

Муниципальное общеобразовательное учреждение
«Средняя общеобразовательная школа имени генерала Захаркина И.Г.»
г. Кременки Жуковского района Калужской области

Принята
педагогическим советом
протокол № 1 от «30» 08 2022 г.

УТВЕРЖДЕНА
приказом № 325
от « 31 » 08 2022 г.

Дополнительная общеобразовательная
общеразвивающая программа
технической направленности
«Программирование и робототехника»

Возраст обучающихся: 14-17 лет

Срок реализации: 1 год

Автор составитель:
Неманов Олег Олегович,
педагог дополнительного
образования

г. Кременки, 2022 г.

ПАСПОРТ ПРОГРАММЫ

Полное название программы	Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование и робототехника»
Автор-составитель программы, должность	Неманов Олег Олегович, педагог дополнительного образования.
Адрес реализации программы	Калужская область. Жуковский район. Г. Кремёнки. МОУ СОШ им. Генерала Захаркина И. Г. Тел. 58820
Вид программы	–по степени авторства (модифицированная) –по форме организации содержания – (модульная) –по уровню освоения – (базовый)
Направленность	Техническая
Тип программы	Общеразвивающая, общеобразовательная
Срок реализации программы	1 год (144 часа)
Возраст детей	14-17 лет
Форма реализации программы	групповая
Форма организации образовательной деятельности	объединение
Название объединения	«Программирование и робототехника»
Педагоги, реализующие программу	Неманов Олег Олегович

РАЗДЕЛ 1.

«КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ»

1.1 Пояснительная записка

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования.

Для школьников этот курс – возможность развить навыки программирования на языке С («Си»). Программирование – это стержень как базового, так и профильного курсов информатики. В рамках предлагаемого курса «Мехатроника и программирование» изучение основ программирования на языке С и мехатроники – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Исключительно велика роль программирования для формирования мышления школьников, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Мехатроника дополнительно к перечисленному способствует развитию пространственного мышления, навыкам конструирования 3х мерных объектов реального мира и сложных взаимосвязанных конструкций. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения по отношению к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования – необходимое условие подготовки современных школьников. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Направленность программы: техническая.

Вид программы: модифицированная, разноуровневая, комплексная.

Отличительные особенности программы: постепенное создание индивидуального проекта через изучение связанных между собой тем от занятия

к занятию; занятия проходят так, что учащемуся сначала предлагается решить проблему самим, а потом объясняются наиболее эффективные методы и приёмы её решения, а так же минусы, выбранного учащимся решения; педагог не разъясняет всё сразу, а действует через наводящие вопросы и подсказки, позволяя прийти к решению самостоятельно.

Новизна программы: использование консервативных, проверенных временем методов, вместо модных и неэффективных.

Педагогическая целесообразность: порядок тем и форма их преподавания формирует у учащегося собственный опыт в решении каждой проблемы, позволяя в процессе решения совершать собственные ошибки, исправлять их и находить правильные пути для достижения поставленной цели; это (метод проб и ошибок) даёт более глубокое понимание изучаемых механизмов и процессов, а участие направляющего воздействия педагога способствует более быстрому течению данного процесса, чем если бы учащийся оставался с проблемой один на один.

Адресат программы: инициативный школьник, интересующийся техническими вопросами, любящий выдвигать интересные и не стандартные предположения и экспериментировать с целью их проверки.

Особенности организации образовательного процесса

Объем программы: 144 часа.

Сроки программы: 1 год.

Режим занятий: 4 ч/н, 144 ч/г.

Формы обучения: очная.

Форма организации образовательной деятельности: групповая.

Методы обучения: словесные, наглядные, практические, проектные.

Формы проведения занятий: теоретические, практические.

Состав группы, особенности набора: постоянный.

1.2. Цель и задачи программы

Цель программы:

Изучение значения алгоритмизации как метода познания окружающего мира, базовых понятий теории алгоритмов; овладение навыками составления и реализации алгоритмов в виде программ, написанных на языке программирования С, конструирования электронно-механических механизмов.

Задачи программы:

- познакомиться с понятиями «алгоритм», «язык программирования»;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования С;
- изучить основные конструкции языка программирования С;
- научиться работать с графическими средствами языка программирования С;
- научиться отлаживать и тестировать программы, делать выводы о работе

этих программ;

- познакомиться с основными электронными компонентами для построения схем;

- научиться проектировать и конструировать электронно-механические устройства; научиться программировать МК.

1.3. Содержание программы

Учебный план и содержание программы

Данная таблица содержит учебный план вместе с кратким содержанием программы.

№ п/п	Наименование разделов и тем и их содержание	Количество часов			Формы аттеста ции/ контро ля
		Всего	Теория	Практ ика	
1	Знакомство, техника безопасности. История UNIX, история Linux, история СПО, устройство ЭВМ, виды ЯП, устройство и виды ОС	2	2		
2	Задача и ее решение. Стадии решения задачи. Алгоритм как решение задачи. Способ записи алгоритмов с помощью псевдокода. Исполнитель алгоритма. Множество допустимых команд исполнителя, детализация алгоритма.	2	2		
3	Практический пример решения задачи. Исполнитель «Тупой художник». Запись алгоритма в нескольких видах (псевдокод, блок-схема). Ошибки в алгоритме и способе его записи. Перевод алгоритма в программу (кодирование). Компьютер как исполнитель. Понятие о языке программирования.	2	1	1	
4	Основные возможности графической библиотеки («команды «тупого художника»).	2	1	1	

	Система координат экрана. Решение задач на составление алгоритмов и их запись в различных формах.				
5	Практика: проект «мультфильм» (линейные алгоритмы и их программирование). Элементарные операции в редакторе. Набор и отладка первой программы. Ошибки кодирования. Воспроизведение алгоритмических ошибок и ошибок кодирования.	2		2	
6	Смысловое разделение частей алгоритма. Концепция процедурного программирования. Функции без параметров. Оформление функций. Вызов функции. Пошаговая отладка в программе с функциями. Именованые функции. Понятие стиля программирования.	2	2		
7	Практика: рефакторинг проекта «мультфильм» с применением функций без параметров.	2		2	
8	Функции с параметрами, возможности, которые они предоставляют. Грамотное использование параметров функции (взаимная независимость). Числовой тип данных. Отладка программы с переменными и параметрами. Понятие о библиотеке функций.	2	2		
9	Практика: рефакторинг проекта «мультфильм» с применением функций с параметрами.	2		2	
10	Повторяющиеся действия в алгоритмах. Разбор циклических алгоритмов. Цикл while. Работа с переменными в цикле. Отладка программы с циклами. Ошибки при работе с циклами.	2	1	1	
11	Практика: доработка проекта «мультфильм» с использованием циклических алгоритмов.	2		2	

12	Мехатроника. Введение.	2	2		
13	Библиотеки функций. Размещение библиотеки в отдельном файле (или множестве файлов) и подключение к главной программе с помощью директивы <code>#include</code> . Оформление библиотеки и написание документации с помощью системы <code>doxygen</code> .	2	1	1	
14	Практика: Создание библиотеки героев с документацией и обмен библиотеками между учащимися. Создание короткого мультфильма с использованием только чужих героев и написание рецензии на используемую библиотеку. Доработка собственной библиотеки, после получения рецензии на неё.	2		2	
15	Работа с мышью. Загрузка и использование изображений. Условные операторы и их использование.	2	1	1	
16	Практика: доработка проекта «мультфильм» до проекта «простая игра».	2		2	
17	Проект «Маячок». Изучаем основы электричества и законов электричества, принципиальных схем. Теория управления электричеством. Мигаем светодиодом.	2	1	1	
18	Задача о функции, изменяющей значения своих параметров (на примере функции движения объекта-шарика). Передача указателей на переменные и работа с ними.	2	1	1	
19	Практика: рефакторинг проекта «игра» с использованием функций, принимающих указатели.	2		2	
20	Проект «Маячок с нарастающей яркостью». Изучаем ШИМ.	2	1	1	
21	Группировка данных, принадлежащих одному объекту, в структуры. Объявление структур, инициализация и	2	1	1	

	использование структурных переменных. Передача структурных переменных в функции. Использование указателей на константные структуры для ускорения передачи параметров.				
22	Практика: рефакторинг проекта «игра» с использованием структур.	2		2	
23	Проект «Светильник с управляемой яркостью». Изучаем работу делителя напряжения и вспоминаем теорию его работы. Учимся работать с потенциометром.	2	1	1	
24	Написание документации к игре и выпуск версии программы. Создание презентации по программе. Технология хорошего доклада.	2	1	1	
25	Практика: написание документации, создание презентации и доклад на уроке-конференции.	2		2	
26	Проект «Терменвокс». Учимся работать с пьезодинамиком.	2	1	1	
27	Построение графиков заданных элементарных функций. Задача о построении графика функции. Алгебраическая система координат. Пересчёт координат из одной координатной системы в другую. Задача об исследовании элементарной функции (интервалы возрастания и убывания, точки экстремума, нарушение области определения, скорость изменения функции).	2	1	1	
28	Практика: проект «графики функций». Изучение элементарных функций с параметрами. Особенности визуального исследования.	2		2	
29	Проект «Ночной светильник». Учимся работать с фоторезистором.	2	1	1	
30	Возврат значений в функции. Функция как «чёрный ящик». Множество значений функции, допустимые и	2	1	1	

	исключительные значения. Понятие надёжности функции. Концепция защитного программирования. Функции для пересчёта координат. Проект «Знакомые графики» с функциями и модулем пересчёта координат.				
31	Практика: рефакторинг проекта «графики» с использованием возврата значений и функциями пересчёта координат.	2		2	
32	Проект «Пульсар». Изучаем устройство и работу биполярного транзистора. Учимся управлять большой нагрузкой с помощью биполярного транзистора.	2	1	1	
33	Кнопочный интерфейс и сложности, связанные с ним. Реализация кнопок с помощью структур. Сменные (skinnable) интерфейсы.	2	1	1	
34	Практика: Разработка кнопочного интерфейса и простого менеджера кнопок.	2		2	
35	Проект «Бегущий огонёк». Учимся работать со светодиодными сборками.	2	1	1	
36	Выбор одной функции из семейства родственных. Способы выбора (нумерация целыми числами, использование enum, использование указателей на функции). Запись типа переменной – указателя на функцию.	2	1	1	
37	Практика: рефакторинги проекта «графики»: с применением enum и с применением указателей на функции.	2		2	
38	Проект «Мерзкое пианино». Учимся работать с тактовыми кнопками. Изучаем эффект дребезга контактов, схемы со стягивающим резистором и подтягивающим резистором.	2	1	1	
39	Графики неравенств (тесты на принадлежность точки заданной	2	1	1	

	области). Дальнейшее использование принципа визуализации.				
40	Практика: проект «графики неравенств». Построение алгоритма «от определения». Вложенные циклы	2		2	
41	Проект «Миксер». Учимся работать с коллекторным двигателем, полевым транзистором и выпрямительным диодом.	2	1	1	
42	Массивы. Алгоритмы работы с массивами (ввод, распечатка). Размер массива. Передача массива в функцию. Ошибка выхода за границы массива и её контроль с помощью assert. Структура «вектор» (содержащая массив и его длину, это пока не std::vector, а очень упрощённая самодельная версия). Разбор задач на алгоритмы работы с массивами.	2	1	1	
43	Практика: Реализация простейших функций для работы с массивами (заполнение массивов, сумма элементов, последовательный поиск элемента).	2		2	
44	Проект «Кнопочный переключатель». Учимся бороться программно с дребезгом контактов. Делаем из тактовой кнопки триггер.	2	1	1	
45	Проблема проверки правильности алгоритмов. Применение автоматического тестирования (юнит-тестирования).	2	1	1	
46	Практика: реализация юнит-тестирования для уже написанных алгоритмов. Работа с файлами. Реализация библиотеки функций для работы с массивами (уже написанные функции + среднее значение, циклические сдвиги вправо-влево, обращение, двоичный поиск и т. д.)	2		2	
47	Проект «Светильник с кнопочным управлением». Учимся управлять яркостью светильника с помощью 2х тактовых кнопок.	2	1	1	

48	Строки как вид массивов. Максимальная и текущая длины строки. Основные алгоритмы над строками (копирование, сравнение, поиск буквы и подстроки, подсчёт слов, удаление подстроки, удаление пробелов, поиск и замена подстроки).	2	1	1	
49	Практика: проект «строковая библиотека».	2		2	
50	Проект «Кнопочные ковбои». Практика использования кнопок и пьезодинамика.	2	1	1	
51	Алгоритмы быстрого поиска (Рабина-Карпа, Бойера-Мура). Для сильных групп – алгоритм Кнута-Морриса-Прата.	2	1	1	
52	Практика: проект «строковая библиотека», реализация алгоритмов быстрого поиска с юнит-тестированием.	2		2	
53	Проект «Секундомер». Учимся использовать семисегментный индикатор(цифра).	2	1	1	
54	Перевод чисел в строковую форму и обратно в разных системах счисления. Представление отрицательных чисел. Сложение и вычитание в различных системах счисления.	2	2		
55	Практика: реализация функций перевода с юнит-тестированием. Использование функций перевода для шифрования текста.	2		2	
56	Проект «Счётчик нажатий». Учимся использовать сдвиговый регистр вместе с семисегментным индикатором.	2	1	1	
57	Матрицы (многомерные массивы) и массивы строк. Основные навыки работы с многомерными массивами. Представление матриц в машине (расположение в оперативной памяти). Работа с динамической памятью.	2	1	1	

58	Практика: решение задач, связанных с матрицами.	2		2	
59	Проект «Комнатный термометр». Учимся использовать термистор.	2	1	1	
60	Сортировка в массивах. Основные виды сортировки. Разбор задач, связанных с сортировкой. Эффективность алгоритма сортировки. Алгоритм сортировки выбором (через анализ человеческих действий).	2	1	1	
61	Практика: реализация сортировки выбором с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности. Сортировка стихотворных строк со сравнением «начиная с начала» и «начиная с конца». Словари рифм.	2		2	
62	Проект «Метеостанция». Учимся передавать данные с МК на компьютер.	2	1	1	
63	Алгоритм сортировки пузырьком (через определение отсортированной последовательности). Эффективность алгоритма сортировки пузырьком. Адаптивный характер алгоритма. Шейкерная сортировка.	2	2		
64	Практика: реализация сортировки «пузырьком» с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности.	2		2	
65	Проект «Пантограф». Учимся использовать конденсатор.	2	1	1	
66	Другие алгоритмы сортировки (простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая). Сравнение эффективности алгоритмов.	2	1	1	
67	Практика: проект «сравнение сортировок». Реализация алгоритмов с	2		2	

	юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Разработка полноценной программы с интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки. Написание документации и создание презентации. Выступление на уроке-конференции с анализом одного из алгоритмов и демонстрацией программы.				
68	Проект «Тестер батареек». Учимся работать с текстовым экраном.	2	1	1	
	Всего	144			

1.4 Планируемые результаты

Часть 1. Язык С

Учащиеся должны знать / понимать:

- назначение и основные команды среды разработки;
- общую структуру программы;
- понятие типа данных и их виды;
- понятие переменной;
- операции с переменными;
- циклы;
- условные выражения;
- массивы;
- структуры(struct);
- назначение и использование библиотек;

Учащиеся должны уметь:

- пользоваться средой разработки;
- составлять и запускать программы;
- организовывать вывод данных;
- использовать переменные;
- правильно определять подходящий тип данных для переменной;
- использовать циклы;
- использовать условные выражения;
- использовать массивы;
- использовать структуры данных;
- создавать и использовать библиотеки.

Часть 2. Мехатроника

Учащиеся должны знать / понимать:

- основы электроники;
- базовые электронные компоненты;
- принципы построения электронных схем;
- устройство МК;

Учащиеся должны уметь:

- выбирать необходимый элемент для использования в схеме;
- проектировать и составлять схемы;
- разрабатывать программу для МК;
- загружать программу в МК.

РАЗДЕЛ № 2
«КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ
УСЛОВИЙ»

2.1 Календарный учебный график

№ учебной недели	Наименование разделов и тем	Кол -во часо в
1	Знакомство, техника безопасности, история UNIX, история Linux, история СПО, устройство ЭВМ, виды ЯП, устройство и виды ОС	2
	Задача и ее решение. Стадии решения задачи. Алгоритм как решение задачи. Способ записи алгоритмов с помощью псевдокода. Исполнитель алгоритма. Множество допустимых команд исполнителя, детализация алгоритма.	2
2	Практический пример решения задачи. Исполнитель «Тупой художник». Запись алгоритма в нескольких видах (псевдо-код, блок-схема). Ошибки в алгоритме и способе его записи. Перевод алгоритма в программу (кодирование). Компьютер как исполнитель. Понятие о языке программирования.	2
	Основные возможности графической библиотеки («команды «тупого художника»). Система координат экрана. Решение задач на составление алгоритмов и их запись в различных формах.	2
3	Практика: проект «мультфильм» (линейные алгоритмы и их программирование). Элементарные операции в редакторе. Набор и отладка первой программы. Ошибки кодирования. Воспроизведение алгоритмических ошибок и ошибок кодирования.	2
	Смысловое разделение частей алгоритма. Концепция процедурного программирования. Функции без параметров. Оформление функций. Вызов функции. Пошаговая отладка в программе с функциями. Именованные функции. Понятие стиля программирования.	2
4	Практика: рефакторинг проекта «мультфильм» с применением функций без параметров.	2
	Функции с параметрами, возможности, которые они предоставляют. Грамотное использование параметров функции (взаимная независимость). Числовой тип данных. Отладка программы с переменными и параметрами. Понятие о библиотеке функций.	2
5	Практика: рефакторинг проекта «мультфильм» с применением функций с параметрами.	2

	Повторяющиеся действия в алгоритмах. Разбор циклических алгоритмов. Цикл while. Работа с переменными в цикле. Отладка программы с циклами. Ошибки при работе с циклами.	2
6	Практика: доработка проекта «мультфильм» с использованием циклических алгоритмов.	2
	Мехатроника. Введение.	2
7	Библиотеки функций. Размещение библиотеки в отдельном файле (или множестве файлов) и подключение к главной программе с помощью директивы include. Оформление библиотеки и написание документации с помощью системы doxygen.	2
	Практика: Создание библиотеки героев с документацией и обмен библиотеками между учащимися. Создание короткого мультфильма с использованием только чужих героев и написание рецензии на используемую библиотеку. Доработка собственной библиотеки, после получения рецензии на нее.	2
8	Работа с мышью. Загрузка и использование изображений. Условные операторы и их использование.	2
	Практика: доработка проекта «мультфильм» до проекта «простая игра».	2
9	Проект «Маячок».	2
	Задача о функции, изменяющей значения своих параметров (на примере функции движения объекта-шарика). Передача указателей на переменные и работа с ними.	2
10	Практика: рефакторинг проекта «игра» с использованием функций, принимающих указатели.	2
	Проект «Маячок с нарастающей яркостью».	2
11	Группировка данных, принадлежащих одному объекту, в структуры. Объявление структур, инициализация и использование структурных переменных. Передача структурных переменных в функции. Использование указателей на константные структуры для ускорения передачи параметров.	2
	Практика: рефакторинг проекта «игра» с использованием структур.	2
	Проект «Светильник с управляемой яркостью».	2
12	Написание документации к игре и выпуск версии программы. Создание презентации по программе. Технология хорошего доклада.	2
	Практика: написание документации, создание презентации и доклад на уроке-конференции.	2
	Проект «Терменвокс».	2
13	Построение графиков заданных элементарных функций. За-	2

	дача о построении графика функции. Алгебраическая система координат. Пересчет координат из одной координатной системы в другую. Задача об исследовании элементарной функции (интервалы возрастания и убывания, точки экстремума, нарушение области определения, скорость изменения функции).	
	Практика: проект «графики функций». Изучение элементарных функций с параметрами. Особенности визуального исследования.	2
14	Проект «Ночной светильник».	2
	Возврат значений в функции. Функция как «черный ящик». Множество значений функции, допустимые и исключительные значения. Понятие надежности функции. Концепция защитного программирования. Функции для пересчета координат. Проект «Знакомые графики» с функциями и модулем пересчета координат.	2
15	Практика: рефакторинг проекта «графики» с использованием возврата значений и функциями пересчета координат.	2
	Проект «Пульсар».	2
16	Кнопочный интерфейс и сложности, связанные с ним. Реализация кнопок с помощью структур. Сменные (skinnable) интерфейсы.	2
	Практика: Разработка кнопочного интерфейса и простого менеджера кнопок.	2
17	Проект «Бегущий огонёк».	2
	Выбор одной функции из семейства родственных. Способы выбора (нумерация целыми числами, использование enum, использование указателей на функции). Запись типа переменной – указателя на функцию.	2
18	Практика: рефакторинги проекта «графики»: с применением enum и с применением указателей на функции.	2
	Проект «Мерзкое пианино».	2
19	Графики неравенств (тесты на принадлежность точки заданной области). Дальнейшее использование принципа визуализации.	2
	Практика: проект «графики неравенств». Построение алгоритма «от определения». Вложенные циклы	2
20	Проект «Миксер».	2
	Массивы. Алгоритмы работы с массивами (ввод, распечатка). Размер массива. Передача массива в функцию. Ошибка выхода за границы массива и ее контроль с помощью assert. Структура «вектор» (содержащая массив и его длину, это пока не std::vector, а очень упрощенная самодельная версия). Разбор задач на алгоритмы работы с массивами.	2

21	Практика: Реализация простейших функций для работы с массивами (заполнение массивов, сумма элементов, последовательный поиск элемента).	2
	Проект «Кнопочный переключатель».	2
22	Проблема проверки правильности алгоритмов. Применение автоматического тестирования (юнит-тестирования).	2
	Практика: реализация юнит-тестирования для уже написанных алгоритмов. Работа с файлами. Реализация библиотеки функций для работы с массивами (уже написанные функции + среднее значение, циклические сдвиги вправо-влево, обращение, двоичный поиск и т. д.)	2
23	Проект «Светильник с кнопочным управлением».	2
	Строки как вид массивов. Максимальная и текущая длины строки. Основные алгоритмы над строками (копирование, сравнение, поиск буквы и подстроки, подсчет слов, удаление подстроки, удаление пробелов, поиск и замена подстроки).	2
24	Практика: проект «строковая библиотека».	2
	Проект «Кнопочные ковбой».	2
25	Алгоритмы быстрого поиска (Рабина-Карпа, Бойера-Мура). Для сильных групп – алгоритм Кнута-Морриса-Прата.	2
	Практика: проект «строковая библиотека», реализация алгоритмов быстрого поиска с юнит-тестированием.	2
26	Проект «Секундомер».	2
	Перевод чисел в строковую форму и обратно в разных системах счисления. Представление отрицательных чисел. Сложение и вычитание в различных системах счисления.	2
27	Практика: реализация функций перевода с юнит-тестированием. Использование функций перевода для шифрования текста.	2
	Проект «Счётчик нажатий».	2
28	Матрицы (многомерные массивы) и массивы строк. Основные навыки работы с многомерными массивами. Представление матриц в машине (расположение в оперативной памяти). Работа с динамической памятью.	2
	Практика: решение задач, связанных с матрицами.	2
	Проект «Комнатный термометр».	2
29	Сортировка в массивах. Основные виды сортировки. Разбор задач, связанных с сортировкой. Эффективность алгоритма сортировки. Алгоритм сортировки выбором (через анализ человеческих действий).	2
	Практика: реализация сортировки выбором с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности. Сортировка стихотворных строк со	2

	сравнением «начиная с начала» и «начиная с конца». Словари рифм.	
30	Проект «Метеостанция».	2
	Алгоритм сортировки пузырьком (через определение отсортированной последовательности). Эффективность алгоритма сортировки пузырьком. Адаптивный характер алгоритма. Шейкерная сортировка.	2
31	Практика: реализация сортировки «пузырьком» с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности.	2
	Проект «Пантограф».	2
32	Другие алгоритмы сортировки (простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая). Сравнение эффективности алгоритмов.	2
	Практика: проект «сравнение сортировок». Реализация алгоритмов с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Разработка полноценной программы с интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки. Написание документации и создание презентации. Выступление на уроке-конференции с анализом одного из алгоритмов и демонстрацией программы.	2
33-35	Проект «Тестер батареек».	2
	Экскурсии	
	Всего	144

2.2 Условия реализации программы

Аппаратное обеспечение:

1. IBM PC-совместимый компьютер.
2. Процессор не ниже Pentium Core i3.
3. Оперативная память не меньше 4 Гб.

Программное обеспечение:

1. Операционная система: linux-based.
2. Среда разработки: окружение ОС.

2.3 Формы аттестации (контроля)

Время проведения	Цель проведения	Форма контроля
Начальный или входной контроль		
В начале учебного года	Определение уровня развития учащихся, их способностей	Беседа, опрос
Текущий контроль		
В течение всего учебного года	Определение степени усвоения обучающимися учебного материала. Определение готовности детей к восприятию нового материала. Повышение ответственности и заинтересованности обучающихся в обучении. Выявление отстающих и опережающих обучение. Подбор наиболее эффективных методов и средств обучения	Педагогическое наблюдение, опрос, контрольное занятие. Самостоятельная работа.
Итоговый контроль		
В конце учебного года или курса	Определение изменения уровня развития детей, их творческих способностей. Определение результатов обучения. Ориентирование обучающихся на дальнейшее обучение. Получение сведений для совершенствования образовательной программы и методов обучения.	Выставка, конкурс, творческая работа, опрос, презентация творческих работ, демонстрация моделей

2.4 Оценочные материалы

В конце курса оценивается качество выполненного индивидуального проекта учащегося, выбранного им изначально. Оценка проекта производится как внешне - путём запуска и использования программы/механизма учащегося по её назначению; так и внутренне - путём изучения исходного кода программы и/или внутреннего устройства механизма.

2.5 Методические материалы

В качестве методических материалов необходимо использовать материалы Дединского И.Р., которые можно загрузить с его сайта в сети Интернет - <http://ded32.net.ru/> .

Список литературы

Для педагога:

1. <http://ded32.net.ru/>
2. <https://www.opennet.ru/docs/RUS/ansi-c/>
3. Электроника шаг за шагом, Р.А.Сворень, Издание 4, 2001
4. Электроника для начинающих, Чарльз Платт, Издание 2, 2017

Для учащегося:

1. <https://www.opennet.ru/docs/RUS/ansi-c/>
2. <http://c.kremenki.su>
3. <https://kpolyakov.spb.ru/school/c.htm>
4. Электроника шаг за шагом, Р.А.Сворень, Издание 4, 2001
5. Электроника для начинающих, Чарльз Платт, Издание 2, 2017
6. <http://wiki.amperka.ru>
7. [Роботехника для детей и родителей, С. А. Филиппов](#)