

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.П. АСТАФЬЕВА»
(КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики
Выпускающая кафедра: информатики и информационных технологий в
образовании

Машукова Ксения Владимировна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Обучение программированию в психолого-педагогических классах
через разработку дидактических материалов в среде Scratch**

Направление подготовки:
44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль) образовательной программы:
Математика и Информатика

ДОПУСКАЮ К ЗАЩИТЕ
Заведующий кафедрой
профессор, доктор пед. наук Н.И. Длак

10.06.2025

(дата, подпись)

Научный руководитель
канд. пед. наук, доцент И.А. Яшина

10.06.2025

Дата защиты

17.06.2025

Обучающийся
К.В. Машукова

Машу

Оценка удовлетворительно

Пронисью

Красноярск 2025

Содержание

Введение.....	3
Глава 1. Теоретические основы обучения программированию в психолого-педагогических классах.....	7
1.1. Особенности обучения программированию учащихся психолого-педагогического профиля.....	7
1.2. Дидактический потенциал среды Scratch для психолого-педагогического образования.....	17
Глава 2. Практическая разработка и апробация комплекса дидактических материалов по Scratch для психолого-педагогических классов.....	26
2.1. Структура и содержание комплекса дидактических материалов.....	26
2.2. Апробация комплекса дидактических материалов в программе «Школа юного педагога».....	40
Заключение.....	44
Библиографический список.....	46
Приложение А.....	50
Приложение Б.....	52
Приложение В.....	54

Введение

Современные технологии, окружающие человека в любой сфере деятельности, внедрились в нашу жизнь, как незаменимые механизмы. Искусственный интеллект, роботизированные системы, виртуальная реальность и другие технологии научились заменять людей. В различной деятельности человека, связанной, например, с производственными процессами или автоматизацией рутинных задач, человеческий ресурс встает на второй план, а в ближайшие 10-15 лет, возможно, перестанет иметь необходимость в принципе.

Но существуют и другие профессиональные области, в которых до сих пор существует необходимость человеческого фактора. Профессии из таких областей, как медицина, образование и социальная работа являются человекоцентрированными, то есть ориентированными на самих же людей. Люди данных профессий занимают важную роль в нашей жизни и напрямую влияют на уровень жизни страны в целом. Таким образом, перед системой образования стоит задача ускорения процесса подготовки будущих учителей, врачей и социальных работников. Особую актуальность имеет вопрос нехватки работников сферы просвещения.

Данная проблема нашла свое решение во внедрении концепции профильных классов в общеобразовательные организации, осуществляющие образовательную деятельность по программам среднего общего образования. Ожидаемый результат федерального проекта состоит в решении сразу нескольких проблем: неразумный подход к выбору будущей профессии и учебного заведения у выпускников школы, в результате чего появляется другая проблема - массовый отток студентов в другие учреждения. Такое решение проблемы, как создание, профильных классов позволит будущим выпускникам школ своевременно определиться с предметами для сдачи единых государственных экзаменов, а также определиться с выбором учебного заведения и специальности, по которой они хотят обучаться.

Возвращаясь, к человекоцентрированным профессиям, упоминаемым выше, хочется отметить аспект выбора таких направлений в профессиональной деятельности. Люди, планирующие работать в сфере «человек-человек» должны быть уверены в том, что данная область подходит им по разным параметрам. Таким образом, профориентационный аспект также имеет значимость в концепции профильных классов, в частности, классов с психолого-педагогическим уклоном.

Говоря об особенностях обучения психолого-педагогических классов, стоит отметить, что упор происходит на предметы гуманитарного цикла, тогда как в современных реалиях важно иметь разносторонние качества, которые могут пригодиться в быстро изменяющихся условиях жизни. «Soft skills» - гибкие навыки, обладая которыми человек развивается на протяжении всей жизни и может приумножать свои знания и умения, что немаловажно в современных реалиях. Это также имеет значение для современных педагогов, обучающих детей поколения «альфа», как для координаторов, способных правильно организовать самообучение.

Одни из важнейших современных гибких навыков это критическое мышление, умение работать в команде, управление временем и самоорганизация. Такие навыки способно развивать обучение информационным технологиям (ИТ). Огромная роль в изучении ИТ уделяется программированию и этому есть свои причины: развитие алгоритмического мышления, развитие умственных способностей. Программирование на различных формальных языках расширяет область возможностей выпускника школы. Умея программировать, человек учится просчитывать действия на несколько шагов вперед. Этот навык может существенно пригодиться в профессиональной сфере педагога: учитель любого профиля должен уметь правильно подобрать или создать электронные ресурсы для работы с учащимися на каждый этап урока, будь то этап мотивации или этап применения первично полученных знаний на практике. Средством, с помощью которого возможно обучение программированию

психолого-педагогических классов может быть платформа Scratch. Визуальная среда имеет технические возможности для того, чтобы научиться создавать любые учебные ресурсы: наличие объектно-ориентированного языка программирования высокого уровня, наличие графического редактора, встроенных инструментов для создания медиа-ресурсов. Среда способствует связи всех теоретических знаний об алгоритмах и программировании за счет решения одной практической задачи - создание компьютерной игры, тренажера или интерактивной самостоятельной работы.

Таким образом, **актуальность** данной работы обусловлена требованиями образовательных стандартов к ИТ-компетенциям выпускников и необходимостью подготовки обучающихся, владеющих навыками программирования. Отсюда возникает **противоречие** в том, что программированию необходимо обучать школьников с гуманитарной направленностью, но специальной методической литературы по обучению классов психолого-педагогического профиля программированию недостаточно.

Таким образом, **проблему** можно сформулировать в виде вопроса «Как обучать программированию в психолого-педагогических классах?»

Цель работы: разработка комплекса дидактических материалов по обучению программированию в среде Scratch обучающихся психолого-педагогических классов.

Объект работы: процесс обучения программированию в профильных психолого-педагогических классах.

Предмет работы: комплекс дидактических материалов для обучения программированию в среде Scratch в психолого-педагогических классах.

Задачи:

1. Выявить особенности обучения программированию классов психолого-педагогического профиля.

2. Определить дидактический потенциал среды Scratch для разработки учебных материалов психолого-педагогической направленности.
3. Разработать структуру и содержание комплекса дидактических материалов по обучению программированию в среде Scratch, ориентированного на профессиональные задачи психолого-педагогических классов.
4. Апробировать разработанный комплекс в рамках программы «Школа юного педагога» для обучающихся психолого-педагогических классов.

На основе задач, поставленных выше, составлена структура работы, в которой первая глава содержит теоретический аспект работы, вторая глава - практическую часть.

Глава 1. Теоретические основы обучения программированию в психолого-педагогических классах

1.1. Особенности обучения программированию учащихся психолого-педагогического профиля

Современное образование ставит перед педагогами задачу формирования у обучающихся не только предметных знаний, но и ключевых компетенций, таких как логическое мышление, алгоритмизация, креативность и проектная деятельность. Особую значимость эти навыки приобретают в психолого-педагогических классах (ППК), где учащиеся ориентированы на дальнейшую профессиональную деятельность в сфере педагогики и психологии. Введение элементов программирования в учебный процесс таких классов способствует развитию когнитивных способностей, системного мышления и цифровой грамотности, что соответствует требованиям ФГОС среднего общего образования. Обновленный Федеральный государственный образовательный стандарт среднего общего образования [30] в части преподавания информатики отражает современные тенденции цифровой трансформации общества и актуализирует подходы к формированию ключевых компетенций учащихся. Документ подчеркивает необходимость развития не только технических навыков работы с информационными технологиями, но и метапредметных умений, связанных с критическим мышлением, алгоритмизацией и цифровой грамотностью. В отличие от предыдущих редакций, новый стандарт делает особый акцент на практико-ориентированном характере обучения, интеграции информатики с другими предметными областями и формировании осознанного подхода к использованию цифровых технологий в повседневной и профессиональной деятельности.

Содержательный аспект стандарта выделяет несколько ключевых направлений в изучении информатики. Одним из приоритетов становится развитие вычислительного мышления, включающего способность к декомпозиции задач, распознаванию паттернов, абстрагированию и

алгоритмическому решению проблем. Эти навыки должны формироваться не изолированно, а в контексте реальных жизненных ситуаций, что соответствует общей тенденции на практическую направленность современного образования. Особое внимание уделяется основам программирования, причем стандарт подчеркивает вариативность в выборе языков и сред, рекомендуя учитывать возрастные особенности учащихся и специфику образовательной организации.

Методологическая часть документа акцентирует внимание на деятельностном подходе в обучении. Информатика должна преподаваться не как набор теоретических знаний, а как инструмент решения практических задач. Это предполагает широкое использование проектной деятельности, case-study, учебных исследований и других интерактивных форм работы. Стандарт рекомендует применять дифференцированные задания, позволяющие учитывать индивидуальные образовательные потребности учащихся, что особенно важно в условиях разнородности классов по уровню подготовки.

Оценочная система в рамках нового ФГОС также претерпела изменения. Наряду с традиционными формами контроля знаний, стандарт предусматривает оценку сформированности практических умений через выполнение комплексных проектов, решение кейсов и участие в цифровых конкурсах. Особое значение придается формирующему оцениванию, позволяющему отслеживать прогресс учащихся и своевременно корректировать образовательную траекторию. В документе подчеркивается важность развития навыков самооценки и рефлексии у школьников. Интеграционный потенциал информатики выделяется в стандарте как одна из ключевых характеристик предмета. Предполагается, что знания и умения, полученные на уроках информатики, должны находить применение в других учебных дисциплинах — от естественнонаучных до гуманитарных. Это соответствует междисциплинарному характеру современных цифровых технологий и требованиям к функциональной грамотности выпускников.

Обучение программированию оказывает значительное влияние на когнитивное и личностное развитие учащихся. Как отмечает И. Б. Рогожкина, программирование способствует развитию логического и алгоритмического мышления, а также формированию навыков структурированного решения задач [23]. Эти выводы подтверждаются исследованиями в области педагогики и психологии, где подчеркивается, что освоение основ программирования не только расширяет технические компетенции, но и стимулирует творческий потенциал обучающихся.

Важным аспектом является использование современных методик, таких как геймификация, которая повышает мотивацию школьников. Борисов Н. А. и Харюнин А. С. указывают, что игровые элементы в обучении программированию способствуют более глубокому усвоению материала, снижая уровень стресса и повышая вовлеченность учащихся [3].

Особое место в обучении алгоритмизации занимают визуальные среды программирования, такие как Scratch. Лукьянова Е. С. подчеркивает, что подобные платформы позволяют учащимся наглядно представлять логику кода, что особенно важно на начальных этапах обучения [11]. Визуальное программирование снижает когнитивную нагрузку и делает процесс обучения более доступным для школьников разного уровня подготовки.

Кроме того, современные образовательные тенденции предполагают интеграцию программирования в психолого-педагогические программы. Никодимова Е. А. и Митросенко С. В. отмечают, что внедрение элементов программирования в педагогические классы способствует развитию метапредметных компетенций, включая критическое мышление и проектную деятельность [13, 14].

Таким образом, обучение программированию не только формирует технические навыки, но и способствует развитию ключевых компетенций XXI века, что подтверждается исследованиями в области педагогики и психологии. Как отмечается в Педагогическом энциклопедическом словаре, эффективное обучение должно учитывать индивидуальные особенности

учащихся и использовать интерактивные методы, обеспечивающие устойчивую мотивацию [1].

Обучение программированию в ППК имеет ряд особенностей, обусловленных как возрастными психологическими характеристиками старшеклассников, так и их профессиональной направленностью. «Психолого-педагогические классы - объединение обучающихся образовательной организации, характерологическими признаками которого являются: избирательный принцип комплектования состава учащихся; профилирование обучения за счет включения в учебный план предметов психолого-педагогической и гуманитарной направленности; обеспечение деятельностного подхода в обучении на основе активного освоения и использования школьниками элементов педагогических технологий; наличие отлаженной структуры взаимодействия с организациями образования и другими социальными партнерами» [18].

Согласно исследованиям Л.С. Выготского, развитие мышления в юношеском возрасте тесно связано с формированием абстрактно-логических структур, что делает данный период благоприятным для освоения алгоритмических концепций [6]. Контекстное обучение в понимании Л.С. Выготского предполагает создание «социальной ситуации развития», где программирование перестает быть абстрактной дисциплиной, а становится инструментом решения профессионально значимых задач. В психолого-педагогических классах это означает необходимость проектирования учебных ситуаций, где элементы программирования естественным образом вплетаются в контекст будущей профессиональной деятельности. Например, создание интерактивных психологических тестов или дидактических игр для младших школьников позволяет учащимся осваивать алгоритмические конструкции не как самоцель, а как средство решения педагогических задач. Такой подход полностью соответствует выготсковскому принципу «обучение через деятельность», где знания приобретают личностный смысл через их практическое применение. Особое

значение в свете теории Выготского приобретает вопрос о знаково-символических средствах в обучении программированию. Визуальные среды программирования, такие как Scratch, могут рассматриваться как современные психологические орудия, опосредующие развитие алгоритмического мышления. Выготский отмечал, что новые психические функции сначала формируются во внешней деятельности с использованием внешних знаков, и лишь затем становятся внутренними психическими процессами. Это объясняет эффективность визуальных языков программирования для начального этапа обучения - они выступают как внешние опоры, которые постепенно интериоризируются, превращаясь во внутренние мыслительные структуры. Социальный аспект теории Выготского также имеет важное значение для методики обучения программированию. Согласно его концепции, высшие психические функции первоначально складываются как формы коллективного поведения, как средства социального взаимодействия, и лишь впоследствии становятся индивидуальными функциями самого ребенка. Это означает, что обучение программированию в психолого-педагогических классах должно активно использовать формы совместной деятельности: парное программирование, групповые проекты, коллективное обсуждение решений. Такой подход не только соответствует природе формирования мышления, но и отвечает профессиональной направленности ППК, где коммуникативные навыки имеют первостепенное значение. Принцип единства аффекта и интеллекта, разработанный Выготским, особенно важен при обучении программированию гуманитариев. Согласно этому принципу, эффективное обучение возможно только тогда, когда познавательные процессы эмоционально насыщены, лично значимы. Это объясняет, почему традиционные методики обучения программированию, основанные на решении абстрактных задач, часто оказываются малоэффективными в психолого-педагогических классах. Вместо этого необходимо создавать учебные ситуации, где программирование связано с решением эмоционально

значимых, профессионально ориентированных проблем. Теория Выготского помогает понять и преодолеть типичные трудности, возникающие у учащихся гуманитарного профиля при освоении программирования. Его концепция о соотношении научных и житейских понятий объясняет, почему формальные определения алгоритмических конструкций часто остаются для гуманитариев пустыми абстракциями. Согласно Выготскому, научные понятия должны «вращаться» в систему житейского опыта учащегося, что требует специальной работы по установлению связей между абстрактными алгоритмическими конструкциями и конкретными профессиональными ситуациями.

Однако, как отмечает И.Г. Семакин, традиционные методы преподавания программирования, ориентированные на сложные синтаксические конструкции, могут вызывать у учащихся когнитивную перегрузку [26]. Методическая система обучения программированию, разработанная Игорем Геннадьевичем Семакиным, представляет собой целостный подход, сочетающий фундаментальные принципы информатики с возрастными особенностями школьников. В основе его концепции лежит идея поэтапного формирования алгоритмического мышления через систему специально организованных учебных задач. Автор рассматривает программирование не как самоцель, а как мощное средство развития логического и алгоритмического мышления, что особенно актуально для психолого-педагогических классов, где технические навыки должны гармонично сочетаться с гуманитарной направленностью. Центральное место в методике занимает принцип «от простого к сложному», реализуемый через тщательно выстроенную систему упражнений. Особенностью его подхода является то, что элементарные понятия программирования (переменная, ветвление, цикл) вводятся не как абстрактные конструкции, а как инструменты решения конкретных содержательных задач. Этот методический прием особенно важен для учащихся гуманитарного профиля, так как позволяет избежать формального усвоения синтаксических конструкций без понимания их семантики. Автор настаивает на том, что каждая новая

языковая конструкция должна вводиться только тогда, когда становится очевидной ее необходимость для решения практической задачи. Для психолого-педагогических классов этот аспект особенно значим, так как позволяет учащимся видеть целостную картину применения программирования в профессиональной деятельности, а не фрагментарные технические приемы. Он подчеркивает важность таких аспектов, как структурирование кода, выбор осмысленных имен переменных, комментирование программы - то есть тех элементов, которые часто недооцениваются в школьной практике, но критически важны для профессиональной деятельности. Этот компонент его методики особенно актуален для будущих педагогов и психологов, так как развивает системность мышления и внимание к деталям - качества, необходимые в их будущей профессии. Это делает ее особенно ценной для психолого-педагогических классов, где акцент должен делаться не на освоении конкретного синтаксиса, а на развитии общих способов алгоритмической деятельности. В то же время Семакин подчеркивает важность практической реализации алгоритмов, так как только через собственноручное написание программ достигается подлинное понимание алгоритмических принципов. Для психолого-педагогических классов это особенно значимо, так как уровень математической и алгоритмической подготовки учащихся может существенно варьироваться. Методика предусматривает возможность выбора заданий в соответствии с индивидуальными возможностями и интересами учащихся, что способствует формированию положительной мотивации к изучению программирования. Он выделяет такие распространенные проблемы, как формальное усвоение синтаксиса без понимания семантики, трудности в декомпозиции задач, неумение находить и исправлять ошибки. Для каждой из этих проблем предлагаются конкретные методические решения, основанные на принципах постепенного усложнения материала и многократного возвращения к ключевым понятиям на новом уровне.

Особую ценность для психолого-педагогических классов представляет разработанная Митчеллом Резником концепция «творческих циклов» (creative learning spiral), описывающая итеративный процесс создания проектов: воображение → создание → игра → обмен → рефлексия → новое воображение [12]. Этот подход идеально соответствует задачам формирования профессиональных компетенций будущих педагогов, так как развивает не только технические навыки, но и метакогнитивные способности - умение планировать, оценивать, корректировать свою деятельность.

В связи с этим возникает необходимость в адаптации методик обучения, учитывающих психологические особенности обучающихся ППК.

Важным аспектом является мотивация учащихся. По теории А. Маслоу, самореализация и профессиональное самоопределение выступают ключевыми потребностями старшеклассников [12]. Теория мотивации, разработанная Абрахамом Гарольдом Маслоу, представляет собой важную основу для понимания психологических аспектов обучения программированию, особенно в контексте психолого-педагогических классов. Автор предложил иерархическую модель потребностей, известную как «пирамида Маслоу», которая раскрывает взаимосвязь между базовыми и высшими потребностями человека. В образовательном процессе, включая изучение программирования, эта теория помогает объяснить, как внутренние и внешние факторы влияют на вовлеченность и успешность учащихся. Потребность в уважении и признании, занимающая четвертую ступень в иерархии, играет ключевую роль в образовательном процессе. Ученый утверждал, что положительная обратная связь, возможность демонстрировать свои достижения и получать одобрение от педагогов и сверстников способствуют формированию устойчивой мотивации. В обучении программированию это может выражаться в системе поощрений, публичных презентациях проектов или участии в конкурсах, что особенно актуально для психолого-педагогических классов, где социальное признание часто является мощным стимулом. Высший уровень пирамиды — потребность в

самоактуализации — непосредственно связан с творческим и профессиональным ростом. Согласно идеям А.Г. Маслоу, именно на этом этапе человек стремится реализовать свой потенциал через значимую деятельность. Для учащихся психолого-педагогических классов программирование может стать инструментом самореализации, если связать его с будущей профессией. Например, создание образовательных приложений, интерактивных тестов или игр для детей позволяет превратить технические навыки в средство решения педагогических задач. Автор теории подчеркивал, что деятельность, наполненная личностным смыслом, вызывает наибольшую внутреннюю мотивацию. Критики подхода исследователя отмечают, что иерархия потребностей не всегда носит строгий линейный характер. Однако в контексте обучения программированию теория остается актуальной, так как помогает педагогам выстраивать учебный процесс с учетом психологических особенностей учащихся. Например, если студент испытывает стресс из-за внешних обстоятельств (неудовлетворенность базовых потребностей), его способность к освоению сложных алгоритмов может снижаться. Применяя идеи А.Г. Маслоу в преподавании программирования, важно создавать условия, при которых учащиеся могут последовательно удовлетворять свои потребности — от базовых до высших. Это включает комфортную среду обучения, поддержку социальных взаимодействий, признание достижений и возможности для творческой реализации. Такой подход особенно важен в психолого-педагогических классах, где мотивация к изучению технических дисциплин часто зависит от их гуманитарной и профессиональной значимости. В контексте ППК программирование может быть представлено не как самоцель, а как инструмент решения педагогических и психологических задач, например, разработка обучающих игр или тренажеров. Такой подход согласуется с концепцией контекстного обучения А.А. Вербицкого, согласно которой знания усваиваются эффективнее, если связаны с будущей профессиональной деятельностью [5].

Старший школьный возраст (15-18 лет) характеризуется завершением формирования основных когнитивных структур и переходом к абстрактно-логическому мышлению [25]. Согласно исследованиям Л.С. Выготского, именно в этот период наиболее эффективно развивается способность к теоретическому обобщению и системному анализу [5]. Эти психологические особенности создают благоприятную почву для изучения программирования, которое по своей природе требует умения выявлять закономерности, способности к декомпозиции задач и навыков логического моделирования.

Таким образом, современные исследования подтверждают необходимость разработки специализированных методик для ППК, учитывающих когнитивные профили учащихся, профессиональную контекстуализацию задач и использование визуальных сред как базового инструмента. Такой подход позволит эффективно формировать цифровые компетенции у будущих педагогов и психологов, обеспечивая их готовность к профессиональной деятельности в условиях цифровой трансформации образования.

Проведенный анализ теоретических основ обучения программированию в среде Scratch для учащихся психолого-педагогических классов позволяет сделать ряд принципиальных выводов. Теоретические исследования и методические разработки демонстрируют, что Scratch представляет собой уникальную образовательную платформу, которая оптимально соответствует как возрастным особенностям старшеклассников, так и специфике их профессиональной ориентации. Культурно-исторический подход Л.С. Выготского и теория мотивации А.Г. Маслоу, рассмотренные в первом параграфе, создают прочную психолого-педагогическую основу для построения образовательного процесса, подчеркивая важность социального взаимодействия, профессиональной контекстуализации и личностной значимости учебной деятельности.

1.2. Дидактический потенциал среды Scratch для психолого-педагогического образования

Современная методическая литература, посвященная обучению программированию в среде Scratch, представляет собой обширный пласт исследований, охватывающий различные аспекты образовательного процесса - от теоретических основ до конкретных методических приемов. Глубокий анализ этих работ позволяет выявить как общие тенденции, так и специфические подходы, характерные для разных научных школ и педагогических традиций. В данной статье мы подробно рассмотрим ключевые направления современных исследований, уделяя особое внимание сравнительному анализу различных методических систем и их эффективности в образовательной практике.

В российской педагогической науке значительный вклад в разработку методики обучения программированию в Scratch внес Е.Д. Патаракин. Его работы отличаются системным подходом к интеграции Scratch в образовательный процесс. Автор выделяет три основных этапа освоения среды: 1) ознакомительный (создание простых анимаций), 2) алгоритмический (работа с базовыми структурами программирования), 3) проектный (разработка комплексных приложений) [18]. Автор особое внимание уделяет формированию цифровых компетенций, подчеркивая, что Scratch позволяет развивать не только алгоритмическое мышление, но и навыки сетевого взаимодействия, цифрового творчества и критического восприятия информации. В его методике значительное место отводится рефлексивной деятельности учащихся - анализу созданных проектов, выявлению использованных алгоритмических паттернов, оценке эффективности различных подходов к решению задач.

И.Г. Семакин в своих исследованиях делает акцент на системности обучения программированию в Scratch. Его методическая система построена на принципе последовательного усложнения заданий и четкой структуризации учебного материала. Автор разработал детальную

классификацию упражнений по типам формируемых умений: от простейших операций с объектами до сложных алгоритмов обработки данных. Особую ценность представляет предложенная Семакиным система типовых задач для отработки основных алгоритмических конструкций: линейных программ, ветвлений, циклов, процедур и обработки событий. В отличие от многих других авторов, Семакин уделяет особое внимание формированию «чистоты кода» - навыков структурирования программы, выбора осмысленных имен переменных, комментирования кода, что, по его мнению, закладывает основы профессиональной культуры программирования [26].

Ю.А. Первин в своих работах развивает направление, связанное с использованием Scratch для формирования вычислительного мышления [9]. Автор предлагает оригинальную систему заданий, направленных на развитие четырех ключевых компонентов: 1) декомпозиции задач, 2) распознавания паттернов, 3) абстрагирования, 4) алгоритмизации. Первин подчеркивает важность межпредметных связей, разрабатывая задания, которые интегрируют программирование с математикой, физикой, биологией и другими учебными дисциплинами. Особый интерес представляет его концепция «программирования как грамотности», согласно которой навыки алгоритмического мышления в современном мире становятся столь же необходимыми, как традиционные навыки чтения и письма.

В.С. Корнилов и В.И. Цыганов в своих совместных исследованиях рассматривают Scratch как инструмент формирования цифровой грамотности в широком смысле [10]. Авторы разработали комплексную систему оценки образовательных результатов, включающую: 1) технические критерии (корректность кода, использование алгоритмических структур, оптимальность решений), 2) содержательные критерии (логичность сценария, научная достоверность), 3) дизайн-критерии (удобство интерфейса, эстетика оформления), 4) коммуникативные критерии (ясность инструкций, качество презентации). Их методика особое внимание уделяет вопросам цифровой

безопасности и этики, формированию ответственного отношения к интеллектуальной собственности и персональным данным.

Л.Л. Босова (2022) в своих последних работах развивает направление дифференцированного обучения программированию в Scratch [4]. Автор предлагает детальную классификацию учащихся по типам познавательных стратегий (визуалы, кинестетики, логики, креативщики) и разрабатывает соответствующие методические подходы для каждой группы. Босова подчеркивает важность индивидуальных образовательных траекторий, предлагая систему «разноуровневых проектов» - заданий одной тематической направленности, но разной степени сложности. Ее методика включает также специальные приемы для работы с одаренными детьми и учащимися, испытывающими трудности в обучении.

В исследованиях Людмилы Леонидовны Босовой, посвященных дифференцированному обучению программированию в среде Scratch, представлена комплексная методическая система, учитывающая когнитивные особенности учащихся различных возрастных групп и уровней подготовки. Автор исходит из принципиального положения о том, что традиционный «усредненный» подход к преподаванию программирования неэффективен в условиях реальной школьной практики, где классы характеризуются значительной неоднородностью по уровню цифровой грамотности, математической подготовки и познавательных стратегий. Босова разрабатывает модель дифференцированного обучения, основанную на тщательной диагностике индивидуальных особенностей учащихся и создании соответствующих образовательных траекторий. Центральное место в ее подходе занимает классификация учащихся по типам познавательных стратегий, где выделяются четыре основные группы: визуалы, предпочитающие графическое представление информации; кинестетики, лучше усваивают материал через практические действия; логики, ориентированные на формальные структуры; и креативщики, стремящиеся к нестандартным решениям. Для каждой из этих групп автор предлагает

специфические методические приемы работы в Scratch, позволяющие максимально использовать сильные стороны учащихся и постепенно развивать менее выраженные способности.

Особое внимание в работах автора уделяется системе разноуровневых заданий, организованных по принципу «спирали» - последовательного усложнения материала с постоянным возвратом к ключевым понятиям на новом уровне. Автор разрабатывает три основных уровня сложности проектов в Scratch: стартовый (базовые алгоритмические конструкции), продвинутый (комбинированные алгоритмы) и творческий (комплексные проекты с элементами исследования). Каждый уровень включает вариативные задания, позволяющие учащимся выбирать задачи в соответствии со своими интересами и возможностями. Л.Л. Босова подчеркивает важность плавного перехода между уровнями, для чего в ее методике предусмотрены специальные «мостиковые» задания, помогающие учащимся преодолеть когнитивный барьер при переходе на следующую ступень сложности.

Важным компонентом методики Л.Л. Босовой является система поддержки и сопровождения учащихся, испытывающих трудности в освоении программирования. Автор предлагает комплекс мер, включающий: адаптированные инструкции с пошаговыми подсказками, визуальные схемы алгоритмов, шаблоны проектов и систему парной работы. Особый интерес представляет разработанная Босовой методика «обратного проектирования», когда учащиеся анализируют готовые работающие проекты, модифицируют их параметры и постепенно проникают в логику их создания. Этот подход особенно эффективен для кинестетиков и визуалов, составляющих значительную часть учащихся в общеобразовательных классах.

Для одаренных детей автор предлагает систему расширенных заданий, включающих элементы научно-исследовательской деятельности: анализ эффективности алгоритмов, сравнение различных подходов к решению задач, оптимизацию кода. Автор разрабатывает специальные методические приемы

стимулирования творческой активности, такие как «открытые задания» (без строго определенного решения), «проекты-вызовы» (с постепенно усложняющимися требованиями) и «коллаборативные задачи» (требующие совместной работы над сложным проектом).

Особую ценность в методике Л.Л. Босовой представляет разработанная система оценивания, которая учитывает не только конечный результат (работающий проект), но и процесс его создания: умение планировать работу, находить и исправлять ошибки, оптимизировать решения. Автор предлагает комплекс критериев, позволяющих объективно оценить прогресс каждого учащегося относительно его стартового уровня, что создает ситуацию успеха и способствует формированию положительной мотивации. Босова особо подчеркивает важность формирующего оценивания, когда обратная связь дается в процессе работы и помогает учащимся корректировать свои действия.

Теоретической основой подхода Л.Л. Босовой служат принципы личностно-ориентированного обучения, адаптивной педагогики и теории множественного интеллекта Гарднера. Практическая реализация ее методики показала эффективность в условиях реального учебного процесса, особенно в классах с выраженной неоднородностью состава учащихся. Разработанные Босовой приемы и методы позволяют создать инклюзивную образовательную среду, где каждый ученик может развивать навыки программирования в соответствии со своими индивидуальными особенностями и темпом усвоения материала.

Учебное пособие «Основы программирования для психолого-педагогических классов» под редакцией И.А. Яшиной, Д.А. Бархатовой и Э.А. Нигматулиной представляет собой ценный ресурс для обучения программированию, особенно для педагогов и учащихся, не имеющих технического бэкграунда. В пособии особое внимание уделяется использованию среды Scratch как инструмента для создания интерактивных образовательных проектов. Scratch выбран благодаря своей визуальной

простоте, что делает программирование доступным для начинающих, включая школьников и педагогов [13].

Методический подход авторов начинается с фундаментального объяснения базовых концепций алгоритмизации. В пособии подчеркивается, что понимание алгоритмов является ключевым для освоения программирования, причем акцент делается на их практической значимости в повседневной жизни. Приводятся многочисленные примеры из реального мира – от простейших бытовых алгоритмов (приготовление пищи по рецепту) до сложных систем (работа GPS-навигации или светофоров). Такой подход позволяет учащимся осознать универсальность алгоритмического мышления и его применимость в различных сферах деятельности.

Особое внимание в пособии уделяется свойствам алгоритмов: дискретности, детерминированности, результативности, конечности и массовости. Авторы не просто перечисляют эти характеристики, но и наглядно демонстрируют их на конкретных примерах, что значительно облегчает понимание материала. Для представления алгоритмов предлагаются три основных метода: словесное описание, графические блок-схемы и непосредственное программное кодирование. Такой многоаспектный подход позволяет учащимся с разным типом мышления найти наиболее понятный для себя способ работы с алгоритмами.

Центральное место в анализируемом пособии занимает раздел, посвященный визуальному программированию в среде Scratch. Авторы представляют эту платформу как идеальный инструмент для начального обучения, подчеркивая ее главное преимущество – возможность создавать программы без необходимости запоминания сложного синтаксиса. Scratch описывается как своеобразный «конструктор», где программирование сводится к комбинированию визуальных блоков, что делает процесс обучения интуитивно понятным и увлекательным.

В пособии подробно рассматриваются дидактические возможности Scratch для создания разнообразных учебных проектов. Особый акцент

делается на разработке интерактивных элементов, таких как образовательные викторины, где программный персонаж задает вопросы, а учащийся вводит ответы с последующей автоматической проверкой. Приводятся пошаговые инструкции по созданию таких проектов, начиная от добавления спрайтов и фонов до программирования логики взаимодействия. Отдельное внимание уделяется работе с мультимедийными компонентами – добавлению звукового сопровождения и анимационных эффектов, что значительно повышает вовлеченность учащихся в учебный процесс.

Практическая составляющая пособия представлена разнообразными заданиями, которые позволяют закрепить полученные знания. Среди них – создание анимированных инструкций, где каждый шаг сопровождается соответствующим визуальным рядом, разработка интерактивных презентаций по различным темам, а также более сложные проекты, сочетающие элементы геймификации и обучения. Важно отметить, что авторы не ограничиваются техническими аспектами работы в Scratch, но и предлагают методические рекомендации по интеграции созданных проектов в учебный процесс.

Методический аппарат пособия построен на принципе «свертывания информации», когда учебный материал представлен в виде четко структурированных вопросов и кратких, но содержательных ответов. Такой подход позволяет учащимся быстро находить нужную информацию и сразу применять ее на практике. Например, раздел по созданию викторин содержит не только технические инструкции, но и педагогические рекомендации по формулировке вопросов и организации обратной связи.

Особого внимания заслуживает проектный метод обучения, который активно пропагандируют авторы пособия. Создание законченных, работающих проектов с первых занятий значительно повышает мотивацию учащихся и позволяет им сразу увидеть практическое применение своих знаний. При этом подчеркивается важность творческого подхода – учащимся

предлагается не просто воспроизводить готовые примеры, но и разрабатывать собственные уникальные проекты.

В заключительной части пособия рассматриваются современные тенденции в обучении программированию, в частности использование искусственного интеллекта как вспомогательного инструмента. Авторы знакомят читателей с платформой replit, демонстрируя, как искусственный интеллект может помогать в написании кода, поиске ошибок и объяснении сложных концепций.

Анализируя пособие в целом, можно отметить его несомненные достоинства: доступность изложения, практическую направленность, актуальность рассматриваемых технологий. Материал адаптирован для педагогов и учащихся без специальной технической подготовки, что делает его ценным ресурсом для массового образования. Вместе с тем, стоит отметить, что переход от визуального программирования к текстовому (например, к Python) представлен недостаточно подробно, что могло бы усилить преемственность в обучении.

Пособие успешно решает свою главную задачу – делает программирование доступным и интересным для широкого круга учащихся. Оно может служить отличной стартовой точкой для знакомства с миром программирования, а также источником методических идей для педагогов.

Сравнительный анализ различных методических подходов позволяет сделать ряд важных выводов. Во-первых, современная методическая литература демонстрирует переход от технократического подхода (программирование как цель) к гуманистическому (программирование как средство развития личности). Во-вторых, наблюдается тенденция к интеграции различных аспектов обучения - технических, когнитивных, социальных и творческих.

Выводы по первой главе:

В параграфе 1.1 было установлено, что обучение программированию в психолого-педагогических классах способствует не только формированию

цифровых компетенций, но и развитию алгоритмического мышления, логики и креативности. Особое значение имеет интеграция программирования с психолого-педагогическими дисциплинами, что позволяет будущим педагогам осваивать современные образовательные технологии и применять их в профессиональной деятельности. Исследования подтверждают, что использование геймификации и визуальных сред повышает мотивацию учащихся и снижает когнитивную нагрузку, что особенно важно для профильного обучения.

Параграф 1.2 посвящен анализу возможностей среды Scratch в психолого-педагогическом образовании. Было выявлено, что данная платформа обладает значительным дидактическим потенциалом благодаря интуитивно понятному интерфейсу, позволяющему обучать программированию без углубленного изучения синтаксиса, возможности проектной деятельности, что способствует развитию soft skills (командная работа, критическое мышление), адаптивности для учащихся с разным уровнем подготовки, включая детей с ОВЗ.

Объединяющим фактором рассмотренных теоретических положений выступает их ориентация на развитие не только узкопредметных навыков программирования, но и метапредметных компетенций, критически важных для будущих педагогов и психологов: системного мышления, способности к алгоритмизации профессиональных задач, проектной культуры и цифровой грамотности. Теоретический анализ подтверждает, что среда Scratch благодаря своей гибкости и многофункциональности позволяет реализовать принципы личностно-ориентированного обучения, обеспечивая возможность выбора индивидуальных образовательных траекторий в соответствии с познавательными стилями и профессиональными интересами учащихся.

Глава 2. Практическая разработка и апробация комплекса дидактических материалов по Scratch для психолого-педагогических классов

2.1. Структура и содержание комплекса дидактических материалов

Комплекс дидактических материалов для обучения программированию в среде Scratch учащихся психолого-педагогических классов представляет собой целостную методическую систему, разработанную с учетом современных требований цифровой педагогики и особенностей профессиональной подготовки будущих педагогов. Структурно комплекс организован как многоуровневая иерархическая система, включающая пять тематических модулей, каждый из которых решает конкретные образовательные задачи и одновременно вносит вклад в формирование профессиональных компетенций.

Модуль 1. Введение в программирование и Scratch.

Модуль 2. Основы алгоритмизации.

Модуль 3. Работа со звуком.

Модуль 4. Создание игр.

Модуль 5. Scratch в образовании.

Основу комплекса составляет модульный принцип организации, позволяющий реализовать гибкий подход к обучению с возможностью адаптации содержания под различные образовательные контексты и индивидуальные потребности учащихся. Первый модуль («Введение в программирование и Scratch») выполняет вводную функцию, формируя базовые навыки работы в среде и демонстрируя ее педагогический потенциал. Второй модуль («Основы алгоритмизации») закладывает фундамент вычислительного мышления через освоение ключевых алгоритмических конструкций. Третий модуль («Работа со звуком») расширяет технические возможности учащихся, уделяя особое внимание развитию аудиальных компонентов образовательного контента. Четвертый

модуль («Создание игр») раскрывает потенциал игрофикации в обучении, а пятый модуль («Scratch в образовании») интегрирует полученные знания и навыки в профессиональный контекст.

Каждый модуль представляет собой законченный дидактический блок, включающий три взаимосвязанных компонента: теоретический материал с профессионально-ориентированными примерами, систему практических заданий разного уровня сложности и методические рекомендации по практическому применению созданных проектов в образовательном процессе. Теоретическая часть модулей построена по принципу контекстного обучения, где абстрактные понятия программирования иллюстрируются конкретными педагогическими ситуациями. Например, понятие «цикл» объясняется через аналогию с повторяющимися этапами урока, а «ветвление» - через анализ различных сценариев педагогического взаимодействия.

Практические задания разработаны с учетом принципа постепенного усложнения и дифференциации по трем уровням: базовый (репродуктивный), продвинутый (частично-поисковый) и творческий (исследовательский). Особое внимание уделено профессиональной направленности заданий - от создания простых анимаций педагогических ситуаций до разработки комплексных дидактических игр и тренажеров. Все задания сопровождаются подробными методическими комментариями, объясняющими их значение для профессионального становления будущего педагога.

Методический компонент комплекса включает не только рекомендации по выполнению заданий, но и конкретные советы по интеграции созданных проектов в реальный образовательный процесс.

Важной отличительной особенностью комплекса является его двойная направленность. С одной стороны, он обеспечивает системное освоение фундаментальных понятий программирования: от базовых алгоритмических структур до сложных проектных решений. С другой - целенаправленно формирует навыки создания цифровых дидактических материалов, соответствующих современным требованиям к профессиональной

деятельности педагога. Такой подход позволяет преодолеть традиционный разрыв между техническими и гуманитарными аспектами подготовки будущих учителей.

Все учебные материалы разработаны с учетом принципов современного педагогического дизайна. В их числе: последовательность изложения от простого к сложному с постоянным возвратом к ключевым понятиям на новом уровне; визуализация сложных понятий через профессиональные аналогии; возможность выбора индивидуальной образовательной траектории за счет вариативности заданий; баланс между индивидуальной и коллективной формами работы.

Особое место в структуре комплекса занимает система оценивания, которая включает как традиционные критерии качества программирования (корректность кода, использование алгоритмических структур, оптимальность решений), так и специально разработанные показатели педагогической ценности создаваемых цифровых продуктов. К последним относятся: соответствие возрастным особенностям целевой аудитории, методическая обоснованность, потенциал практического применения, качество визуализации учебного материала.

Комплекс также включает дополнительные ресурсы для учителей: методические рекомендации по организации занятий, банк типовых ошибок и способы их коррекции, примеры лучших ученических работ. Для учащихся предусмотрены справочные материалы: глоссарий терминов, памятки по работе в Scratch, коллекция полезных скриптов и алгоритмов.

Реализация данного комплекса в учебном процессе позволяет не только сформировать у будущих педагогов базовые навыки программирования, но и развить профессионально значимые качества: проектное мышление, способность к созданию цифрового образовательного контента, понимание принципов визуализации учебной информации, готовность к инновациям в педагогической деятельности.

Методические рекомендации по использованию комплекса дидактических материалов по Scratch для психолого-педагогических классов представляют собой систему практических советов по организации эффективного учебного процесса. При реализации модульного принципа обучения важно начинать с базовых понятий первого модуля, постепенно переходя к более сложным проектам, обеспечивая при этом постоянное возвращение к ключевым концепциям на новом уровне в каждом последующем модуле. Особое внимание следует уделять межмодульным связям, наглядно демонстрируя развитие тем от простого к сложному. Дифференцированный подход предполагает использование заданий трех уровней сложности - базового, продвинутого и творческого, позволяя учащимся выбирать задания в соответствии с их индивидуальными возможностями и интересами. Для учащихся, испытывающих трудности, рекомендуется применять шаблоны и пошаговые инструкции, тогда как для продвинутых учеников эффективными будут открытые задания с возможностью усложнения проектов.

Профессиональная ориентация занятий требует активного использования педагогического контекста при объяснении программистских понятий, приведения аналогий из школьной практики и обсуждения реального применения создаваемых проектов в образовательном процессе. Организация проектной деятельности должна быть направлена на создание практически полезных разработок, с последующей их защитой и оценкой педагогической ценности. Рекомендуется создавать банк полезных проектов для их дальнейшего использования в школе. В методическом плане важно чередовать различные формы организации деятельности: краткие объяснения нового материала (не более 15-20 минут), практическую работу за компьютерами, парное программирование с распределением ролей «учитель» - «ученик», а также групповые обсуждения и анализ проектов. Среди активных методов обучения особую эффективность показывают кейс-метод с

разбором реальных педагогических ситуаций, проектный подход, взаимное обучение учащихся и элементы геймификации.

Система оценивания должна быть комплексной и включать технические критерии (корректность кода, использование алгоритмов), профессиональные критерии (педагогическая ценность проекта), критерии оформления (удобство интерфейса, дизайн) и учет личностных достижений относительно стартового уровня каждого ученика. Разнообразные формы оценивания - формирующее оценивание в процессе работы, самооценка по чек-листам, взаимооценка проектов и итоговая демонстрация лучших работ - позволяют получить объективную картину учебных достижений. При работе с разными категориями учащихся для испытывающих трудности рекомендуется предоставлять пошаговые инструкции, использовать метод «делай как я» с демонстрацией на экране, разрешать работу с готовыми шаблонами и организовывать помощь более сильных учеников. Для продвинутых учащихся эффективными будут дополнительные усложненные задания, поощрение экспериментирования и поиска нестандартных решений, привлечение к помощи одноклассникам и возможность работы над индивидуальными проектами.

Внеурочная деятельность может быть организована через создание клуба Scratch-разработчиков, проведение конкурсов образовательных проектов, формирование школьной галереи лучших работ и организацию мастер-классов от успешных учеников. Техническое обеспечение должно включать подготовку цифровых ресурсов: коллекции полезных скриптов и алгоритмов, библиотеки звуков и изображений для образовательных проектов, видеоинструкций по сложным темам и примеров качественных ученических работ. Система поддержки может быть реализована через консультационные часы для помощи учащимся, форум или чат для вопросов и обсуждений, а также банк типовых ошибок и способов их исправления. Для профессионального развития учителя важно постоянно анализировать эффективность каждого занятия, вести журнал успешных методических

приемов, обмениваться опытом с коллегами и участвовать в профессиональных сообществах Scratch-педагогов. Параллельно следует развивать собственную цифровую грамотность: осваивать новые возможности Scratch, изучать лучшие практики применения в образовании, создавать собственные образовательные проекты и экспериментировать с новыми форматами заданий. Реализация этих рекомендаций позволит максимально эффективно использовать потенциал комплекса дидактических материалов для формирования как программистских навыков, так и профессиональных педагогических компетенций у учащихся психолого-педагогических классов.

**Методические рекомендации по организации и проведению урока
«Введение в программирование в среде Scratch» для обучающихся
психолого-педагогических классов**

Аннотация

Данные методические рекомендации предназначены для учителей информатики, работающих с учащимися психолого-педагогических классов. Материал содержит практические советы по организации учебного процесса с использованием модульной системы обучения, направленной на формирование как программистских навыков, так и профессиональных педагогических компетенций. Рекомендации учитывают особенности восприятия учащихся гуманитарного профиля и ориентированы на создание практически значимых образовательных проектов. Материал содержит подробный план-конспект первого урока по теме «Введение в программирование в среде Scratch» продолжительностью 45 минут.

Цель урока: сформировать первичное представление о программировании через создание профессионально-ориентированного мини-проекта в Scratch.

Задачи урока:

1. Познакомить учащихся с интерфейсом Scratch (сцена, спрайты, блоки, скрипты).
2. Научить использовать базовые блоки (движение, внешность, события). Сформировать понимание алгоритмической последовательности (линейный сценарий).
3. Показать связь программирования с педагогической деятельностью (создание обучающих материалов).
4. Развивать алгоритмическое мышление (пошаговое решение задач).
5. Совершенствовать пространственное воображение (работа со сценой и спрайтами).

6. Воспитывать интерес к программированию через профессиональный контекст.

Тип урока: урок открытия нового знания.

Вид урока: практикум с элементами проектной деятельности.

Оборудование: ПК учителя, ПК обучающихся, Scratch 3.0, интерактивная доска.

Содержание методических рекомендаций

1. Конспект урока;
2. Дифференцированные задания трех уровней сложности;
3. Практические советы по организации работы в классе;
4. Критерии оценки результатов урока.

Конспект урока

№	Этап урока	Цель этапа	Действия учителя	Результаты
1.	Организационный этап.	Создание оптимальных условий для продуктивной учебной деятельности.	Приветствие, проверка готовности рабочих мест Краткое введение в тему: «Сегодня мы начнём знакомство с языком программирования Scratch, который поможет вам в будущей профессии создавать обучающие материалы и игры для детей».	У учащихся сформирован познавательный интерес к программированию через профессиональный контекст.
2.	Этап мотивации к учебной деятельности.	Выработка на личностно значимом уровне внутренней готовности	Демонстрация 3 профессионально-ориентированных проектов: интерактивный тест	Выявлен дефицит знаний: отсутствие навыков

		выполнения нормативных требований учебной деятельности.	«Стиль преподавания», анимация «Конфликт в классе», игра «Распределение ролей в группе». Вопрос для обсуждения: «Как эти программы могут облегчить работу педагога?» Формулировка темы.	работы с базовыми блоками Scratch.
3.	Актуализация и фиксирование индивидуального затруднения.	Подготовка мышления учащихся, организация осознания ими внутренней потребности к построению учебных действий и фиксирование каждым из них индивидуального затруднения в пробном действии.	Обсуждение: Какие конкретно действия не получились? Почему возникли проблемы? Что нужно узнать, чтобы решить задачу? Вывод: Нам нужно изучить: Основные блоки движения и внешности, принцип соединения блоков, кнопки запуска программы.	Четко определены проблемные точки: незнакомый интерфейс, отсутствие понимания логики соединения блоков, неумение управлять порядком действий.
4.	Выявление места и причины затруднений.	Организация анализа учащимися возникшей ситуации и на этой основе выявление места и причины	План действий: 1. Изучаем интерфейс (сцена, спрайты, блоки) 2. Осваиваем 5 ключевых блоков: <i>«идти 10 шагов»</i>	Усвоена структура типового проекта в Scratch: Стартовый блок, действия персонажа,

		затруднения, осознания того, в чем именно состоит недостаточность их знаний, умений или способностей.	<p>«говорить Привет 2 сек»</p> <p>«следующий костюм»</p> <p>«когда щелкнут по зеленому флажку»</p> <p>«ждать 1 сек»</p> <p>3. Создаем алгоритм: Появиться → Поздороваться → Пройти → Исчезнуть.</p>	взаимодействие объектов, составлен алгоритм базовой анимации.
5.	Самостоятельная работа с проверкой по эталону.	Переход извне внутрь нового способа действия и исполнительская рефлексия (коллективная и индивидуальная) достижения цели пробного учебного действия, применение нового знания в типовых заданиях.	<p>Задание: Создать анимацию «Первое сентября» (учитель встречает ученика)</p> <p>Эталонный скрипт: когда флажок нажат показаться говорить «Здравствуйте, ребята!» 2 сек идти 50 шагов ждать 1 сек сменить костюм на «улыбка».</p>	Создана работающая анимация с педагогическим сюжетом. Освоены базовые операции: добавление/изменение спрайтов, управление движением.
6.	Включение в систему знаний.	Повторение и закрепление ранее изученного и подготовка к изучению следующих разделов курса, выявление границы	<p>Связь с профессией: Где в работе педагога пригодятся такие анимации? (презентации, тренажеры, инструкции). Какие психологические темы можно</p>	Осознаны межпредметные связи: Программирование + педагогика + психология. Сформулированы перспективы

		применимости нового знания и использование его в системе изученных ранее знаний, повторение учебного содержания, необходимого для обеспечения содержательной непрерывности, включение нового способа действий в систему знаний.	иллюстрировать? (эмоции, конфликты, командная работа) Перспектива: «На следующем уроке научимся делать интерактивные тесты».	применения навыков.
7.	Рефлексия учебной деятельности.	Самооценка учащимися результатов своей учебной деятельности, осознание метода построения и границ применения нового способа действия.	Вопросы: 1. Какое задание было самым полезным? 2. Где возникли трудности? 3. Пригодятся ли эти навыки в вашей профессии? Самооценка по чек-листу (Приложение А)	Самооценка приобретённых умений, осознание профессиональной ценности навыка.

Дифференцированные задания трех уровней сложности

Базовый уровень (репродуктивный)

Задание: «Создай анимацию *Учитель приветствует класс*»

Требования:

1. Используй готовые спрайты из библиотеки.
2. примени три блока
 - когда флажок нажат

- сказать «Здравствуйте!» 2 сек
- идти 50 шагов

Критерии успеха:

- ✓ Спрайт появляется и движется
 - ✓ Есть текст приветствия
 - ✓ Проект запускается без ошибок.
-

Продвинутый уровень (частично-поисковый)

Задание: «Разработать интерактивную сцену *Диалог учителя и ученика*»

Требования:

1. Добавь 2 персонажа (учитель + ученик);
2. Реализуй последовательность: учитель задает вопрос (блок сказать) → ученик отвечает через 2 секунды (блок ждать) → оба меняют костюмы (улыбка/грусть).

Критерии успеха:

- ✓ Логичная очередность действий
 - ✓ Использованы 5+ типов блоков
 - ✓ Есть элемент творчества (фон, костюмы)
-

Творческий уровень (исследовательский)

Задание: «Создай мини-тренажер *Решение конфликта в классе*»

Требования:

1. Реализуй 3 варианта развития событий (ветвление): удачное разрешение (персонажи улыбаются); эскалация конфликта (персонажи сердятся); вмешательство педагога (появление третьего спрайта).
2. Добавь звуковые эффекты (аплодисменты/шум)

Критерии успеха:

- ✓ Рабочая система условий (если-то);

- ✓ 2+ смены фона;
- ✓ Понятный педагогический сюжет.

Практические советы по организации работы в классе

При подготовке к уроку особое внимание следует уделить технической базе. Необходимо заранее проверить работу всех компьютеров и установку Scratch 3.0, а также подготовить запасные USB-флеш-накопители с портативной версией программы на случай технических сбоев. Целесообразно создать общую папку с тщательно подобранными ресурсами - фонами и спрайтами педагогической тематики, которые помогут учащимся быстрее включиться в работу.

Организация групповой работы требует продуманного подхода. Эффективным решением будет распределение ролей в парах: один ученик выступает в роли «Педагога», отвечая за методическое содержание проекта, другой - «Программиста», непосредственно работая в среде Scratch. Через 10-15 минут рекомендуется поменять учащихся ролями, что позволит каждому попробовать разные аспекты работы. Для создания комфортной рабочей атмосферы можно организовать гибкое учебное пространство, выделив зону «Экспертов» с компьютерами, оборудованными дополнительными мониторами для демонстрации удачных решений, и зону «Совета» - специально отведенное место для группового обсуждения возникающих проблем.

Дифференцированный подход к обучению предполагает различные стратегии работы с учащимися разного уровня подготовки. Для начинающих полезно использовать цветовую маркировку блоков, наклеив соответствующие стикеры на клавиатуры, предоставлять готовые фрагменты кода с пропусками и разрешать использование шаблонов персонажей. Продвинутым ученикам можно предложить более сложные задания: введение дополнительных параметров вроде таймеров и счетчиков баллов, работу с усложненными педагогическими кейсами или доступ к расширенным блокам, включая переменные и клонирование.

Эффективное управление временем урока - залог успешного освоения материала. Оптимальная структура 45-минутного занятия может включать организационный момент и эмоциональный настрой (первые 5 минут), демонстрацию примера и постановку задачи (5-15 минуты), основную практическую работу (15-35 минуты), разделенную на три этапа: создание базовой анимации (10 минут), добавление профессионального контекста (7 минут) и обмен проектами в парах (3 минуты). Завершать урок следует рефлексией и обсуждением перспектив (35-45 минуты).

Для оперативной помощи учащимся стоит разработать систему поддержки. Визуальные подсказки в виде плаката «5 шагов при ошибке» (чтение системного сообщения, проверка соединения блоков, использование подсказки F1, обращение к соседу и только потом - к учителю) помогут развить самостоятельность в решении проблем. Мотивацию можно поддерживать через «Минуту славы» - демонстрацию лучших фрагментов проектов на большом экране, а также элементы геймификации - вручение специальных значков «За креативность» или «За педагогический подход».

Профессиональная ориентация урока достигается через анализ реальных образовательных приложений, обсуждение психологических принципов, заложенных в ученические проекты, и организацию peer-to-peer оценки с позиции будущего педагога.

Завершать урок следует организованно, оставив 5 минут на сохранение проектов в облачное хранилище, приведение рабочих мест в порядок и возможность для стеснительных учеников задать индивидуальные вопросы. Такой комплексный подход создаст комфортную среду для освоения программирования, учитывающую специфику психолого-педагогических классов и способствующую формированию профессионально значимых компетенций.

Проведенный анализ структуры и содержания комплекса дидактических материалов для обучения программированию в среде Scratch в психолого-педагогических классах позволяет констатировать, что

разработанный комплекс представляет собой целостную методическую систему, органично сочетающую технические аспекты программирования с профессиональной подготовкой будущих педагогов. Особую ценность комплексу придает его модульная структура, обеспечивающая последовательное освоение материала от базовых понятий до сложных проектных заданий, что соответствует принципам дидактики и возрастным особенностям учащихся. Важнейшей характеристикой материалов является их профессиональная ориентированность, проявляющаяся в подборе заданий, максимально приближенных к реальной педагогической практике - созданию интерактивных учебных пособий, психологических тестов и дидактических игр. Методическое обеспечение комплекса отличается тщательно продуманной системой дифференциации, позволяющей учитывать индивидуальные особенности и уровень подготовки каждого обучающегося. Особого внимания заслуживает разработанная система оценивания, сочетающая традиционные критерии качества программирования с показателями педагогической ценности создаваемых проектов. Комплекс не только формирует цифровые компетенции учащихся, но и способствует развитию профессионально значимых качеств - системного мышления, способности к алгоритмизации педагогических задач и проектной культуры.

2.2. Апробация комплекса дидактических материалов в программе «Школа юного педагога»

Разработанный комплекс дидактических материалов прошел частичную апробацию в рамках «Школы юного педагога», проводимой на базе ИМФИ КГПУ им. В.П. Астафьева. Нами был проведен мастер-класс по теме: «Разработка образовательного контента в среде Scratch по информатике». Очное участие принимали ученики 10 «П» класса (психолого-педагогический профиля) МАОУ СШ «Комплекс Покровский».

Дистанционное участие принимали обучающиеся других школ Красноярска и Красноярского края. Школьникам было предложено создать

тренажер или игру на выбор. Темы для проектов были взяты из школьного курса математики и информатики.

Мастер-класс начался с входного опроса, который позволил оценить уровень знакомства участников с программированием и средой Scratch. Результаты опроса представлены в таблице.

Таблица 1. Результаты входного опроса

В каком классе вы учитесь?	В каких средах программирования вы работали на уроках информатики?	Вы когда-нибудь создавали программы в среде Scratch ?	Что вы ожидаете от мастер-класса?
8-9	КуМир, Python	Нет	Познакомиться со средой Scratch для дальнейших разработок, Узнать что-то новое
10-11	КуМир, Pascal	Нет	Узнать что-то новое
10-11	КуМир, Pascal, Python	Нет	Узнать как создавать образовательный контент
10-11	КуМир, Pascal	Нет	Узнать что-то новое, Узнать как создавать образовательный контент
10-11	КуМир, Python	Нет	Познакомиться со средой Scratch для дальнейших разработок, Узнать как создавать образовательный контент
10-11	Pascal, Python	Нет	Познакомиться со средой Scratch для дальнейших разработок, Узнать что-то новое
10-11	Другое	Да	Узнать что-то новое
10-11	КуМир, Python	Нет	Узнать что-то новое
10-11	КуМир, Python	Нет	Узнать что-то новое
10-11	КуМир, Python	Нет	Узнать что-то новое
10-11	КуМир, Python	Нет	Узнать что-то новое
10-11	Python	Нет	Узнать что-то новое

Это помогло адаптировать дальнейшую работу под потребности аудитории. Участникам была представлена краткая теоретическая часть, включающая обзор интерфейса Scratch, основных блоков команд и их функционального назначения. Особый акцент был сделан на простоте и доступности среды, что делает ее идеальным инструментом для использования в педагогической практике, даже при отсутствии у учащихся углубленных навыков программирования.

Практическая часть мастер-класса включала создание мини-проекта — интерактивной анимации, в которой персонажи обменивались приветствиями. Участники, работая в группах по 2–3 человека, последовательно выполняли задания: выбирали и настраивали спрайты, создавали диалог между персонажами, добавляли анимацию движения и смену костюмов. Каждый этап сопровождался подробными инструкциями, что позволило даже новичкам успешно завершить проект (Приложение В).

Важной частью мероприятия стало обсуждение педагогического потенциала Scratch. Участники рассмотрели примеры готовых разработок, таких как обучающие игры и викторины, а также узнали о возможностях взаимодействия с мировым сообществом Scratch, где можно найти готовые решения для учебных целей. Примеры работ представлены в приложении Б.

Проведенный мастер-класс подтвердил эффективность предложенного комплекса дидактических материалов. Участники не только освоили базовые принципы работы в Scratch, но и увидели его практическую ценность для создания учебных материалов. Полученные обратные связи свидетельствуют о высоком интересе к дальнейшему изучению программирования и его интеграции в будущую профессиональную деятельность. Результаты представлены в виде таблицы.

Таблица 2. Обратная связь

Устроил ли вас результат практической работы?	Оцените по пятибалльной шкале следующие аспекты занятия: [Интересный материал]	Оцените по пятибалльной шкале следующие аспекты занятия: [Полезный материал]	Оцените по пятибалльной шкале следующие аспекты занятия: [Понятные объяснения]	Оцените по пятибалльной шкале следующие аспекты занятия: [Продуктивная практика]	Расскажите о том, что вам понравилось больше всего.
Да, полностью	3	3	3	3	Все понравилось.
Да, полностью	3	3	3	3	Преподаватель и
Да, полностью	5	5	5	5	Практическая часть
Да, но не полностью	5	4	5	3	Работать со звуком
Да, но не полностью	5	5	5	5	Материал доступно объяснён

Выводы по второй главе:

Проведенный анализ подтверждает, что структура и содержание комплекса дидактических материалов полностью соответствуют целям подготовки будущих педагогов в условиях цифровой трансформации образования, обеспечивая формирование как технических навыков программирования, так и профессиональных компетенций в области создания цифровых образовательных ресурсов.

Таким образом, апробация в рамках «Школы юного педагога» продемонстрировала, что Scratch является мощным инструментом для развития цифровых компетенций будущих педагогов, сочетая в себе доступность, творческий потенциал и широкие возможности для применения в образовательном процессе.

Заключение

Ключевым теоретическим положением выступает целесообразность применения профессионально-ориентированного подхода при обучении программированию обучающихся психолого-педагогического профиля. В данном случае технические компетенции формируются не изолированно, а как средство решения актуальных профессиональных задач - начиная от разработки учебных материалов и заканчивая созданием диагностических методик. Подобная методологическая установка, базирующаяся на проанализированных концептуальных положениях, способствует устранению искусственного противопоставления технических и гуманитарных аспектов подготовки, обеспечивая обучающимся системное понимание значения цифровых инструментов в педагогической практике.

Проведенная работа по разработке и апробации методического комплекса для обучения программированию в среде Scratch в психолого-педагогических классах позволила сделать ряд значимых выводов.

Во-первых, было подтверждено, что Scratch является эффективным инструментом для формирования цифровых компетенций у обучающихся психолого-педагогических классов. Его визуально-блочный интерфейс, интуитивная понятность и низкий порог входа делают среду идеальной для учащихся гуманитарного профиля, позволяя преодолеть психологические барьеры перед изучением программирования.

Во-вторых, разработанный комплекс дидактических материалов продемонстрировал свою эффективность в ходе апробации в рамках программы «Школа юного педагога». Модульная структура, профессиональная ориентация заданий и система дифференцированного подхода позволили участникам с разным уровнем подготовки успешно освоить базовые принципы программирования и увидеть их практическую ценность для педагогической деятельности.

Особую значимость имеет выявленная возможность интеграции программирования с психолого-педагогическими дисциплинами. Создание

интерактивных тестов, дидактических игр и обучающих анимаций не только развивает алгоритмическое мышление учащихся, но и формирует навыки проектирования цифровых образовательных ресурсов - важнейшей компетенции современного педагога.

Перспективы дальнейшего исследования видятся в разработке расширенного курса, включающего более сложные проекты с элементами искусственного интеллекта и анализа данных, а также в создании методики оценки влияния изучения программирования на формирование профессиональных педагогических компетенций.

Библиографический список

1. Абдулгалимов Г. Л., Холмогорова Е. Г. Обучение будущих учителей физики и информатики системам разработки Scratch и TinkerCad // ББК 74.262 Ф503. — С. 25.
2. Бим-Бад Б. М. Педагогический энциклопедический словарь. — М. : Инфра-М, 2002. — 312 с.
3. Борисов Н. А., Харюнин А. С. Использование геймификации при обучении школьников основам программирования. — 2018. — № 1.
4. Босова Л. Л. Тенденции развития школьного курса информатики и ИКТ // URL: <http://saratov.ito.edu.ru/2011/section/173/93160> (дата обращения: 10.06.2025).
5. Вербицкий А. А. Педагогические технологии контекстного обучения : науч.-метод. пособие. — Вып. 1. — 2-е изд. — М. : РИЦ МГГУ им. М. А. Шолохова, 2011. — 52 с.
6. Выготский Л. С. История развития высших психических функций // Собр. соч. : в 6 т. — М. : Педагогика, 1983. — Т. 3 : Проблемы развития психики. — С. 68.
7. Евстифеева А. В. Основы программирования в Scratch для учащихся с ОВЗ на уроках информатики // Современные проблемы математики, физики и физико-математического образования. — 2021. — С. 303–306.
8. Захаров С. Е. Scratch как эффективный инструмент интеграции информатики и английского языка.
9. Киргизова Е. В., Сергаева Н. О. Организация внеурочной деятельности школьников по информатике с использованием среды программирования Scratch // Информатизация образования: теория и практика. — 2022. — С. 263–265.

10. Корнилов В. С., Цыганов В. И. Методические аспекты обучения школьников информатике с применением системы управления контентом «Joomla» // Вестн. Моск. гор. пед. ун-та. Сер.: Информатика и информатизация образования. — 2013. — № 1. — С. 65–71.
11. Лукьянова Е. С. Особенности использования визуальной среды «Scratch» при обучении алгоритмизации и программированию в основной школе // Студенческий электронный журнал СтРИЖ. — 2019. — № 4-2. — С. 60–67.
12. Маслоу А. Мотивация и личность : пер. с англ. — 3-е изд. — СПб. : Питер, 2019. — 400 с.
13. Митросенко С. В. Педагогический класс: новое прочтение традиционной формы ориентации на выбор педагогической профессии // Проблемы современного педагогического образования. — 2016. — № 51-3. — С. 204–211.
14. Никодимова Е. А. Создание психолого-педагогических классов в региональной системе образования // Вологодский институт развития образования. — 2021. — № 2. — С. 11–15.
15. Ораева О. О., Бабаева А. Б. О возможностях расширения учебной программы факультативного занятия «Пропедевтика основ алгоритмизации и программирования в визуальной среде программирования Scratch» для учащихся V–VI классов. — 2021.
16. Организация деятельности психолого-педагогических классов : учеб.-метод. пособие / под ред. Е. Г. Врублевской, Л. А. Даринской. — М. : Академия Минпросвещения России, 2021. — 392 с.
17. Основы программирования для психолого-педагогических классов / под ред. И. А. Яшиной, Д. А. Бархатовой, Э. А. Нигматулиной. — Красноярск : Краснояр. гос. пед. ун-т им. В. П. Астафьева, 2025. — 108 с.

18. Патаракин Е. Д. Педагогический дизайн социальной сети Scratch // Открытое образование. — 2013. — № 2. — URL: <https://cyberleninka.ru/article/n/pedagogicheskiy-dizayn-sotsialnoy-seti-scratch> (дата обращения: 11.06.2025).
19. Первин Ю. А. Методика раннего обучения информатике : метод. пособие. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2008.
20. Пиаже Ж. Психология интеллекта. — СПб. : Питер, 2004. — 192 с.
21. Простак О. Ю. О дидактических материалах содержательной линии «Основы алгоритмизации и программирования» с использованием визуального языка программирования Scratch : метод. пособие. — М. : БИНОМ. Лаборатория знаний, 2023. — 64 с.
22. Резник М. Спираль обучения. 4 принципа развития детей и взрослых. — М. : Манн, Иванов и Фербер, 2020. — 320 с.
23. Рогожкина И. Б. Развивающий эффект обучения программированию: психолого-педагогические аспекты // Психология. Журнал ВШЭ. — 2012. — № 2.
24. Розов К. В. Интеграция виртуальной среды Scratch, платформы Arduino и искусственного интеллекта как фактор повышения эффективности информационно-технологической подготовки педагогических кадров // От идеи к практике: социогуманитарное знание в цифровой среде. — 2021. — С. 132–138.
25. Сафонов В. И. Особенности изучения программирования с использованием среды Scratch // Актуальные проблемы методики обучения информатике и математике в современной школе. — 2021. — С. 245–247.

26. Семакин И. Г., Шестаков А. П. Основы алгоритмизации и программирования. Практикум : учеб. пособие для студ. учреждений сред. проф. образования. — М. : Академия, 2013. — 144 с.
27. Соловьева Т. А. Методика преподавания информатики в школе : учеб. пособие. — СПб. : Питер, 2021. — 256 с.
28. Топчу Е. Использование среды программирования Scratch для реализации компетентностного подхода в обучении математике и информатике // *Știință, educație, cultură*. — 2025. — Т. 1. — С. 448–451.
29. Ушаков Д. В. Психология интеллекта и одаренности. — М. : Изд-во Ин-та психологии РАН, 2011. — 464 с.
30. ФГОС Среднее общее образование [Электронный ресурс]. — URL: <https://fgos.ru/fgos/fgos-soo/> (дата обращения: 11.06.2025).

**Критерии оценки обучающихся на уроке «Введение в
программирование на Scratch»**

Лист самооценки проекта в Scratch

ФИО

ученика: _____

Класс: _____

Дата: _____

Шкала оценивания:

«5» - отлично выполнено;

«4» - хорошо, есть недочеты;

«3» - требуется доработка;

«2» - выполнено частично.

Критерии оценки	Показатели выполнения	Баллы (0-5)	Комментарий
1. Техническое исполнение			
Использование базовых блоков	Применено 5+ типов блоков		
Логика программы	Алгоритм работает без ошибок		
Интерфейс проекта	Удобное управление, приятный дизайн		
2. Содержание			
Полнота реализации	Выполнены все требования задания		
Творческие элементы	Добавлены оригинальные улучшения		

3. Профессиональная направленность			
Педагогическая ценность	Четко прослеживается связь с профессией		
Возможность применения	Конкретные варианты использования		
4. Рефлексия			
Какие трудности возникали?			
Что получилось лучше всего?			
Что вы хотели бы усовершенствовать?			

Итоговый балл: _____/35

Оценка:

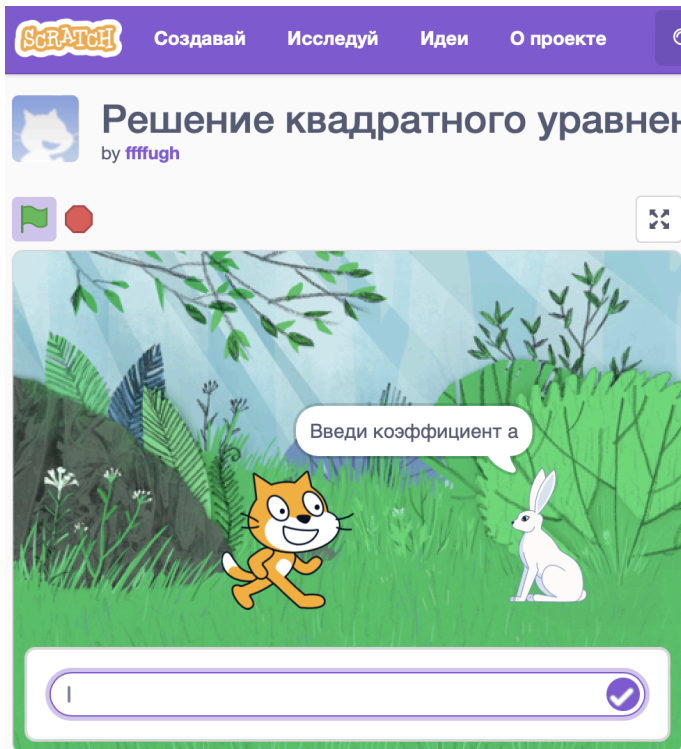
«5» - 30-35 баллов;

«4» - 24-29 баллов;

«3» - 16-23 баллов;

«2» - менее 16 баллов.

Примеры готовых проектов



Scratch Создавай Исследуй Идеи О проекте Поиск Присоединяйся Вход

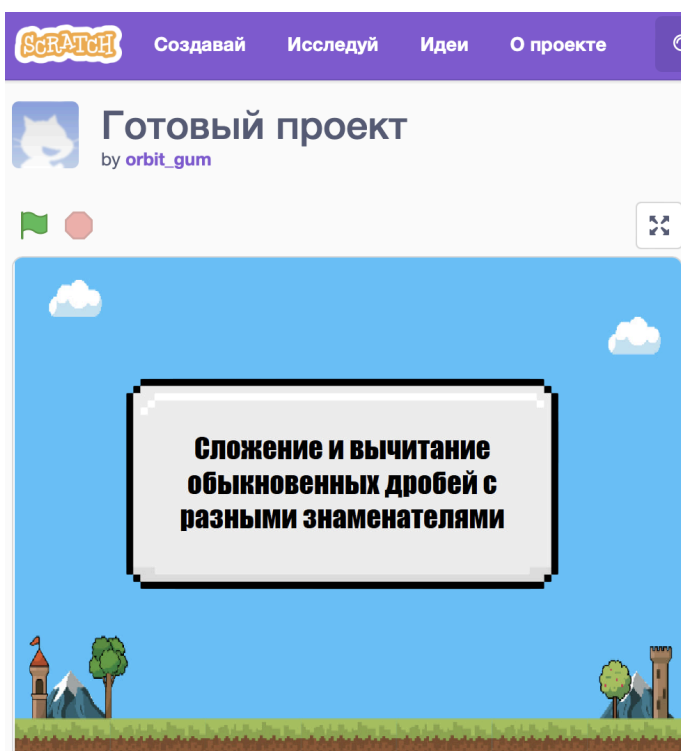
Решение квадратного уравнения

by fffugh

Войти внутрь проекта

Инструкции

Примечания и благодарности



Scratch Создавай Исследуй Идеи О проекте Поиск Присоединяйся Вход

Готовый проект


by orbit_gum




Войти внутрь проекта

Инструкции


Примечания и благодарности


SCRATCH Создай Исследуй Идеи О проекте Поиск Присоединяйся Вход

 **МК_Задачи**
by [MK_scr_Krsk](#) [Войти внутрь проекта](#)

Давай проверим, хорошо ли ты умеешь решать задачи!

 Спасибо [lashina_Irina](#) за исходный проект [Заготовка_МК_Задачи](#).

 Спасибо [Dary83](#) за исходный проект [викторина](#).

Инструкции

Решай интересные задачи и вписывай ответы

Инструкция по дополнению проекта:

“Сложение и вычитание дробей с разными знаменателями”

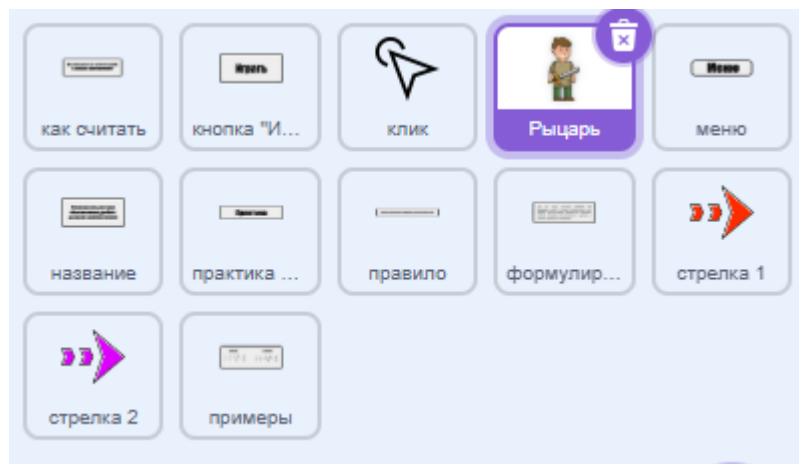
Описание проекта: Проект предназначен для проверки знаний по теме **“Сложение и вычитание дробей с разными знаменателями”**, включает две части теоретическую и практическую. Ваша задача выполнить 4 шага и разобраться, как работает проект

1 Шаг

Откройте заготовку проекта: <https://scratch.mit.edu/projects/1119649692/>

2 Шаг


Ниже представлены спрайты, которые используются в проекте:




Поменяйте спрайт “Рыцарь” на спрайт “Маг” (изображение можно скачать из папки) или на любой другой понравившийся спрайт из библиотеки спрайтов



Как это быстро сделать?

1. Перейдите в костюмы спрайта
2. Нажмите на кнопку  в этой вкладке (смотрите на рис.1 ниже)
3. Далее нажимаем на “Загрузить костюм”
4. Проверьте имя костюма “Маг”

5. Удалите костюм “Рыцарь”  нажав на сам костюм, а далее на значок мусора

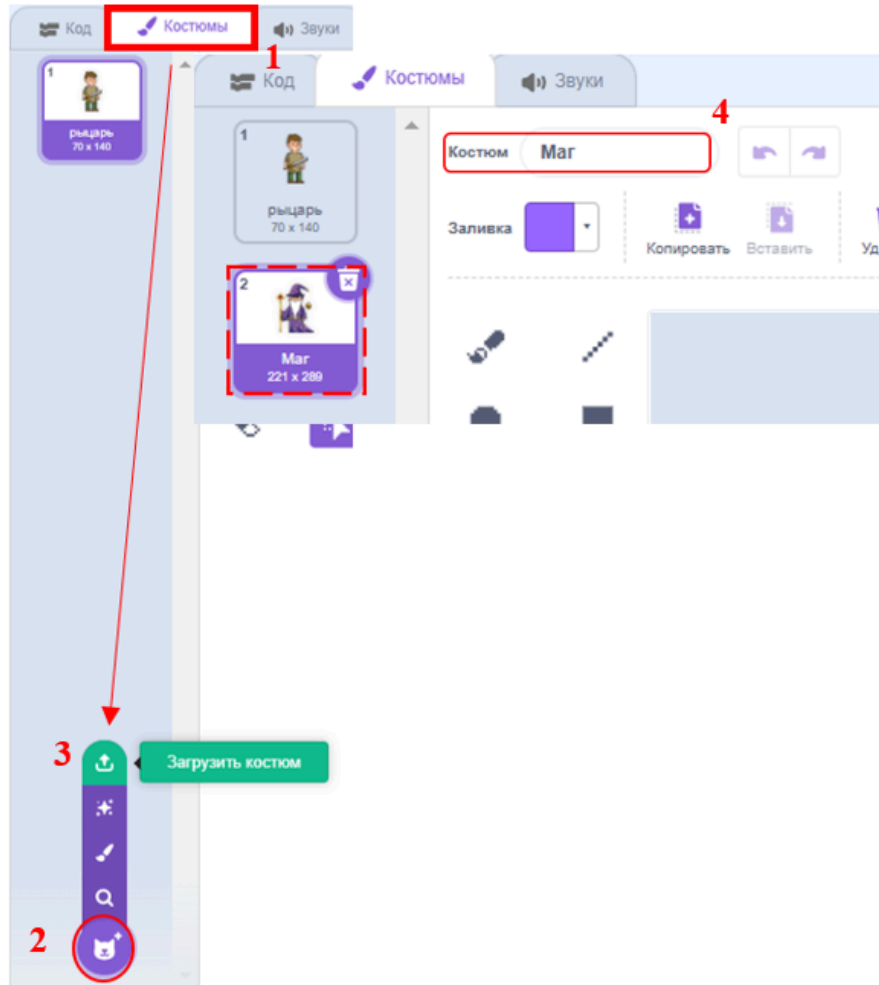


Рис.1

Шаг 3

Поменяйте главный фон на другой: из библиотеки можно загрузить или с папки с фонами (аналогичные действия, как в пункте выше, а также смотрите на рис.2)

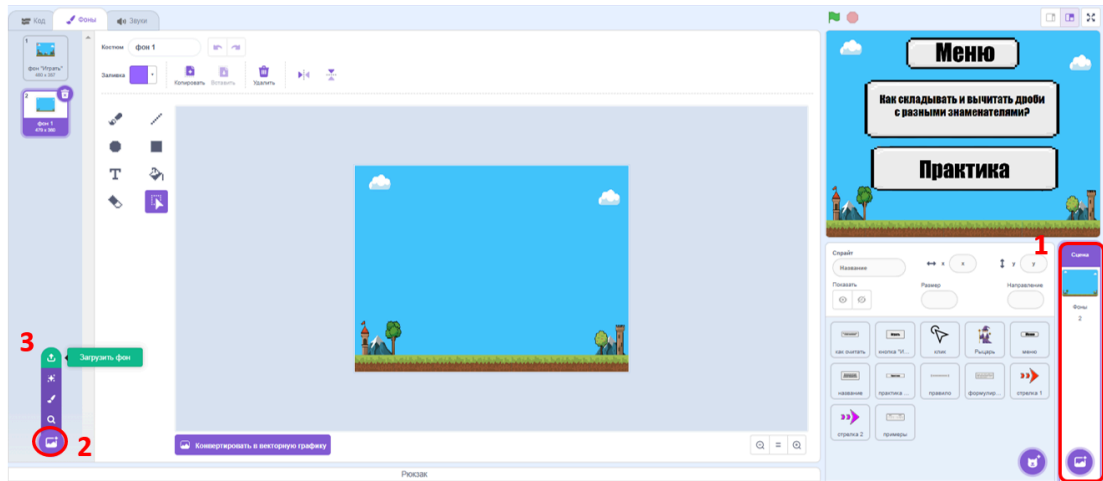
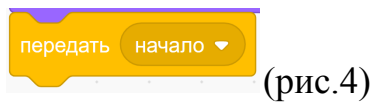


Рис.2

Шаг 4

1. Поменяйте приветственные слова новому спрайту “Маг” или выбранному из библиотеки (рис.3)
2. Ведущий будет задавать вопросы, необходимо ему написать скрипт для 3 вопросов. Добавьте вопросы ведущему, используя пример кода (рис.5), обязательно вставьте свои скрипт перед блоком



(рис.4)

Рис.3

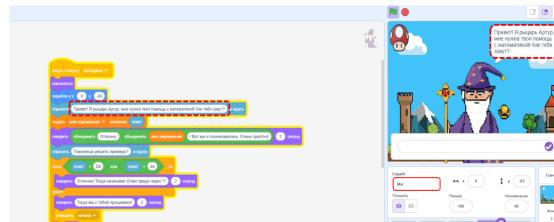


Рис.4

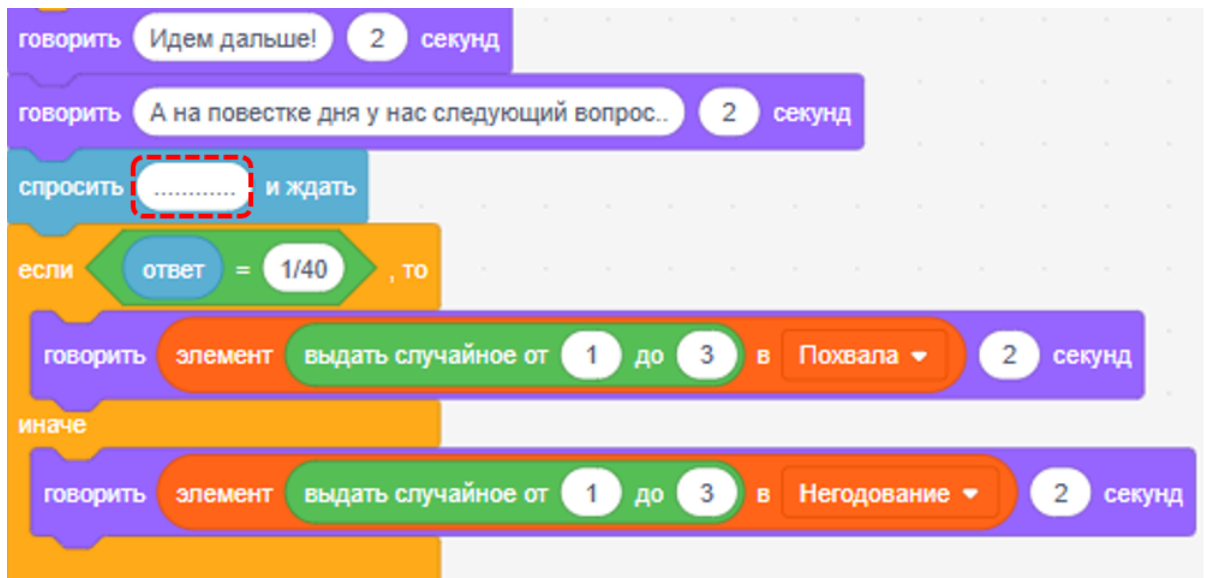
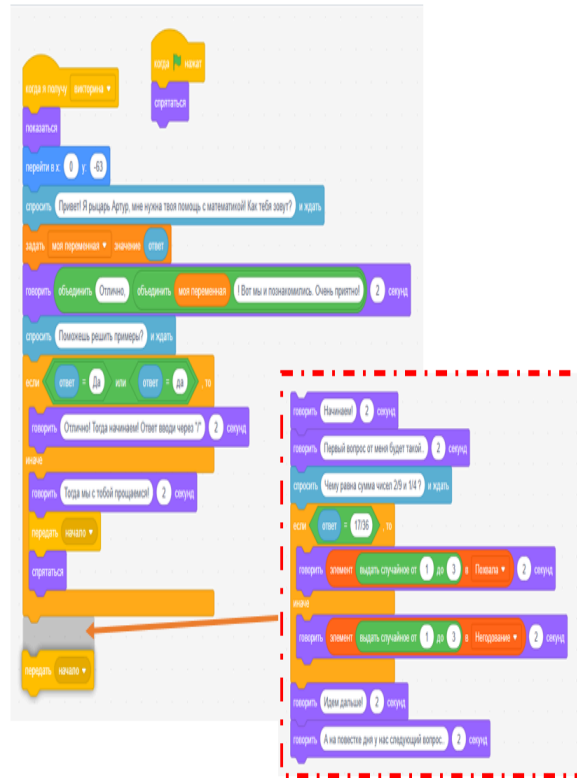


Рис.5