

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ МУНИЦИПАЛЬНОГО  
ОБРАЗОВАНИЯ «ГОРОД СЕВЕРОБАЙКАЛЬСК»  
МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ «ДОМ ДЕТСКОГО ТВОРЧЕСТВА «ЭВРИКА»

«ПРИНЯТО»  
На педагогическом совете  
МАОУ ДО «ДТТ «Эврика»

Протокол № 01  
От «24» 08 2022 года



«УТВЕРЖДЕНО»  
Директор МАОУ ДО «ДТТ «Эврика»  
М.О. Фибих Фибих М.О.

Приказ № 53  
От «24» 08 2022 года

«ПРИНЯТО»  
На педагогическом совете  
МАОУ «Гимназия №5»  
имени Л.Н. Усыниной

Протокол № 2  
От «27» 08 2022 года



«УТВЕРЖДЕНО»  
Директор МАОУ «Гимназия №5»  
имени Л.Н. Усыниной  
Е.Г. Бутаков Бутаков Е.Г.

Приказ № 248/1  
От «  »    2022 года

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

**«ОСНОВЫ АЛГОРИТМИКИ И ЛОГИКИ»**

Направленность: *техническая*

Возраст: 6-12 лет  
Срок реализации: 1 год

Автор-разработчик: педагог дополнительного образования  
Давыдкина Вера Владиславовна

г. Северобайкальск, 2022

## I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Основы алгоритмизации и программирования являются важной составляющей курса информатики средней школы. В Федеральном государственном образовательном стандарте основного общего образования (ФГОС ООО) указано, что одной из целей изучения курса информатики является развитие у учащихся основ алгоритмического мышления. Под способностью алгоритмически мыслить понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата. Для того чтобы записать алгоритм решения задачи, необходим какой-то формальный язык, например блок-схемы. В примерной программе по информатике предполагается рассмотрение основных алгоритмических конструкций: ветвление, цикл, вспомогательный алгоритм. Также стоит отметить, что основы алгоритмизации в дальнейшем выступают базой для обучения программированию.

### **1. Основные характеристики программы:**

**1.1. Дополнительная общеразвивающая программа «Основы алгоритмики и логики»** (далее - **Программа**) реализуется в соответствии с **технической направленностью** образования и предназначена для использования в системе дополнительного образования детей. Обучающиеся в ходе занятий приобщаются к знаниям в области информационных технологий, формируют логическое мышление.

**1.2. Актуальность программы** обусловлена широким внедрением информационных технологий в образовательные процессы и обычную жизнь каждого человека. Данная программа способствует развитию мотивации к получению новых знаний, возникновению интереса к программированию как к инструменту самовыражения в творчестве, помогает в самоопределении и выявлении профессиональной направленности личности. Изучение основных принципов практики написания программ. Для обучения была выбрана среда разработки Scratch. Данный выбор обусловлен тем, овладев даже минимальным набором операций, учащиеся смогут создавать законченные проекты. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования настолько прост и доступен, что Scratch рассматривается в качестве средства обучения не только старших, но и 4 младших школьников. Знания и умения, приобретенные в результате освоения курса, являются базой для обучения программированию

**1.3. Отличительные особенности программы** - программа модифицированная, за основу взята рабочая программа, разработанная Министерством Просвещения Российской Федерации «реализация дополнительной общеобразовательной программы по тематическому направлению основы алгоритмики и логики с использованием оборудования центра цифрового образования детей «IT-куб», а также обобщен опыт введения данного направления на протяжении 1 года.

**1.4. Педагогическая целесообразность** программы состоит в том, что изучая программирование в среде Scratch, у обучающихся формируется не только логическое мышление, но и навыки работы с мультимедиа; создаются условия для активного, поискового учения, предоставляются широкие возможности для разнообразного программирования.

**1.5. Цель программы** – развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций, а также пропедевтика будущего изучения программирования на одном из современных языков.

### **1.6. Задачи:**

Обучающие (предметные):

- формирование умения построения различных видов алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- формирование умения использовать инструменты среды Scratch для решения поставленных задач;
- формирование умения построения различных алгоритмов в среде Scratch для решения поставленных задач;
- формирование навыков работы со структурой алгоритма.

Развивающие (метапредметные):

- развитие алгоритмического и логического мышления;
- развитие умений постановки задачи, выделения основных объектов, математические модели задачи;
- развитие умения поиска необходимой учебной информации;
- формирование представления об этапах решения задачи;
- формирование алгоритмического подхода к решению задач;
- формирование ключевых компетенций проектной и исследовательской деятельности;
- формирование мотивации к изучению программирования;
- формирование умения целеполагания;
- формирование умения прогнозировать свои действия и действия других участников группы;
- формирование умения самоконтроля и самокоррекции.

Воспитательные (личностные):

- формирование профессионального самоопределения;
- формирование уважительного отношения к интеллектуальному труду;
- формирование смыслообразования;
- формирование умения работать индивидуально и в группе для решения поставленной задачи;
- формирование трудолюбия, упорства, желания добиваться поставленной цели;
- формирование информационной культуры.

Задачи - это результаты тех конкретных стадий реализации программы, которые поддаются фиксации, детализации и измерению. В этом смысле задачи технологичны, они в совокупности есть некий фундамент, на основе которого и происходит **достижение суммарного качественного сдвига в развитии ребёнка**. Можно сказать, что задачи – это те конкретные пролонгированные результаты реализации программы, суммарным выражением которых и является поставленная цель.

**1.7. Возраст учащихся, которым адресована программа** Программа рассчитана на учащихся в возрасте от 6 до 14 лет, не требует предварительных знаний и входного тестирования.

**1.8. Формы занятий:**

Аудиторные (в т.ч. самостоятельные) занятия.

Занятия проводятся по группам и индивидуально.

**2. Объем программы:**

**2.1. Объем программы** – продолжительность программы 72 часа

**2.2. Срок реализации программы** – 36 недель, 1 год, необходимых для освоения программы. Занятия проводятся в группах до 12 человек.

**2.3. Режим занятий:** Занятия проводятся 1 раз в неделю по 2 академических часа с перерывом 10 минут для проветривания помещения и отдыха обучающихся. Продолжительность одного занятия – 45 минут.

### **3. Планируемые результаты**

Личностные результаты:

- формирование профессионального самоопределения;
- формирование уважительного отношения к интеллектуальному труду;
- формирование смыслообразования.

Познавательные УУД:

- развитие алгоритмического и логического мышления;
- развитие умений постановки задачи, выделения основных объектов, математические модели задачи;
- развитие умения поиска необходимой учебной информации;
- формирование представления об этапах решения задачи;
- формирование алгоритмического подхода к решению задач;
- формирование ключевых компетенций проектной и исследовательской деятельности;
- формирование мотивации к изучению программирования.

Регулятивные УУД:

- формирование умения целеполагания;
- формирование умения прогнозировать свои действия и действия других участников группы;
- формирование умения самоконтроля и самокоррекции.

Коммуникативные УУД:

- формирование умения работать индивидуально и в группе для решения поставленной задачи;
- формирование трудолюбия, упорства, желания добиваться поставленной цели;
- формирование информационной культуры.

Предметные результаты:

- формирование умения построения различных видов алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- формирование умения использовать инструменты среды Scratch для решения поставленных задач;
- формирование умения построения различных алгоритмов в среде Scratch для решения поставленных задач;
- формирование навыков работы со структурой алгоритма.

### **3.2. Способы и формы проверки результатов**

Во время проведения курса предполагается текущий, промежуточный и итоговый контроль. Текущий контроль осуществляется регулярно во время проведения каждого лабораторного занятия. Он заключается в ответе учащихся на контрольные вопросы, демонстрации полученных скриптов в среде Scratch, фронтальных опросов учителем. В тематическом планировании предполагается проведение двух промежуточных контрольных работ. Форма итогового контроля – проектная деятельность. Продукт – презентация и защита собственного проекта. Для оценивания проекта могут быть разработаны специальные оценочные листы. Ниже представлен пример оценочного листа:

Таблица 1 – Лист оценивания проекта

<b>Критерий оценивания</b>	<b>1 группа</b>	<b>--- группа</b>
Актуальность темы		
Соответствие содержания проекта		
Дизайн		
Техническая сложность		
Оригинальность		
Возможность применения проекта в школе		
Наличие соответствующего музыкального сопровождения		
Уровень проработанности проекта		
Итоговое количество баллов		

## **II. СОДЕРЖАНИЕ ПРОГРАММЫ**

### **2.1. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

#### **1-ый год обучения**

<b>№</b>	<b>Наименование разделов и тем</b>	<b>Всего часов</b>	<b>Теоретические занятия</b>	<b>Практические занятия</b>	<i>Форма контроля</i>
1	Введение в программу	2	1	1	Наблюдение, беседа
2	Тема 1 Знакомство со средой Scratch	4	2	2	Беседа, Текущая форма контроля.
3	Тема 2 Линейные алгоритмы	8	2	6	Текущая форма контроля
4	Тема 3 Работа с переменными	8	4	4	Текущая форма контроля
5	Тема 4 Условные алгоритмы	10	2	8	Текущая форма контроля
6	Тема 5 Контрольная работа	2	-	2	Индивидуальный контроль, практическая работа, тестовое задание
7	Тема 6 Циклические алгоритмы	12	4	8	Текущая форма контроля.
8	Тема 7 Работа со списками	8	4	4	Текущая форма контроля

9	Тема 8 Создание подпрограмм	4	2	2	Текущая форма контроля
10	Тема 9 Контрольная работа	2	-	2	Индивидуальный контроль, практическая работа, тестовое задание
11	Тема 10 Разработка индивидуального или группового проекта	10	2	8	Практическая работа.
12	Тема 11 Защита индивидуальных или групповых проектов	2	-	2	Анализ выполненных работ. Защита творческих проектов.
	Итого	72			

## 2.2. СОДЕРЖАНИЕ ТЕМ ПРОГРАММЫ

### Тема 1: Введение в программу

**Теория** - цели и задачи курса. Влияние работы с компьютером на организм человека, его физическое состояние. Правила работы и требования охраны труда при работе на ПК, правила поведения и требований безопасности в кабинете информатики.

**Практика** - Демонстрация возможностей Scratch с помощью готового проекта.

**Формы контроля** - Наблюдение, беседа

### Тема 2: Знакомство со средой Scratch.

**Теория** - изучение основных элементов интерфейса среды Scratch.

**Практика** - приёмы работы со спрайтами, приёмы работы с фоном, составление простых скриптов из различных блоков.

**Формы контроля** – беседа, текущая форма контроля.

### Тема 3: Линейные алгоритмы.

**Теория** - основные приёмы составления линейных алгоритмов в среде Scratch.

**Практика** - решение задач на составление линейных алгоритмов.

**Формы контроля** – Промежуточный.

### Тема 4: Работа с переменными

**Теория** - основные приёмы добавления переменных в среде Scratch, использование основных блоков для работы с переменными, основные приёмы составления программ с использованием переменных в среде Scratch.

**Практика** - решение задач с переменными.

**Формы контроля** – Текущая форма контроля.

**Тема 5:** Условные алгоритмы.

**Теория** - ознакомление с понятием «условный алгоритм», основные приёмы составления условных алгоритмов в среде Scratch.

**Практика** - решение задач на условные алгоритмы, использование основных блоков для составления условных алгоритмов в среде Scratch.

**Формы контроля** – Текущая форма контроля.

**Тема 6:** Контрольная работа.

**Практика:** проверка полученных навыков по темам «Линейные алгоритмы», «Условные алгоритмы», решение задач.

**Формы контроля** – индивидуальный контроль, практическая работа, тестовое задание.

**Тема 6:** Циклические алгоритмы.

**Теория** - ознакомление с понятием «циклический алгоритм», основные приёмы составления циклических алгоритмов в среде Scratch, использование основных блоков для составления циклических алгоритмов в среде Scratch.

**Практика** - решение задач с циклами.

**Формы контроля** – Промежуточный.

**Тема 7:** Работа со списками.

**Теория** - ознакомление с понятием «список» в среде Scratch, создание списка, работа с блоками по обработке списков, основные приёмы составления программ по работе со списками в среде Scratch.

**Практика** - работа с блоками по обработке списков, решение задач со списками.

**Формы контроля** – Текущая форма контроля.

**Тема 8:** Создание подпрограмм.

**Теория** - ознакомление с возможностью создания подпрограмм в среде Scratch. Раздел «Другие блоки», создание блока, параметры блока.

**Практика** - создание блоков-подпрограмм.

**Формы контроля** – Текущая форма контроля.

**Тема 9:** Контрольная работа.

**Практика** - проверка полученных навыков по темам «Циклические алгоритмы», «Работа со списками», решение задач.

**Формы контроля** – индивидуальный контроль, практическая работа, тестовое задание.

**Тема 10:** Разработка индивидуального или группового проекта.

**Практика** - разработка индивидуального или группового проекта в среде Scratch.

**Формы контроля** – Практическая работа, наблюдение.

**Тема 11:** Защита индивидуальных или групповых проектов

**Практика** - защита индивидуальных или групповых проектов.

**Формы контроля** – анализ выполненных работ, защита творческих проектов.

### **III. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ И ФОРМЫ АТТЕСТАЦИИ**

#### **3.1. Методическое обеспечение программы:**

Методы и приемы работы:

- сенсорного восприятия (лекции, просмотр видеофрагментов);
- практические (лабораторные работы, проекты);
- коммуникативные (дискуссии, беседы, ролевые игры);
- комбинированные (самостоятельная работа учащихся);
- проблемный (создание на уроке проблемной ситуации).

**Форма подведения итогов реализации программы:** защита проектов.

Использование метода проектов позволяет обеспечить условия для развития у ребят навыков самостоятельной постановки задач и выбора оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи.

Формы контроля и анализа результатов освоения программы: - обсуждение педагогом и воспитанником результатов выполнения определенных работ и их оценка; - защита проекта на итоговом занятии.

#### **3.2. Условия реализации программы**

Для организации работы центра «ИТ-Куб» в распоряжении «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «ИТ-Куб» от 12.02.2021 рекомендуется следующее оборудование лаборатории:

- ноутбук — рабочее место преподавателя;
- рабочее место обучающегося;
- жёсткая, неотключаемая клавиатура: наличие;
- русская раскладка клавиатуры: наличие;
- диагональ экрана: не менее 15,6 дюймов;
- разрешение экрана: не менее 1920×1080 пикселей;
- количество ядер процессора: не менее 4;
- количество потоков: не менее 8;
- базовая тактовая частота процессора: не менее 1 ГГц;
- максимальная тактовая частота процессора: не менее 2,5 ГГц;
- кэш-память процессора: не менее 6 Мбайт;
- объём установленной оперативной памяти: не менее 8 Гбайт;
- объём поддерживаемой оперативной памяти (для возможности расширения): не менее 24 Гбайт;
- объём накопителя SSD: не менее 240 Гбайт;
- время автономной работы от батареи: не менее 6 часов;
- вес ноутбука с установленным аккумулятором: не более 1,8 кг;
- внешний интерфейс USB стандарта не ниже 3.0: не менее трёх свободных;
- внешний интерфейс LAN (использование переходников не предусмотрено): наличие;
- наличие модулей и интерфейсов (использование переходников не предусмотрено): VGA, HDMI;
- беспроводная связь Wi-Fi: наличие с поддержкой стандарта IEEE 802.11n или современнее;

- веб-камера: наличие;
- манипулятор «мышь»: наличие;
- предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общесистемных приложений: наличие), МФУ, веб-камера, интерактивный моноблочный дисплей, диагональ экрана: не менее 65 дюймов, разрешение экрана: не менее 3840×2160 пикселей, оборудованный напольной стойкой.

В центре «ИТ-Куб» действует несколько лабораторий, в том числе лаборатория для осуществления направления «Основы алгоритмики и логики».

### 3.3. Список литературы

1. Босова Л. Л. Информатика. 8 класс : учебник. / Босова Л. Л. — М.: БИНОМ. Лаборатория знаний, 2016. — 176 с.
2. Винницкий Ю. А. Scratch и Arduino для юных программистов и конструкторов./ Винницкий Ю. А. — СПб.: БХВ-Петербург, 2018. — 176 с.
3. Голиков Д. В. Scratch для юных программистов. / Голиков Д. В. — СПб.: БХВ-Петербург, 2017. — 192 с.
4. Лаборатория юного линуксоида. Введение в Scratch. — <http://younglinux.info/scratch>
5. Маржи М. Scratch для детей. Самоучитель по программированию. / Маржи М. — пер. с англ. М. Гескиной и С. Таскаевой. — М.: Манн, Иванов и Фербер, 2017. — 288 с.
6. Пашковская Ю. В. Творческие задания в среде Scratch: рабочая тетрадь для 5—6 классов. / Пашковская Ю. В. — М., 2018. — 195 с.
7. Первин Ю. А. Методика раннего обучения информатике. / Первин Ю. А. — М.: БИНОМ. Лаборатория знаний, 2008. — 228 с.
8. Поляков К. Ю. Информатика. 7 класс (в 2 частях) : учебник. Ч. 1 / Поляков К. Ю., Еремин Е. А. — М.: БИНОМ. Лаборатория знаний, 2019. — 160 с.
9. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-методическое пособие. / Рындак В. Г., Дженжер В. О., Денисова Л. В. — Оренбург: Оренб. гос. ин-т. менеджмента, 2009. — 116 с.
10. Свейгарт Эл. Программирование для детей. Делай игры и учи язык Scratch! / Свейгарт Эл. — М.: Эксмо, 2017. — 304 с.
11. Семакин, И. Г. Информатика и ИКТ: учебник для 9 класса. / Семакин, И. Г., Залогова, Л. А. и др. М: БИНОМ. Лаборатория знаний, 2014. — 171 с.
12. Торгашева Ю. В. Первая книга юного программиста. Учимся писать программы на Scratch. / Торгашева Ю. В. — СПб.: Питер, 2016. — 128 с.
13. Уфимцева П. Е. Обучение программированию младших школьников в системе дополнительного образования с использованием среды разработки Scratch / Уфимцева П. Е., Рожина И. В. // Наука и перспективы. — 2018. — № 1. — С. 29—35.
14. <https://scratch.mit.edu/> Сообщество Scratch.

**ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

**СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП**

Сертификат 603332450510203670830559428146817986133868575873

Владелец Фибих Мария Олеговна

Действителен с 11.04.2022 по 11.04.2023