной проблемы на подзадачи, построению, реализации и отладке алгоритмов, а также к рефлексивной оценке и оптимизации полученных решений. На основе анализа научных подходов были выделены основные компоненты алгоритмического мышления: когнитивный, процедурно-деятельностный и рефлексивный, а также уровни его развития – репродуктивный, эвристический и творческий.

Проведённое исследование показало, что наиболее перспективными средствами развития алгоритмического мышления учащихся основной школы являются игровая форма обучения, иммерсивные образовательные среды и

ВКР УдГУ

программирование. Особое внимание было уделено возможностям использования среды Minecraft как средства создания наглядного, мотивирующего и практико-ориентированного образовательного пространства. Было установлено, что последовательный переход от визуального взаимодействия с объектами игровой среды к текстовому программированию на языке Python позволяет снизить когнитивную нагрузку учащихся и сосредоточить их внимание на логике построения алгоритмов.

На основе результатов теоретического исследования была разработана педагогическая технология развития алгоритмического мышления учащихся 8-х классов. Технология основана на сочетании системно-деятельностного, метапредметного, проблемного и игрового подходов и включает четыре взаимосвязанных этапа: иммерсивно-мотивационный, экспериментально-моделирующий, абстрактно-конструкторский и рефлексивно-интегрирующий. В рамках технологии была осуществлена модернизация содержания модуля «Алгоритмы и программирование», разработана система прогрессивно усложняющихся алгоритмических задач, а также созданы средства обучения, обеспечивающие её практическую реализацию. В качестве основного средства обучения была разработана образовательная карта «Алгоритмический мир» в среде Minecraft, дополненная комплектом дидактических заданий и алгоритмических кейсов.

Для проверки эффективности разработанной педагогической технологии была организована опытно-экспериментальная работа на базе МБОУ «СОШ № 19» города Ижевска с учащимися 8-х классов. Диагностика уровня развития алгоритмического мышления проводилась по трём направлениям: оценка когнитивного, процедурно-деятельностного и рефлексивного компонентов. Анализ результатов констатирующего и контрольного этапов эксперимента показал положительную динамику по всем исследуемым показателям.

В ходе реализации технологии учащиеся продемонстрировали повышение уровня логического анализа, способности выявлять

ВКР УдГУ

закономерности, планировать последовательность действий и осуществлять декомпозицию задач. Существенные изменения были зафиксированы в развитии навыков построения, реализации и отладки алгоритмов. Повысилась способность учащихся осуществлять рефлексивный анализ собственной деятельности, выявлять ошибки, сравнивать различные способы решения задач и выбирать наиболее эффективные варианты. Кроме того, качественный анализ результатов показал рост учебной мотивации, самостоятельности и осознанности при выполнении алгоритмических задач.

Полученные результаты подтвердили выдвинутую гипотезу исследования о том, что развитие алгоритмического мышления учащихся 8-х классов будет более эффективным при использовании педагогической технологии, основанной на системе прогрессивно усложняющихся алгоритмических задач, интеграции игровой иммерсивной среды Minecraft и программирования на языке Python, а также организации поэтапного перехода от визуального взаимодействия к текстовому программированию.

Таким образом, цель исследования достигнута, поставленные задачи решены в полном объёме. Разработанная педагогическая технология доказала свою эффективность и может быть рекомендована для использования в образовательной практике учителей информатики при изучении модуля «Алгоритмы и программирование». Перспективы дальнейших исследований связаны с расширением возможностей применения иммерсивных образовательных сред для развития алгоритмического мышления учащихся различных возрастных групп и совершенствованием методического обеспечения данного направления обучения.

ВКР УлГУ

ИСТОЧНИКИ ИНФОРМАЦИИ

1. Амтхауэр, Р. Тест структуры интеллекта : руководство по использованию / Р. Амтхауэр ; адаптация Л. А. Ясюковой. – Санкт-Петербург : ГП «Иматон», 2002. – 48 с.

2. Босова, Л. Л. Информатика. 7–9 классы : сборник рабочих программ / Л. Л. Босова, А. Ю. Босова. – Москва : БИНОМ. Лаборатория знаний, 2021. – 472 с.

3. Босова, Л. Л. Развитие алгоритмического мышления школьников на уроках информатики / Л. Л. Босова, Т. В. Бурмистрова // Информатика и образование. – 2019. – № 6. – С. 12–19.

4. Вербицкий, А. А. Активное обучение в высшей школе: контекстный подход / А. А. Вербицкий. – Москва : Высшая школа, 1991. – 207 с.

5. Выготский, Л. С. Психология развития человека / Л. С. Выготский. – Москва : Смысл : Эксмо, 2005. – 1136 с.

6. Гальперин, П. Я. Лекции по психологии : учебное пособие / П. Я. Гальперин. – Москва : КДУ, 2007. – 400 с.

7. Горячев, А. В. Информатика. Алгоритмика. 5–6 классы : методическое пособие / А. В. Горячев, Н. И. Суворова. – Москва : Баласс, 2018. – 144 с.

8. Ершов, А. П. Избранные труды / А. П. Ершов ; под ред. Ю. Л. Ершова. – Новосибирск : Наука, 1994. – 496 с.

9. Ершов, А. П. Школьная информатика в СССР: от грамотности к культуре / А. П. Ершов // Информатика и образование. – 1987. – № 6. – С. 3–11.

10. Загвязинский, В. И. Методология и методы психолого-педагогического исследования / В. И. Загвязинский, Р. Атаханов. – Москва : Академия, 2012. – 208 с.

ВКР У ДГУ

11. Крылов, С. С. Государственная итоговая аттестация по информатике: анализ результатов и методические рекомендации / С. С. Крылов // Педагогическая информатика. – 2024. – № 2. – С. 34–42.

12. Кузнецов, А. А. Информатика. Методические рекомендации к преподаванию в условиях цифровой трансформации / А. А. Кузнецов, С. А. Бешенков. – Москва : Просвещение, 2023. – 192 с.

13. Лапчик, М. П. Методика преподавания информатики : учебник / М. П. Лапчик, И. Г. Семакин, Е. К. Хеннер ; под ред. М. П. Лапчика. – 5-е изд., стер. – Москва : Академия, 2018. – 416 с.

14. Львовский, В. А. Применение Minecraft в образовании: зарубежный опыт и российские практики / В. А. Львовский, Е. С. Никитина // Информационные технологии в образовании. – 2022. – № 4 (63). – С. 56–64.

15. Минпросвещения России. Федеральный государственный образовательный стандарт основного общего образования : утвержден приказом Минпросвещения России от 31.05.2021 № 287. – Москва, 2021. – 56 с.

16. Минпросвещения России. Федеральная рабочая программа основного общего образования по информатике (для 7–9 классов). – Москва, 2023. – 118 с.

17. Морзе, Н. В. Методика навчання інформатики : підручник / Н. В. Морзе. – Київ : Видавнича група BHV, 2019. – 480 с. (на укр. яз.)

18. Окулов, С. М. Программирование в алгоритмах / С. М. Окулов. – Москва : БИНОМ. Лаборатория знаний, 2012. – 344 с.

19. Павлов, Д. И. Иммерсивные технологии в школьном образовании: возможности и риски / Д. И. Павлов // Психологическая наука и образование. – 2023. – Т. 28, № 1. – С. 85–96.

20. Правительство РФ. Национальная программа «Цифровая экономика Российской Федерации» : утверждена распоряжением Правительства РФ от 28.07.2017 № 1632-р– Москва, 2017.

ВКР УдГУ

21. Причинин, А. Е. Педагогические технологии в профессиональном образовании / А. Е. Причинин. – Ижевск : Удмуртский университет, 2023. – 184 с.

22. Селевко, Г. К. Энциклопедия образовательных технологий : в 2 т. Т. 1 / Г. К. Селевко. – Москва : НИИ школьных технологий, 2006. – 816 с.

23. Семакин, И. Г. Информатика. 8 класс : учебник / И. Г. Семакин, Л. А. Залогова, С. В. Русаков [и др.]. – Москва : БИНОМ. Лаборатория знаний, 2021. – 208 с.

24. Смирнова, Е. О. Психологические особенности современных подростков: цифровое поколение / Е. О. Смирнова, Н. Г. Салмина // Вестник Московского университета. Серия 14: Психология. – 2019. – № 4. – С. 88–106.

25. Солдатова, Г. У. Цифровая социализация в контексте возрастного развития / Г. У. Солдатова, Е. И. Рассказова // Психологический журнал. – 2021. – Т. 42, № 6. – С. 32–43.

26. Степин, В. С. Теоретическое знание / В. С. Степин. – Москва : Прогресс-Традиция, 2000. – 744 с.

27. Тихонова, М. А. Формирование алгоритмического мышления учащихся основной школы в процессе обучения программированию : диссертация … кандидата педагогических наук : 13.00.02 / Тихонова Мария Алексеевна. – Москва, 2017. – 194 с.

28. Тихонова, М. А. От визуального к текстовому программированию: методический конструктор / М. А. Тихонова // Информатика и образование. – 2020. – № 1. – С. 24–31.

29. Усова, А. В. Формирование у школьников научных понятий в процессе обучения / А. В. Усова. – Москва : Педагогика, 1986. – 176 с.

30. Хеннер, Е. К. Информационные компетенции и алгоритмическое мышление: системный подход / Е. К. Хеннер // Образование и наука. – 2016. – № 5 (134). – С. 25–41.

ВКР УдГУ

31. Хуторской, А. В. Метапредметное содержание и результаты образования: как реализовать ФГОС / А. В. Хуторской // Народное образование. – 2014. – № 4. – С. 157–166.

32. Хуторской, А. В. Педагогическая инноватика / А. В. Хуторской. – Москва : Академия, 2012. – 256 с.

33. Цветкова, М. С. Информатика. Формирование алгоритмического мышления: от начальной до старшей школы / М. С. Цветкова, О. Б. Богомолова // Информатика в школе. – 2018. – № 9. – С. 4–13.

34. Шапиро, К. В. Среда Minecraft как инструмент развития пространственного и алгоритмического мышления / К. В. Шапиро // Игровые технологии в образовании. – 2022. – № 3. – С. 15–24.

35. Эльконин, Д. Б. Психология обучения младшего школьника / Д. Б. Эльконин. – Москва : Знание, 1974. – 64 с.

36. Blakemore, S. J. The adolescent brain / S. J. Blakemore. – London : Routledge, 2018. – 245 p.

37. Brennan, K. New frameworks for studying and assessing the development of computational thinking / K. Brennan, M. Resnick // Proceedings of the 2012 Annual Meeting of the American Educational Research Association. – Vancouver, 2012. – P. 1–25.

38. Diamond, A. Executive functions / A. Diamond // Annual Review of Psychology. – 2013. – Vol. 64. – P. 135–168.

39. Flavell, J. H. Metacognitive development / J. H. Flavell // Metacognition: Knowing about knowing / ed. by J. Metcalfe, A. Shimamura. – Cambridge, MA : MIT Press, 1994. – P. 85–103.

40. Kuhn, D. The development of scientific thinking / D. Kuhn // Handbook of child psychology. 6th ed. Vol. 2. – New York : Wiley, 2006. – P. 371–425.

41. Prensky, M. Digital natives, digital immigrants / M. Prensky // On the Horizon. – 2001. – Vol. 9, No. 5. – P. 1–6.

ВКР УдГУ

42. Resnick, M. Scratch: programming for all / M. Resnick, J. Maloney, A. Monroy-Hernández [et al.] // Communications of the ACM. – 2009. – Vol. 52, No. 11. – P. 60–67.

43. Steinberg, L. Age of opportunity: lessons from the new science of adolescence / L. Steinberg. – Boston : Houghton Mifflin Harcourt, 2014. – 272 p.

44. Twenge, J. M. iGen: why today's super-connected kids are growing up less rebellious, more tolerant, less happy – and completely unprepared for adulthood / J. M. Twenge. – New York : Atria Books, 2017. – 352 p.

45. Wing, J. M. Computational thinking / J. M. Wing // Communications of the ACM. – 2006. – Vol. 49, No. 3. – P. 33–35.

46. Zelazo, P. D. The development of executive function in childhood / P. D. Zelazo, S. M. Carlson // Handbook of developmental cognitive neuroscience. 2nd ed. – Cambridge, MA : MIT Press, 2012. – P. 575–590.

47. HeadHunter. Тренды на рынке труда 2024 : аналитический обзор [Электронный ресурс] / HeadHunter. – Режим доступа: https://hh.ru/article/trendy2024 .

48. Minecraft Education Edition. Официальный сайт [Электронный ресурс]. – Режим доступа: https://education.minecraft.net/

49. Python Software Foundation. The Python Tutorial [Электронный ресурс]. – Режим доступа: https://docs.python.org/3/tutorial/

50. Stack Overflow. 2024 Developer Survey [Электронный ресурс] / Stack Overflow. – Режим доступа: https://survey.stackoverflow.co/2024/).

51. Яндекс. Как мы проверяем алгоритмические навыки на собеседованиях [Электронный ресурс] / Блог Яндекса. – Режим доступа: https://yandex.ru/blog/interviews/algorithms.

52. Code.org. Час кода: методические материалы [Электронный ресурс]. – Режим доступа: https://code.org/.

53.Федеральный институт педагогических измерений (ФИПИ). Единый государственный экзамен по информатике: аналитические и методические

ВКР УлГУ

материалы [Электронный ресурс]. – Режим доступа: https://fipi.ru/ege/analiticheskie-i-metodicheskie-materialy

ВКР УдГУ

ПРИЛОЖЕНИЕ А

Фрагмент учебной программы по предмету «Информатика», модуль «Алгоритмы и программирование» (8 класс)

1. Пояснительная записка

Данный фрагмент рабочей программы разработан в рамках модернизации модуля «Алгоритмы и программирование» для 8-го класса. Программа ориентирована на развитие алгоритмического мышления учащихся через систему прогрессивно усложняющихся алгоритмических задач.

Основой реализации является интеграция:

- иммерсивной игровой среды Minecraft,
- языка программирования Python,
- поэтапного перехода от визуального программирования к текстовому,
- системы практико-ориентированных заданий.

Программа направлена на формирование у учащихся умений анализировать условие задачи, выполнять декомпозицию, проектировать алгоритмы, осуществлять отладку и проводить рефлексию результатов.

2. Цели изучения модуля

Развитие алгоритмического мышления учащихся 8-х классов в процессе изучения модуля «Алгоритмы и программирование» через практическую деятельность в иммерсивной игровой среде и программирование на языке Python.

3. Планируемые результаты обучения

Учащиеся должны:

Личностные результаты:

- проявлять устойчивую учебную мотивацию к изучению программирования;
- развивать самостоятельность в решении задач.

Метапредметные результаты:

- осуществлять анализ условия задачи;
- выполнять декомпозицию;
- планировать последовательность действий;
- осуществлять рефлексию и анализ результата.

Предметные результаты:

- понимать и выполнять линейные, циклические и вложенные алгоритмы;
- использовать конструкции Python (циклы, условия, строки);
- применять алгоритмические конструкции в игровой среде Minecraft.

4. Содержание учебного модуля (тематическое планирование)

№	Тема занятия	Основное содержание	Кол-во часов
1	Введение в Minecraft: Python и диалоговая отладка	Знакомство с игровой средой, выполнение линейных алгоритмов, работа с маршрутом агента	1

ВКР УлГУ

2	Цикл с условием while	Построение моста заданной длины, анализ повторяющихся действий	1
3	Цикл с параметром for	Автоматическая расстановка объектов через равные интервалы	1
4	Вложенные циклы	Построение прямоугольных структур (стены, площадки)	1
5	Работа со строками	Создание простых игровых команд, обработка текстовых данных	1
6	Анализ и оптимизация алгоритмов	Оптимизация готовых алгоритмов, уменьшение избыточности кода	1

5. Организация учебной деятельности

Каждое занятие включает:

- постановку проблемной ситуации в игровой среде Minecraft;
- анализ условия задачи;
- проектирование алгоритма;
- реализацию решения;
- отладку и исправление ошибок;
- рефлексию и обсуждение результата.

6. Формы и методы обучения

Методы обучения:

- проблемное обучение;
- учебный эксперимент;
- проектная деятельность;
- рефлексия;
- сравнительный анализ.

Формы организации:

- фронтальная работа;
- парная работа;
- групповая работа;
- индивидуальная деятельность;
- коллективное обсуждение.

7. Средства обучения

- иммерсивная игровая среда Minecraft;
- язык программирования Python;
- библиотека mcpi;
- задания в игровой форме;
- алгоритмические задачи различного уровня сложности.

8. Особенности реализации программы

Программа реализует принцип постепенного усложнения:

- от линейных алгоритмов к циклическим;
- от циклов к вложенным конструкциям;
- от визуального выполнения к текстовому программированию;
- от выполнения алгоритма к его анализу и оптимизации.

9. Ожидаемый результат

ВКР УдГУ

Ожидается повышение уровня алгоритмического мышления учащихся, выражающееся в способности:

- самостоятельно анализировать задачи;
- проектировать алгоритмы;
- реализовывать их в программной среде;
- выполнять отладку и оптимизацию решений;
- применять алгоритмические принципы в новых ситуациях.

ВКР УДГУ

ПРИЛОЖЕНИЕ Б

Сборник алгоритмических задач

Тема 1. Введение в Minecraft: Python и диалоговая отладка

Задача 1. Проведи агента домой

Условие.

Игровой агент находится в начале маршрута.

Необходимо с помощью блоков визуального программирования провести агента до финиша по отмеченной дорожке.

Учащийся должен:

определить последовательность действий;

расположить команды в правильном порядке;

проверить корректность выполнения маршрута.

Задача 2. Найди ошибку

Условие.

Агент должен пройти по маршруту

В программе одна из команд записана неверно.

Необходимо:

запустить алгоритм;

определить место возникновения ошибки;

исправить последовательность действий.

Тема 2. Цикл с условием while

Задача 3. Построй деревянный мост

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
x, y, z = mc.player.getTilePos()
length = 10
i = 0
while i < length:
    mc.setBlock(x + i, y - 1, z, 5)
    i += 1
```

Измените программу так, чтобы мост имел длину 20 блоков.

Задача 4. Мост до другого берега

Условие.

Ширина реки неизвестна.

Необходимо построить мост до тех пор, пока под агентом находится вода.

Подсказка.

Используйте цикл while.

Тема 3. Цикл с параметром for

Задача 5. Забор вокруг участка

Условие.

Постройте прямой забор длиной 15 блоков.

Используйте цикл for.

```
for i in range(15):
    mc.setBlock(x + i, y, z, 85)
```

ВКР УлГУ

Задача 6. Аллея фонарей
Условие.
Необходимо установить 10 фонарей через одинаковые промежутки.
Самостоятельно определите шаг установки.

Задача 7. Цветная дорожка
Условие.
Постройте дорожку длиной 20 блоков.
Каждый второй блок должен иметь другой цвет.

Задача 8. Лестница
Условие.
Постройте лестницу из 10 ступеней.
Каждая следующая ступень должна располагаться на один блок выше предыдущей.

Тема 4. Вложенные циклы
Задача 9. Пол для дома
Условие.
Постройте квадратный пол размером 10×10 блоков.
Используйте вложенные циклы.

Задача 10. Стена крепости
Условие.
Постройте стену:
длина 15 блоков;
высота 5 блоков.

Задача 11. Башня
Условие.
Создайте башню:
основание 7×7;
высота 12 блоков.
Для решения разбейте задачу на подзадачи.

Задача 12. Шахматная площадка
Условие.
Постройте поле 8×8 из двух видов блоков.
Необходимо чередовать блоки по принципу шахматной доски.

Тема 5. Работа со строками
Задача 13. Команда приветствия
Условие.
Попросите пользователя ввести своё имя.

ВКР УлГУ

Выведите сообщение:
Привет, <имя игрока>!

Задача 14. Построй дом по команде
Условие.
Пользователь вводит слово:
дом
После этого программа строит небольшой дом.

Задача 15. Волшебные команды
Условие.
Реализуйте команды:
мост
башня
фонари
Каждая команда должна запускать отдельный алгоритм.

Задача 16. Система управления строителем
Условие.
Создайте текстовое меню:
1 — построить мост
2 — построить стену
3 — построить башню
Пользователь выбирает действие.
Программа выполняет соответствующий алгоритм.

Тема 6. Анализ и оптимизация алгоритмов
Задача 17. Найди лишние команды
Условие.
Дан алгоритм:
mc.setBlock(x, y, z, 5)
mc.setBlock(x + 1, y, z, 5)
mc.setBlock(x + 2, y, z, 5)
mc.setBlock(x + 3, y, z, 5)
mc.setBlock(x + 4, y, z, 5)
Перепишите его с использованием цикла.

Задача 18. Сравни два решения
Условие.
Предложены два алгоритма построения стены.
Необходимо:
определить более эффективный вариант;
объяснить выбор.

Задача 19. Оптимизация строительства башни

ВКР УлГУ

Условие.
Дан длинный алгоритм строительства башни.
Необходимо:
сократить количество строк программы;
использовать циклы;
сохранить результат работы.

ВКР УдГУ

ПРИЛОЖЕНИЕ В

Диагностический материал для определения уровня алгоритмического мышления учащихся 8-х классов

Блок 1. Диагностика когнитивного компонента

Задание 1 (4 балла)

Установите закономерность и продолжите последовательность:

3, 6, 12, 24, ____, ____

Ответ: 48, 96.

Ключ оценивания:

оба числа указаны верно — 4 балла;

одно число верно — 2 балл;

ответ неверен — 0 баллов.

Задание 2 (6 балла)

Для того чтобы приготовить чай необходимо:

Налить воду в чайник.

Включить чайник.

Подождать, пока вода закипит.

Налить кипяток в кружку.

Положить пакетик чая.

Укажите действие, которое расположено не на своём месте.

Правильный ответ:

Пункт 5 должен быть выполнен до пункта 4.

Ключ оценивания:

указана ошибка и объяснена причина — 6 балла;

указана только ошибка — 3 балла;

ответ частично верен — 1 балл;

неверный ответ — 0 баллов.

Задание 3 (6 балла)

Выберите наиболее рациональный способ решения задачи.

Задача: необходимо разложить 50 тетрадей по 5 штук в каждую стопку.

□ Выполнять подсчёт вручную для каждой стопки.

□ Использовать повторяющееся действие до окончания всех тетрадей.

□ Раскладывать тетради в случайном порядке.

Правильный ответ:

Использовать повторяющееся действие.

Ключ оценивания:

правильный выбор и обоснование — 6 балла;

правильный выбор без обоснования — 3 балла;

частично верное объяснение — 1 балл;

неверный ответ — 0 баллов.

Задание 4 (4 балла)

Для получения доступа в библиотеку необходимо:

если читатель зарегистрирован, выдать книгу;

если не зарегистрирован, направить на регистрацию.

ВКР УлГУ

Какая алгоритмическая конструкция должна использоваться?
□ Следование
□ Ветвление
□ Повторение
Ответ: ветвление.
Ключ оценивания:
верно — 4 балла;
неверно — 0 баллов.
Максимум за когнитивный компонент — 20 баллов

Блок 2. Диагностика процедурно-деятельностного компонента
Задание 5 (2 балла)
Составьте алгоритм подготовки рабочего места к выполнению практической работы на компьютере.
Пример решения:
Включить компьютер.
Войти в систему.
Подготовить необходимые материалы.
Запустить нужную программу.
Проверить готовность к работе.
Ключ оценивания:
алгоритм полный и логически корректный — 2 балла;
имеются незначительные недочёты — 1 балл;
алгоритм некорректен — 0 баллов.
Задание 6 (3 балла)
Составьте алгоритм определения чётности числа.
Пример решения:
Ввести число.
Найти остаток от деления на 2.
Если остаток равен 0 — число чётное.
Иначе число нечётное.
Ключ оценивания:
алгоритм полностью корректен — 3 балла;
допущена одна неточность — 2 балла;
идея решения присутствует — 1 балл;
неверный ответ — 0 баллов.
Задание 7 (2 балла)
Найдите ошибку в алгоритме.
1. Ввести число.
2. Разделить число на 2.
3. Если остаток равен 1, число чётное.
4. Вывести результат.
Ответ:
Если остаток равен 1, число нечётное.
Ключ оценивания:

ВКР УдГУ

ошибка найдена и исправлена — 2 балла;
ошибка указана частично — 1 балл;
неверно — 0 баллов.

Задание 8 (3 балла)

Предложите собственный алгоритм подготовки к контрольной работе.

Ключ оценивания:
последовательность действий логична, самостоятельна и полна — 3 балла;
имеются отдельные нарушения логики — 2 балла;
представлен фрагментарный алгоритм — 1 балл;
ответ отсутствует — 0 баллов.

Максимум за процедурно-деятельностный компонент — 10 баллов

Блок 3. Диагностика рефлексивного компонента

Задание 9 (3 балла)

Сравните два варианта решения задачи.

Правильный ответ:
Более эффективен вариант Б, поскольку предусматривает анализ материала, структурирование информации и контроль усвоения.

Ключ оценивания:
выбран вариант Б и приведено обоснование — 3 балла;
выбран вариант Б без объяснения — 2 балла;
частичное объяснение — 1 балл;
неверный ответ — 0 баллов.

Задание 10 (3 балла)

Ученик составил алгоритм сборки модели, однако результат оказался неверным.

Какие действия необходимо выполнить для поиска ошибки?

Пример ответа:
Проверить порядок действий.
Найти шаг, после которого появилась ошибка.
Сравнить алгоритм с образцом.
Исправить ошибку.
Повторно выполнить алгоритм.

Ключ оценивания:
указана полная процедура анализа ошибки — 3 балла;
приведены основные действия — 2 балла;
ответ частично верен — 1 балл;
неверный ответ — 0 баллов.

Задание 11 (4 балла)

Предложите способ улучшения алгоритма.

1. Прочитать задачу.
2. Прочитать задачу ещё раз.
3. Прочитать задачу третий раз.
4. Начать решение.

Пример ответа:
Прочитать задачу.

ВКР УлГУ

Выделить известные данные.
Определить цель.
Составить план решения.
Выполнить решение.
Проверить результат.
Ключ оценивания:
предложен рациональный улучшенный алгоритм с обоснованием — 4 балла;
алгоритм улучшен, но обоснование неполное — 3 балла;
имеются отдельные улучшения — 2 балла;
ответ частично верен — 1 балл;
неверный ответ — 0 баллов.

Определение уровня алгоритмического мышления

Уровень	Критерий
Репродуктивный	хотя бы один компонент имеет низкий уровень
Эвристический	все компоненты сформированы не ниже среднего уровня
Творческий	все компоненты сформированы на высоком уровне

ВКР УлГУ

ПРИЛОЖЕНИЕ Г

Фотоматериалы с занятий





ВКР УДГ








```
from mcpi.minecraft import Minecraft
mc= minecraft.create()
pos = mc.player.getPos()
 x=int(pos.x)
 y=int(pos.y)
.z=int(pos.z)
.l= 5
block=1
i=0
while i<l:
    mc.setBlock(x+i, y, z, block)
```

ВКР УДГ