

**Автономная некоммерческая организация высшего образования
«Поволжская академия образования и искусств имени Святителя Алексия,
митрополита Московского»**

Кафедра математики и информатики

Направление подготовки 44.03.01 Педагогическое образование
Направленность (профиль) Информатика и информационные технологии

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**Разработка практикума для программы внеурочной деятельности
по информатике «Основы программирования в среде Scratch»**

Выполнил студент
4 курса группы ИТ– 401
очной формы обучения
Жолудев Никита Сергеевич

Научные руководители:
Бахусова Елена Васильевна,
доцент, канд. пед. наук, зав.
кафедрой математики и
информатики

Давыдова Анастасия
Александровна, старший
преподаватель кафедры
математики и информатики

Допустить к защите:

Заведующий кафедрой

математики и информатики _____ Е. В. Бахусова

« ____ » _____ 2025 г.

Тольятти
2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
Глава 1 Нормативные и педагогические аспекты разработки методического сопровождения для программ внеурочной деятельности по информатике.....	11
1.1 Анализ нормативной документации по разработке методического сопровождения для реализации программ внеурочной деятельности по информатике для учащихся средней школы	11
1.2 Методические аспекты реализации курса внеурочной деятельности «Основы программирования» для 5–6 классов в контексте современных требований ФГОС	20
1.3 Отбор содержания для практикума «Основы программирования в среде Scratch»	26
1.4 Психолого–педагогические подходы к развитию алгоритмического мышления обучающихся на уроках информатики.....	28
Выводы по главе 1	33
Глава 2 Разработка цифрового практикума «Основы программирования в среде Scratch»	34
2.1 Структура и содержание практикума «Основы программирования в среде Scratch»	34
2.2 Разработка электронного образовательного ресурса для реализации практикума «Основы программирования в среде Scratch»	40
2.3 Методические рекомендации для учителей по использованию практикума «Основы программирования в среде Scratch» в учебном процессе	46
Выводы по главе 2.....	50
Глава 3 Опытнo–экспериментальная работа по развитию алгоритмического мышления детей среднего школьного возраста во внеурочной деятельности посредством реализации программы «Основы программирования»	51
3.1 Цель и задачи педагогического эксперимента по развитию алгоритмического мышления у обучающихся 5–х классов	51
3.2 Констатирующий и формирующий этапы педагогического эксперимента .	56
3.3 Контрольный этап педагогического эксперимента	63
Выводы по главе 3	67
ЗАКЛЮЧЕНИЕ	68

БИБЛИОГРАФИЧЕСКИЙ СПИСОК	70
ПРИЛОЖЕНИЯ.....	76

ВВЕДЕНИЕ

В современном мире информационных технологий развитие алгоритмического мышления является одним из ключевых факторов успешной социализации и профессиональной реализации молодого поколения. О необходимости развития особого вида мышления у обучающихся заговорили ещё в советское время. В 1981 году академик Ершов Андрей Петрович на 3–й Всемирной конференции ИФИП (Международной федерации по обработке информации) и ЮНЕСКО по применению ЭВМ в обучении выдвинул лозунг: «Программирование – вторая грамотность», подчеркнув тем самым важность изучения программирования в современном технологичном мире. В своих учебниках по информатики А.П. Ершов ввел раздел, посвящённый изучению алгоритмизации [13].

Согласно Федеральному образовательному стандарту общего образования одной из целей обучения информатике является развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе [3].

Значимость развития алгоритмического мышления школьников подтверждается работами ведущих отечественных и зарубежных ученых, таких как А.Г. Кушниренко [17], Л.Л. Босова [2], Ю.А. Первин, П.Я. Гальперин, И.Г. Семакин, Т.Н. Лебедева [18], а также зарубежных исследователей, таких как Сеймур Пейперт, Митчелл Резник [35]. Эти авторы подчеркивают, что алгоритмическое мышление способствует развитию логических, аналитических и творческих способностей, необходимых для решения задач в условиях информационного общества.

В 2007 году в Массачусетском технологическом институте под руководством Митчелла Резника был разработан блочный язык программирования Scratch [32]. Данная разработка позволила с помощью

конструирования из блоков получать рабочий код. Эта среда, благодаря своей наглядности, подходит для детей среднего школьного возраста, у которых преобладает наглядно–образное мышление. Этот продукт можно сегодня использовать для обучения школьников основам программирования на занятиях внеурочной деятельности.

В 2022 году Институтом стратегии развития образования Российской академии образования разработана примерная рабочая программа курса внеурочной деятельности «Основы программирования» для 5–6 классов [6]. В связи с тем, что документ достаточно новый, учебников и учебно–методических пособий в соответствии с данным документом ещё не разработано.

С 2023 года школы начали работать по новым программам, но столкнулись с проблемой отсутствия учебных материалов, соответствующих данным документам. Актуальность темы развития алгоритмического мышления обучающихся на уроках информатики при реализации программы внеурочной деятельности «Основы программирования» для детей среднего школьного возраста обусловлена следующими факторами:

- важностью информатики как одного из основных направлений образования, способствующего развитию алгоритмического мышления;
- недостаточной разработанностью методик и программ внеурочной деятельности, направленных на развитие алгоритмического мышления у школьников;
- необходимостью формирования навыков и умений, необходимых для успешной социализации и профессиональной ориентации в условиях информационного общества.

Цель исследования: разработать практикум «Основы программирования в среде Scratch» для занятий внеурочной деятельности в 5 классе.

Объект исследования: учебно–методическое обеспечение занятий внеурочной деятельности по информатике в 5 классе.

Предмет исследования: практикум для занятий внеурочной деятельности «Основы программирования в среде Scratch».

Задачи исследования:

- изучить психолого–педагогическую литературу по проблеме развития алгоритмического мышления детей среднего школьного возраста во внеурочной деятельности;
- разработать практикум «Основы программирования в среде Scratch» для реализации программы внеурочной деятельности «Основы программирования» для 5 класса;
- апробировать практикум «Основы программирования в среде Scratch» на занятиях по внеурочной деятельности с обучаемыми 5–х классов;
- организовать и провести педагогический эксперимент по развитию алгоритмического мышления обучаемых;
- провести анализ эффективности развития алгоритмического мышления детей среднего школьного возраста посредством реализации практикума «Основы программирования в среде Scratch».

Гипотеза исследования: развитие алгоритмического мышления детей среднего школьного возраста во внеурочной деятельности будет эффективно если:

- 1) работа ведется с учетом возрастных и индивидуальных особенностей обучающихся;
- 2) разработан практикум для внеурочной деятельности по информатике «Основы программирования в среде Scratch» на основе программы «Основы программирования» с целью развития алгоритмического мышления обучаемых;
- 3) организована целенаправленная систематическая работа по апробации практикума «Основы программирования в среде Scratch» на занятиях внеурочной деятельности по информатике для обучающихся 5 класса.

Методы исследования:

- 1) анализ научно–методической, учебной литературы, учебников и учебных пособий по педагогике, психологии и информатике;
- 2) изучение педагогического опыта по формированию алгоритмического мышления учащихся среднего школьного возраста при изучении информатики;
- 3) педагогическое наблюдение, проведение педагогического эксперимента, статистическая обработка полученных результатов исследования и их анализ.

Структура и объем бакалаврской работы: работа состоит из введения, трех глав, выводов по главам, заключения, библиографического списка и приложений.

В первой главе работы проанализирована нормативная и методическая документация, научная и психолого–педагогическая литература. Сделан отбор содержания для практикума «Основы программирования в среде Scratch». Дан анализ понятия «Алгоритмическое мышление», представленных в работах отечественных и зарубежных авторов в области педагогической науки и психологии.

Вторая глава представляет собой практическую часть работы, включающую разработку практикума и сайта для размещения практикума, используемого при реализации программы внеурочной деятельности «Основы программирования». Описана структура и модель сайта с размещенным практикумом «Основы программирования в среде Scratch».

В третьей главе описаны констатирующий, формирующий и контрольные этапы педагогического эксперимента. Исследование проходило на базе ЧОУ СОШ «Общеобразовательный центр «Школа» г. Тольятти [30]. В апробации приняли участие 43 человека.

В заключении подводятся итоги проделанной работы.

Библиографический список содержит 35 источников информации, использованных при выполнении работы. Работа содержит 25 рисунков, 32 таблицы.

Приложения включают практикум «Основы программирования в среде Scratch», бланки и таблицы с результатами прохождения диагностических методик.

Результаты исследования докладывались на студенческих конференциях:

1. Доклад «Разработка цифрового учебного комплекса для реализации программы внеурочной деятельности «Основы программирования в среде Scratch» для 5 класса», VI Рождественские образовательные чтения тольяттинской епархии: «80–летие Великой Победы: Память и духовный опыт поколений». Проходившие 3 декабря 2024 года в «Поволжской академии образования и искусств имени Святителя Алексия», г. Тольятти.

2. Доклад «Разработка практикума для программы внеурочной деятельности по информатике «Основы программирования в среде Scratch», VIII Региональная молодежная научно–практическая конференция «Поволжский фестиваль студенческой науки». Проходившие 25 марта 2025 года, в «Поволжской академии образования и искусств имени Святителя Алексия», г. Тольятти.

Результаты работы опубликованы в статье «Разработка практикума для программы внеурочной деятельности по информатике «Основы программирования для обучающихся 5 классов» журнала «Поволжский вестник науки» № 2 (36) за 2025 год.

Глава 1 Нормативные и педагогические аспекты разработки методического сопровождения для программ внеурочной деятельности по информатике

1.1 Анализ нормативной документации по разработке методического сопровождения для реализации программ внеурочной деятельности по информатике для учащихся средней школы

В соответствии с Федеральными государственными образовательными стандартами (ФГОС), знакомство школьников с основами информатики начинается уже на уровне начального общего образования. Первые элементы логико–алгоритмического мышления формируются в рамках математического курса, а благодаря интеграции информационно–коммуникационных технологий (ИКТ) в различные учебные дисциплины у учащихся постепенно развиваются ключевые цифровые навыки.

На этапе основного общего образования курс информатики выполняет систематизирующую функцию: он не только опирается на уже имеющийся у школьников опыт применения ИКТ, но и способствует его осмыслению, теоретическому обобщению и дальнейшему развитию. Это создает основу для формирования полноценной ИКТ–компетентности, необходимой как для продолжения обучения, так и для жизни в цифровом обществе.

Цели и задачи изучения предмета «Информатика» в рамках основного и среднего общего образования формируют структуру содержания учебного предмета, которое представлено следующими четырьмя тематическими разделами:

- 1) цифровая грамотность;
- 2) теоретические основы информатики;
- 3) алгоритмы и программирование;
- 4) информационные технологии.

Современные образовательные стандарты (ФГОС) также делают акцент на развитии алгоритмического мышления у обучающихся. В соответствии с ФГОС ООО одним из основных разделов информатики в примерной рабочей программе основного общего образования, является изучение раздела «Алгоритмизации и программирование». Алгоритмическое мышление рассматривается в Федеральном государственном образовательном стандарте основного общего образования (ФГОС ООО) как ключевой компонент метапредметных и предметных результатов в области информатики. Эту мысль повторяет и раскрывает более широко (Примерная образовательная программа) по информатике указывая на возможность формирования базовых логических действий, базовых исследовательских действий, навыков работы с информацией через работу с содержанием предмета и учебной коммуникацией (общением, совместная деятельность) связанного с алгоритмизацией и программированием [8].

Требования, относящиеся к предметным результатам освоения программы базового курса информатики раздела «Алгоритмы и программирование», являются:

- развитие алгоритмического мышления как необходимого условия профессиональной деятельности в современном обществе; понимание сущности алгоритма и его свойств; овладение навыками работы с компьютером как средством управления информацией;

- владение основными понятиями: информация, передача, хранение и обработка информации, алгоритм, модель, цифровой продукт и их использование для решения учебных и практических задач; умение оперировать единицами измерения информационного объема и скорости передачи данных;

- владение основными понятиями: информация, передача, хранение и обработка информации, алгоритм, модель, цифровой продукт и их

использование для решения учебных и практических задач; умение оперировать единицами измерения информационного объема и скорости передачи данных;

– умение составлять, выполнять вручную и на компьютере несложные алгоритмы для управления исполнителями (Черепашка, Чертежник); создавать и отлаживать программы на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык), реализующие несложные алгоритмы обработки числовых данных с использованием циклов и ветвлений; умение разбивать задачи на подзадачи, использовать константы, переменные и выражения различных типов (числовых, логических, символьных); анализировать предложенный алгоритм, определять, какие результаты возможны при заданном множестве исходных значений [3].

Приказ Министерства просвещения Российской Федерации от 12 августа 2022 г. № 732 «О внесении изменений в федеральный государственный образовательный стандарт среднего общего образования» изменения ФГОС СОО и ОО проведены в соответствии с принципом единства образовательного пространства в РФ, заложенном в № ФЗ–273 «Об образовании в Российской Федерации» [5]. С 1 сентября 2023 года в школах России в 5–6 и 7–9 классах по информатике реализуются следующие программы основного общего образования:

1. «Примерная рабочая программа (ООО) по информатике для 5–6 классов (базовый уровень)» [7].
2. «Федеральная рабочая программа (ООО) по информатике для 7– 9 классов (базовый уровень)» [8].



Рисунок 1 – Программы основного общего образования по информатике

Примерная рабочая программа по информатике для 5–6 классов составлена в соответствии с требованиями Федерального государственного образовательного стандарта основного общего образования (ФГОС ООО). Она ориентирована на достижение учащимися ключевых образовательных результатов, предусмотренных стандартом. Программа обеспечивает преемственность с курсом информатики в начальной школе, опираясь на знания и умения, сформированные у обучающихся в рамках ФГОС начального общего образования. При составлении программы учитываются психологические и возрастные особенности школьников 5–6 классов.

В примерной рабочей программе ООО по информатике для 5–6 классов базового уровня на изучение всех разделов в 5 классе отводится 34 часа (1 час в неделю), на изучение раздела «Алгоритмизация и основы программирования» отводится в 5 классе 10 часов. Программа по информатике для 5–6 классов составлена из расчёта общей учебной нагрузки 68 часов за 2 года обучения: 1 час в неделю в 5 классе и 1 час в неделю в 6 классе (Таблица 1).

Таблица – 1 Тематические разделы 5 и 6 класса

Тематические разделы	5 класс	6 класс
Цифровая грамотность	7 часов	4 часа
Теоретические основы информатики	3 часа	6 часов
Алгоритмизация и основы программирования	10 часов	12 часов
Информационные технологии	12 часов	10 часов
Резервное время	2 часа	2 часа
Итого:	34 часа	34 часа

В таблице 2 показаны темы и отводимые на их освоение часы, учебное содержание и основные виды деятельности учащихся.

Таблица 2 – Тематический план раздела «Алгоритмизация и программирование» для 5 классов базового уровня по информатике

Раздел 3. «Алгоритмизация и основы программирования (10 часов)»		
Тема 5. Алгоритмы и исполнители (2 часа)	Понятие алгоритма. Исполнители алгоритмов. Линейные алгоритмы. Циклические алгоритмы.	Раскрывать смысл изучаемых понятий. Приводить примеры неформальных и формальных исполнителей в окружающем мире. Приводить примеры циклических действий в окружающем мире.
Тема 6. Работа в среде программирования (8 часов)	Составление программ для управления исполнителем в среде блочного или текстового программирования. Практические работы 1. Знакомство со средой программирования. 2. Реализация линейных алгоритмов в среде программирования. 3. Реализация циклических алгоритмов в среде программирования.	Раскрывать смысл изучаемых понятий. Анализировать пользовательский интерфейс применяемого программного средства. Определять условия и возможности применения программного средства для решения типовых задач.

Предметные результаты освоения раздела «Алгоритмизация и основы программирования», предусмотренные примерной рабочей программой для 5–6 класса, включают в следующие планируемые результаты:

– понимать содержание понятий: «алгоритм», «исполнитель», «программа управления исполнителем», «искусственный интеллект»;

– составлять программы для управления исполнителем в среде блочного или текстового программирования с использованием последовательного выполнения операций и циклов;

– разбивать задачи на подзадачи;

– составлять программы для управления исполнителем в среде текстового программирования, в том числе с использованием циклов и вспомогательных алгоритмов (процедур) с параметрами.

В соответствии с приказом от 18 июля 2024 г. № 499 об утверждении федерального перечня электронных образовательных ресурсов, допущенных к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования [4]. Установлен перечень электронных образовательных ресурсов, которые можно использовать во время реализации рабочих программ ФГОС, документ содержит описания нижеприведенных данных:

– наименование электронного образовательного ресурса;

– составители электронного образовательного ресурса;

– краткое описание ЭОР включая структуру, предметное содержание и метаданные, позволяющие идентифицировать ЭОР;

– правообладателя электронного образовательного ресурса;

– класс, для которого разрабатывалось.

В соответствии с приказом на уроках информатике можно использовать следующие ЭОР (Таблица 3)

Таблица 3 – ЭОР, используемые на уроках информатике

Наименование ЭОР	Класс	Правообладатель	Ссылка
ЭОР «Домашние задания. Основное общее образование. Информатика» 5 – 9 класс.	5–9	АО Издательство «Просвещение»	https://hw.lecta.ru/?utm_source=prosvet&utm_medium=prosvru&utm_campaign=digital_prosvru

В 7–9 классах предмет «Информатика» (ФГОС ООО) является обязательным для изучения. Для каждого класса предусмотрено резервное учебное время, которое может быть использовано участниками образовательного процесса в целях формирования вариативной составляющей содержания конкретной рабочей программы. При этом обязательная (инвариантная) часть содержания предмета, установленная рабочей программой, и время, отводимое на её изучение, должны быть сохранены полностью (Таблица 4).

Таблица 4 – Тематические разделы 7,8,9 класса

Тематические разделы	Базовый уровень		
	7 класс	8 класс	9 класс
Цифровая грамотность	9 часов	–	6 часов
Теоретические основы информатики	11 часов	12 часов	8 часов
Алгоритмы и программирование;	–	21 час	8 часов
Информационные технологии	13 часов	–	11 часов
Резервное время	1 час	1 час	1 час
Итого:	34 часа	34 часа	34 часа

Общее число часов, рекомендованных примерной рабочей программой ООО для изучения информатики на базовом уровне, – 102 часа: в 7 классе – 34 часа (1 час в неделю), в 8 классе – 34 часа (1 час в неделю), в 9 классе – 34 часа (1 час в неделю).

К концу обучения в соответствии с примерной рабочей программой у учащихся 8 и 9 классов должны быть сформированы следующие результаты, относящиеся к разделу «Алгоритмы и программирование»:

К концу обучения в 8 классе у обучающегося будут сформированы следующие умения:

- раскрывать смысл понятий «исполнитель», «алгоритм», «программа», понимая разницу между употреблением этих терминов в обыденной речи и в информатике;

- описывать алгоритм решения задачи различными способами, в том числе в виде блок–схемы;

- составлять, выполнять вручную и на компьютере несложные алгоритмы с использованием ветвлений и циклов для управления исполнителями, такими как Робот, Черепашка, Чертёжник;

- использовать константы и переменные различных типов (числовых, логических, символьных), а также содержащие их выражения, использовать оператор присваивания;

- использовать при разработке программ логические значения, операции и выражения с ними;

- анализировать предложенные алгоритмы, в том числе определять, какие результаты возможны при заданном множестве исходных значений;

- создавать и отлаживать программы на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык), реализующие несложные алгоритмы обработки числовых данных с использованием циклов и ветвлений, в том числе реализующие проверку делимости одного целого числа на другое, проверку натурального числа на простоту, выделения цифр из натурального числа.

К концу обучения в 9 классе у обучающегося будут сформированы следующие умения:

- разбивать задачи на подзадачи, составлять, выполнять вручную и на компьютере несложные алгоритмы с использованием ветвлений, циклов и вспомогательных алгоритмов для управления исполнителями, такими как Робот, Черепашка, Чертёжник;

– составлять и отлаживать программы, реализующие типовые алгоритмы обработки числовых последовательностей или одномерных числовых массивов (поиск максимумов, минимумов, суммы или количества элементов с заданными свойствами) на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык).

В разделах «Алгоритмизация и основы программирования» и «Алгоритмы и программирование» учебной программы по информатике чётко определены целевые ориентиры обучения. В результате освоения данного курса учащиеся должны:

- понимать основные принципы алгоритмизации и уметь применять их для решения задач;
- владеть базовыми конструкциями программирования (линейные, ветвящиеся и циклические алгоритмы);
- разрабатывать и отлаживать программы на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык);
- анализировать эффективность алгоритмов и оптимизировать их при необходимости.

Обязательная часть учебного плана примерной основной образовательной программы основного общего образования не предусматривает обязательное изучение курса информатики в 5–6 классах. Но время на данный курс образовательная организация может выделить за счёт части учебного плана, формируемой участниками образовательных отношений [7].

Таким образом, обучающиеся могут иметь возможность приступить с изучения раздела «Алгоритмизация и программирование» на занятиях внеурочной деятельности. Для этих целей разработана примерная рабочая программа курса внеурочной деятельности «Основы программирования» [6].

1.2 Методические аспекты реализации курса внеурочной деятельности «Основы программирования» для 5–6 классов в контексте современных требований ФГОС

С 1 сентября 2023 года в школах России в 5–6 классах реализуются Примерная рабочая программа курса внеурочной деятельности «Основы программирования» для 5–6 классов [6]. На рисунке 2 представлена обложка программы.

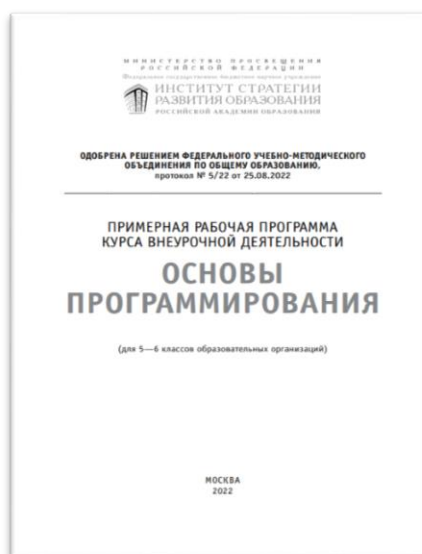


Рисунок 2 – Обложка примерной рабочей программ по информатике

Курс внеурочной деятельности отражает и расширяет содержание четырёх тематических разделов информатики на уровне основного общего образования:

- 1) цифровая грамотность;
- 2) теоретические основы информатики;
- 3) алгоритмы и программирование;
- 4) информационные технологии.

Курс внеурочной деятельности «Основы программирования» содержит следующие разделы, изучаемые в 5 классе:

1. Устройство компьютера (разделы «Цифровая грамотность» «Информационные технологии»).

2. Знакомство со средой визуального программирования Scratch (раздел «Алгоритмы и программирование»).
3. Создание презентаций (раздел «Информационные технологии»);
4. Коммуникация и безопасность в Сети (раздел «Цифровая грамотность»).

Примерная рабочая программа состоит из 4–х разделов и обязывает нас при изучении раздела «Знакомство со средой визуального программирования Scratch» использовать на уроках во внеурочной деятельности блочный язык программирования Scratch.

В соответствии с данным документом, тематический план при изучении раздела «Алгоритмы и программирование» включает темы, представлены в таблице 5.

Таблица 5 – Тематический план раздела «Алгоритмы и программирование» курса внеурочной деятельности «Основы программирования» для 5 класса.

Темы, раскрывающие данный раздел программы, и число часов на их изучение	Содержание программы	Основные виды деятельности обучающегося при изучении темы
Раздел 2. Знакомство со средой визуального программирования Scratch (11 ч)		
Язык программирования	Алгоритмы и языки программирования. Блок–схемы. Линейные алгоритмы. Интерфейс Scratch. Циклические алгоритмы. Ветвление. Среда Scratch: скрипты. Повороты. Повороты и движение. Система координат. Установка начальных позиций. Установка начальных позиций: свойства, внешность. Параллельные скрипты, анимация. Передача сообщений	Определяет по программе, для реше ния какой задачи она предназначена. Программирует линейные, циклические и разветвляющиеся алгоритмы. Осуществляет действия со скриптами

В примерной рабочей программе курса внеурочной деятельности «Основы программирования» для 5–6 классов на изучение всех разделов по информатике в 5 классе отводится 34 часа, 1 час в неделю. Из таблицы мы видим, что для

изучения раздела 2. Знакомство со средой визуального программирования Scratch, обучающие знакомятся с алгоритмами, блок– схемой, а также скриптами и интерфейсом в среде Scratch. На изучение раздела «Знакомство со средой визуального программирования Scratch» отводится в 5 классе 11 часов. Программа курса внеурочной деятельности разработана для проведения занятий в рамках направления «Углублённое изучение учебных предметов». Для каждого класса выделено резервное учебное время, предназначенное для гибкого формирования вариативной части рабочей программы. Эти часы могут быть использованы на повторение материала, а также на представление результатов проектной деятельности учащихся.

Одной из приоритетных целей курса внеурочной деятельности, является развитие алгоритмического и критического мышления, что предполагает способность обучающегося разбивать сложные задачи на более простые подзадачи.

«Задачи курса внеурочной деятельности «Основы программирования» – сформировать у обучающихся:

- понимание принципов устройства и функционирования объектов цифрового окружения, представления об истории и тенденциях развития информатики периода цифровой трансформации современного общества;
- владение основами информационной безопасности;
- знания, умения и навыки грамотной постановки задач, возникающих в практической деятельности, их решение с помощью информационных технологий;
- умения и навыки формализованного описания поставленных задач;
- знание основных алгоритмических структур и умение применять эти знания для построения алгоритмов решения задач по их математическим моделям;

- умения и навыки эффективного использования основных типов прикладных программ (приложений) общего назначения и информационных систем для решения с их помощью практических задач;
- умение грамотно интерпретировать результаты решения практических задач с помощью информационных технологий, применять полученные результаты в практической деятельности» [6].

Примерная рабочая программа курса внеурочной деятельности «Основы программирования» для 5–6 классов представляет собой структурированный курс, направленный на формирование у обучающихся базовых навыков алгоритмического мышления, работы с цифровыми технологиями и основ программирования в среде Scratch. Примерная рабочая программа курса устанавливает количественные и качественные параметры учебного материала для каждого года обучения, включая: планируемые результаты освоения программы обучающимися на уровне основного общего образования и систему оценки достижения этих результатов.

Программа разработана в соответствии с Федеральным государственным образовательным стандартом основного общего образования (ФГОС ООО) и является основой для разработки учителем поурочно–тематического планирования курса внеурочной деятельности. В рамках предметных результатов курс формирует навыки создания и выполнения алгоритмов в среде программирования. «Изучение информатики в 5–6 классах поддерживает непрерывность подготовки школьников в этой области и обеспечивает необходимую теоретическую и практическую базу для изучения курса информатики основной школы в 7–9 классах» [7]. Выделение 11 часов на изучение среды Scratch в 5 классе позволяет учащимся освоить ключевые концепции программирования через практическую деятельность, что создаёт основу для дальнейшего углублённого изучения программирования.

Рассмотрим учебно–методический комплекс (УМК) по информатике под редакцией Л. Л. Босовой. УМК Босовой Л.Л включает в себя:

- авторская учебная программа по информатике для основной школы;
- учебники для каждого года обучения;
- электронное приложение к каждому учебнику;
- рабочие тетради для каждого года обучения;
- методические пособия для учителя по преподаванию курсов 5–6 и 7– 9 классов.

Практикум входит в УМК по информатике для 5–6 классов авторского коллектива под руководством Л. Л. Босовой. Практикум предназначен как для обычных уроков, так и для организации внеурочных занятий таких как: кружковой или факультативной. Также подходит и для индивидуальной работы со школьниками, направленной на формирование информационной культуры школьников, а также на развитие их алгоритмического мышления и творческих способностей в процессе изучения программирования Scratch.

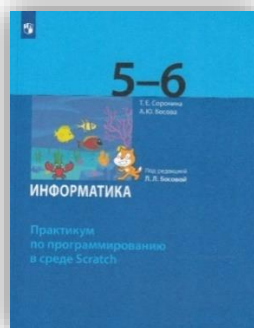


Рисунок 3 – Обложка практикума по программированию в среде Scratch

Материал пособия, построен на основе проектно–исследовательского подхода, который соответствует требованиям ФГОС. Большое количество и разнообразие заданий в практикуме даёт учителю возможность адаптировать материал к фактическому уровню подготовки школьников и выделенному на освоение материала объёму учебного времени.

Практику состоит из 23 тем для изучения Scratch. В каждой теме прилагается QR код, отсканировав который открывает видео урок по данной теме.

1. Знакомство со средой программирования Scratch.
2. Спрайт, цвет и размер пера.
3. Способы смены внешнего вида (костюма).
4. Знакомство с векторным графическим редактором LibreOffice Draw
5. Линейные алгоритмы. Линии.
6. Линейные алгоритмы. Квадраты и прямоугольники.
7. Циклические алгоритмы. Квадраты, линии.
8. Циклические алгоритмы. Разные фигуры.
9. Вложенные циклы. Квадрат из квадратов.
10. Вложенные циклы. Пунктирная линия с поворотом.
11. Анимация на основе готовых костюмов.
12. Сцена как исполнитель. Создаём модель таймера.
13. Одинаковые программы для разных исполнителей.
14. Параллельное выполнение действий несколькими исполнителями.
15. Планирование работы. Таймер.
16. Анимация «Солнечные сутки».
17. Алгоритмы с ветвлением. Условие «если».
18. Циклические конструкции с условием. Шарики в лабиринте.
19. Случайные числа. Исполнитель определяет цвета.
20. Знакомство со слоями. Самолёт сквозь облака.
21. Работа со слоями. Дорога.
22. Взаимодействие исполнителей.
23. Последовательное выполнение команд исполнителями [15].

УМК включает в себя разнообразные компоненты, обеспечивающие полноценное изучение предмета: авторскую учебную программу, электронные приложения, рабочие тетради и методические пособия для учителей.

1.3 Отбор содержания для практикума «Основы программирования в среде Scratch»

В рамках исследования среда Scratch была выбрана в качестве основного инструмента для разработки практикума «Основы программирования в среде Scratch».

Scratch – это визуальная среда программирования, разработка началась ещё в 2003 году командой MIT Media Lab под руководством Митчелла Резника [35]. Основная цель проекта – сделать программирование доступным и увлекательным для детей. Первая публичная версия (Scratch 1.0) вышла в 2007 году, а в 2013 году была выпущена Scratch 2.0 с веб–версией. В 2019 году появился Scratch 3.0, который работает в современных браузерах и поддерживает мобильные устройства. Scratch предлагает интуитивно понятный интерфейс, основанный на блочном программировании [32].

Основные возможности среды Scratch:

Графический редактор для создания и редактирования спрайтов (персонажей) и фонов.

Библиотека звуков и изображений, а также возможность загрузки собственных медиафайлов.

Блоки команды, разделенных на категории: Движение, внешность, звук, события, управление, сенсоры, операторы, переменные и списки.

Scratch поддерживает создание мультимедийных проектов таких как: игры, анимации, интерактивные истории, симуляции интеграция с внешними устройствами (микроконтроллерами Lego Mindstorms, micro: bit). Современная

версия Scratch 3.0 работает в браузере и не требует установки, но есть и офлайн-версия.

Минимальные системные требования для использования среды Scratch:
ОС: Windows 10/macOS 10.13+. Интернет (для онлайн-версии).

Аппаратные требования:

- Процессор: должен быть с тактовой частотой не менее 1,6 ГГц или выше.
- Оперативная память (RAM): 4 ГБ;
- Место на диске: 500 МБ (для офлайн-редактора).

Интернет: требуется для онлайн-версии и загрузки проектов

В рамках изучения языка программирования Scratch разработаны разнообразные методические материалы, включающие:

учебники и учебные пособия, систематизированные издания, содержащие теоретические основы и практические задания: «Scratch для детей» М. Мажед. [20]. «Scratch для юных программистов» Д.В. Голиков [12]. «Программирование для детей на Scratch» К. Вордерман [24]. «Программирование на Scratch 3.0 для детей. Уровень 1» Ю. В. Пашковская [22]. Материалы образовательных проектов «Кодвардс» [27], «Учи.ру») [33], «IT-куб» [28].

Выделим основные особенности визуальной среды Scratch.

1. Визуальное программирование: Scratch использует блоки, которые нужно соединять, как паззлы. Это делает процесс обучения интуитивно понятным и избавляет от необходимости запоминать сложный синтаксис, как в текстовых языках программирования.

2. Образовательная направленность: Scratch был разработан специально для обучения детей и подростков. Он активно используется в школах и на курсах программирования, так как помогает развивать логическое мышление, алгоритмическое мышление и навыки решения задач.

3. Поддержка мультимедиа: в Scratch можно работать с графикой, звуками и музыкой, что делает его привлекательным для создания ярких и интерактивных проектов.

4. Обучение через игру: Scratch превращает процесс программирования в игру, что особенно важно для детей, так как они учатся, не замечая сложностей.

5. Бесплатность и доступность: Scratch полностью бесплатен и работает в браузере, что делает его доступным для всех, у кого есть доступ к интернету.

Современное образование активно интегрирует цифровые технологии, в том числе средства визуального программирования, для развития у школьников и студентов навыков алгоритмического мышления и основ робототехники. Однако для работы с реальными роботами и микроконтроллерами требуется расширение её функционала среды. В условиях импортозамещения и развития отечественных IT-решений особый интерес представляют российские аналоги популярных образовательных платформ. RobboScratch – это разработанная в России расширенная версия Scratch, разработанная компанией РОББО и адаптированная для работы с оборудованием РОББО.

RobboScratch поддерживает программирования такого оборудования как: роботов, датчиков, плат Arduino. Это позволяет использовать знакомый интерфейс Scratch для управления физическими устройствами, что делает обучение более наглядным и практико – ориентированным. Использование данной среды в школах и кружках робототехники поддерживает цель технологического суверенитета и снижает зависимость от зарубежного ПО [31].

1.4 Психолого–педагогические подходы к развитию алгоритмического мышления обучаемых на уроках информатики

В наши дни информационные технологии оказывают значительное воздействие на различные аспекты человеческой деятельности, и для того, чтобы стать грамотным пользователем этих технологий, человеку необходимо иметь

представление об их устройстве и принципах функционирования. Это представление следует формировать с самого раннего возраста, в том числе и при изучении раздела «Алгоритмы и программирование» в школьном курсе информатики.

С 1985 года информатика была введена как обязательный предмет в школах СССР. Первый учебник по информатике фокусировался на разделе «Алгоритмизация и программирование». Обучая этому предмету школьников, к формированию алгоритмического мышления стали подходить более системно.

Как отмечают исследователи С.В. Панюкова и О.А. Прусакова в своей работе: «Становление информатики в качестве школьного учебного предмета с самого начала было ориентировано на развитие различных видов мышления: школьного (в котором автор делает акцент на умение ученика выделять главное и выстраивать логические связи для самостоятельного построения решения) (А.П. Ершов), алгоритмического (А.Г. Кушнеренко), операционного, логического, критического (И.О. Загашев, С.И. Заир–Бек, К. Мередит, И.В. Муштавинская, Р. Поул, Д. Стал, Ч. Темпл, С. Уолтер, Д. Халперн, и др.)» [21].

Формирование алгоритмического мышления признаётся одной из ключевых целей школьного курса информатики (ФГОС ООО). Алгоритмическое мышление является фундаментальным компонентом познавательной деятельности, формирующим у обучающихся способность к структурированию задач, анализу условий и построению последовательных действий для их решения. Федеральные государственные образовательные стандарты (ФГОС), также делают акцент на метапредметные результаты, и предметных результатах [3].

Согласно определению академика РАО А.Г. Асмолова, алгоритмическое мышление входит в блок познавательных универсальных учебных действий,

которые включают: логические действия, знаково–символические действия, действия постановки и решения проблем [9].

В педагогике и психологии выделяются несколько ключевых подходов, которые помогают эффективно развивать алгоритмическое мышление на уроках информатики. Формирование алгоритмического мышления у школьников требует учёта возрастных психологических особенностей. В младшем школьном возрасте преобладает наглядно–образное мышление [23], поэтому используются методы визуализации алгоритмов (блок–схемы). В среднем и старшем звене развивается абстрактно–логическое мышление, что позволяет переходить к более сложным алгоритмическим конструкциям, таким как сочетание алгоритмов ветвления и циклических алгоритмов.

Л. С. Выготский считал, что высшие психические функции (память, внимание, мышление, воображение и др.) развиваются через использование знаков и инструментов (язык, символы, схемы). Алгоритмическое мышление опирается на формальные символы (код, блок–схемы, логические структуры), которые помогают структурировать мысли. В контексте обучения информатике это означает, что для успешного формирования алгоритмического мышления необходимо:

- постепенное усложнение задач – от простых линейных алгоритмов к сложным структурам;
- использование деятельностного подхода – обучение через практику, решение реальных задач, проектирование алгоритмов [10].

Как отмечает доктор педагогических наук С. В. Панюкова, «Основной особенностью алгоритмического мышления считается умение определять последовательность действий (алгоритм), необходимую для решения поставленной задачи» [21].

В рамках исследования определения алгоритмического мышления в школьном курсе информатики представляется необходимым осуществить

сравнительный анализ трактовок данного понятия в работах ведущих отечественных и зарубежных авторов в области педагогической науки и психологии.

Людмила Леонидовна Босова, автор школьных учебников по информатике в своих работах подчеркивает, что развитие алгоритмического мышления – одна из ключевых целей изучения информатики в школе, направленная на формирование умений структурировать информацию, понимать свойства алгоритмов и применять их для решения практических задач [2].

Татьяна Николаевна Лебедева раскрывает понятие алгоритмического мышления, акцентируя его связь с рекурсивными алгоритмами и их изучением в профильных классах средней школы [18].

Она определяет алгоритмическое мышление как совокупность знаний и умений, включающих:

- Понимание структуры алгоритмов: способность анализировать и строить последовательности действий для решения задач.
- Работа с рекурсией: умение применять рекурсивные методы, включая декомпозицию задач, определение базовых случаев и построение рекурсивных решений.
- Критерии сформированности: владение навыками анализа задачи, выделения ключевых элементов, построения алгоритма и его проверки на корректность.

Т.Н. Лебедева подчеркивает, что формирование алгоритмического мышления происходит через выполнение творческих заданий, способствующих развитию логического и системного подхода к решению задач.

Создатель обучающей системы программирования КуМир (Комплект Учебных МИРов), разработанной для обучения школьников, Анатолий Георгиевич Кушниренко рассматривает алгоритмическое мышление как творческий процесс, включающий способность придумывать, обсуждать и

корректировать алгоритмы, а не только их выполнять. Он подчеркивает важность целостного восприятия проблемы, её декомпозиции на части и поиска оптимальных решений [17].

Академик Андрей Петрович Ершов, выдающийся советский ученый в области информатики, ввел понятие «операционный стиль мышления», которое тесно связано с алгоритмическим мышлением. Согласно Ершову, алгоритмическое мышление (или операционный стиль мышления) представляет собой способность поэтапно и осознанно решать задачи, следуя строгой логической последовательности, соответствующей нормам формальной логики. Это включает умение анализировать задачу, разбивать ее на отдельные шаги, составлять план действий и преобразовывать входные данные в выходные, что особенно важно в программировании и решении вычислительных задач [25]. А.П. Ершов подчеркивал, что развитие алгоритмического мышления необходимо для формирования навыков программирования и решения широкого круга задач, требующих структурированного подхода. Это понятие стало основой для изучения алгоритмизации в школьной информатике, где акцент делался на создании алгоритмов и использовании таких инструментов, как блок – схемы и учебные алгоритмические языки.

В работе А.П. Ершова «Программирование – вторая грамотность» (1981) особое внимание уделяется роли программирования в образовании и формировании операционного мышления. Автор раскрывает связь законов программирования с математическим образованием, подчеркивая их значение для развития «операционного и комбинаторного мышления, способности к абстракции, рассуждению и действию» автор приводит пример задачи возведения числа в степень, демонстрирующий переход от знания к действию через последовательные этапы решения. На основе данного примера и описания А.П. Ершовым процесса программирования можно сформулировать определение алгоритмического мышления как способности человека строить

последовательные шаги для решения задач, анализировать условия, контролировать процесс и переходить от знания к действию через логически выстроенные операции [13].

Таким образом, концепция А.П. Ершова подчеркивает значимость алгоритмического мышления как фундаментального навыка, необходимого для развития интеллектуальных способностей учащихся.

Выводы по главе 1

В первой главе проведён анализ нормативных, методических и психолого–педагогических аспектов разработки методического сопровождения для программ внеурочной деятельности по информатике, направленных на развитие алгоритмического мышления у школьников, обучающихся в 5 классах.

Scratch был выбран в качестве основной среды для обучения программированию в 5 классах, так как он:

1. Позволяет визуализировать алгоритмы, что соответствует наглядно–образному мышлению младших подростков.
2. Поддерживает проектную деятельность развивая алгоритмическое мышление.
3. Имеет широкую образовательную экосистему (учебные материалы, сообщество, интеграция с робототехникой через RobboScratch).

Разработанное методическое сопровождение должно учитывать нормативные требования, возрастные особенности учащихся и современные образовательные технологии, включая оптимальный выбор среды программирования, чтобы обеспечить эффективное развитие алгоритмического мышления.

Глава 2 Разработка цифрового практикума «Основы программирования в среде Scratch»

2.1 Структура и содержание практикума «Основы программирования в среде Scratch»

Для разработки практикума «Основы программирования в среде Scratch» выделены основные этапы работы:

- 1) Определение целей и задач практикума «Основы программирования в среде Scratch».
- 2) Проектирование структуры и содержания практикума «Основы программирования в среде Scratch».
- 3) Создание образовательного материала для практикума «Основы программирования в среде Scratch» которые включает теорию, практические задания, контрольные вопросы.
- 4) Выбор платформы для реализации для практикума «Основы программирования в среде Scratch»: определение платформы, на которой будет размещен практикум.
- 5) Проектирование внешнего вида образовательного сайта.

Учитывая рассмотренные особенности Scratch и основываясь, на содержание рабочей программы внеурочной деятельности «Основы программирования» для 5–6 классов, мы составили тематический план и разработали практикум, опишем структуру и содержание практикума «Основы программирования в среде Scratch» (Таблица 6).

Таблица 6 – Содержание практикума «Основы программирования в среде Scratch» и курса внеурочной деятельности «Основы программирования» для 5 класса

Содержание практикума «Основы программирования в среде Scratch»		
Название тем и кол– во часов	Цель занятия	Содержание занятия
1. Знакомство со средой программирования Scratch (1 час)	Ознакомиться с возможностями среды программирования Scratch.	Рассмотрение продукции созданной командой разработчиков Scratch. Исследование функциональных возможностей визуальной среды. Scratch. Разбор основных понятий: спрайт, костюмы, звуки, сцена, сценарий, блоки управления, код, скрипты. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами.		
2. Создание и редактирование объектов (1 час)	Научиться создавать и редактировать спрайты и фоны, использовать такие инструменты как: кисть, заливка, ластик, линия, круг, текст, прямоугольник, изменение формы.	Создание и редактирования спрайта и костюма с использованием инструментов рисования таких как: кисть, заливка, ластик, линия, круг, текст, прямоугольник, изменение формы. Векторный и растровый режимы графики в Scratch. Управление порядком отображения объектов. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Определяет по программе, для решения какой задачи она предназначена.		
3. Направление и движение спрайта (1 час)	Научиться использовать команды поворота для указания направления и движения спрайта; понимать как работает координатная система в Scratch.	Роль спрайта в компьютерной графике и анимации. Понятие координатной системы: оси X и Y. Как спрайты перемещаются, изменение координат. Различные типы движения: линейное, круговое, случайное. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Программирует линейные алгоритмы.		
4. Взаимодействие между объектами (1 час)	Научиться использовать блоки кода для создания взаимодействий между спрайтами; познакомиться с концепцией событийного программирования.	Определение взаимодействий объектов: что такое объекты в Scratch и как они могут взаимодействовать друг с другом. Значение взаимодействий в создании динамичных и интерактивных проектов. Создание анимационных эффектов. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Программирует линейные алгоритмы.		

Продолжение таблицы 6 – Содержание практикума «Основы программирования в среде Scratch» и курса внеурочной деятельности «Основы программирования» для 5 класса

5. Линейный алгоритм (1 час)	Понимать, как устроен линейный алгоритм, научиться создавать проект который будет включать линейный алгоритм.	Примеры простых линейных алгоритмов (например, рисование линии, вывод текста). Демонстрация пошагового выполнения команд. Деятельность обучающегося: Просмотр видео–лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Программирует линейные алгоритмы.		
6. Циклический алгоритм (1 час)	Понимание циклического алгоритма, использование цикла для выполнения повторяющихся действий; создание проекта в котором используют циклы для автоматизации задач и упрощения кода.	Примеры циклических алгоритмов. Демонстрация работы с различными видами циклов. Применение циклов для автоматизации повторяющихся операций. Деятельность обучающегося: Просмотр видео–лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Программирует циклические алгоритмы.		
7. Разветвляющийся алгоритм (1 час)	Понимание разветвляющегося алгоритма, научиться создавать проекты, использующие условные операторы для принятия решений.	Примеры сложных разветвленных алгоритмов (например, игра с выбором вариантов ответов). Демонстрация работы с вложенными условиями. Применение логических операторов (И, ИЛИ, НЕ). Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Программирует разветвляющиеся алгоритмы.		
8. Счетчик (1 час)	Изучение блоков «Переменные» и «Операторы»; знать, что такое переменные и для чего они нужны в программировании; уметь создавать переменные.	Примеры использования переменных в различных алгоритмах (линейных, разветвляющихся, циклических). Демонстрация изменения значений переменных в процессе выполнения программы. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Осуществляет действия со скриптами. Определяет по программе, для решения какой задачи она предназначена.		
9. Создание анимационного проекта (1 час)	Создать анимационный проект. Изучение применение генерации случайных чисел для создания динамичных и интерактивных игровых элементов.	Создание анимационного проекта с использованием оператора генерации случайных чисел. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами.		

Продолжение таблицы 6 – Содержание практикума «Основы программирования в среде Scratch» и курса внеурочной деятельности «Основы программирования» для 5 класса

10. Игры, создаваемые в среде Scratch (1 час)	Научиться разрабатывать и создавать игры в Scratch.	Изучения процесса создания игры в Scratch. Планирование игры (идея, сюжет, персонажи). Примеры создаваемых игр в Scratch. Создание персонажей и объектов. Программирование механики игры (управление, взаимодействия, условия победы/поражения). Тестирование игры. Деятельность обучающегося: Просмотр видео– лекции и презентации. Выполнение практического задания. Прохождение теста.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами. Определяет по программе, для решения какой задачи она предназначена.		
11. Итоговое занятие: Создание собственного проекта– игры (1 час)	Создавать собственную игру в Scratch, используя изученные концепции. Определить планирование и дизайна игры перед началом написания проекта.	Выбор жанра игры (платформер, аркада, головоломка и т.д.). Определение основных механик и взаимодействий в игре. Деятельность обучающегося: Разработка сюжета и темы игры. Создание игры.
Виды деятельности в соответствии с программой: Осуществляет действия со скриптами.		

Практикум «Основы программирования в среде Scratch» предназначен для обучения основам программирования детей среднего школьного возраста для изучения на уроках внеурочной деятельности. Практикум позволит развивать алгоритмическое мышление, развивать навыки решения простых задач по программированию через создание интерактивных проектов: игр, анимационных проектов. Для успешного освоения практикума по Scratch учащиеся должны обладать следующими компетенциями: уметь работать с компьютером на базовом уровне, запускать программы, пользоваться мышью и клавиатурой, сохранять и открывать файлы, иметь минимальный опыт работы в графических редакторах, уметь пользоваться браузером для использования для онлайн–версии Scratch.

В процессе работы с платформой Scratch учащиеся осваивают базовые принципы программирования, такие как циклы, условия и переменные, а также учатся работать с графическими объектами и звуками.

Реализация практикума предусмотрена как в очном, так и в дистанционном формате обучения. Практикум разработан для работы со средой Scratch 3.0. Разработанный нами практикум рассчитан на 11 учебных часов, по 1 часу в неделю. Можно организовать использование практикума по методике «Перевернутый класс». Тип проводимого урока включает в себя теорию и практические задания, контрольные вопросы по каждой теме разработанного практикума. Для работы в среде программирования Scratch присутствует возможность входа в браузер или установка среды на компьютер. Форма контроля – итоговое занятие, создание игры.

Цель практикума: выполнение практических работ и обучения основам программирования с целью развития алгоритмического мышления.

Задачи практикума:

- Освоение интерфейса и базовых возможностей Scratch: Знакомство со средой (спрайты, фон, блоки кода). Работа с графическим редактором (создание и редактирование объектов). Понимание системы координат и управления движением спрайтов.
- Изучение основных алгоритмических структур: Линейные алгоритмы (последовательное выполнение команд). Циклические алгоритмы (повторяющиеся действия с помощью циклов). Разветвляющиеся алгоритмы (условные операторы если, иначе).
- Работа с переменными и взаимодействием объектов: Создание и использование переменных (счетчик, параметры игры). Организация взаимодействия между спрайтами (события, передача сообщений).
- Разработка анимации и интерактивных элементов: Создание анимационных проектов (движение, смена костюмов). Применение генерации случайных чисел для динамичных эффектов.

– Создание завершенного игрового проекта: Планирование игры (сюжет, механики, дизайн). Программирование игровой логики (управление, победа/проигрыш). Тестирование и доработка проекта.

Практикум «Основы программирования в среде Scratch» состоит из четырех модулей и 11 тем.

Модуль 1. «Основы работы в Scratch»: содержит темы «Знакомство со средой программирования Scratch (Тема 1). Создание и редактирование объектов (Тема 2)».

Модуль 2. «Управление движением и взаимодействием»: содержит темы «Направление и движение спрайта (Тема 3). Взаимодействие между объектами (Тема 4)».

Модуль 3. «Основы алгоритмизации»: содержит темы «Линейный алгоритм (Тема 5). Циклический алгоритм (Тема 6). Разветвляющийся алгоритм (Тема 7). Счетчик (Тема 8)».

Модуль 4. «Создание проектов в Scratch»: содержит темы «Создание анимационного проекта (Тема 9). Игры, создаваемые в среде Scratch (Тема 10).

Итоговое занятие: Создание собственного проекта игры (Тема 11)».

В практикуме представлены методические указания по изучению темы.

В конце каждой темы приведены контрольные вопросы.

Практикум «Основы программирования в среде Scratch» представлен в приложении А.

После прохождения практикума «Основы программирования в среде Scratch» обучающиеся смогут:

- 1) Понимать основные виды алгоритмов (циклы, ветвления, переменные).
- 2) Создавать анимационные проекты.
- 3) Работать с графическими элементами и звуком.
- 4) Использовать визуальные блоки для написания программного кода.
- 5) Разрабатывать собственные компьютерные игры.

6) Представлять результаты своей работы в виде законченных цифровых продуктов.

Для того, чтобы убедиться, что практикум «Основы программирования в среде Scratch» развивает алгоритмическое мышление у детей среднего возраста, было проведено исследование на базе школы ООЦ г. Тольятти [30]. Для оценки эффективности практикума проведено исследование с использованием методик Бориса Леонидовича Покровского и Анатолия Залмановича Зака, направленных на диагностику уровня алгоритмического мышления до и после прохождения курса.

2.2 Разработка электронного образовательного ресурса для реализации практикума «Основы программирования в среде Scratch»

С развитием информационно–коммуникационных технологий стали интенсивно развиваться и электронные образовательные ресурсы (ЭОР).

Одним из элементов электронной образовательной среды образовательной организации является электронный образовательный ресурс. Электронный образовательный ресурс (ЭОР) – образовательный ресурс, представленный в электронной цифровой форме и включающий в себя структуру, предметное содержание и метаданные о них [1]. С развитием электронных технологий в современном обществе медиакommunikации и электронные образовательные ресурсы (ЭОР) изменили подход к обучению и методы взаимодействия между преподавателями и обучающимися. Доступность интернета, появление новых коммуникационных технологий взаимодействия, сделали образование более доступным, эффективным и интерактивным [26].

Существует несколько типов ЭОР, среди которых существуют: электронные учебники, видеоуроки, интерактивные тесты и задания [26]. Получается ЭОР это совокупность электронных материалов, предназначенных для использования в образовательной деятельности, которые могут быть

использованы как в традиционном обучении, так и в дистанционных формах образования.

Опишем достоинства использования ЭОР в учебном процессе. Важное достоинство ЭОР – доступность. ЭОР могут быть доступны в любое время и из любого места, что обеспечивает гибкость в обучении. Многие ЭОР включают элементы интерактивности, которые способствуют более активному вовлечению студентов в процесс обучения. Следующее достоинство ЭОР – модифицируемость. ЭОР могут быть легко обновлены и адаптированы под конкретные потребности образовательного процесса. Многообразие форматов. ЭОР могут быть представлены в различных форматах, таких как текст, видео, аудио, анимация и т.д.

Теоретический анализ научных исследований в области педагогического дизайна как анализирует автор Т.С. Кочурина, определяют его как технологию разработки цифровых образовательных ресурсов, включающую анализ, проектирование, реализацию, применение, контроль, оценку эффективности на основе методических, эргономических, функциональных, технико–технологических требований к построению электронной образовательной среды [16].

Анализируя актуальные тенденции в области образования, можно подчеркнуть: педагогический дизайн представляет собой передовые технологии современного обучения. В этом контексте электронные образовательных ресурсы включают в себя от учебно–практического материала и методических рекомендаций до нормативно–правовой информации. Обеспечивая высокое качество образовательного процесса.

Концепция применения педагогических подходов в проектировании образовательных процессов можно увидеть на девяти уровнях обучения, разработанных Робертом Ганье [11]. Эти методологические положения

сохраняют свою актуальность вплоть до настоящего времени рассмотрим данные принципы:

1. Привлечение внимания. Использование мультимедийных возможностей на первом этапе занятия позволит привлечь внимание обучающихся и повысить их мотивацию.

2. Определение целей обучения. Этот подход предполагает определение целей и задач обучения, описание навыков, которые приобретут ученики, а также примеры применения полученных знаний в будущем.

3. Обращение к знаниям, которые уже есть у учеников. Этот принцип акцентирует внимание на применении имеющихся у учеников знаний и навыков, относящихся к изучаемой теме.

4. Представление изучаемого материала. Для знакомства с изучаемым материалом можно использовать картинки, анимацию, видеоматериалы и другие мультимедийные средства.

5. Руководство обучением. Этот подход предусматривает, что ученики должны получать советы и руководства, необходимые для успешного обучения.

6. Проверка новых знаний на практике. Согласно этому принципу, новые знания следует применять на практике, опираясь на уже имеющиеся знания и навыки.

7. Обратная связь. Во взаимодействии учителя и учеников реализуется объективно–субъективная функция. Учитель анализирует учебное поведение школьников, корректируя их ответы и направляя процесс обучения.

8. Оценка выполнения. Данный подход отслеживает общую информацию об успехах учеников в выполнении заданий.

9. Сохранение и перенос полученных умений. Обеспечиваются условия для выполнения практических заданий, в которых применяются как новые, так и ранее полученные знания.

Педагогический дизайн в современном контексте понимается как процесс разработки успешной образовательной среды и формирования эффективного информационно–образовательного пространства.

Разработка образовательного сайта требует использования различных сервисов и инструментов, которые помогут создать, организовать и управлять образовательным контентом. Выбор инструментов и сервисов для разработки образовательного сайта зависит от конкретных задач и целей образовательного процесса. Важно учитывать не только функциональность, но и удобство использования, доступность и возможность интеграции с другими системами. Для подбора эффективных инструментов необходимых для реализации практикума «Основы программирования в среде Scratch» был проведён анализ следующих платформ: Google Sites, Tilda, Wordpress, Craftum (Таблица 7).

Таблица 7 – Анализ образовательных платформ

Название платформы	Описание	Основные преимущества платформ	Основные недостатки платформ
Tilda	Платформа для разработки адаптивных веб–сайтов с акцентом на визуальное оформление	Интуитивно понятный интерфейс, обширная библиотека шаблонов и функциональных блоков	Платные функции, ограниченная функциональность в бесплатной версии
Google Sites	Платформа для создания простых веб–страниц с глубокой интеграцией экосистемы Google	Бесплатное использование, полная совместимость с сервисами Google	Нестабильная работа на территории РФ, риск блокировки аккаунтов, ограниченный доступ к некоторым функциям
Wordpress	Платформа управления контентом для создания сайтов и блогов с поддержкой тем	Высокая степень гибкости и масштабируемости, широкий выбор тем и плагинов	Ограниченный функционал в бесплатной версии, требующий дополнительных плагинов для расширения возможностей
Craftum	Платформа для построения веб–сайтов с использованием готовых шаблонов	Простота освоения, наличие разнообразных шаблонов	Ограниченный функционал в бесплатной версии, низкая интеграция со сторонними сервисами

При анализе платформ для создания цифрового практикума была выбрана платформа Tilda.

Учитывая принципы педагогического дизайна, анализа платформ, был разработан внешний вид образовательного сайта повторяющий структуру и содержание практикума «Основы программирования в среде Scratch».

Перед тем как разработать сайт для реализации практикума была составлена модель для планирования и структуры разрабатываемого продукта. На рисунке 4 – представлена модель цифрового практикума «Основы программирования в среде Scratch»

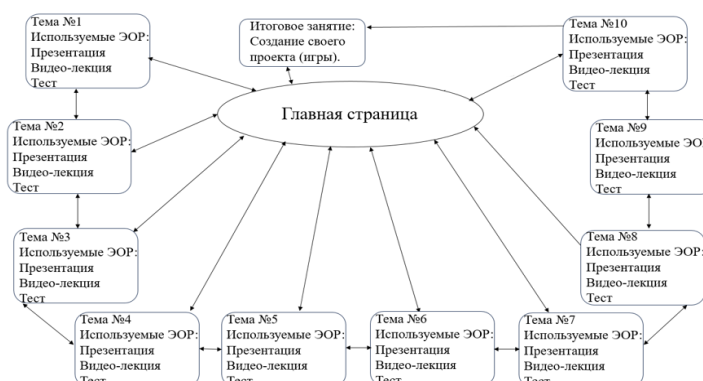


Рисунок 4 – Модель цифрового практикума «Основы программирования в среде Scratch»

Опишем модель цифрового практикума «Основы программирования в среде Scratch». При составлении модели можно заранее протестировать различные концепции и идеи, прежде чем приступить к полноценной разработке сайта. Выявить потенциальные проблемы в навигации и взаимодействии, так же устранить недостатки в структуре и дизайне с сайтом, что снижает вероятность изменений на этапе разработки, тем самым улучшая конечный результат. При разработке цифрового практикума были составлены видео–лекции и презентации к каждой теме практикума «Основы программирования в среде Scratch».

Видео–лекции созданы при помощи программы Bandicam – это программа для записи экрана, которая позволяет пользователям захватывать видео и аудио с компьютера. Она часто используется для создания видеозаписей игр, учебных материалов, презентаций и других видов контента. Вот некоторые ключевые функции Bandicam: запись экрана, запись звука, настройка качества. Видео выложены на платформе Rutube. Презентации составлены и подготовлены в

программе Microsoft PowerPoint, он является одним из самых популярных инструментов для создания презентаций благодаря своей функциональности и удобству использования. Всего создано 10 видео–лекций. Презентаций сделано в количестве 11 штук. Комплекс разработанных ЭОР по темам практикума представлены в таблице 8.

Таблица 8 – Электронные образовательные ресурсы в практикуме «Основы программирования в среде Scratch»

Электронные образовательные ресурсы в практикуме «Основы программирования в среде Scratch»	
Название тем и кол–во часов	Оснащённость ЭОР
1.Знакомство со средой программирования Scratch (1 час)	Презентация. Тест. Видео–лекция по теме «Знакомство со средой программирования Scratch».
2. Создание и редактирование объектов (1 час)	Презентация. Тест. Видео–лекция по теме «Создание и редактирование объектов».
3. Направление и движение спрайта (1 час)	Презентация. Тест. Видео–лекция по теме «Направление и движения спрайта».
4. Взаимодействие между объектами (1 час)	Презентация. Тест. Видео–лекция по теме «Взаимодействия объектов».
5.Линейный алгоритм (1 час)	Презентация. Тест. Видео–лекция теме «Линейный алгоритм».
6.Циклический алгоритм (1 час)	Презентация. Тест. Видео–лекция теме «Циклический алгоритм».
7. Разветвляющийся алгоритм (1 час)	Презентация. Тест. Видео–лекция по теме «Разветвляющийся алгоритм».
8. Счетчик (1 час)	Презентация. Тест. Видео–лекция теме «Счетчик».
9.Создание анимационного проекта (1 час)	Презентация. Тест. Видео–лекция теме «Создание анимационного проекта». Просмотр видео–лекции. Выполнение практического задания.
10.Игры, создаваемые в среде Scratch (1 час)	Презентация. Тест. Видео–лекция «Игры создаваемые в среде Scratch».
11.Итоговое занятие: Создание собственного проекта игры (1 час)	Презентация.

Для того, чтобы обучающиеся закрепили полученные знания после изучения темы, были разработаны тесты по каждой теме. Тесты составлены с помощью платформы «Яндекс формы» [34]. Всего в тесте 5 вопросов и 4 выбора ответа. На рисунке 5 представлена главная страница сайта.

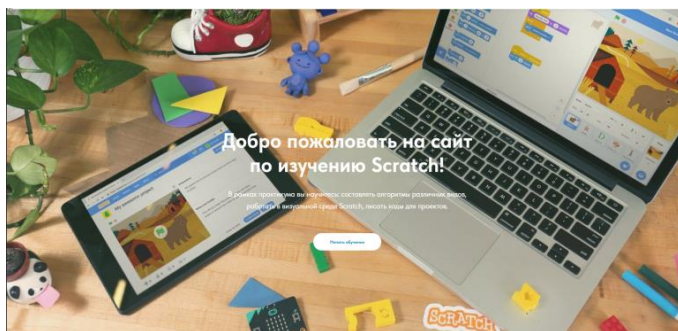


Рисунок 5 – Главная страница сайта

Данный образовательный сайт, помогает определить и организовать основные разделы и страницы практикума, что способствует логичному и интуитивно понятному навигационному перемещению для работы с ним. Указана цель и содержания занятия для каждого занятия. На модели сайта показано расположение электронных образовательных ресурсов и методических материалов. Задания расположены в нужной последовательности для изучения каждой темы практикума.

Сайт с практикумом опубликован на страницах контентного сайта, расположенного по интернет адресу: learn-scratch.ru [29].

2.3 Методические рекомендации для учителей по использованию практикума «Основы программирования в среде Scratch» в учебном процессе

Применение цифровых образовательных ресурсов предоставляет учителю возможность провести урок с использованием различных средств обучения. Это способствует повышению интереса учащихся к учебному материалу и позволяет реализовать разнообразные методы обучения. Разработанный практикум может быть использован учителями информатики при реализации программы внеурочной деятельности «Основы программирования» в 5–классах как в очном формате, так и в дистанционном.

При использовании учителем сайта с материалами практикума в учебном процессе рекомендуется:

- изучить материалы, представленные на сайте, учитывать возможности при изучении той или иной темы;
- организовать контроль за выполнением повторения изученного материала с помощью цифрового образовательного сайта;
- оценивать степень усвоения материала по практикуму «Основы программирования в среде Scratch», делать разбор заданий, вызывающих затруднения.

Электронные материалы, предназначенные для уроков, предоставляют учителю доступ к информационным ресурсам и обеспечивают визуализацию с использованием информационно–коммуникационных технологий (ИКТ). Разработанные материалы для сайта «Основы программирования в среде Scratch», подготовлены к каждому уроку. Материалы позволяют сочетать в себе элементы изучения темы, учитывая цели урока и образовательные потребности учеников. Разработанные видео–лекции включают в себя теоретическую базу для работы с заданиями, которые направлены на закрепление знаний и формирование умений. Разработанный образовательный сайт дает возможность учителю проводить занятия в очном и в дистанционном формате. Задания можно выполнять в классе и дома.

При изучении темы №1 «Знакомство со средой программирования Scratch», обучающиеся должны воспользоваться презентацией по теме «Знакомство со средой программирования Scratch» и посмотреть видео–лекцию по данной теме. Ученики должны выполнить практическое задание. После того как ученики выполнили практическую работу нужно сохранить файл в формате sb3 и прислать или показать работу учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №2 «Создание и редактирование объектов», обучающим нужно воспользоваться презентацией по теме «Создание и редактирование объектов» и посмотреть видео–лекцию по этой теме.

Обучающиеся должны выполнить практическое задание. После выполнения работы обучающиеся должны сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №3 «Направление и движение спрайта», ученикам нужно воспользоваться презентацией по теме «Направление и движение спрайта» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы обучающиеся должны сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №4 «Взаимодействие между объектами», ученикам нужно воспользоваться презентацией по теме «Взаимодействие между объектами», посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы обучающиеся должны сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №5 «Линейный алгоритм», ученикам нужно воспользоваться презентацией по теме «Линейный алгоритм» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №6 «Циклический алгоритм», ученикам нужно воспользоваться презентацией по теме «Циклический алгоритм» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №7 «Разветвляющийся алгоритм», ученикам нужно воспользоваться презентацией по теме «Разветвляющийся алгоритм» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №8 «Счетчик», ученикам нужно воспользоваться презентацией по теме «Счетчик» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №9 «Создание анимационного проекта», ученикам нужно воспользоваться презентацией по теме «Создание анимационного проекта» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

При изучении темы №10 «Игры, создаваемые в среде Scratch», ученикам нужно воспользоваться презентацией по теме «Игры, создаваемые в среде Scratch» и посмотреть видео–лекцию. Обучающиеся должны выполнить практическое задание. После выполнения работы сохранить файл в формате sb3 и прислать учителю. Для закрепления полученных знаний ученикам нужно пройти тест по пройденной теме.

Цель одиннадцатого занятия: создавать собственную игру в Scratch, используя изученные концепции и приемы. Ученики должны определить планирование и дизайн игры перед началом написания проекта, создать собственный сценарий для создания своего проекта (игры). Выбрать жанр игры:

ролевая игра, стратегия, квесты, головоломка т.д. После выполнения работы сохранить файл в формате sb3 работу прислать или показать учителю.

Выводы по главе 2

Во второй главе представлена разработка цифрового практикума «Основы программирования в среде Scratch», направленного на развитие алгоритмического мышления у учащихся 5 классов.

Практикум представляет собой сайт, наполненный электронными образовательными ресурсами, сочетающий теорию и практику, и поддерживает реализацию программы внеурочной деятельности как в очном, так и в дистанционном форматах. Его внедрение в учебный процесс позволит повысить мотивацию учащихся и подготовить их к изучению более сложных языков программирования в 7–9 классах.

Для того, чтобы убедиться, что практикум «Основы программирования в среде Scratch» развивает алгоритмическое мышление у детей среднего возраста, будет проведено исследование на базе школы ООЦ г. Тольятти [30]. Для оценки эффективности практикума запланировано исследование с использованием методик Бориса Леонидовича Покровского и Анатолия Залмановича Зака, направленных на диагностику уровня алгоритмического мышления до и после прохождения практикума.

Глава 3 Опытнo–экспериментальная работа по развитию алгоритмического мышления детей среднего школьного возраста во внеурочной деятельности посредством реализации программы «Основы программирования»

3.1 Цель и задачи педагогического эксперимента по развитию алгоритмического мышления у обучаемых 5–х классов

Для определения эффективности практикума «Основы программирования в среде Scratch» проведен педагогический эксперимент.

Цель педагогического эксперимента: выявление уровня алгоритмического мышления детей среднего школьного возраста до и после проведения занятий внеурочной деятельности по информатике на основе программы «Основы программирования» и разработанного для этой программы практикума «Основы программирования в среде Scratch».

Задачи педагогического эксперимента:

- 1) подобрать диагностические методики в соответствии с возрастными особенностями обучающихся;
- 2) выбрать базу экспериментальной работы и экспериментальную и контрольную группы среди обучаемых 5–х классов;
- 3) провести диагностику развития уровня алгоритмического мышления в экспериментальной и контрольной группах (констатирующий этап работы);
- 4) провести внеурочные занятия программы «Основы программирования» с использованием практикума «Основы программирования в среде Scratch» в экспериментальной группе (формирующий этап педагогического эксперимента);
- 5) провести диагностику развития уровня алгоритмического мышления в экспериментальной и контрольной группах (контрольный этап работы);
- 6) обработать результаты диагностик, сделать выводы.

Для педагогического эксперимента выбраны методики «Установление закономерностей» (Б.Л. Покровский) [19]. и «Игра в обмен» (А.З. Зак) [14].

Исследование проходило на базе ЧОУ СОШ «Общеобразовательный центр «Школа» г. Тольятти (Таблица 9).

Таблица 9 – Информация образовательной организации ЧОУ СОШ «Общеобразовательный центр «Школа» г. Тольятти

Образовательная организация	Частное общеобразовательное учреждение средняя общеобразовательная школа «Общеобразовательный центр «Школа».
Дата создания	5 августа 1992 года.
Учредитель образовательной организации	Сидорова Светлана Ивановна..
Юридический адрес	445028, Самарская область, г.о. Тольятти, б–р Королева, 22
Фактический адрес	445057, Самарская область, г.о. Тольятти, ул. Юбилейная, 77.
Режим/график работы	С 8.00 до 18.00. Перерыв на обед с 12.00 до 12.30. Выходные дни: воскресенье, праздничные дни, установленные законодательством РФ.
Приемная директора	Телефон: (8482) 34– 00– 21
Адреса электронной почты	оос–shkola@mail.ru (общеобразовательная школа), оос–ad@mail.ru (структурное подразделение).
Адрес официального сайта школы	http://оос– school.ru

В исследовании приняли участие 3 класса: 5 «А», 5 «Б», 5 «В». Каждый класс поделен на 2 группы. Группа 1 была взята в качестве экспериментальной группы. Группа 2 была взята в качестве контрольной группы. Сведения о респондентах:

- Учащиеся группы 5 «А» класса поделены на 2 группы: группа 1 состоит из 4 мальчиков и 3 девочек; группа 2 состоит из 5 мальчиков и 4 девочек.
- Учащиеся группы 5 «Б» класса поделены на 2 группы: группа 1 состоит из 2 мальчиков и 6 девочек; группа 2 состоит из 2 мальчиков и 6 девочек.
- Учащиеся группы 5 «В» класса поделены на 2 группы: группа 1 состоит из 3 мальчиков и 2 девочек; группа 2 состоит из 2 мальчиков и 4 девочек.

В экспериментальной группе – 20 человек, в контрольной группе – 23 человека. Всего задействовано 43 обучающихся. Возраст детей составляет от 11 до 12 лет. Возрастной состав представлен следующим образом: 7 детей 2012 года рождения, 36 детей 2013 года рождения. Участие приняли обучающиеся с наиболее стабильной посещаемостью образовательного учреждения. Исследование проводилось в специальном отведенном помещении при помощи выбранных методик. В качестве диагностических методик использованы:

- 1) Методика «Установление закономерностей» Борис Леонидович Покровский;
- 2) Методика «Игра в обмен» Анатолий Залманович Зак.

Результаты полученные в ходе исследования обработаны и проанализированы. Разберём применяемые методики, используемые в педагогическом эксперименте:

Методика №1. Для выявления уровня алгоритмического мышления детей среднего школьного возраста использовалось тестирование по методике «Установление закономерностей» Борис Леонидович Покровский. Методика служит для оценки качеств мышления – способности к анализу и сравнению, умению делать логические построения, легкости возникновения ассоциативных связей, установлении тождества и различия, скорости переключения с одного способа умственного действия на другой, а также для характеристики устойчивости внимания и оперативной памяти.

Цель: выявления уровня алгоритмического мышления развития детей среднего школьного возраста.

Необходимо найти и подчеркнуть слово в строке, которое соответствует закономерности, зашифрованной в наборе знаков перед строкой. Каждая строка содержит 5 слов. Время, отведенное на выполнение задания: 8 минут. Ошибки можно исправлять, если они замечены вовремя. Успешность выполнения задания оценивается по коэффициенту А (Таблица 10).

Таблица 10 – Успешность выполнения задания по коэффициенту А

А	более 100	85 – 100	70 – 84	55 – 69	менее 55
	Очень высокий уровень;	Высокий уровень	Средний уровень	Низкий уровень	Очень низкий

Стимульный материал для тестирования приведен в приложение Б.

Методика №2. Для выявления уровня алгоритмического мышления детей среднего школьного возраста использовалось тестирование по методике «Игра в обмен» Анатолий Залманович Зак. Методика служит для диагностики интеллектуальной готовности к средней школе с помощью решения пространственно–комбинаторных задач. Задачи этого рода представляют собой такие ситуации, когда одно расположение объектов (букв, цифр, знаков) необходимо преобразовать в другое их расположение за требуемое число действий. При этом, в зависимости от используемых правил преобразования исходного расположения в требуемое – взаимообмен объектов местами, перестановки их через занятые места, передвижение их на свободные места.

Цель методики: выявления уровня алгоритмического мышления развития детей среднего школьного возраста.

Методика «Игра в обмен» состоит из 2–х модификаций. Методика основана на решении задач, связанных с преобразованием начального расположения объектов, например, (букв, цифр, знаков) в конечное требуемое расположение, но с одним строгим правилом «взаимообмена мест». За один ход можно поменять местами только два символа, остальные остаются на своих местах. На основе этого правила были разработаны четыре модификации данной методики с тем, чтобы была возможность выявить возможности осуществления аналитического способа теоретического мышления в разных конкретных условиях.

Модификация № 1 особенности этого варианта данной методики заключаются в том, что в начальном и конечном расположениях объектов

используются одни и те же знаки (буквы). Всего дано 8 задач. В задачах 1–4, нужно найти три обмена, имеют шесть вариантов решения. Задачи 5–8 (неизвестны четыре обмена), имеют восемь вариантов решения.

Уровень развития алгоритмического мышления оценивается по шкале от 1 до 8:

- Высокий уровень: 7–8 верных задач;
- Средний уровень: 4–6 верных задач;
- Низкий уровень: 1–3 верных задачи.

Бланк для тестирования приведен в приложение В.

Модификация № 2 особенности этого варианта методики «Игра в обмен» заключается в том, что в задачах (1–3) в начальном и конечном расположениях объектов используются цифры, в задачах (4–6) цифры и буквы, а в задачах (7–10) только буквы. Задачи данной модификации, так же, как и в модификации № 1, имеют много конкретных вариантов решения. Так, в задачах 1, 2 и 3 нужно иметь в виду, что в каждом действии меняются только соседние цифры. В задачах 4, 5 и 6 в каждом действии меняются местами только цифры, стоящие на одной вертикали (друг над другом). В задачах 7 и 8 меняются местами буквы, расположенные друг от друга через два места. В задачах 9 и 10 меняются местами буквы, расположенные по краям и в середине.

Уровень развития алгоритмического мышления оценивается по шкале от 1 до 10:

- Высокий уровень: 9–10 верных задач;
- Средний уровень: 5–8 верных задач;
- Низкий уровень: 1–4 верных задачи.

Бланк для тестирования приведен в приложение Г.

3.2 Констатирующий и формирующий этапы педагогического эксперимента

Методика №1 «Установление закономерностей» Борис Леонидович Покровский. На констатирующем этапе было проведено диагностическое исследование уровня развития алгоритмического мышления в экспериментальной и контрольной группах по методике №1 «Установление закономерностей» Борис Леонидович Покровский. Результаты прохождения методики представлены в приложение Д. На рисунке 6 приведены результаты выполнения методики в графическом виде, экспериментальной и контрольной группы.

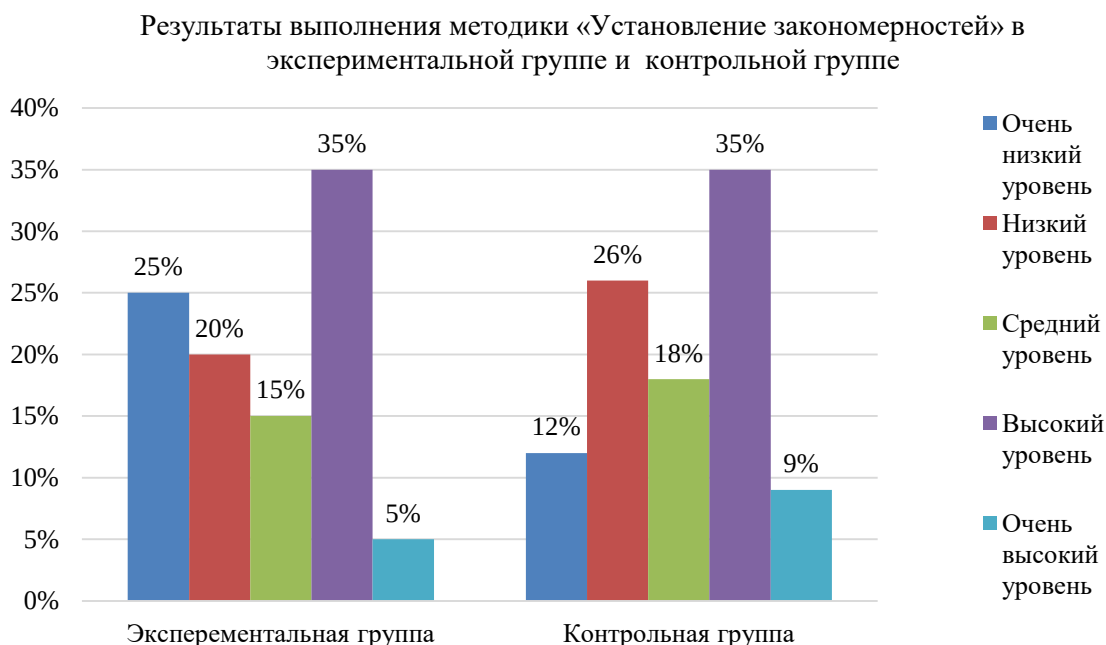


Рисунок 6 – Результаты выполнения методики №1 «Установление закономерностей» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что в экспериментальной группе:

- очень низкий уровень алгоритмического мышления у 25% учеников;
- низкий уровень алгоритмического мышления у 20% учеников;
- средний уровень алгоритмического мышления у 15% учеников;

- высокий уровень алгоритмического мышления у 35% учеников;
 - очень высокий уровень алгоритмического мышления у 5 % учеников;
- в контрольной группе:
- очень низкий уровень алгоритмического мышления у 12% учеников;
 - низкий уровень алгоритмического мышления у 26% учеников;
 - средний уровень алгоритмического мышления у 18% учеников;
 - высокий уровень алгоритмического мышления у 35% учеников;
 - очень высокий уровень алгоритмического мышления у 9 % учеников.

По результатам прохождения первой методики большинство учеников имеют высокий уровень алгоритмического мышления. Однако в экспериментальной и контрольной группе мало учащихся с очень высоким уровнем развития алгоритмического мышления.

Методика №2 (модификация №1). «Игра в обмен» Анатолий Залманович Зак. На констатирующем этапе было проведено диагностическое исследование уровня развития алгоритмического мышления в экспериментальной и контрольной группах по методике №2 (модификация №1). «Игра в обмен» Анатолий Залманович Зак. Результаты прохождения методики представлены в приложение Е. На рисунке 7 приведены результаты выполнения методики в графическом виде, экспериментальной и контрольной группы.



Рисунок 7 – Результаты выполнения методики №2 (модификация №1) «Игра в обмен» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что в экспериментальной группе:

- низкий уровень алгоритмического мышления у 35% учеников;
- средний уровень алгоритмического мышления у 45% учеников;
- высокий уровень алгоритмического мышления у 20% учеников;

в контрольной группе:

- низкий уровень алгоритмического мышления у 21% учеников;
- средний уровень алгоритмического мышления у 53% учеников;
- высокий уровень алгоритмического мышления у 26% учеников.

По результатам прохождения второй методики большинство учеников имеют средний уровень алгоритмического мышления. Также стоит отметить, что в экспериментальной группе уровень алгоритмического мышления ниже чем в контрольной группе.

Методика №2 (модификация №2). «Игра в обмен» Анатолий Залманович Зак. Результаты прохождения методики представлены в приложение Ж.



Рисунок 8 – Результаты выполнения методики №2 (модификация №2)
«Игра в обмен» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что
в экспериментальной группе:

- низкий уровень алгоритмического мышления у 55% учеников;
- средний уровень алгоритмического мышления у 45% учеников;
- высокий уровень алгоритмического мышления у 0% учеников;

в контрольной группе:

- низкий уровень алгоритмического мышления у 24% учеников;
- средний уровень алгоритмического мышления у 76% учеников;
- высокий уровень алгоритмического мышления у 0% учеников.

По результатам третьего эксперимента большинство учеников имеют
средний уровень алгоритмического мышления.

В связи с тем, что в экспериментальной группе наблюдается значительное
количество учащихся с низким уровнем алгоритмического мышления,
необходимо организовать работу по реализации практикума для программы

внеурочной деятельности по информатике «Основы программирования в среде Scratch».

Апробация практикума «Основы программирования в среде Scratch» проводилась проходило на базе ЧОУ СОШ «Общеобразовательный центр «Школа» г. Тольятти. В эксперименте принимали участие 2 группы: экспериментальная (группа 1) – учащиеся 5 «А», 5 «Б», 5 «В» класса и контрольная (группа 2) – учащиеся 5 «А», 5 «Б», 5 «В» класса.

Цель формирующего этапа эксперимента являются работа с экспериментальной группой направленная на развитие алгоритмического мышления в процессе реализации уроков во внеурочной деятельности по информатике в 5 классе.

Цель практикума: развитие алгоритмического мышления, у детей среднего школьного возраста.

Задачи практикума:

- Освоение интерфейса и базовых возможностей Scratch: Знакомство со средой (спрайты, фон, блоки кода). Работа с графическим редактором (создание и редактирование объектов). Понимание системы координат и управления движением спрайтов.

- Изучение основных алгоритмических структур: Линейные алгоритмы (последовательное выполнение команд). Циклические алгоритмы (повторяющиеся действия с помощью циклов). Разветвляющиеся алгоритмы (условные операторы если, иначе).

- Работа с переменными и взаимодействием объектов: Создание и использование переменных (счетчики, параметры игры). Организация взаимодействия между спрайтами (события, передача сообщений).

- Разработка анимации и интерактивных элементов: Создание анимационных проектов (движение, смена костюмов). Применение генерации случайных чисел для динамичных эффектов.

– Создание завершённого игрового проекта: Планирование игры (сюжет, механики, дизайн). Программирование игровой логики (управление, победа/проигрыш). Тестирование и доработка проекта.

Планируемые результаты:

- 1) Понимать основные виды алгоритмов (циклы, ветвления, переменные).
- 2) Создавать анимационные проекты.
- 3) Работать с графическими элементами и звуком.
- 4) Использовать визуальные блоки для написания программного кода.
- 5) Разрабатывать собственные компьютерные игры.
- 6) Представлять результаты своей работы в виде законченных цифровых продуктов.

Занятия проводились один раз в неделю. Работа по развитию алгоритмического мышления проводилась в экспериментальной группе в течение месяца.

Разработанный комплекс занятий, описанный в главе 2 был направлен на развитие алгоритмического мышления, освоение базовых концепций программирования и создание интерактивных проектов. Апробация проводилась в рамках четырех модулей, включающих 11 тем, каждая из которых сочетала теоретическую и практическую части, а также контрольные вопросы для закрепления материала.

В процессе апробации применялись разнообразные методы обучения:

- Теоретическая подготовка: изучение основных понятий (спрайты, сцены, скрипты, блоки, переменные и т.д.) через объяснения и наглядные материалы, включая презентации.
- Практические задания: индивидуальные и групповые задания по созданию скриптов, анимаций, игр и других проектов в среде Scratch.
- Проектный метод: разработка собственных проектов.

- Систематическое повторение: каждое занятие начиналось с актуализации материала предыдущих уроков.

- Наглядный метод: использование интерактивных презентаций, демонстраций интерфейса Scratch и примеров готовых проектов.

- Контроль знаний: терминологические диктанты и контрольные вопросы для оценки усвоения материала.

Обучение было организовано в визуальной среде программирования Scratch, которая позволила учащимся на практике освоить основы алгоритмизации, циклы, условные операторы, переменные и событийное программирование.

В самом начале апробации некоторые дети путались в последовательности блоков, возникали сложности с пониманием координатной системы, последовательности блоков и условных операторов, но наглядные примеры и своевременная помощь педагога быстро устранили затруднения. Задания по созданию простых проектов вызвали интерес благодаря возможности выбирать персонажей и наблюдать их движение. Занятия проходили в доброжелательной и мотивирующей обстановке. Дети с энтузиазмом включались в работу, активно делились идеями, помогали друг другу и с удовольствием создавали проекты.

Использование презентаций и тестов, способствовали глубокому усвоению знаний. Занятия проходили интересно, с высоким уровнем вовлеченности и положительными результатами. Дети не только освоили основы программирования, но и развили алгоритмическое мышление, навыки сотрудничества. Успех апробации подтверждает эффективность комплекса занятий и его пригодность для дальнейшего использования.

К концу обучения все учащиеся успешно прошли обучение, и представили собственные проекты по последнему занятию, продемонстрировав уверенное владение Scratch.

3.3 Контрольный этап педагогического эксперимента

Проведение диагностики развития уровня алгоритмического мышления в экспериментальной и контрольной группах контрольный этап работы проводился по тем же методикам, которые были использованы на констатирующем этапе эксперимента.

Цель: Выявить динамику уровня развития алгоритмического мышления детей среднего школьного возраста.

Методика №1 «Установление закономерностей» Борис Леонидович Покровский. На контрольном этапе было проведено диагностическое исследование уровня развития алгоритмического мышления в экспериментальной и контрольной группах по методике №1 «Установление закономерностей» Борис Леонидович Покровский. Результаты прохождения методики представлены в приложение И. На рисунке 9 приведены результаты выполнения методики в графическом виде, экспериментальной и контрольной группы.

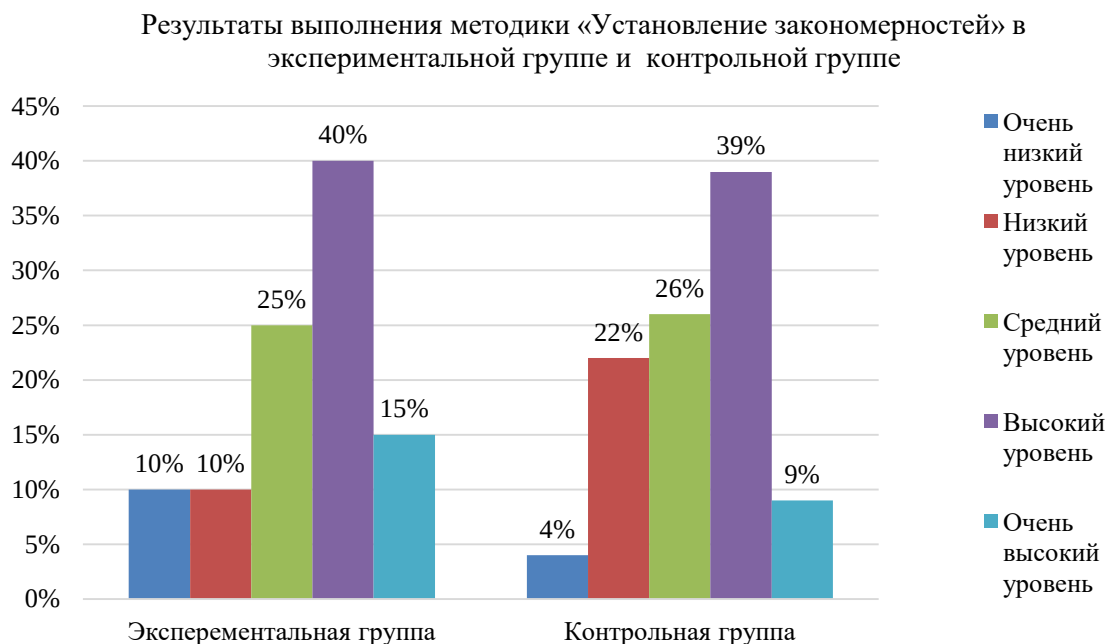


Рисунок 9 – Результаты выполнения методики №1 «Установление закономерностей» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что в экспериментальной группе:

- очень низкий уровень алгоритмического мышления снизился на 15%;
- низкий уровень алгоритмического мышления снизился на 10%;
- средний уровень алгоритмического мышления повысился на 10%;
- высокий уровень алгоритмического мышления повысился на 5%;
- очень высокий уровень алгоритмического мышления повысился на 10%;

в контрольной группе:

- очень низкий уровень алгоритмического мышления снизился на 8%;
- низкий уровень алгоритмического мышления снизился на 4%;
- средний уровень алгоритмического мышления повысился на 8%;
- высокий уровень алгоритмического мышления повысился на 4%;
- очень высокий уровень алгоритмического мышления без изменений.

Методика №2 (модификация №1). «Игра в обмен» Анатолий Залманович Зак. На контрольном этапе было проведено диагностическое исследование уровня развития алгоритмического мышления в экспериментальной и контрольной группах по методике №2 (модификация №1). «Игра в обмен» Анатолий Залманович Зак. Результаты прохождения методики представлены в приложение К. На рисунке 10 приведены результаты выполнения методики в графическом виде, экспериментальной и контрольной группы.

Цель: выявить динамику уровня развития алгоритмического мышления детей среднего школьного возраста.



Рисунок 10 – Результаты выполнения методики №2 (модификация №1) «Игра в обмен» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что в экспериментальной группе:

- низкий уровень алгоритмического мышления снизился на 30%;
- средний уровень алгоритмического мышления повысился на 15%;
- высокий уровень алгоритмического мышления повысился на 15%;

в контрольной группе:

- низкий уровень алгоритмического мышления снизился на 8%;
- средний уровень алгоритмического мышления повысился на 4%;
- высокий уровень алгоритмического мышления повысился на 4%.

Методика №2 (модификация №2). «Игра в обмен» Анатолий Залманович Зак. Результаты прохождения методики представлены в приложение Л.

Цель: Выявить динамику уровня развития алгоритмического мышления детей среднего школьного возраста.



Рисунок 11 – Результаты выполнения методики №2 (модификация №2)
«Игра в обмен» в экспериментальной группе и контрольной группах

Подводя итоги педагогического эксперимента, следует отметить, что
в экспериментальной группе:

- низкий уровень алгоритмического мышления снизился на 20%;
- средний уровень алгоритмического мышления повысился на 5%;
- высокий уровень алгоритмического мышления повысился на 15%;

в контрольной группе:

- низкий уровень алгоритмического мышления повысился на 4%;
- средний уровень алгоритмического мышления снизился на 4%;
- высокий уровень алгоритмического мышления без изменений.

Подводя итоги на основе данных, полученных с помощью пройденных методик, можно заключить, что уровень развития алгоритмического мышления у учащихся экспериментальной группы значительно повысился по сравнению с их предыдущими результатами. Это свидетельствует о успешности формирующего этапа и подтверждает эффективность применения практикума «Основы программирования в среде Scratch».

Выводы по главе 3

Проведенный педагогический эксперимент по развитию алгоритмического мышления у учащихся 5–х классов с использованием практикума «Основы программирования в среде Scratch» подтвердил его эффективность. На констатирующем этапе диагностика выявила, что большинство учащихся как в экспериментальной, так и в контрольной группах имели средний и низкий уровни алгоритмического мышления. При этом в экспериментальной группе наблюдалась более выраженная потребность в развитии данного вида мышления. Формирующий этап, включавший систематические занятия по программированию в Scratch, способствовал значительному прогрессу у учащихся экспериментальной группы. Результаты контрольного этапа показали: снижение доли учащихся с низким уровнем алгоритмического мышления; рост числа учащихся со средним, высоким уровнем и очень высоким уровнем. В контрольной группе, где занятия по практикуму «Основы программирования в среде Scratch» не проводились, динамика оказалась незначительной или отсутствовала.

Таким образом, практикум «Основы программирования в среде Scratch» доказал свою эффективность в развитии алгоритмического мышления у учащихся среднего школьного возраста.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута цель, а именно, разработан и апробирован практикум «Основы программирования в среде Scratch» для занятий внеурочной деятельности в 5 классе, направленный на развитие алгоритмического мышления учащихся. Гипотеза исследования подтвердилась: систематическое использование разработанного практикума на занятиях по внеурочной деятельности способствует развитию алгоритмического мышления у учащихся 5–х классов.

В первой главе проведён теоретический анализ нормативных, методических и психолого–педагогических аспектов разработки методического сопровождения для программ внеурочной деятельности по информатике. Изучены требования Федеральных государственных образовательных стандартов (ФГОС) и примерной рабочей программы по информатике, подчеркивающие значимость алгоритмического мышления. Проанализированы работы А.П. Ершова, А.Г. Кушниренко, Л.Л. Босовой, Т.Н. Лебедевой, раскрывающие подходы к развитию алгоритмического мышления. Выполнен отбор содержания для практикума на основе нормативных требований, с учётом возрастных особенностей учащихся и возможностей среды Scratch как инструмента визуального программирования.

Во второй главе осуществлена практическая разработка цифрового практикума «Основы программирования в среде Scratch». Спроектирована структура практикума, включающая четыре модуля и 11 тематических занятий, охватывающих основы работы в Scratch. Краткое содержание тем практикума: управление движением и взаимодействием объектов, изучение линейных, циклических и разветвляющихся алгоритмов, а также создание игровых проектов. Разработан образовательный сайт на платформе Tilda для размещения

практикума, включающего теоретические материалы, практические задания и контрольные вопросы.

В третьей главе описан педагогический эксперимент, проведённый на базе ЧОУ СОШ «Общеобразовательный центр «Школа» г. Тольятти с участием 43 учащихся 5–х классов, направленный на проверку эффективности практикума. На констатирующем этапе определён исходный уровень алгоритмического мышления учащихся, который в большинстве случаев был низким или средним. На формирующем этапе экспериментальная группа занималась по разработанному практикуму, а контрольная по стандартной программе. На контрольном этапе проведена повторная диагностика, показавшая значительный рост уровня алгоритмического мышления в экспериментальной группе (снижение доли учащихся с низким уровнем и увеличение доли со средним и высоким уровнями) по сравнению с контрольной группой, где динамика была минимальной. Анализ результатов подтвердил эффективность практикума в развитии алгоритмического мышления обучаемых.

Разработанный практикум «Основы программирования в среде Scratch» доказал свою эффективность в развитии алгоритмического мышления учащихся среднего школьного возраста и может быть рекомендован для внедрения в образовательный процесс в рамках внеурочной деятельности по информатике в 5 классе. Материалы практикума, доступные на образовательном сайте, обеспечивают гибкость применения в различных форматах обучения и создают основу для дальнейшего углубления знаний в области программирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р 53620–2009 Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения: национальный стандарт Российской Федерации / Федеральное агентство по техническому регулированию и метрологии. – Москва: Стандартинформ, 2009. – URL: <https://docs.cntd.ru/document/1200082196> (дата обращения: 03.04.2025). – Текст: электронный.
2. Информатика. Углубленный уровень: реализация требований ФГОС ООО / Министерство просвещения Российской Федерации. – Москва: Edsoo, 2023. – 87 с. – URL: [https://edsoo.ru/wp-content/uploads/2023/08/ Информатика-углубленный-уровень-Реализация-требований-ФГОС-ООО.pdf](https://edsoo.ru/wp-content/uploads/2023/08/Информатика-углубленный-уровень-Реализация-требований-ФГОС-ООО.pdf) (дата обращения: 27.03.2025). – Текст: электронный.
3. Об утверждении федерального государственного образовательного стандарта основного общего образования: приказ Министерства просвещения Российской Федерации от 31 мая 2021 г. № 287 // Единое содержание общего образования. – URL: https://edsoo.ru/wp-content/uploads/2023/08/Приказ-№-287-от-31.05.2021-ФГОС_ООО.pdf (дата обращения: 22.12.2024). – Текст: электронный.
4. Об утверждении федерального государственного образовательного стандарта среднего общего образования: приказ Министерства просвещения Российской Федерации от 18 июля 2024 г. № 499 // Контур.Норматив:– URL: <https://normativ.kontur.ru/document?moduleId=1&documentId=476578> (дата обращения: 25.12.2025). – Текст: электронный.
5. Приказ Министерства просвещения Российской Федерации от 12 августа 2022 г. № 732 «Об утверждении федерального государственного образовательного стандарта среднего общего образования» — URL:

<https://normativ.kontur.ru/document?moduleId=1&documentId=431499> (дата обращения: 10.02.2025). — Текст: электронный.

6. Примерная рабочая программа внеурочной деятельности «Основы программирования для 5–6 классов» / Министерство просвещения Российской Федерации. — Москва: Edsoo, 2022. — 45 с. — URL: https://edsoo.ru/wp-content/uploads/2023/08/ВУД_ПРП-КВД-Основы-программирования-для-5-6-классов_Новая.pdf (дата обращения: 05.01.2025). — Текст: электронный.

7. Примерная рабочая программа основного общего образования по учебному предмету «Информатика». Базовый уровень (для 5–6 классов) / Министерство просвещения Российской Федерации. — Москва, 2022. — URL: https://fgosreestr.ru/uploads/files/b8e65c9ad4d29aef63_aa38c3e8533f0.pdf (дата обращения: 20.12.2024). — Текст: электронный.

8. Федеральная рабочая программа по учебному предмету «Информатика». 7–9 классы (базовый уровень): рабочая программа / Минпросвещения России. — Москва: Edsoo, 2023. — URL: https://edsoo.ru/wp-content/uploads/2023/08/15_ФРП-Информатика-7-9 классы_база.pdf (дата обращения: 20.12.2024). — Текст: электронный.

9. Асмолов, А. Г. Формирование универсальных учебных действий в основной школе: от действия к мысли. Система заданий: пособие для учителя / под ред. А. Г. Асмолова. — Москва: Просвещение, 2010. — 159 с.

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

11. Ганье, Р.М. Условия обучения / Р.М. Ганье. — 4-е изд.— Нью-Йорк: Holt, Rinehart and Winston, 1985. — 429 с.

12. Голиков, Д. В. Scratch для юных программистов / Д. В. Голиков. — Санкт-Петербург: БХВ-Петербург, 2017. — 192 с.

13. Ершов, А. П. Программирование— вторая грамотность / А. П. Ершов // Избранные труды. — Новосибирск: Наука, 1994. — С. 32–40. — 416 с.

14. Зак, А. З., Развитие и диагностика мышления подростков и старшеклассников // А. З. Зак. – Москва; Обнинск: ИГ–СОЦИН, 2010. – С. 194–207. – 350 с.
15. Информатика. 5–6 классы: практикум по программированию в среде Scratch / авт.– сост. Л. Л. Босова [и др.]; под ред. Л. Л. Босовой. – Москва: БИНОМ. Лаборатория знаний, 2019. – 128 с.– Режим доступа– – URL: [https://files.lbz.ru/pdf/978–5–9963–4189–4o.pdf] (дата обращения: 10.01.2025).
16. Кочурина, Т. С. «Педагогический дизайн»: сущность и структура [Текст] / Т. С. Кочурина // НИЦ «Вестник науки», 2022. – № 1(1). – С. 21–29.
17. Кушниренко, А. Г. Основы информатики и вычислительной техники: пробный учебник для средних учебных заведений / А. Г. Кушниренко, П. В. Лебедев, Р. А. Сворень. – Москва: Просвещение, 1990. – 224 с.
18. Лебедева, Т. Н. Формирование алгоритмического мышления школьников в процессе обучения рекурсивным алгоритмам в профильных классах средней общеобразовательной школы: диссертация кандидата педагогических наук: Лебедева Татьяна Николаевна. – Екатеринбург, 2005. – 215 с.
19. Логутова, Е. В. Диагностика познавательного развития: учебное пособие; Оренбургский государственный университет. – Оренбург: ОГУ, 2021. – С. 97–100. – 142 с.
20. Маржи, М. Scratch для детей. Самоучитель по программированию / М. Маржи; пер. с англ. М. Гескиной, С. Таскаевой. – Москва: Манн, Иванов и Фербер, 2017. – 288 с.
21. Панюкова, С. В. Психолого–педагогические аспекты развития мышления при реализации проблемного обучения информатике / С. В. Панюкова, О. А. Прусакова // Рязанский государственный радиотехнический университет. 2023.– № 1. – С. [29–32]. – Режим доступа – URL: [https://cyberleninka.ru/article/n/psihologo–pedagogicheskie–aspekty–razvitiya–

myshleniya-pri-realizatsii-problemnogo-obucheniya-informatike/viewer] (дата обращения: 25.03.2025). – Текст: электронный.

22. Пашковская Ю. В. Программирование на Scratch для детей. Уровень 1, 2024. – Режим доступа–URL: [<https://www.litres.ru/book/u-v-pashkovskaya-89459/programmirovanie-na-scratch-3-0-dlya-detey-uroven-1-70502377/>] (дата обращения: 12.02.2025).

23. Пиаже, Ж. Речь и мышление ребенка / Ж. Пиаже; [пер. с фр.]. – Санкт–Петербург: Союз, 1997. – 256 с

24. Программирование для детей / К. Вордерман, Дж. Вудкок, Ш. Макаманус [и др.]; пер. с англ. С. Ломакина. – Москва: Манн, Иванов и Фербер, 2015. – 224 с.

25. Стась, А. Н. Развитие алгоритмического мышления в процессе обучения будущих учителей информатики / А. Н. Стась, Н. Ф. Долганова // Информатика и образование. – 2012. – № 7. – С. 241–244. – Режим доступа–URL: <https://cyberleninka.ru/article/n/razvitie-algoritmicheskogo-myshleniya-v-protssesse-obucheniya-buduschih-uchiteley-informatiki/viewer> (дата обращения: 30.03.2025). – Текст: электронный.

26. Фомина, О. Е. Медиакоммуникации и электронные образовательные ресурсы в современном образовании [Текст] / О. Е. Фомина // НИЦ «Вестник науки». – 2024. – № 12 (81). – С. 1145–1149.

27. CodeWards: платформа для обучения программированию. – Текст: электронный. – URL: <https://codewards.ru/> (дата обращения: 14.02.2025).

28. IT-Cube: образовательный центр. – Текст: электронный. – URL: <https://it-cube-0-2.vercel.app/> (дата обращения: 14.02.2025).

29. learn-Scratch.ru: сайт. – Текст: электронный. – URL: <https://learn-scratch.ru> (дата обращения: 15.02.2025).

30. ООЦ Школа: сайт. – Текст: электронный. – URL: <https://ooc-school.ru/> (дата обращения: 20.02.2025).

31. Robbo Scratch: сайт. – Текст: электронный. – URL: <https://scratch.robbo.ru/> (дата обращения: 18.02.2025).

32. Scratch: (официальный сайт визуального языка программирования) / разраб. MIT Media Lab; ред. М. Резник. – Режим доступа– URL: <https://scratch.mit.edu/> (дата обращения: 05.01.2025). – Текст: электронный.

33. Образовательная платформа «Учи.ру» – Режим доступа– URL: <https://uchi.ru/> (дата обращения: 15.02.2025).

34. Яндекс.Формы: Обзор возможностей // Справка Яндекс. – Режим доступа: <https://yandex.ru/support/forms/ru/overview> (дата обращения: 13.03.2025).

35. Резник, М. Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play/ М. Резник. – Кембридж, Массачусетс: The MIT Press, 2017. – 208 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Модуль 1. «Основы работы в Scratch»

Тема 1 «Знакомство со средой программирования Scratch»

Цель занятия – Ознакомиться с возможностями среды программирования Scratch.

Методические рекомендации по изучению темы «Знакомство со средой программирования Scratch»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Scratch – это визуальная среда программирования, разработанная для обучения детей и начинающих основам алгоритмизации и создания интерактивных проектов. Разработан командой MIT Media Lab в 2007 году.

В Scratch используется блочное программирование (перетаскивание блоков вместо написания кода). В Scratch можно создавать такие проекты, как анимации, игры, алгоритмы для изучения основ программирования.

Версии Scratch, созданные разработчиками, включают: Scratch (офлайн-редактор), Scratch Online (редактор в браузере), Scratch Junior (упрощённая версия Scratch), Scratch S4A для микроконтроллеров (например, взаимодействие с Lego Mindstorms EV3, Arduino).

Основные понятия Scratch:

Спрайт – объект (персонаж, предмет), который выполняет действия.

Сцена – фон, на котором происходит действие. Можно менять задний план. Добавлять спрайты, и сцена можно из библиотеки или рисовать самостоятельно.

Скрипт – набор блоков в Scratch, составляющий программу, которая управляет действием спрайтов или сцены. Он представляет собой последовательность команд, определяющих, что и как будет делать объект. С помощью скриптов можно запрограммировать действия спрайтов и сцены.

Область скриптов – это основное рабочее пространство в Scratch, где создаются программы для спрайтов и сцены. В этой области составляются скрипты из визуальных блоков, с помощью перетаскивая и соединения их между собой.

Костюмы – разные изображения спрайта (анимация, смена внешнего вида спрайта).

Блоки – визуальные элементы программирования в Scratch, из которых собираются скрипты (алгоритмы). Блоки можно перетаскивать, соединять между собой и создавать программы. В Scratch используются следующие блоки: «Движение», «Внешний вид», «Звук», «События», «Управления», «Сенсоры», «Операторы», «Переменные».

Звуки в Scratch позволяют добавлять аудиоэффекты и музыку к проектам.

Сценарий в Scratch – это последовательность блоков кода, которые определяют поведение спрайта или сцены, позволяя задавать логику проекта.

К исполнителям в Scratch относятся спрайты и среда (фон), с которыми работает пользователь.

Система команд исполнителя – это набор всех доступных блоков (команд), которые можно использовать для управления спрайтами и сценой в Scratch. Эти команды объединены в категории и позволяют программировать движение, внешний вид, звуки, и другие действия.

Окно исполнителя – область в интерфейсе Scratch, где отображается название спрайта, текущие координаты спрайта, угол поворота (направление движения), размер спрайта в процентах, показать/спрятать объект, текущий используемый костюм и фон, добавление спрайта, удаление спрайта.

Проект в Scratch – это результат работы в среде Scratch, который можно сохранить и опубликовать.

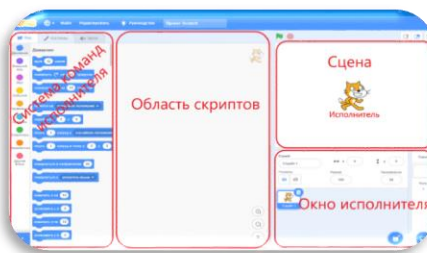


Рисунок – 12 Основные блоки интерфейса в Scratch

Практическая часть.

1. Подготовка проекта.

Запустите Scratch на компьютере. Оставьте «Спрайт 1» или выберете другого спрайта, для этого нажмите перейдите к окну исполнителя нажмите «Выбрать спрайт» выберите персонажа, например, «Dinosaur4». В разделе «Сцена» выберите «Выбрать фон», например, «Blue Sky 2».

2. Создание скрипта для движения спрайта.

Задача: выбранный спрайт должен ходить из стороны в сторону (влево–вправо). Во вкладке «Код» (СКИ) для спрайта добавьте блоки:

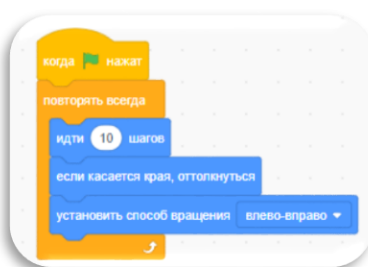


Рисунок – 13 Скрипт для движения спрайта

3. Разбор команд.

– «Когда флажок нажат» (блок события) – это стартовый блок, который запускает выполнение скрипта при нажатии на зелёный флаг.

«Повторять всегда» (блок управления) – создает бесконечный цикл команды внутри будут выполняться снова и снова без остановки. Чтобы остановить программу, нужно нажать красный кружок (стоп).

– «Идти (10) шагов» (блок движения) – заставляет спрайт двигаться в текущем направлении на 10 шагов.

– «Если касается края, оттолкнуться» (блок движения) – проверяет, достиг ли спрайт границы сцены. Если да – спрайт меняет направление на противоположное.

– «Установить направление влево–вправо» (блок движения) – определяет, куда будет поворачиваться спрайт, когда меняется его направление движения.

4. Практическое задание.

Добавьте в проект, спрайт (Crab) который двигается по сцене (Beach Rio). Используйте для этого скрипт для движения спрайта. Измените программу, чтобы спрайт (Crab) перемещался быстрее.

5. Сохранение проекта на компьютер.

Нажмите кнопку «Файл» в верхнем меню, выберите «Сохранить на свой компьютер». Выберите папку для сохранения проекта под именем «Проект 1». Файл сохранится в формате (.sb3).

6. Контрольные вопросы.

Таблица – 11 Контрольные вопросы по теме «Знакомство со средой программирования Scratch»

Вопросы	Варианты ответа
№ 1 Какой элемент в Scratch отвечает за фон, на котором происходят действия?	а) Спрайт <u>б) Сцена</u> в) Блок г) Костюм
№ 2 Как называется объект, который двигается в проекте Scratch?	а) Блок б) Сцена <u>в) Спрайт</u> г) Звук
№ 3 Какой блок в Scratch запускает скрипт при нажатии зелёного флага?	а) Когда клавиша (пробел) нажата <u>б) Когда флажок нажат</u> в) Всегда г) Повторить 10
№ 4 Что такое «Костюмы» в Scratch?	а) Разные фоны для сцены <u>б) Разные изображения спрайта (для смены внешнего вида)</u> в) Блоки управления движением г) Звуковые эффекты
№ 5 Какой блок заставляет спрайт двигаться на 10 шагов?	а) Повернуть на 15 градусов б) Изменить размер на 10 <u>в) Идти 10 шагов</u> г) Ждать 1 секунду

Тема 2 «Создание и редактирование объектов»

Цель занятия – научиться создавать и редактировать спрайты и фоны, использовать такие инструменты как: кисть, заливка, ластик, линия, круг, текст, прямоугольник, изменение формы.

Методические рекомендации по изучению темы «Создание и редактирование объектов»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Спрайты – это графические объекты, которые используются в проектах. В Scratch предусмотрено изменение существующих костюмов. Каждый спрайт может иметь несколько костюмов, которые могут меняться для создания анимации или визуальных эффектов. Костюмы можно добавлять и удалять. Редактирование спрайтов и костюмов включает в себя: рисование новых спрайтов с помощью инструментов графического редактора таких как: кисть, заливка, ластик, линия, круг, текст, прямоугольник, изменение формы.

В Scratch есть два режима работы с изображениями: векторный и растровый. Они отличаются способом представления изображения, и каждый имеет свои преимущества. В растровом режиме изображение состоит из пикселей (точек). Можно детально прорисовывать каждый пиксель. В векторном режиме изображение строится из геометрических форм (линий, кривых, фигур). Масштабирование графического объекта в это режиме идет без потери качества. Чтобы перейти в разные режимы графики нужно нажать «Конвертировать в (векторную, растровую) графику» в редакторе графических объектов. Команда «Показать» или «Спрятать» используется чтобы управлять видимостью спрайта. Управление порядком отображения объектов на сцене. В Scratch для работы с графикой объектов доступны специальный редактор: Для спрайтов редактор «Костюмы», который позволяет создавать и редактировать внешний вид персонажа. В этом редакторе объекты можно переместить на

передний, задний план. Содержит инструменты рисования. Для сцены редактор «Фоны», предназначен для оформления фона проекта, имеет аналогичные инструменты рисования.

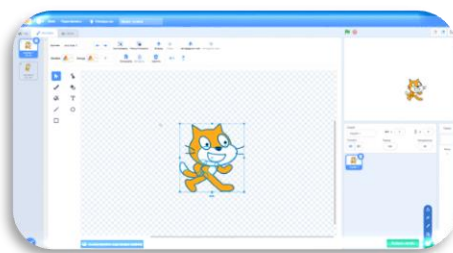


Рисунок – 14 Редактор графических объектов

Практическая часть.

1. Подготовка проекта.

Запустите Scratch на компьютере. Используя инструменты (кисть, заливка, ластик, линия, круг, прямоугольник, текст), создайте собственного спрайта с несколькими костюмами. В области окна исполнителя, нажмите «Выбрать спрайт» выберете «Нарисовать». создайте простой объект (например, мяч или персонажа) с несколькими костюмами. Нарисуйте солнце с помощью инструмента «Круг» (удерживайте Shift для идеальной окружности). Добавьте текст с помощью инструмента «Текст» во вкладке фон для объектов. Нарисуйте собственный фон, например, (голубое небо с облаками, зелёный луг, горы) или выберите подходящей фон из библиотеки («Blue Sky 2», «Bench With View» и т. д.). Возможны альтернативные варианты для подготовки проекта.

2. Редактирование спрайта и создание скрипта для движения.

Задача: создайте анимацию движения спрайта. Пример спрайт (Cat). Костюм первый (cat-a), костюм второй (cat-b). Во вкладке «Код» (СКИ) для спрайта добавьте блоки:



Рисунок – 15 Скрипт для движения спрайта с изменением костюма

3. Разбор новых команд.

– «Следующий костюм» (блок внешний вид) – переключает спрайт на следующий по порядку костюм. Если текущий костюм – последний, то после выполнения команды спрайт вернётся к первому костюму в списке.

– «Ждать () секунд» (блок управление) – приостанавливает выполнение скрипта на указанное количество секунд. После паузы скрипт продолжает работу.

4. Практическое задание.

Добавьте спрайт, (Giga Walking), который будет двигаться. Используйте для этого скрипт для движения спрайта. Измените программу, чтобы спрайт (Giga Walking) перемещался плавнее. Переключитесь между режимами («Векторный» и «Растровый»). В векторном режиме измените один костюм спрайта (Giga Walking) на растровый режим.

5. Сохранение проекта на компьютер.

Нажмите «Файл» в верхнем меню, выберите «Сохранить на свой компьютер». Выберите папку для сохранения проекта под именем «Проект 2». Файл сохранится в формате (.sb3). Для чтобы открыть сохраненный проект на компьютере в формате (.sb3), нужно нажать «Файл». После выберите «Загрузить с компьютера» выберете ваш файл в формате (.sb3).

6. Контрольные вопросы

Таблица – 12 Контрольные вопросы по теме «Создание и редактирование объектов»

Вопросы	Варианты ответа
№ 1 Какой инструмент используется для стирания частей изображения в графическом редакторе Scratch?	а) Кисть <u>б) Ластик</u> в) Заливка г) Выделение
№ 2 Что такое костюм в Scratch?	<u>а) Один из вариантов внешнего вида спрайта</u> б) Фон сцены в) Элемент интерфейса г) Инструмент для рисования
№ 3 Каким образом можно изменить фон сцены в Scratch?	<u>а) Через меню «Фоны»</u> б) Через блок кода в) Только вручную нарисовав новый фон г) Все вышеперечисленное

№ 4 Что происходит при использовании инструмента «заливка» в графическом редакторе Scratch?	а) Заполнение замкнутой области выбранным цветом б) Удаление выбранного цвета в) Создание нового слоя г) Прозрачное заполнение
№ 5 Как сохранить созданный в Scratch рисунок для последующего использования?	а) Через функцию экспорта в формате PNG б) Через команду «Сохранить как» в меню «Файл» в) Сохранить только как часть проекта г) Нельзя сохранить отдельно от проекта

Модуль 2. «Управление движением и взаимодействием»

Тема 3 «Направление и движение спрайта»

Цель занятия – научиться использовать команды поворота для указания направления и движения спрайта; понимать, как работает координатная система в Scratch.

Методические рекомендации по изучению темы «Направление и движение спрайта»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

В Scratch спрайты (персонажи или объекты) могут двигаться по сцене, используя команды движения и поворота. Сцена в Scratch – это двумерная плоскость с координатами X (горизонталь) и Y (вертикаль). Центр сцены имеет координаты (0, 0). Ось X: Положительные значения (вправо) от 0 до 240.

Отрицательные значения (влево) от 0 до – 240. Ось Y: Положительные значения (вверх) от 0 до 180. Отрицательные значения (вниз) от 0 до – 180.

Примеры координат: (0, 0) – центр, (100, 50) – справа вверх, (– 150, – 100) – слева вниз. Направление движения задаётся в градусах: 0 градусов – вверх; 90 градусов – вправо; 180 градусов – вниз; – 90 градусов – влево.

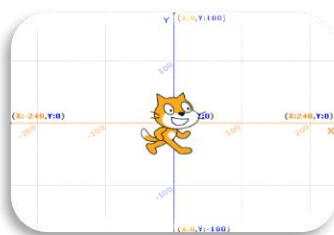


Рисунок 16 – Система координат в Scratch

Практическая часть.

1. Подготовка проекта.

Запустите Scratch на компьютере. Перейдите к окну исполнителя выберите спрайт (Cat) и фон (Ху– grid), который имеет 4 разных костюма. Выберите подходящий.

2. Создание скрипта для движения спрайта.

Задача: спрайт (Cat) должен двигаться по направлению движения (вверх–вниз) от выбранной сцены.

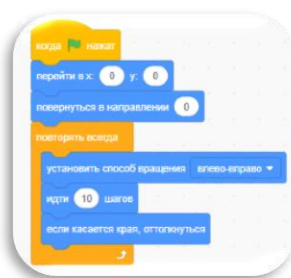


Рисунок – 17 Скрипт для направления спрайта

3. Разбор команд.

«Перейти в x: () y: ()» (блок движения) – выполняет мгновенное перемещение спрайта в указанные координаты на сцене.

«Повернуться в направлении 0» (блок движения) – 0 градусов обозначает направление вверх. 90 градусов – вправо, 180 градусов – вниз, – 90 градусов – влево.

«Плыть () секунд в точку x: () y:» (блок движения) – плавно перемещает спрайт к заданным координатам за указанное время. Создает эффект анимации движения.

«Повернуть на () градусов» (блок движения) – поворачивает спрайт по часовой стрелке на указанный угол. Не изменяет позицию, только направление.

«Повернуться к (указатель мыши)» – автоматически разворачивает спрайт в сторону указателя мыши или другого объекта. Угол рассчитывается динамически.

4. Практическое задание.

Создайте проект, выберите: спрайт (Dot), сцена (Xy– grid– 30px).

Создайте скрипт направления движения (влево–вправо) на основе предыдущего скрипта. После этого, измените программу, чтобы спрайт двигался по кругу.

5. Сохранение проекта на компьютер.

Нажмите «Файл» в верхнем меню, выберите «Сохранить на свой компьютер». Выберите папку для сохранения проекта под именем «Проект 3». Файл сохранится в формате (.sb3).

6. Контрольные вопросы.

Таблица – 12 Контрольные вопросы по теме «Направление и движение спрайта»

Вопросы	Варианты ответа
№ 1 Какое направление соответствует 180° в Scratch?	<u>а) Вниз</u> б) Влево в) Вправо г) Вверх
№ 2 Как изменится направление движения спрайта после выполнения команды поверни на (90) градусов, если изначально спрайт был направлен вправо?	<u>а) Спрайт будет двигаться вверх.</u> б) Спрайт будет двигаться вниз. в) Спрайт будет продолжать двигаться вправо. г) Спрайт будет двигаться влево.
№ 3 Каким образом можно заставить спрайт двигаться по кругу?	<u>а) Использовать бесконечный цикл с последовательными поворотами и движениями вперед.</u> б) Использовать команду круговое движение. в) Использовать команду вращай на () градусов. г)Использовать команду движение по траектории
№ 4 Как заставить спрайт следовать за указателем мыши?	а) Использовать команду следуй за мышью. <u>б) Использовать команду «повернуться к (указателю мыши)».</u> в) Использовать команду пойд к (). г)Использовать команду установи направление на ().
№ 5 Что означает термин «Спрайт» в контексте программирования ?	а) Статический фон. <u>б) Объект, который может перемещаться и взаимодействовать с другими объектами.</u> в) Звуковые эффекты. г) Набор инструментов для разработки игр.

Тема 4 «Взаимодействие между объектами»

Цель занятия – научиться использовать блоки кода для создания взаимодействий между спрайтами; познакомиться с концепцией событийного программирования.

Методические рекомендации по изучению темы «Взаимодействие между объектами»:

– Изучить теоретическую часть;

- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

В Scratch объекты называются спрайтами и фонами. Спрайты – это графические элементы, которые могут двигаться, изменять внешний вид, издавать звуки и взаимодействовать друг с другом. Фон – это статическое изображение, которое задает сцену для проекта. Спрайты могут быть персонажами, предметами или любыми другими элементами, которые участвуют в программе. Взаимодействие между объектами в Scratch – это процесс, при котором спрайты обмениваются информацией или реагируют друг на друга. Это может быть:

Физическое взаимодействие: например, столкновение двух спрайтов (персонаж касается препятствия).

Обмен сообщениями: один спрайт отправляет сигнал другому, чтобы запустить действие.

Изменение свойств: один спрайт изменяет положение или размер.

Реакция на события: спрайт выполняет действия в ответ на определенные события (например, нажатие клавиши или касание другого спрайта). Взаимодействия делают проекты динамичными и интерактивными, позволяя разрабатывать анимации, где спрайты синхронно выполняют действия. Создавать игры, где персонажи реагируют на действия игрока или друг друга (например, игрок собирает монеты, уворачивается от столкновений). Моделировать процессы, такие как физические столкновения или диалоги. Без взаимодействий проекты были бы статичными, так как каждый спрайт действовал бы независимо.

Событийное программирование – это подход, при котором действия в программе выполняются в ответ на определенные события. Например, команды «Когда клавиша нажата», «Когда спрайт нажат». Получение сообщения от другого спрайта команда «Когда я получу сообщение». Касание спрайта другим спрайтом или цветом, отвечает команда «Касается цвета».

Практическая часть.

1. Подготовка проекта.

Запустите Scratch на компьютере. Выберите спрайт (Dot) из библиотеки Scratch или нарисуйте свой спрайт. Добавьте спрайт «Препятствие» (например, выберите «Star» или другой объект, подходящий для роли препятствия). Установите фон (Moon). Выберите спрайт «Dot». Добавьте скрипт для управления спрайтом с помощью команды блока (события) «клавиша стрелка (вверх, низ, влево, вправо) нажата?» вставив в команду «если () то», 4 раза для каждой стороны.

2. Создание скрипта для смены костюма спрайта.

В команду «повторять всегда» вставьте получившийся скрипт для управления спрайтом. Для каждой стороны каждой стороны добавьте направление и количество шагов в количестве (15). После этого создайте скрипт для смен костюма спрайта «Dot». Вставьте в команду «повторять всегда», и блок (внешний вид) команды «Изменить костюм на (dot-a), (dot-b)». В команду «если () то», нужно поместить команду «Касается (Star)», после этого добавить команду (передать hit). Скрипт для (Star) в команду «повторять всегда» вставьте команды: «показаться» и «плыть 2 секунд к случайному положению». После этого добавьте скрипт получения сообщений из команд: «если («касается цвета») то», выберите цвет контура спрайта «Dot». После этого присоедините команду «спрятаться» и «стоп (этот скрипт)». Не забудьте поставить команду запуска скрипта.



Рисунок – 18 Скрипт смены костюма спрайта «Dot»

3. Практическое задание.

Модернизируйте существующий проект или создайте новый проект по собственному замыслу, с другими персонажами. В существующем проекте, измените количество шагов, управление спрайтом на другие клавиши. Сохранение проекта на компьютер.

4. Сохраните проект под именем «Проект 4» в личной папке.

5. Контрольные вопросы.

Таблица – 13 Контрольные вопросы по теме «Взаимодействие между объектами»

Вопросы	Варианты ответа
№ 1 Как добавить управление спрайтом с помощью клавиш?	а) <u>Использовать команду, когда флаг нажат и условия если «клавиша (стрелка) нажата?»</u> б) Использовать только команду передать сообщение в) Написать код без блоков г) <u>Использовать только команду ждать 1 секунду</u>
№ 2 Какой блок позволяет объекту двигаться вперед на определенное количество шагов?	а) <u>Идти (10) шагов</u> б) Повернуться вправо (15) градусов в) Изменить у на (– 10) г) <u>Отправить команду «Привет»</u>
№ 3 Как заставить спрайт остановиться после завершения скрипта ?	а) Использовать команду «спрятаться» б) Использовать команду «убрать графические элементы» в) Использовать команду «перейти на случайное положение» г) <u>Использовать команду «стоп все(этот скрипт)»</u>
№ 4 Что делает команда «если касается края, оттолкнуться»	а) <u>Заставляет спрайт отражаться от края сцены, когда он достигает границы</u> б) Останавливает движение спрайта, если он достигнет границы сцены в) Позволяет спрайту проходить сквозь границу сцены г) <u>Увеличивает скорость спрайта при достижении границы сцены</u>
№ 5 Какое событие используется для запуска скрипта при нажатии на команду «Когда зеленый флаг нажат»?	а) Когда нажмут клавишу пробел б) Когда начнется сцена в) <u>Когда щелкнут по зеленому флажку</u> г) Когда произойдет столкновение

Модуль 3. «Основы алгоритмизации».

Тема 5 «Линейный алгоритм».

Цель занятия – понимать, как устроен линейный алгоритм, научиться создавать проект, который будет включать линейный алгоритм.

Методические рекомендации по изучению темы «Линейный алгоритм»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Линейный алгоритм представляет собой простейший тип алгоритма, в котором действия выполняются в строгой последовательности, одно за другим. Это означает, что каждая операция следует за предыдущей в заранее определённом порядке, и ни одна из них не пропускается, не повторяется. В среде Scratch, где программирование осуществляется путём соединения визуальных блоков, линейный алгоритм выглядит как цепочка действий, выполняемых от начала до конца. Действия выполняются в том порядке, в котором они заданы. В Scratch такие алгоритмы могут управлять спрайтами (персонажами) или изменять визуальные элементы сцены.

Практическая часть.

1. Подготовка проекта.

Выберите спрайт (Pencil). Подключите дополнение (перо) расположенное в левом нижнем углу (СКИ). После команды запуска скрипта, используйте команду «Стереть всё», «Перейти в $x:(0)$ $y:(0)$ », «Опустить перо», «Установить размер пера (10)», «Установить для пера цвет (Зеленый)». Ниже используйте команды «Повернуться на $(90, 180, - 90, 0)$ градусов», для каждой стороны поворота спрайта. После этого, добавьте для каждой стороны поворота команду «идти 100 шагов», «Ждать (1) секунд», «Поднять перо».



Рисунок 19 – Фрагмент линейного алгоритма направления спрайта

2. Практическое задание.

Модифицируйте текущий скрипт, измените цвет пера. Выберите другие геометрические фигуры, такие как: прямоугольник, треугольник, пятиугольник, шестиугольник, с изменением цвета пера на каждой стороне.

3. Сохраните проект под именем «Проект 5» в личной папке.

4. Контрольные вопросы.

Таблица – 14 Контрольные вопросы по теме «Линейный алгоритм»

Вопросы	Варианты ответа
№ 1 Что такое линейный алгоритм?	а) Алгоритм, включающий в себя условия и циклы <u>б) Последовательность действий, которая выполняется строго одна за другой без условий и циклов</u> в) Алгоритм, основанный исключительно на использовании циклов г) Алгоритм, который всегда завершает свою работу за конечное время
№ 2 Какой блок используется для ожидания определенного времени перед выполнением следующего действия?	<u>а) ждать (1) сек</u> б) ожидать (1) сек в) задержка (1) сек г) паузу (1) сек
№ 3 Какой блок используется для завершения работы всех скриптов в проекте?	а) Остановить скрипт <u>б) Остановить все</u> в) Закончить проект г) Выйти из программы
№ 4 Что произойдёт, если вы используете блок «Поднять перо» во время движения спрайта?	а) Спрайт начнёт рисовать более толстую линию. <u>б) Спрайт перестанет оставлять след на сцене.</u> в) Спрайт изменит цвет линии на случайный. г) Спрайт повернётся на 90 градусов.
№ 5 Какой блок позволяет настроить толщину линии, рисуемой пером?	а) «Изменить цвет пера» <u>б) «Установить размер пера»</u> в) «Очистить перо» г) «Повернуться в направлении

Тема 6 «Циклический алгоритм».

Цель занятия – понимание циклического алгоритма, использование цикла для выполнения повторяющихся действий; создание проекта в котором используют циклы для автоматизации задач и упрощения кода.

Методические рекомендации по изучению темы «Циклический алгоритм»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Циклический алгоритм – это алгоритм, в котором определённая последовательность действий повторяется несколько раз до выполнения заданного условия или определённое количество раз. В программировании циклы позволяют автоматизировать повторяющиеся задачи, экономя время и упрощая код. В Scratch, визуальной среде программирования, циклические алгоритмы реализуются с помощью специальных блоков, которые позволяют повторять команды. Это делает создание игр, анимаций и других проектов

более эффективным. В Scratch доступны несколько видов циклических конструкций, которые можно найти в блоке «Управление»:

Цикл с заданным количеством повторений, команда «Повторить (n) раз», используется, когда известно точное количество повторений.

Бесконечный цикл, команда «Повторять всегда», выполняет действия бесконечно, пока программа не будет остановлена.

Цикл с условием, команда «Повторять, пока не ()»: Выполняет действия до тех пор, пока не выполнится заданное условие.

Практическая часть.

1. Подготовка проекта.

Выберите спрайт (Gobo) и фон (Jurassic). После команды запуска, выберите команду «Перейти в x:(0) y: (– 120)». Используя цикл с заданным количеством повторений, выберите команду «Повторить (4) раз», поместите в цикл команду «Изменить y на (50, – 50)», «Ждать 0.3 секунд». После этого выберите цикл с условием, команды «Повторять, пока не (Касается край)», внутри этого цикла добавьте «Изменить x на 5». Добавьте команду «Говорить (я на месте) 2 секунд». Используйте бесконечный цикл, команду «Повторять всегда». Внутри этого цикла используйте команды: «Повернуть на (15) градусов», «Ждать 0.1 секунд».



Рисунок 20 – Фрагмент циклического алгоритма движения спрайта

2. Практическое задание.

Используя все три типа циклов в Scratch (цикл с заданным количеством повторений, бесконечный цикл и цикл с условием), создайте проект, в которой спрайт начинает двигаться с координат x:(30) y: (– 50), выполняя прыжки (влево–вправо). Во время прыжков спрайт должен менять костюмы и

воспроизводить звук «Meow». После достижения края спрайт воспроизведёт звук «Pur».

3. Сохраните проект под именем «Проект 6» в личной папке.

4. Контрольные вопросы.

Таблица – 15 Контрольные вопросы по теме «Циклический алгоритм»

Вопросы	Варианты ответа
№ 1 Что такое циклический алгоритм?	а) Алгоритм, который выполняется один раз <u>б) Алгоритм, который выполняется несколько раз до выполнения определенного условия</u> в) Алгоритм, который не имеет конца г) Алгоритм, который выполняет случайные действия
№ 2 Какая команды используется для создания бесконечного цикла?	а) «Если () то» б) «Повторять» <u>в) «Повторять всегда»</u> г) «Ждать»
№ 3 Какая из команд позволяет повторять действия определенное количество раз?	а) «Повторять до» б) «Повторять» <u>в) «Повторить 10 раз»</u> г) «Повторять всегда»
№ 4 Что произойдет, если в бесконечном цикле не будет условия выхода?	а) Цикл завершится через несколько секунд б) Цикл выполнится один раз <u>в) Программа может зависнуть или перестать реагировать</u> г) Цикл выполнится и завершится автоматически
№ 5 Какой подход лучше всего использовать для оптимизации циклических алгоритмов?	а) Использовать много блоков «Сказать» <u>б) Минимизировать количество повторений</u> в) Увеличить количество шагов в каждом цикле г) Использовать только бесконечные циклы

Тема 7 «Разветвляющийся алгоритм».

Цель занятия – понимание разветвляющегося алгоритма, научиться создавать проекты, использующие условные операторы для принятия решений.

Методические рекомендации по изучению темы «Разветвляющийся алгоритм»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Разветвляющийся алгоритм – это алгоритм, в котором выполнение программы зависит от выполнения определённых условий. В зависимости от того, истинно или ложно условие, программа выбирает одну из ветвей выполнения. В Scratch такие алгоритмы реализуются с помощью команд

условных операторов «если (), то», «если (), то иначе» и логических операторов (И, ИЛИ, НЕ). Команда «если (условие) то» – выполняет действия, если условие истинно. Команда «если (условие) то, иначе» – выполняет одно действие при истинном условии, другое – при ложном. Логические операторы:

И – истинно, если оба условия верны. ИЛИ – истинно, если хотя бы одно условие верно. НЕ – меняет значение условия на противоположное (истину на ложь и наоборот). Вложенные условия– условия внутри других условий, позволяющие проверять более сложные комбинации.

Практическая часть.

1. Подготовка проекта.

Выберите спрайт (Avery) и фон (School). После команды запуска, выберите блок сенсоры, команду «Спросить (Сколько тебе лет?) и ждать». Выберите команду «если (), то иначе». Используйте оператор (ответ >10), после вставьте команды «Спросить (Сегодня не учебный день?) и ждать», «Если (оператор (ответ =да)), то», команда «Говорить (Ты можешь начать игру) (2) секунд».



Рисунок 21 – Фрагмент разветвляющегося алгоритма спрайта

После (Иначе) используйте команду «Спросить (Родители разрешили играть?) и ждать», выберите команду «если ((ответ =да)) то», «Говорить (Вы можете играть в игру) (2) секунд». Используйте команду «Стоп (все)», чтобы остановить скрипт.

2. Практическое задание.

Создайте проект по собственному замыслу, в котором спрайт принимает решение продолжить дальше на основе условий с использованием операторов «если», «иначе», и логических операторов (И, ИЛИ, НЕ). Добавьте визуальные

эффекты, например, смену костюмов спрайта (другой костюм при выборе первого ответа) блок «Внешний вид».

3. Сохраните проект под именем «Проект 7» в личной папке.

4. Контрольные вопросы.

Таблица – 16 Контрольные вопросы по теме «Разветвляющийся алгоритм»

Вопросы	Варианты ответа
№1 Что такое разветвляющийся алгоритм?	а) <u>Алгоритм, включающий в себя условия и циклы.</u> б) Последовательность действий, которая выполняется строго одна за другой без условий и циклов. в) Алгоритм, основанный исключительно на использовании циклов. г) Алгоритм, который всегда завершает свою работу за конечное время.
№2 Какая команда в Scratch используется для проверки условия?	а) <u>«Если () тогда»</u> б) Проверить условие в) Условие выполнено г) Выполнить если
№3 Как можно объединить два условия в одной команде в Scratch?	а) Использовать оператор «условие» и «условие» б) Использовать команду объединение условий. в) Использовать оператор «Не ()» г) <u>Использовать оператор (или)</u>
№4 Что произойдет, если в бесконечном цикле не будет условия выхода?	а) Цикл завершится через несколько секунд б) Цикл выполнится один раз в) <u>Программа может зависнуть или перестать реагировать</u> г) Цикл выполнится и завершится автоматически
№5 Как можно проверить равенство двух значений в Scratch?	а) <u>Использовать оператор = внутри команды «если () тогда».</u> б) Использовать оператор == команды «если () тогда». в) Использовать команду равенство. г) Использовать команда сравнить.

Тема 8 «Счетчик».

Цель занятия – изучение блоков «Переменные» и «Операторы». Знать, что такое переменные и для чего они нужны в программировании; уметь создавать переменные.

Методические рекомендации по изучению темы «Счетчик»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Переменные в программировании – это специальные (контейнеры), которые используются для хранения данных, таких как числа, текст или другие значения. В Scratch переменные позволяют сохранять информацию, которую можно использовать и изменять в процессе выполнения программы.

Переменные позволяют запомнить информацию, например, счет в игре или имя игрока. С помощью переменных можно изменять данные в программе, например, уровень сложности. Переменные позволяют использовать одно и то же значение в разных частях программы без необходимости его повторного ввода. Значения переменных могут измениться в процессе работы программы.

Практическая часть.

1. Подготовка проекта.

Выберите спрайт 1 (Ghost), спрайт 2 (Star), фон (Woods). После команды запуска, создайте переменную (очки) для спрайта 1. Поместите в цикл «Повторять всегда», команду «идти (5) шагов» и «повернуться к (указателю мыши)». Используйте команду условного оператора «если (касается (star)), то», переменную (очки) на 1, после этого используйте команду «говорить (звезда собрана) (1) секунд». Ниже проверяется условие переменной, используйте команду «если (очки = (10)), то», «Говорить (Все собрал) (1) секунд». Для спрайта 2 (Star), используйте команду «Спрятаться». В цикл «Повторять всегда», поместите команды: «Ждать (2) секунд», «Перейти на (случайное положение)», «Показаться».



Рисунок 22 – Фрагмент скрипта с использованием переменной (Очки)

2. Практическое задание.

Создайте проект по собственному замыслу, в котором переменная используется в линейных и разветвляющихся алгоритмах.

3. Сохраните проект под именем «Проект 8» в личной папке.

4. Контрольные вопросы.

Таблица – 17 Контрольные вопросы по теме «Счетчик»

Вопросы	Варианты ответа
№1 Что такое переменная в Scratch?	а) Структура данных, хранящая текст <u>б) Значение, которое может изменяться в процессе выполнения программы</u> в) Фиксированное значение, которое невозможно изменить г) Команда, выполняющая действие
№2 Как создать новую переменную в Scratch?	а) Нажать на кнопку «События» <u>б) Использовать команду «Создать переменную»</u> в) Нажать на кнопку «Сценарий» г) Перетащить команду «Значение»
№3 Какой тип данных может храниться в переменной?	а) Только числа б) Только текст <u>в) Числа, текст и логические значения</u> г) Только изображения
№4 Как можно использовать переменные в Scratch?	а) Для хранения постоянных значений <u>б) Для отслеживания состояния игры или значений</u> в) Для создания графики г) Для управления звуками
№5 Как можно отобразить значение переменной на экране?	а) Использовать команду «Скрыть» <u>б) Поставить галочку напротив используемой переменной</u> в) Использовать команду «Скрыть переменную» г) Использовать команду «Изменить»

Модуль 4. «Создание проектов в Scratch»

Тема 9 «Создание анимационного проекта».

Цель занятия – создание анимационного проекта. Изучение применения генерации случайных чисел для создания динамичных и интерактивных игровых элементов.

Методические рекомендации по изучению темы «Создание анимационного проекта»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Анимационные проекты в Scratch строятся на основе спрайтов, сцен, скриптов и событий. Создание анимационного проекта предполагает использование спрайтов (персонажей или объектов), их движения, изменения внешнего вида и взаимодействия. Использование оператора генерации случайных чисел добавляет элемент непредсказуемости, что делает анимацию более динамичной и увлекательной, делая её более интересной. Анимационные

проекты в Scratch строятся на четырёх ключевых компонентах: Спрайты – персонажи или объекты, которые двигаются и взаимодействуют. Сцены (фоны) – фон, на котором происходит действие. Скрипты – блоки кода, управляющие поведением спрайтов. События – запускающие скрипты (например, нажатие клавиши или клик по спрайту). Способы анимации спрайтов: смена костюмов – создание иллюзии движения, движение и поворот – перемещение спрайта или объектов. Эффекты графики – изменение размера, прозрачности или цвета спрайта. Клоны – создание множества одинаковых объектов.

Практическая часть.

1. Подготовка проекта.

Выберите спрайт (Star), фон (Woods). После команды запуска, выберите команду «спрятаться». После этого в бесконечный цикл «повторять всегда», добавьте команды: «ждать «Выдать случайное от (0.5) до (2) секунд», «создать клон (самого себя)». Второй скрипт отвечает за создание клонов спрайта (Star). Для этого используя команды: «когда, я начинаю как клон», «перейти на (случайное положение)», «Показаться». Используя команду «повторять (20) раз», выберите команды «играть звук (collect) до конца», «изменить у на (– 10)», «изменить эффект (цвет) на (выдать случайное число от (– 10) до (10))», «ждать (1) секунд».

После этого выберите команду «Удалить клон».

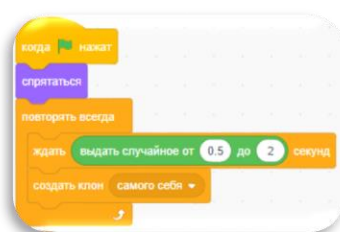


Рисунок 23 – Скрипт с использованием оператора генерации случайных чисел

2. Практическое задание.

Модернизируйте проект. Используйте редактор Scratch для создания собственных костюмов, чтобы сделать проект уникальным. Добавьте несколько спрайтов, случайные движения, размеры и эффекты для каждого

спрайта. Используйте переменные для отслеживания состояния (например, количества собранных звёзд).

3. Сохраните проект под именем «Проект 9» в личной папке.

4. Контрольные вопросы.

Таблица – 18 Контрольные вопросы по теме «Создание анимационного проекта»

Вопросы	Варианты ответа
№1 Какой блок в Scratch используется для генерации случайного числа	а) «Выдать случайное от () до () секунд» б) «Случайный выбор» в) «Случайное значение» г) «Генерировать случайное число»
№2 Какой диапазон можно задать для генерации случайного числа в Scratch?	а) Только от 0 до 100 б) <u>От любого числа до любого другого числа</u> в) Только от – 10 до 10 г) От 1 до 5
№3 Что произойдет, если вы используете команду «Выдать случайное от () до () секунд» несколько раз?	а) Всегда будет одно и то же число б) <u>Каждый раз будет генерироваться новое случайное число в заданном диапазоне</u> в) Блок не будет работать г) Генерируется число, которое всегда больше 10
№4 Как можно использовать случайные числа в игре?	а) <u>Для создания случайных событий, таких как появление спрайтов</u> б) Для управления скоростью спрайтов в) Для хранения постоянных значений г) Для создания графики
№5 Как можно изменить диапазон генерации случайного числа в Scratch?	а) Изменить настройки проекта б) <u>Изменить параметры в команде «Выдать случайное от () до () секунд»</u> в) Невозможно изменить диапазон г) Изменить переменные в коде

Тема 10 «Игры, создаваемые в среде Scratch».

Цель занятия – научиться разрабатывать и создавать игры в Scratch.

Методические рекомендации по изучению темы «Игры, создаваемые в среде Scratch»:

- Изучить теоретическую часть;
- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Создание игры в Scratch – это структурированный процесс, который включает несколько этапов. Каждый этап требует внимания к деталям, чтобы конечный продукт был интересным, функциональным и понятным для игроков. Создание игры в Scratch можно поделить на несколько этапов: 1) Определение

концепции игры, жанра и целей. 2) Создание визуальных элементов—персонажей (спрайтов), фонов и объектов. 3) Настройка механик игры, включая управление, взаимодействия между объектами и логику игры. 4) Проверка игры на ошибки, баланс сложности и удобство для игрока.

Практическая часть.

1. Подготовка проекта.

Определение концепции игры, жанра и целей. «Космическое приключение» – это аркадная игра, в которой игрок управляет космическим кораблем, отклоняя курс корабля от астероида. Всего дано 10 очков здоровья, игрок должен маневрировать, избегать препятствий, и стараться не проиграть. Нужно набрать 15 звезд, чтобы завершить игру победой. Код разделен на три основных спрайта: космический корабль (управляемый игроком), астероид, звезда (объект для сбора). Каждый спрайт имеет свои скрипты, которые реализуют механики, соответствующие концепции и жанру аркады.

2. Практическое задание.

Выберите спрайт 1 (Rocketship), спрайт 2 (Rocks), спрайт 3 (Star), фон (Galaxy). Создайте скрипт управления спрайтом рассмотренный в теме 3 «Направление и движение спрайта» для спрайта (Rocketship). После команды запуска, выберите команды «перейти в x: (0) y: (0)». Используя команду «повторять всегда», добавьте команды «следующий костюм» и «ждать (1) секунд».

Для спрайта спрайт 2 (Rocks), создайте переменную (Очки здоровья). После команды запуска, выберите переменную указав значение (0). Используя команду «повторять всегда», добавьте команды «если «касается Rocketship», то», «изменить (Очки здоровья) на (1)», «ждать (1) секунд». Для проверки условий добавим команды, «если (очки здоровья>6), то», «говорить (Будь аккуратней (1) секунд»; «если (очки здоровья>10), то», «говорить (Игра окончена) (1) секунд», «играть звук (Boom Cloud) до конца», «стоп (все)». Для передвижения спрайта используйте следующий скрипт:



Рисунок – 24 Скрипт для движения спрайта

Для спрайта 3 (Star), создайте переменную (Количество звёзд). После команды запуска, выберите переменную указав значение (0). Используйте команду «спрятаться». В команду «повторять всегда» добавьте следующие команды: «если (Количество звезд <15), то», «перейти на (случайное положение)», «показаться», «плыть (выжать случайное от (1) до (10) секунд) к (случайное положение)», «ждать (0.5) секунд». Для изменения значений переменной (Кол– во звёзд) используйте следующий скрипт:



Рисунок – 25 Скрипт для изменения значений переменной (Кол– во звёзд)

3. Сохраните проект под именем «Проект 10» в личной папке.

4. Контрольные вопросы.

Таблица – 19 Контрольные вопросы по теме «Игры, создаваемые в среде Scratch»

Вопросы	Варианты ответа
№1 Какой элемент является основным в создании игры в Scratch?	а) Фон б) Спрайт в) Звук г) Переменная
№2 Как можно добавить звук в игру Scratch?	а) Использовать блок «Проиграть звук» б) Звуки нельзя добавлять в Scratch в) Использовать команду «играть звук () до конца» г) Использовать команду «Скрыть»
№3 Как можно использовать переменные в играх Scratch?	а) Для хранения значений, таких как очки или здоровье б) Для управления фоном в) Невозможно использовать переменные в играх г) Для создания новых спрайтов

№4 Как можно поделиться своей игрой, созданной в Scratch, с другими пользователями?	а) Опубликовать проект на официальном сайте Scratch. б) Отправить ссылку на проект друзьям. в) Скачать проект и передать его на флешке. г) <u>Все вышеперечисленные способы.</u>
№5 Какой команда используется для проверки нескольких условий одновременно?	а) команда «если () то» б) команда « <u>() и ()</u> » в) команда «повторить пока ()» г) команда «ждать до ()»

Тема 11 «Итоговое занятие: Создание собственного проекта– игры».

Цель занятия – создавать собственную игру в Scratch, используя изученные концепции. Определить планирование и дизайна игры перед началом написания проекта.

Изучить теоретическую часть;

- Выполнить практическую часть;
- Ответить на контрольные вопросы;

Теоретическая часть

Жанр игры определяет ее стиль, механики и общую игровую обстановку. Выбор жанра важен, так как он задает направление для всех последующих этапов разработки. В Scratch можно создавать игры различных жанров (платформер, аркада, головоломка и т.д.). Платформер: игрок управляет персонажем, который перемещается по платформам, прыгает, избегает препятствий и собирает объекты. Аркада: ориентирована на реакцию и накопление очков. Головоломка: требует решения задач и логического мышления. На этапе планирования игры важно четко определить, какие действия игрок будет выполнять и как объекты в игре будут реагировать друг на друга.

Практическая часть.

1. Практическое задание.

Используя изученный материал, определите планирование и дизайн игры перед началом написания проекта, создайте собственный сценарий для создания своего проекта (игры). Выберите жанр игры: (платформер, аркада, головоломка и т.д.).

2. Сохраните проект под именем «Проект 11» в личной папке.

ПРИЛОЖЕНИЕ Б

Таблица 20 – Стимульный материал методики №1 «Установление закономерностей» Борис Леонидович Покровский

1.	X^{*+*}	клад пила рана жижа вода
2.	$*X.^{*}$	репа шлак соус крик луна
3.	$*=.$	крик обод тара воск рама
4.	$?:^{?}=$	лупа хата обои град лапа
5.	$+^{?}*^{?}$	лицо река желе квас окно
6.	$X=XO^{+}$	старт сосна пепел ребро ответ
7.	$*+^{?}^{?}+$	маска касса месса сумма лассо
8.	$0=*X=$	слово вахта олово казак жажда
9.	$?XO:^{?}$	аванс холод стихи рупор чашка
10.	$=*X=*^{*}$	тайна кашка шапка ванна ответ
11.	$*:^{0}:XO$	молоко фараон ананас метеор мнение
12.	$=X^{?}X:^{!}$	статья талант стакан рапира сорока
13.	$?-0=-0$	пурпур филиал шедевр скалка фосфор
14.	$X:^{*}+=0$	почерк стакан полоса ангина старик
15.	$?+00+=$	разлад окорок капкан сундук список
16.	$X^{*}0^{*}X^{*}0$	балаган порошок леденец колокол соломка
17.	$+ = 0^{?}0^{?}*^{*}$	барабан пулемет чувство филолог фамилия
18.	$*+0=+:^{!}$	гипофиз конвоир гиацинт кипарис лазарет
18.	$*+0=+:^{!}$	гипофиз конвоир гиацинт кипарис лазарет
19.	$O=*^{*}=O:^{!}$	шеренга кантата милиция веревка колокол
20.	$*:^{0}!X:^{!}$	поворот гравюра складка селедка адмирал
21.	$!=+^{*}+0=:$	синоптик градация оболочка фантазия биология
22.	$?^{*}0=+^{*}0:$	анаконда прокурор снижение внимание описание
23.	$?=!-OX-!^{!}$	антрекот квартира антрацит скакалка практика
24.	$!^{?}+^{?}=X^{?}0$	катафалк царапина бурундук анфилада карандаш
25.	$!+=XOOX:$	пассажир оперетта коллизии рассылка комиссия

ПРИЛОЖЕНИЕ В

Бланк для тестирования методика №2 «Игра в обмен» (Модификация 1)

Анатолий Залманович Зак.

Тренировочные задачи:

1. НКП – КНП (одно действие).

№ 1. 1) КНП

2. Р К М ТВ – К Р В Т М (два действия).

№ 2. 1) КРМТВ 2) КРВТМ

Основные задачи:

1. М Б Т Н К Р – Н К Р М Б Т (три действия).

№ 1. 1) 2) 3)

2. Р В Ш К Л Д – К Л Д Р В Ш (три действия).

№ 2. 1) 2) 3)

3. И А У О Е Ю Я – Е Ю Я О И А У (три действия).

№ 3. 1) 2) 3)

4. К Р В Г Н С П – Н С П Г К Р В (три действия).

№ 4. 1) 2) 3)

5. Р Д К Ш В Ф М Ч – В Ф М Ч Р Д К Ш (четыре действия).

№ 5. 1) 2) 3) 4)

6. П С Н Г Л В Р К – Л В Р К П С Н Г (четыре действия).

№ 6. 1) 2) 3) 4)

7. Р К Н С Ш Т Б М Д – Т М Б Д Ш Р К Н С (четыре действия).

№ 7. 1) 2) 3) 4)

8. Ч М Ф В Ж Ш К Д Р – Ш К Д Р Ж Ч М Ф В (четыре действия).

№ 8. 1) 2) 3) 4)

ПРИЛОЖЕНИЕ Г

Бланк для тестирования методика №2 «Игра в обмен» (Модификация 2)
Анатолий Залманович Зак.

Тренировочная задача 1

Условие	Решение
---------	---------

6 2 7 1 – 2 6 1 7	1) 2 6 7 1, 2) 2 6 1 7
-------------------	------------------------

Основные задачи:

1) 9 2 3 8 – 2 9 8 3 (два действия) № 1. 1) 2)

2) 6 1 2 7 8 4 – 1 6 7 2 4 8 (три действия) № 2. 1) 2) 3)

3) 2 4 7 1 8 6 9 3 – 4 2 1 7 6 8 3 9 (четыре действия) № 3. 1) 2) 3) 4)

Тренировочная задача 2

Условие	Решение
---------	---------

3 7 1 8 – К Н Н К	1) 3713 2) 3773
-------------------	-----------------

8 1 7 3 – Х Л Л Х	1) 8178 2) 8118
-------------------	-----------------

Основные задачи

4) 8 2 3 6 – Т Х Х Т

6 3 2 8 – Р С С Р (два действия)

5) 5 1 2 7 6 4 – П Л К К Л П

4 6 7 2 1 5 – С Н М М Н С (три действия)

6) 1 2 8 6 5 3 9 4 – Г Б Н Ч Ч Н Б Г

4 9 3 5 6 8 2 1 – В Р Т С С Т Р В (четыре действия)

7) В Т Р М Н – М Н Р В Т (два действия)

8) Г Л Б Х Р К – Х Р К Г Л Б (три действия)

9) М Ш Л Г Р – Д С Ш М (два действия)

10) Н Х Л Г Р М – М Р Г Л Х Н (три действия)

ПРИЛОЖЕНИЕ Д

Методика №1 «Установление закономерностей» Борис Леонидович Покровский. Результаты прохождения констатирующего этапа в экспериментальной и контрольной группе.

Таблица 21 – Результаты прохождения методики экспериментальной группы

Ученик	Д (кол-во слов по ключу для просмотренного материала)	Пропуски	Ошибки	С (общее кол- во всех просмотренных слов)	Успешность
№1	24	6	10	95	51
№2	14	2	6	60	36
№3	24	2	0	95	87
№4	17	1	0	70	63
№5	32	2	4	125	104
№6	31	2	6	120	97
№7	14	2	1	60	48
№8	23	2	0	90	85
№9	23	1	0	90	86
№10	32	2	8	125	93
№11	25	0	8	100	75
№12	23	1	1	95	90
№13	32	2	6	125	98
№14	19	3	2	75	53
№15	17	1	2	70	63
№16	23	1	3	90	76
№17	20	1	3	80	66
№18	19	1	2	75	64
№19	14	1	5	60	52
№20	20	0	2	80	72

Таблица 22 – Результаты прохождения методики контрольной группы

Ученик	Д (кол-во слов по ключу для просмотренного материала)	Пропуски	Ошибки	С (общее кол- во всех просмотренных слов)	Успешность
№1	32	4	2	125	101
№2	32	4	2	125	102
№3	32	4	5	125	94
№4	32	6	5	125	87
№5	32	3	7	125	93
№6	32	2	5	125	101
№7	25	4	5	100	70
№8	20	4	5	80	51
№9	32	5	14	125	73
№10	24	5	5	95	61
№11	26	0	2	105	97
№12	25	2	12	100	64

№13	25	4	9	100	63
№14	32	3	6	125	95
№15	19	0	2	75	68
№16	21	2	11	85	50
№17	26	2	4	105	84
№18	25	2	1	100	88
№19	20	2	6	80	55
№20	20	2	7	80	53
№21	32	5	7	125	86
№22	24	6	6	95	57
№23	25	2	4	100	79

ПРИЛОЖЕНИЕ Е

Методика №2 «Методика №1 «Игра в обмен» (Модификация №1)
Анатолий Залманович Зак. Результаты прохождения констатирующего этапа в экспериментальной и контрольной группе.

Таблица 23 – Результаты выполнения заданий в экспериментальной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 1).

(Модификация № 1) Методика «Игра в обмен» (Зак А.3.)										
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Общий балл	Уровень успешности
1	+	+	+	+	–	–	–	–	4/8	Средний
2	+	+	+	+	+	+	–	–	6/8	Средний
3	+	+	+	+	+	+	+	–	7/8	Высокий
4	+	+	+	–	–	–	–	–	3/8	Низкий
5	+	+	+	–	–	–	–	–	3/8	Низкий
6	+	+	+	+	+	+	+	+	8/8	Высокий
7	+	+	+	+	+	–	–	–	5/8	Средний
8	+	+	+	+	+	+	+	+	8/8	Высокий
9	+	+	+	+	+	+	–	–	6/8	Средний
10	+	+	+	+	+	+	–	–	6/8	Средний
11	+	+	+	+	+	+	–	–	6/8	Средний
12	+	+	–	–	–	–	–	–	2/8	Низкий
13	+	+	+	+	+	–	–	–	5/8	Средний
14	+	+	+	–	–	–	–	–	3/8	Низкий
15	+	+	+	+	+	+	+	–	7/8	Высокий
16	+	+	–	–	–	–	–	–	2/8	Низкий
17	+	+	+	–	–	–	–	–	3/8	Низкий
18	+	+	+	+	+	–	–	–	5/8	Средний
19	+	–	+	+	–	–	–	–	3/8	Низкий
20	+	+	+	+	+	+	–	–	6/8	Средний

Таблица 24 – Результаты выполнения заданий в контрольной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация №1).

(Модификация № 1) Методика «Игра в обмен» (Зак А.3.)										
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Общий балл	Уровень успешности
1	+	+	–	+	–	–	–	–	3/8	Низкий
2	+	+	+	+	+	+	–	–	6/8	Средний
3	+	+	+	+	+	–	+	+	6/8	Средний
4	+	+	+	+	+	+	–	–	6/8	Средний
5	+	+	+	+	–	–	–	–	3/8	Низкий
6	+	+	+	+	+	+	–	–	6/8	Средний
7	+	+	+	+	+	+	+	–	7/8	Высокий
8	+	+	+	+	+	+	+	–	7/8	Высокий
9	+	+	+	+	+	–	–	+	6/8	Средний
10	+	+	+	+	+	+	+	–	7/8	Высокий

11	+	+	+	+	+	+	—	+	7/8	Высокий
12	+	+	+	+	+	+	—	—	6/8	Средний
13	+	+	+	+	+	+	—	—	6/8	Средний
14	+	+	+	+	+	—	—	—	5/8	Средний
15	+	+	—	—	—	—	—	—	2/8	Низкий
16	+	+	+	+	+	—	—	—	5/8	Средний
17	+	+	+	+	—	—	—	—	4/8	Средний
18	+	+	+	+	+	+	—	—	6/8	Средний
19	+	+	+	—	—	—	—	—	3/8	Низкий
20	+	+	+	+	+	+	+	—	7/8	Высокий
21	+	+	+	+	+	+	—	—	6/8	Средний
22	+	+	+	—	+	+	+	+	7/8	Высокий
23	+	+	—	+	—	—	—	—	3/8	Низкий

ПРИЛОЖЕНИЕ Ж

Методика №2 «Методика №2«Игра в обмен» (Модификация №1)
Анатолий Залманович Зак. Результаты прохождения констатирующего этапа в
экспериментальной и контрольной группе.

Таблица 25– Результаты выполнения заданий в экспериментальной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 2).

[illegible]

Таблица 26 – Результаты выполнения заданий в контрольной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 2).

[illegible]

[illegible]

ПРИЛОЖЕНИЕ И

Методика №1 «Установление закономерностей» Борис Леонидович Покровский. Результаты прохождения контрольного этапа в экспериментальной и контрольной группе.

Таблица 27 – Результаты прохождения методики экспериментальной группы

Ученик	Д (кол–во слов по ключу для просмотренного материала)	Пропуски	Ошибки	С (общее кол– во всех просмотренных слов)	Успешность
№1	25	4	7	100	65
№2	19	1	5	75	56
№3	25	1	3	100	89
№4	21	3	0	85	72
№5	32	2	3	125	102
№6	31	1	5	120	94
№7	23	2	2	90	75
№8	24	1	0	95	88
№9	23	1	0	90	86
№10	32	1	7	125	99
№11	25	0	6	100	80
№12	27	0	1	110	102
№13	32	1	4	125	105
№14	19	3	2	75	53
№15	24	1	2	95	84
№16	23	1	2	90	79
№17	25	1	1	100	92
№18	23	2	1	90	79
№19	15	1	2	65	54
№20	23	0	1	90	86

Таблица 28 – Результаты прохождения методики контрольной группы

Ученик	Д (кол–во слов по ключу для просмотренного материала)	Пропуски	Ошибки	С (общее кол– во всех просмотренных слов)	Успешность
№1	32	2	2	125	106
№2	32	5	1	125	102
№3	32	5	3	125	96
№4	32	4	4	125	97
№5	32	3	6	125	95
№6	32	3	4	125	100
№7	25	3	2	100	82
№8	20	1	2	80	69

№9	32	3	5	125	97
№10	23	3	5	95	67
№11	27	1	2	110	98
№12	27	3	8	100	75
№13	25	1	2	95	84
№14	32	3	5	125	97
№15	19	0	2	75	68
№16	21	3	9	85	51
№17	26	2	4	105	84
№18	26	2	3	105	86
№19	23	3	2	90	73
№20	24	0	4	95	81
№21	32	3	7	125	92
№22	25	2	6	100	72
№23	25	2	3	100	82

ПРИЛОЖЕНИЕ К

Методика №2 «Методика №1 «Игра в обмен» (Модификация №1)
Анатолий Залманович Зак. Результаты прохождения контрольного этапа в экспериментальной и контрольной группе.

Таблица 29– Результаты выполнения заданий в экспериментальной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 1).

(Модификация № 1) Методика «Игра в обмен» (Зак А.3.)										
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Общий балл	Уровень успешности
1	+	+	+	+	+	+	–	–	6/8	Средний
2	+	+	+	+	+	+	+	–	7/8	Высокий
3	+	+	+	+	+	+	+	–	7/8	Высокий
4	+	+	+	+	+	–	–	–	5/8	Средний
5	+	+	+	+	–	–	–	–	4/8	Средний
6	+	+	+	+	+	+	–	–	6/8	Средний
7	+	+	+	+	+	+	–	–	5/8	Средний
8	+	+	+	+	+	+	+	+	8/8	Высокий
9	+	+	+	+	+	+	+	–	7/8	Высокий
10	+	+	+	+	+	+	–	–	7/8	Средний
11	+	+	+	+	+	+	–	–	6/8	Средний
12	+	+	+	+	+	+	+	–	7/8	Высокий
13	+	+	+	+	+	+	–	–	6/8	Средний
14	+	+	+	+	–	–	–	–	4/8	Средний
15	+	+	+	+	+	+	–	–	6/8	Средний
16	+	+	–	–	–	–	–	–	2/8	Низкий
17	+	+	+	+	+	–	–	–	5/8	Средний
18	+	+	+	+	+	+	–	–	6/8	Высокий
19	+	+	+	+	–	–	–	–	4/8	Средний
20	+	+	+	+	+	+	+	–	7/8	Высокий

Таблица 30 – Результаты выполнения заданий в контрольной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 1).

(Модификация № 1) Методика «Игра в обмен» (Зак А.3.)										
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Общий балл	Уровень успешности
1	+	+	+	+	–	–	–	–	4/8	Средний
2	+	+	+	+	+	+	+	–	7/8	Высокий
3	+	+	+	+	+	–	+	+	6/8	Средний
4	+	+	+	+	+	+	–	–	6/8	Средний
5	+	+	+	+	–	–	–	–	3/8	Низкий
6	+	+	+	+	+	+	–	–	6/8	Средний
7	+	+	+	+	+	+	+	–	7/8	Высокий
8	+	+	+	+	+	+	+	–	7/8	Высокий

9	+	+	+	+	+	—	—	+	6/8	Средний
10	+	+	+	+	+	+	+	—	7/8	Высокий
11	+	+	+	+	+	+	—	+	7/8	Высокий
12	+	+	+	+	+	+	—	—	6/8	Средний
13	+	+	+	+	+	+	—	—	6/8	Средний
14	+	+	+	+	+	—	—	—	5/8	Средний
15	+	+	—	—	—	—	—	—	2/8	Низкий
16	+	+	+	+	+	—	—	—	5/8	Средний
17	+	+	+	+	—	—	—	—	4/8	Средний
18	+	+	+	+	+	+	—	—	6/8	Средний
19	+	+	+	+	—	—	—	—	4/8	Средний
20	+	+	+	+	+	+	+	—	7/8	Высокий
21	+	+	+	+	+	+	—	—	6/8	Средний
22	+	+	+	—	+	+	+	+	7/8	Высокий
23	+	+	+	+	—	—	—	—	4/8	Средний

ПРИЛОЖЕНИЕ Л

Методика №2 «Игра в обмен» (Модификация №2) Анатолий Залманович Зак. Результаты прохождения контрольного этапа в экспериментальной и контрольной группе.

Таблица 31 – Результаты выполнения заданий в экспериментальной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 2).

(Модификация № 2) Методика «Игра в обмен» (Зак. А. 3)												
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Задача №9	Задача №10	Общий балл	Уровень успешности
1	+	+	+	+	+	+	+	+	+	–	9/10	Высокий
2	+	+	+	+	+	–	–	–	–	–	5/10	Средний
3	+	+	+	–	+	+	–	–	–	–	5/10	Средний
4	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
5	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
6	+	+	+	+	+	–	–	–	–	–	5/10	Средний
7	+	+	+	+	+	+	+	+	+	–	9/10	Высокий
8	+	+	+	+	+	–	–	–	–	–	5/10	Средний
9	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
10	+	+	+	+	+	–	–	–	–	–	5/10	Средний
11	+	+	+	+	+	–	–	–	–	–	5/10	Средний
12	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
13	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
14	+	+	+	+	+	+	+	+	+	–	9/10	Высокий
15	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
16	+	+	+	+	–	–	–	–	–	–	4/10	Низкий
17	+	+	+	+	+	–	–	–	–	–	5/10	Средний
18	+	+	+	+	+	+	–	–	–	–	6/10	Средний
19	+	+	+	+	+	–	–	–	–	–	5/10	Средний
20	+	+	+	+	+	–	–	–	–	–	5/10	Средний

Таблица 32 – Результаты выполнения заданий в контрольной группе по методике «Игра в обмен» (Зак. А. 3) (Модификация № 2).

(Модификация № 2) Методика «Игра в обмен» (Зак. А. 3)												
Ученики	Задача №1	Задача №2	Задача №3	Задача №4	Задача №5	Задача №6	Задача №7	Задача №8	Задача №9	Задача №10	Общий балл	Уровень успешности
1	+	+	+	+	+	+	–	–	–	–	6/10	Средний
2	+	+	+	+	+	+	–	–	–	–	6/10	Средний
3	+	+	+	+	+	–	–	–	–	–	5/10	Низкий
4	+	+	+	+	+	+	+	–	–	–	7/10	Средний
5	+	+	+	+	+	+	–	–	–	–	6/10	Средний
6	+	+	+	+	+	–	–	–	–	–	5/10	Средний
7	+	+	+	+	+	+	+	+	–	–	7/10	Средний
8	+	+	+	+	+	–	–	–	–	–	5/10	Средний
9	+	+	+	+	+	–	–	–	–	–	5/10	Средний
10	+	+	+	+	+	–	–	–	–	–	5/10	Средний

11	+	+	+	+	+	+	—	—	—	—	6/10	Средний
12	+	+	+	+	—	—	—	—	—	—	4/10	Низкий
13	+	+	+	+	+	—	—	—	—	—	5/10	Средний
14	+	+	+	+	+	+	+	—	—	—	7/10	Средний
15	+	+	+	—	—	—	—	—	—	—	3/10	Низкий
16	+	+	+	+	—	—	—	—	—	—	4/10	Низкий
17	+	+	+	+	—	—	—	—	—	—	4/10	Низкий
18	+	+	+	+	+	+	—	—	—	—	6/10	Средний
19	+	+	+	+	+	—	—	—	—	—	5/10	Средний
20	+	+	+	+	+	+	+	—	—	—	7/10	Средний
21	+	+	+	+	+	+	—	—	—	—	6/10	Средний
22	+	+	+	+	+	+	—	—	—	—	6/10	Средний
23	+	+	+	+	+	—	—	—	—	—	5/10	Средний