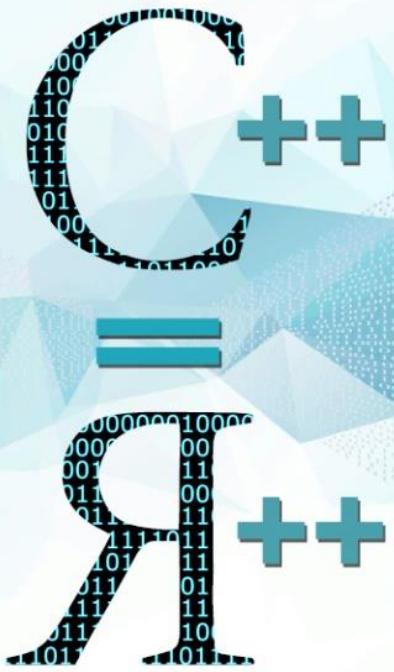
СЕТЬ ЦЕНТРОВ ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ «IT-КУБ»

IT-Cube.Миасс



Муниципальное автономное учреждение
дополнительного образования
«Дом детского творчества «Юность» имени академика В.П.Макеева»
(МАУ ДО «ДДТ «Юность» им. В.П. Макеева»)

Принята на заседании
Методического совета
«25» августа 2021 г.
Протокол № 1

Утверждена:
Директор МАУ ДО «ДДТ
«Юность» им. В.П. Макеева»
_____ /Темур Л.В./
«25» 08 2021 г.

АКЦЕПТОВАНА «ФОНДОМ НОВЫХ ФОРМ РАЗВИТИЯ ОБРАЗОВАНИЯ»

IT-направление «Базовые навыки программирования на С-подобных языках»

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

Возраст обучающихся:10-14 лет
Срок реализации программы: 1 год
216 часов

Автор-составитель:
Ефремова Ирина Александровна
педагог дополнительного образования

Миасс, 2021

Содержание

Раздел I. Комплекс основных характеристик программы

1.1 Пояснительная записка 3

1.2 Учебный план 7

1.3 Планируемые результаты 10

Раздел II. Комплекс основных характеристик программы

2.1 Методическое обеспечение 10

2.2 Список литературы 14

Календарный учебный график 17

Адаптированный учебный план для детей с ОВЗ 21

Дистанционный модуль 25

Летний модуль 27

Методические материалы 29

Оценочный лист освоения программы 45

Оценочный лист творческих проектов 46

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «С++=Я++» (далее - Программа) имеет техническую направленность и составлена на основании нормативных документов федерального и регионального уровней, а также на основании нормативных актов МАУ ДО «ДДТ «Юность» им. В.П.Макеева».

Создание Программы обосновано потребностью современного российского общества в высококвалифицированных специалистах в области программирования.

В наше время компьютеры стали настолько распространенными, что используются практически повсеместно, а программирование является востребованным во всех сферах деятельности человека. Количество компьютерных программ и игр с каждым днем увеличивается, но при этом растет и уровень этих приложений. Запросы и развитие современного общества требуют не только увеличения технического качества приложений, но и их интеллектуального и творческого содержания.

Программа направлена на:

- воспитание и обучение творческих, амбициозных, креативных подростков, уже ориентированных на будущую профессию;
- развитие творческой и научно-технической базы у обучающихся для их дальнейшего личностного роста, как в учебном плане, так и в социальном.

Программа способствует развитию у детей аналитического, логического и творческого мышления, а также их познавательных, исследовательских и коммуникативных способностей.

В ходе обучения дети под руководством педагога будут учиться создавать различные виды действующего и актуального программного обеспечения, что позволит им подготовиться к современным реальным жизненным требованиям.

Отличительная особенность

Программа построена таким образом, что обучающимся с первого занятия будет представлена возможность реализовать свои теоретические знания на практике и постепенно сформировать навыки правильного программирования от простого к сложному. Таким образом обучающиеся привыкают к правилам синтаксиса языка и используемым конструкциям. Комбинированная форма занятий, когда теоретический материал сопровождается примерами и последовательно закрепляется на практике, позволяет обучающимся не утратить интерес к занятиям.

Для закрепления теоретического материала и демонстрации практического применения полученных знаний обучающимся будет предложена возможность запрограммировать микроконтроллеры конструкторов на базе Arduino и ТРИК.

Программа дает возможность получения конкретного наглядного результата (продукта) обучения, что особенно важно для детей подросткового возраста.

Направленность Программы помогает в создании и изобретении современных и актуальных программ и роботизированных устройств, которые смогут не только использоваться в частной, повседневной жизни обучающихся ребят, но и стать основой для серьезного программного обеспечения. Это позволяет создать условия для формирования у обучающихся компетенций будущего и подготовить к выбору профессий технической направленности для предприятий города. Программа содержит адаптированный учебный план для детей с ОВЗ, дистанционный модуль и летний модуль.

Педагогическая целесообразность

Основой Программы является проектная деятельность, состоящая из индивидуальной (под руководством педагога-наставника) и групповой работы по созданию и разработке алгоритмов различной сложности и написания программ, на основе этих алгоритмов. Это создает условия для

развития личностных качеств и умений, необходимых современному человеку: логическое, системное и творческое мышление, умение работать самостоятельно и в команде.

В процессе работы над программными проектами обучающиеся получают дополнительные знания из различных научных и технических областей. В программе делается упор на межпредметные связи, которые расширяют возможности для разработки программных продуктов.

Персонализация образовательного маршрута осуществляется на основе изобретения и создания программ различной сложности, исходя из уровня подготовки обучающихся.

Цель и задачи программы

Цель: обучить основным принципам и этапам программирования и разработки программного обеспечения на основе языка программирования С++ для последующего решения поставленных технических задач, а также реализации творческих проектов.

Задачи:

Образовательные:

- сформировать умение осуществлять алгоритмизацию задачи через освоение различных методов решения задач на языке программирования С++;
- обучить практическому применению основных конструкций языка программирования С++ путем разработки и написания программ различной направленности и сложности;
- обучить практическому применению полученных знаний для программирования конструкторов на базе Arduino и ТРИК;
- способствовать овладению базовых понятий и терминов английского языка технической направленности, необходимых в программировании.

Развивающие:

- развить память, внимание, аналитическое и логическое мышление;
- сформировать навыки проектной и презентационной деятельности;
- развить интеллектуальные, технические и творческие способности обучающихся;
- сформировать навыки самообучения.

Воспитательные:

- сформировать устойчивый интерес к занятиям технического и математического направления;
- сформировать способность к самоорганизации и самоконтролю;
- воспитать терпение, способность преодолевать трудности;
- повысить коммуникативную культуру обучающихся, культуру общения, взаимопонимания, взаимопомощи.

Организация образовательного процесса

Программа адресована обучающимся от 10 до 14 лет и предполагает наличие у обучающихся школьных знаний в области математики и склонность к алгоритмическому мышлению, а также заинтересованности в области программирования и исследовательской деятельности.

Количество обучающихся в группе 12 человек. Объем программы: 216 часов.

Режим занятий: 216 часов в год (2 занятия в неделю длительностью 3 академических часа с двумя 10-минутными перерывами).

Срок реализации программы – 1 год.

Форма обучения: очная с использованием дистанционных технологий.

Уровень освоения программы базовый.

Язык освоения программы: русский.

Программа состоит из двух модулей: базовый модуль (108 часов), углубленный модуль (108 часов).

Для достижения высокого уровня освоения программы необходимо, помимо урочной

деятельности, заниматься домашней самостоятельной работой для закрепления полученных знаний, а также для разработки и написания индивидуального творческого проекта.

Планируемые результаты

Личностные:

- научатся ставить перед собой задачи и решать их, используя имеющиеся знания и овладевая новыми;
- сформируют навыки самообучения и личностного роста;
- научатся самостоятельно организовывать и контролировать свою деятельность;
- повысят уровень развития памяти, внимания, аналитического и логического мышления;
- сформируют устойчивый интерес и стремление к продолжению обучения программированию.

Метапредметные результаты:

- получат навыки самостоятельной работы с технической и научной документацией (мануалы и описания используемых компонент и библиотек);
- приобретут новые знания по математическим предметам и закрепят уже имеющиеся;
- узнают основные термины и понятия английского языка технической направленности, необходимые для программирования;
- приобретут навыки осуществления проектной и презентационной деятельности;
- научатся продуктивно взаимодействовать в паре, в группе, в коллективе на основе взаимопонимания и взаимопомощи.

Предметные результаты:

- сформируют и закрепят интерес к профессиям, связанным с программированием;
- сформируют представление о роли компьютерного программного обеспечения, его возможностях и задачах;
- научатся алгоритмизировать поставленную задачу и реализовывать разработанные алгоритмы на языке программирования С++;
- выучат основные принципы программирования и разработки программного обеспечения на языке программирования С++;
- реализуют полученные знания с помощью программирования микроконтроллеров для конструкторов на базе Arduino и ТРИК;
- создадут индивидуальный проект в виде программного обеспечения для персонального компьютера или роботизированной системы на основе конструкторов на базе Arduino и ТРИК.

Учебный план

№ п/п	Наименование разделов и тем	Общее кол-во часов	В том числе:		Формы контроля
			теоретических	практических	
1. *Базовый модуль*					
1.	1.1. Знакомство с С++	27	9	18	Входное тестирование Контрольные задания и тесты
2.	1.2 Простейшие конструкции языка С++	30	10	20	Контрольные задания и тесты
3.	1.3 Составные типы данных языка С++. Введение в	18	6	12	Контрольные задания и тесты

	робототехнику				
4.	1.4 Функции в языке С++. Работа с программируемыми контроллерами. Создание интерактивной инсталляции «Умный город»	33	11	22	Контрольные задания и тесты
		108	36	72	
2. *Углубленный модуль*					
5.	2.1. Работа с динамической памятью	21	7	14	Контрольные задания и тесты
6.	2.2. Основы объектно-ориентированного программирования	57	19	38	Контрольные задания и тесты
7.	2.3. Стандартные библиотеки языка С++	30	10	20	Контрольные задания и тесты
		108	36	54	
	Итого:	**216**	**77**	**139**	

Содержание учебного плана

Базовый модуль

Раздел №1. Знакомство с С++.

Теория: знакомство с планом работы, целями и задачами программы. Правила техники безопасности и противопожарной защиты. Основы синтаксиса языка программирования С++. Основные операторы, типы данных.

Практика: входное тестирование, написание и запуск первых программ.

Раздел №2. Простейшие конструкции языка С++.

Теория: операторы отношений и логические операторы, управляющие конструкции, циклы, массивы.

Практика: создание простых программ, включающих в себя взаимодействие с пользователем, использование различных типов данных языка, циклические структуры и условные структуры, приемы работы с массивами.

Робототехника: знакомство с конструктором, основными деталями и принципами крепления. Стандартные конструкции роботов.

Обсуждение темы индивидуального итогового проекта.

Раздел №3. Составные типы данных языка С++. Введение в робототехнику

Теория: составные типы данных, структуры, перечисления, а также базовые алгоритмы работы с этими структурами.

Практика: создание программ с использованием изученных структур языка. Реализация изученных базовых алгоритмов на практике.

Робототехника: Изучение датчиков, сенсоров. Создание простейших механизмов.

Обсуждение и выбор способа реализации индивидуального итогового проекта.

Раздел №4. Функции в языке С++. Работа с программируемыми контроллерами.

Теория: принципы и правила использования функций в языке С++.

Практика: написание собственных функций и использование их в коде программы.

Робототехника: работа с конструкторами «Робоняша» и «Йодо» (разработчик «Амперка»). Сборка модели по технологическим картам. Создание интерактивной инсталляции «Умный город».

Разработка алгоритма индивидуального итогового проекта.

Углубленный модуль

Раздел №1. Работа с динамической памятью.

Теория: правила работы с динамической памятью в языке С++.

Практика: закрепление изученного материала на практике - использование динамической памяти в программах.

Робототехника: основы работы с программируемыми микроконтроллерами. Использование встроенных возможностей микроконтроллера.

Начало разработки индивидуального итогового проекта.

Раздел №2. Основы объектно-ориентированного программирования.

Теория: основы объектно-ориентированного программирования.

Практика: создание классов в программах. Реализация различных видов конструкторов, деструктора и методов класса, принципов наследования, инкапсуляции и полиморфизма.

Робототехника: разработка алгоритмов поведения роботизированной системы. Создание программ на основе этих алгоритмов. Программирование микроконтроллера.

Обсуждение вопросов по реализации индивидуального итогового проекта.

Раздел №3. Стандартные библиотеки языка С++.

Теория: изучение возможностей стандартных библиотек языка С++.

Практика: использование на практике методов стандартных библиотек языка С++.

Робототехника: создание роботизированных механизмов на базе микропроцессора TRIK, Arduino.

Защита итоговых индивидуальных проектов.

Ожидаемые результаты программы

В результате реализации программы, обучающиеся будут

Знать:

- основные типы данных языка С++;
- условные и циклические конструкции, их виды и возможности;
- основные структуры языка, правила их использования в программе;
- правила написания функций и их использования в программе;
- возможности работы с динамической памятью;
- принципы объектно-ориентированного программирования и их реализацию на практике;
- принципы создания программного обеспечения для персонального компьютера;
- основные виды микроконтроллеров и способы их программирования;
- основные понятия и термины английского языка технической направленности, необходимые в программировании.

Уметь:

- разрабатывать алгоритм решения поставленной задачи и реализовывать его с помощью языка С++;
- собирать роботизированную систему из конструкторов на базе Arduino и ТРИК;
- ставить перед собой задачи и решать их, используя имеющиеся знания и овладевая

новыми;

- самостоятельно организовывать и контролировать свою деятельность;
- самостоятельно работать с технической и научной документацией (мануалы и описания используемых компонент и библиотек);
- продуктивно взаимодействовать в паре, в группе, в коллективе.

Методическое обеспечение программы

Примеры базовых кейсов и конспектов преподавателя представлены в Приложении 6 данного документа.

В Программе используются следующие педагогические технологии:

- технология развивающего обучения;
- личностно-ориентированная технология;
- технология проблемного обучения;
- технология проектной деятельности.

В Программе применяются следующие методы и приемы обучения:

- информационный рассказ, для изложения теоретического материала по теме;
- беседа, для создания условий для обмена мнениями;
- демонстрация наглядного материала, для применения полученных знаний по программированию в робототехнике;
- изучение источников, для формирования навыков самостоятельной работы с документацией по программированию;
- наставничество, для помощи в решении возникающих проблем у обучающихся;
- индивидуальная работа, для закрепления учащимися полученных знаний.
- кейс-метод;
- мозговой штурм;
- датаскаутинг.

В связи со спецификой теоретической и практической деятельности обучающихся и преобладанием практических занятий используются следующие формы организации деятельности:

- учебно-практическое — обучающиеся слушают информационный рассказ по теме занятия, который одновременно подкрепляется демонстрацией и обсуждением практических примеров.
- практическое – выполнение обучающимися практических заданий для закрепления знаний по теме занятия. Занятия по робототехнике. Работа над индивидуальными творческими проектами;
- Workshop (рабочая мастерская — групповая работа, где все участники активны и самостоятельны);
- контрольная — выполнение обучающимися контрольных тестов и заданий с целью определения уровня усвояемости изученного материала.
- итоговая — защита итогового проекта.

Педагогический контроль

Входной контроль проводится в начале учебного года, для выявления уровня знаний, умений и навыков у обучающихся. Задания для входного контроля представлены в Приложении 1.

Промежуточный контроль проводится после каждого пройденного раздела по выполнению практических заданий, результаты контроля заносятся в оценочный лист (Приложение 8).

Информация, полученная в ходе промежуточного контроля, позволяет оценить степень усвояемости информации, в случае пробелов в знаниях уделить время на дополнительные разъяснения и работу.

Итоговый контроль заключается в защите индивидуального творческого проекта, и проводится в конце учебного года (Приложение 9).

Оценочные материал

В процессе занятий результаты промежуточного контроля учащихся будут отражаться в оценочном листе (Приложение 8).

После каждого пройденного раздела преподаватель оценивает:

- работу на уроке, участие в обсуждениях, дискуссиях;
- самостоятельную работу на практических занятиях и внеурочно;
- качество выполнения практических заданий;
- выполнение контрольных заданий и тестов.

Каждый критерий оценивается по бальной системе от 1 до 5, где:

- 1 — работа не выполнена;
- 2 — работа выполнена частично и не соответствует поставленному заданию;
- 3 — работа соответствует поставленному заданию, но выполнена частично;
- 4 — работа выполнена полностью, но есть небольшие недочеты;
- 5 — работа выполнена полностью и соответствует тематике поставленной задачи.

Баллы за каждый критерий суммируются и получается общий итоговый балл, по которому производится оценка усвояемости пройденного материала.

Итоговый балл:

- 10 и меньше соответствует низкому уровню освоения программы;
- 11-15 – средний уровень освоения программы;
- 16 и выше – высокий уровень освоения программы.

Оценка итогового индивидуального творческого проекта также проводится по нескольким факторам, которые отражаются в оценочном листе итогового проекта (Приложение 9). Оценочные критерии:

- соответствие проекта заданию;
- оригинальность идеи и содержания;
- сложность проекта;
- качество исполнения проекта;
- качество презентации проекта.

Каждый критерий оценивается по бальной системе от 1 до 5, где:

- 1 — критерий не выполнен;
- 2 — критерий выполнен менее, чем наполовину;
- 3 — критерий выполнен наполовину;
- 4 — критерий выполнен, но есть небольшие недостатки;
- 5 — критерий выполнен полностью.

Баллы за каждый критерий суммируются и получается общий итоговый балл, по которому формируется оценка за итоговый проект.

Итоговый балл:

- 10 и меньше — неудовлетворительное выполнение итогового проекта;
- 11-20 – хорошее выполнение итогового проекта;
- 21 и выше – отличное выполнение итогового проекта.

Все учащиеся по окончанию обучения будут поощряться грамотами согласно достигнутого результата.

Материально-техническое обеспечение

- компьютеры с установленной операционной системой Windows для каждого обучающегося и для педагога;
- среда разработки Eclipse IDE for C/C++ Developers;
- локальная сеть, в которую объединены все компьютеры в кабинете;
- доступ в Интернет;
- 3 набора конструкторов с непрограммируемыми микроконтроллерами;
- набор конструктора «Роботрек» с программируемым микроконтроллером уровня «Стажер А»;
- ресурсный набор конструктора «Трекдуино» с программируемым микроконтроллером;
- ресурсный набор конструктора «ТРИК» с программируемым микроконтроллером.

Список литературы и интернет-источников

1. Сайт программы:
https://it-cube.ddt-miass.ru/?page_id=550

2. Абельсон Х., Сассман Дж., Сассман Дж. Структура и интерпретация компьютерных программ. – Изд. КДУ, 2012 г.

3. Бьярне Стауструп. Программирование: принципы и практика использования C++. – Изд. Вильямс, 20016 г.

4. Брюс Эккель, Чак Эллисон. Философия C++. Практическое программирование. – Изд. Питер, 2004 г.

5. Бхаргава Адитья. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих – Изд. Питер, 2019 г.

6. Герберт Шилдт. C++: базовый курс. – Изд. Вильямс, 2008 г.

7. Стивен Прата. Язык программирования C++. Лекции и упражнения. – Изд. Вильямс, 2012 г.

8. Стив Макконнелл. Совершенный код. – Изд. Русская Редакция, Microsoft Press, 2017 г.

Приложение 1
к дополнительной общеобразовательной
общеразвивающей программе

Вопросы и задания для проведения входного контроля

1. Расскажите немного о себе:

- есть ли дома собственный компьютер/ноутбук?
- какое программное обеспечение на нем стоит?
- какие языки программирования ты уже знаешь?
- как относятся родители к твоему увлечению программированием?

2. Что такое двоичная система счисления и пример ее использования?

Ответ: Двоичная система счисления — система, основанная на степенях числа 2, в которой для записи числа используются только цифры 0 и 1. Двоичная система используется практически во всех современных компьютерах и прочих вычислительных электронных устройствах.

3. В классе учится 29 человек, среди которых 17 мальчиков. Известно, что 12 человек из класса являются отличниками, а 7 девочек не являются отличницами. Сколько мальчиков не являются отличниками?

Ответ: 7 мальчиков

4. Определите значение переменной d после выполнения следующего фрагмента программы (все переменные целого типа):

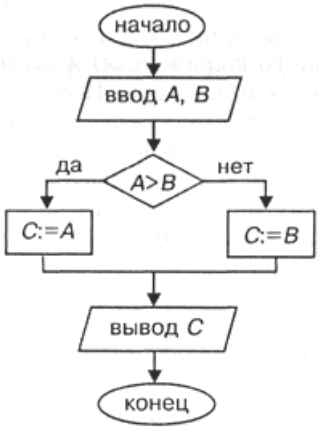
Паскаль	Си

a := 10;
b := 6;
a := a + b;
b := a mod 3;
d := a - b;
	a = 10;
	b = 6;
	a = a + b;
	b = a % 3;
	d = a - b;

Ответ:15

5. Даны два числа, запишите алгоритм их сортировки по возрастанию. Желательно в виде блок-схемы, но можно и в виде текста по пунктам.

Ответ:



Приложение 2
к дополнительной общеобразовательной
общеразвивающей программе

Дополнительная общеобразовательная общеразвивающая программа «С++=Я++» имеет техническую направленность и составлена в соответствии с:

1. Закона РФ «Об образовании в Российской Федерации» (№273-ФЗ от 29.12.2012);
2. Федеральный закон от 02.07.2021 № 320-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации»;
3. Концепции развития дополнительного образования детей (утвержденная распоряжением Правительства РФ от 04.09.2014 г. №1726-р);
4. Приказа Министерства просвещения РФ от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
5. Постановления Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
6. Постановление Главного государственного санитарного врача Российской Федерации от 24.03.2021 № 10 "О внесении изменений в санитарно-эпидемиологические правила СП 3.1/2.4.3598-20 "Санитарно-эпидемиологические требования к устройству, содержанию и организации работы образовательных организаций и других объектов социальной инфраструктуры для детей и молодежи в условиях распространения новой коронавирусной инфекции (COVID-19)", утвержденные постановлением Главного государственного санитарного врача Российской Федерации от 30.06.2020 № 16";
7. Распоряжение Минпросвещения России от 25.12.2019 N Р-145 "Об утверждении методологии (целевой модели) наставничества обучающихся для организаций, осуществляющих образовательную деятельность по общеобразовательным, дополнительным общеобразовательным и программам среднего профессионального образования, в том числе с применением лучших практик обмена опытом между обучающимися";
8. Приказ Министерства просвещения Российской Федерации от 30.09.2020 № 533 "О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам", утвержденный приказом Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196;
9. Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 "Об утверждении Целевой модели развития региональных систем дополнительного образования детей";
10. Приказ Министерства образования и науки РФ от 23 августа 2017 г. N 816 "Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ";

11. Методических рекомендаций по проектированию дополнительных общеразвивающих программ, разработанных Министерством образования и науки России совместно с ГАОУ ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования» и АНО дополнительного профессионального образования «Открытое образование» (письмо Министерства образования и науки России от 18.11.2015

№ 09-3242 «О направлении информации»);
12. Стратегии развития воспитания в Российской Федерации на период до 2025 года (Распоряжение Правительства РФ от 25 мая 2015г. № 996-р);
13. Закона Челябинской области от 29.08.2013 года № 515-ЗО «Об образовании в Челябинской области»;
14. Локальных нормативно-правовых актов актов Муниципального автономного учреждения дополнительного образования «Дом детского творчества «Юность» имени академика В.П. Макеева».

Приложение 3
к дополнительной общеобразовательной общеразвивающей
программе

Календарный учебный график на 2021-2022 учебный год (1 группа, 1 год обучения) по программе «С++=Я++»
Место проведения занятий: МАУ ДО «ДДТ «Юность» им. В.П. Макеева» г. Миасс, пр. Макеева, 39

№ п\п	Число, месяц	Форма занятия	Кол-во часов	Тема занятия	Форма контроля
Базовый модуль					
Первый раздел: «Знакомство с С++»					
1.	сентябрь	учебно-практическое	3	Вводное занятие. Техника безопасности. Знакомство учащихся с образовательной программой, план работы объединения. Знакомство с языком программирования, история, возможности, область применения. Входное тестирование.	Входное тестирование
2.	сентябрь	учебно-практическое	3	Структура кода на С++. Обзор простых элементов языка. Процесс сборки программы. Среда программирования Eclipse. Написание и запуск первой программы.	наблюдение
3.	сентябрь	учебно-практическое	3	Библиотека iostream. Взаимодействие программы с пользователем.	практическое задание, опрос
4.	сентябрь	учебно-практическое	3	Представление данных в памяти. Системы счисления. Двоичная система счисления. Представление разных типов данных в двоичной системе. Представление данных в памяти. Системы счисления. Двоичная система счисления.	опрос
5.	сентябрь	учебно-практическое	3	Примитивные типы данных. Литералы. Модификаторы типов.	практическое задание
6.	сентябрь	учебно-практическое	3	Преобразование типов. Арифметические операции.	практическое задание
7.	сентябрь	учебно-практическое	3	Отладка программ. Точки останова, стек вызовов, отслеживание переменных.	опрос
8.	сентябрь	учебно-	3	Базовые принципы тестирования программы.	практическое

		практическое			задание
9.	октябрь	учебно-практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
Второй раздел: «Простейшие конструкции языка С++».					
10.	октябрь	учебно-практическое	3	Операторы отношений и логические операторы. Инструкции выбора: if, else, switch. Тернарный оператор.	опрос
11.	октябрь	учебно-практическое	3	Циклы: for, while-do, do-while, break, continue.	практическое задание
12.	октябрь	учебно-практическое	3	Знакомство с робототехникой и средой программирования Robotrack IDE.	практическое задание
13.	октябрь	учебно-практическое	3	Введение в робототехнику. Работа с технической документацией. Принципы конструирования и правила работы с набором «Стажер».	наблюдение
14.	октябрь	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
15.	октябрь	контрольное	3	Составные типы данных. Массивы. Основные правила и приемы работы с массивами. Строки как пример массива.	Контрольный тест/ задание
16.	октябрь	практическое	3	Многомерные массивы.	самостоятельная работа
17.	октябрь	практическое	3	Введение в робототехнику. Работа с технической документацией. Создание конструкции робота с использованием заранее запрограммированного контроллера.	опрос
18.	ноябрь	учебно-практическое	3	Практическое занятие по освоению пройденного материала.	наблюдение
19.	ноябрь	учебно-практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
Третий раздел: «Составные типы данных языка С++. Введение в робототехнику»					
20.	ноябрь	практическое	3	Введение в робототехнику. Обработка сигналов.	опрос

21.	ноябрь	учебно-практическое	3	Составные типы данных. Структуры. Перечисления. Объединения.	практическое задание
22.	ноябрь	учебно-практическое	3	Указатели и ссылки.	практическое задание
23.	ноябрь	учебно-практическое	3	Практика по робототехнике. Работа с конструкторами «Робоняша» и «Йодо».	практическое задание
24.	ноябрь	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
25.	ноябрь	контрольное	3	Контрольное задание по освоению пройденного материала.	Контрольный тест/ задание
Четвертый раздел: «Функции в языке С++. Работа с контроллерами»					
26.	ноябрь	практическое	3	Функции. Локальные и глобальные переменные. Область видимости переменных. Объявление и использование функции. Входные и выходные параметры функций.	опрос
27.	декабрь	практическое	3	Функции. Способы передачи переменных в функцию. Указатели как входные и выходные параметры функций. Ссылки как входные и выходные параметры функций.	самостоятельная работа
28.	декабрь	учебно-практическое	3	Пространство имен и его использование.	практическое задание
29.	декабрь	учебно-практическое	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	практическое задание
30.	декабрь	учебно-практическое	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	практическое задание
31.	декабрь	учебно-практическое	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	практическое задание
32.	декабрь	учебно-практическое	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	практическое задание
33.	декабрь	практическое	3	Знакомство с программируемыми контроллерами Trackduino.	практическое

				Работа по методической разработке «Умный город».	задание
34.	декабрь	контрольное	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	самостоятельная работа
35.	январь	практическое	3	Изучение возможностей набора «Стажер». Работа по методической разработке «Умный город».	самостоятельная работа
36.	январь	практическое	3	Презентация интерактивной инсталляции «Умный город»	Защита проектов
Углубленный модуль					
Первый раздел: «Работа с динамической памятью»					
37.	январь	учебно-практическое	3	Дополнительные сведения о типах и операторах. Спецификаторы классов памяти. Статические переменные. Ключевые слова typedef и sizeof.	наблюдение
38.	январь	учебно-практическое	3	Работа с динамической памятью. Выделение, распределение, освобождение. Переменные в динамической памяти.	практическое задание
39.	январь	учебно-практическое	3	Динамические массивы.	практическое задание
40.	январь	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
41.	февраль	контрольное	3	Контрольное задание по освоению пройденного материала.	Контрольный тест/ задание
42.	февраль	практическое	3	Практика по робототехнике. Работа с программируемыми контроллерами.	практическое задание
43.	февраль	практическое	3	Обсуждение вопросов по реализации индивидуального проекта каждого из обучающихся.	практическое задание
Второй раздел: «Основы объектно-ориентированного программирования»					
44.	февраль	учебно-практическое	3	Введение в ООП. Принципы ООП. Классы и объекты.	опрос
45.	февраль	учебно-	3	Введение в классы. Конструкторы и деструктор. Методы класса.	практическое

		практическое		Ключевое слово this.	задание
46.	февраль	учебно-практическое	3	Конструктор копирования. Перегрузка конструктора и методов класса.	практическое задание
47.	февраль	учебно-практическое	3	Принципы создания классов и объектов. Классы и динамическое выделение памяти.	практическое задание
48.	февраль	учебно-практическое	3	Указатели и ссылки на объекты.	практическое задание
49.	март	практическое	3	Практика по робототехнике. Работа с программируемыми контроллерами.	практическое задание
50.	март	учебно-практическое	3	Объекты как входные и выходные параметры функций.	наблюдение
51.	март	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
52.	март	контрольное	3	Контрольное задание по освоению пройденного материала.	Контрольный тест/ задание
53.	март	практическое	3	Практика по робототехнике. Работа с программируемыми контроллерами.	практическое задание
54.	март	практическое	3	Помощь в реализации индивидуального проекта каждого из обучающихся.	практическое задание
55.	март	учебно-практическое	3	Наследование. Базовый класс и класс наследник. Множественное наследование.	практическое задание
56.	март	учебно-практическое	3	Инкапсуляция. Спецификаторы доступа.	опрос
57.	апрель	учебно-практическое	3	Дружественные классы и функции.	практическое задание
58.	апрель	учебно-практическое	3	Полиморфизм. Виртуальные классы и функции.	практическое задание
59.	апрель	практическое	3	Практика по робототехнике. Работа с программируемыми контроллерами.	практическое задание
60.	апрель	учебно-практическое	3	Статические переменные и методы класса.	опрос

61.	апрель	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
62.	апрель	контрольное	3	Контрольное задание по освоению пройденного материала.	Контрольный тест/ задание
Третий раздел: «Стандартные библиотеки языка С++»					
63.	апрель	практическое	3	Практика по робототехнике. Работа с программируемыми контроллерами.	практическое задание
64.	апрель	практическое	3	Варианты защиты и презентации индивидуального итогового проекта.	практическое задание
65.	апрель	учебно-практическое	3	Шаблоны функций и шаблоны классов. Статическое и динамическое приведение типов.	практическое задание
66.	май	учебно-практическое	3	Обработка исключительных ситуаций.	практическое задание
67.	май	учебно-практическое	3	Умные указатели.	практическое задание
68.	май	учебно-практическое	3	Работа с файлами. Чтение, запись.	практическое задание
69.	май	практическое	3	Практическое занятие по освоению пройденного материала.	практическое задание
70.	май	контрольное	3	Контрольное задание по освоению пройденного материала.	Контрольный тест/ задание
71.	май	практическое	3	Подготовка к сдаче индивидуальных итоговых проектов.	консультация
72.	май	итоговое	3	Сдача индивидуальных итоговых проектов.	Сдача проектов

Приложение 4
к дополнительной общеобразовательной
общеразвивающей программе

Адаптированный учебный план для детей с ОВЗ

№	Раздел	Общее кол-во учебных часов	Теория	Практика	Формы контроля
1	Знакомство с С++	27	9	18	Входное тестирование Контрольные задания и тесты
2	Простейшие конструкции языка С++	30	10	20	Контрольные задания и тесты
3	Составные типы данных языка С++. Введение в робототехнику	18	6	12	Контрольные задания и тесты
4	Функции в языке С++. Работа с программируемыми контроллерами	33	11	22	Контрольные задания и тесты
5	Работа с динамической памятью	21	7	14	Контрольные задания и тесты
6	Основы объектно-ориентированного программирования	57	19	38	Контрольные задания и тесты
7	Стандартные библиотеки языка С++	30	10	20	Контрольные задания и тесты
	Всего часов	**216**	**77**	**139**	

Содержание адаптированного учебного плана

Раздел №1. Знакомство с С++.

Теория: знакомство с планом работы, целями и задачами программы. Правила техники безопасности и противопожарной защиты. Основы синтаксиса языка программирования С++. Основные операторы, типы данных.

Практика: входное тестирование, написание и запуск первых программ.

Раздел №2. Простейшие конструкции языка С++.

Теория: операторы отношений и логические операторы, управляющие конструкции, циклы, массивы.

Практика: создание простых программ, включающих в себя взаимодействие с пользователем, использование различных типов данных языка, циклические структуры и условные структуры, приемы работы с массивами.

Робототехника: знакомство с конструктором, основными деталями и принципами крепления. Стандартные конструкции роботов.

Обсуждение темы индивидуального итогового проекта.

Раздел №3. Составные типы данных языка С++. Введение в робототехнику

Теория: составные типы данных, структуры, перечисления, а также базовые алгоритмы работы с этими структурами.

Практика: создание программ с использованием изученных структур языка. Реализация изученных базовых алгоритмов на практике.

Робототехника: Изучение датчиков, сенсоров. Создание простейших механизмов.

Разработка интерактивной инсталляции «Умный город»

Обсуждение и выбор способа реализации индивидуального итогового проекта.

Раздел №4. Функции в языке С++. Работа с программируемыми контроллерами.

Теория: принципы и правила использования функций в языке С++.

Практика: написание собственных функций и использование их в коде программы.

Робототехника: работа с конструкторами «Робоняша» и «Йодо» (разработчик «Амперка»). Сборка модели по технологическим картам.

Разработка алгоритма индивидуального итогового проекта.

Раздел №5. Работа с динамической памятью.

Теория: правила работы с динамической памятью в языке С++.

Практика: закрепление изученного материала на практике - использование динамической памяти в программах.

Робототехника: основы работы с программируемыми микроконтроллерами. Использование встроенных возможностей микроконтроллера.

Начало разработки индивидуального итогового проекта.

Раздел №6. Основы объектно-ориентированного программирования.

Теория: основы объектно-ориентированного программирования.

Практика: создание классов в программах. Реализация различных видов конструкторов, деструктора и методов класса, принципов наследования, инкапсуляции и полиморфизма.

Робототехника: разработка алгоритмов поведения роботизированной системы. Создание программ на основе этих алгоритмов. Программирование микроконтроллера.

Обсуждение вопросов по реализации индивидуального итогового проекта.

Раздел №7. Стандартные библиотеки языка С++.

Теория: изучение возможностей стандартных библиотек языка С++.

Практика: использование на практике методов стандартных библиотек языка С++.

Робототехника: создание роботизированных механизмов на базе микропроцессора TRIK, Arduino.

Защита итоговых индивидуальных проектов.

Приложение 5
к дополнительной общеобразовательной
общеразвивающей программе

Дистанционный модуль

Переход на дистанционный режим обучения возможен при наличии у обучающихся:

- компьютера с установленной операционной системой Windows версии не ниже 7 для каждого обучающегося и для педагога;
- ПО для работы с онлайн-платформой Zoom;
- доступа в Интернет.

Цель: обучить основным принципам и этапам программирования и разработки программного обеспечения на основе языка программирования С++ для последующего решения поставленных технических задач, а также реализации творческих проектов.

Задачи:

Образовательные:

- сформировать умение осуществлять алгоритмизацию задачи через освоение различных методов решения задач на языке программирования С++;
- обучить практическому применению основных конструкций языка программирования С++ путем разработки и написания программ различной направленности и сложности;
- обучить практическому применению полученных знаний для программирования конструкторов на базе Arduino и ТРИК;
- способствовать овладению базовых понятий и терминов английского языка технической направленности, необходимых в программировании.

Развивающие:

- развить память, внимание, аналитическое и логическое мышление;
- сформировать навыки проектной и презентационной деятельности;
- развить интеллектуальные, технические и творческие способности обучающихся;
- сформировать навыки самообучения.

Воспитательные:

- сформировать устойчивый интерес к занятиям технического и математического направления;
- сформировать способность к самоорганизации и самоконтролю;
- воспитать терпение, способность преодолевать трудности;
- повысить коммуникативную культуру обучающихся, культуру общения, взаимопонимания, взаимопомощи.

Режим занятия:

Занятия проводятся 2 раза в неделю.
Продолжительность занятия 45 минут с перерывом 10 минут;
45 минут - работа в онлайн режиме;
90 минут – в офлайн режиме (индивидуальная работа и консультирование).

Форма организации занятий:

1. Онлайн-платформа Zoom, Youtube, - проведение теоретической части занятий, консультации (индивидуальные и групповые), организация работы в группах и т.д.
2. Google - формы – контроль и диагностика освоения образовательной программы (онлайн-тест).
3. Google -диск организация групповой / проектной работы с помощью облачных папок.
4. В офлайн режиме - посредством социальных сетей, мессенджеров обучающимся передается видео, видеоинструкция, презентационный материал с инструкцией выполнения заданий

Формы отслеживания и фиксации образовательных результатов

1. фотоматериалы;
2. отзывы детей и родителей, материалы диагностики (планы опросов);
3. размещение ссылок на архивы и видео работ в чатах Viber, WatsUp, ВКонтакте / в альбоме группы социальной сети ВКонтакте.

Формы контроля

1. Прохождение онлайн-тестов на платформе «OnlineTestPad»
2. Выполнение практических заданий
3. Создание индивидуальных приложений
4. Google-формы - контроль и диагностика освоения образовательной программы (онлайн-тест).

Интернет-ресурсы

1. Инструкция по использованию платформы Zoom для детей и родителей https://studyhome.online/storage/files/instruction_Zoom.pdf
2. Инструкция по регистрации на сервисах Google https://support.google.com/accounts/answer/27441?hl=ru
3. Бесплатный многофункциональный онлайн сервис для проведения тестирования и обучения https://onlinetestpad.com/ru

Приложение 6
к дополнительной общеобразовательной
общеразвивающей программе

Летний модуль

Цель: познакомить с основными принципам и этапами программирования и разработки программного обеспечения на основе языка программирования С++ и с программируемыми контроллерами.

Учебный план

№	Раздел	Общее кол-во учебных часов	Теория	Практика	Формы контроля
1	Знакомство с С++	6	2	4	Входное тестирование Контрольные задания и тесты
2	Простейшие конструкции языка С++	6	2	4	Контрольные задания и тесты
3	Составные типы данных языка С++. Введение в робототехнику	10	4	6	Контрольные задания и тесты
4	Функции в языке С++. Работа с программируемыми контроллерами	12	6	6	Контрольные задания и тесты
5	Итоговое занятие	2	1	1	Защита проектов
	Всего часов	**36**	**15**	**21**	

Содержание учебного плана

Раздел №1. Знакомство с С++.

Теория: знакомство с планом работы, целями и задачами программы. Правила техники безопасности и противопожарной защиты. Основы синтаксиса языка программирования С++. Основные операторы, типы данных.

Практика: входное тестирование, написание и запуск первых программ.

Раздел №2. Простейшие конструкции языка С++.

Теория: операторы отношений и логические операторы, управляющие конструкции, циклы, массивы.

Практика: создание простых программ, включающих в себя взаимодействие с пользователем, использование различных типов данных языка, циклические структуры и условные структуры, приемы работы с массивами.

Робототехника: знакомство с конструктором, основными деталями и принципами крепления. Стандартные конструкции роботов.

Обсуждение темы индивидуального итогового проекта.

Раздел №3. Составные типы данных языка С++. Введение в робототехнику

Теория: составные типы данных, структуры, перечисления, а также базовые алгоритмы работы с этими структурами.

Практика: создание программ с использованием изученных структур языка. Реализация изученных базовых алгоритмов на практике.

Робототехника: Изучение датчиков, сенсоров. Создание простейших механизмов.

Обсуждение и выбор способа реализации индивидуального итогового проекта.

Раздел №4. Функции в языке С++. Работа с программируемыми контроллерами.

Теория: принципы и правила использования функций в языке С++.

Практика: написание собственных функций и использование их в коде программы.

Робототехника: работа с конструкторами «Робоняша» и «Йодо» (разработчик «Амперка»). Сборка модели по технологическим картам.

Разработка алгоритма индивидуального итогового проекта.

Планируемые результаты

В результате освоения модуля обучающиеся будут

Знать:

- основные типы данных языка C++;
- условные и циклические конструкции, их виды и возможности;
- основные структуры языка, правила их использования в программе;
- правила написания функций и их использования в программе;
- основные виды микроконтроллеров и способы их программирования;
- основные понятия и термины английского языка технической направленности, необходимые в программировании.

Уметь:

- разрабатывать алгоритм решения поставленной задачи и реализовывать его с помощью языка C++;
- собирать роботизированную систему из конструкторов на базе Arduino и ТРИК;
- ставить перед собой задачи и решать их, используя имеющиеся знания и овладевая новыми;
- самостоятельно организовывать и контролировать свою деятельность;
- самостоятельно работать с технической и научной документацией (мануалы и описания используемых компонент и библиотек);
- продуктивно взаимодействовать в паре, в группе, в коллективе.

Приложение 7
к дополнительной общеобразовательной
общеразвивающей программе

Методические материалы по программе

Кейс для раздела №1 «Знакомство с С++» базового модуля

Цель: определить качество усвоения материала обучающимися в игровой форме, в частности:
- знание различных типов данных;
- владение понятием переменная и использование различных операций с переменной;
- использование разных стандартных функций библиотеки ввода-вывода;
- творческий подход в решении поставленной задачи.

Тема: создание программы, которая взаимодействует с пользователем.

Длительность: два занятия по 45 минут с перерывом в десять минут.

Организация: обучающиеся получают задания в распечатанном виде и выполняют их индивидуально. Следует учитывать, что каждый ребенок может творчески подходить к выполнению задания и выполнять его с помощью различных алгоритмов и операторов.

Оценка результативности: в задании представлено всего 4 задания, каждое задание оценивается в один балл. Таким образом максимальный балл за выполнение итогового задания — 4 балла. Все оценки заносятся в форму из Приложения 3. Дополнительным баллом можно оценить творческий подход ребенка к поставленной задаче.

Задача преподавателя: преподаватель должен не только проверить итог работы кода, но и посмотреть сам код, насколько он написан правильно.

Необходимое оборудование и материалы: среда программирования Eclipse.

Предполагаемый результат:
- закрепление полученных знаний с помощью самостоятельной деятельности;
- умение самостоятельно построить алгоритм решения поставленной задачи;

Артефакт: готовая программа на языке С++.

Текст тестового задания

Доброе утро! Сегодня твой первый день в лаборатории по разработке искусственного интеллекта. И твое первое задание будет создание программы, которая сможет поговорить с человеком.

Итак, когда человек садится за компьютер, то сначала нужно познакомится. Поэтому первое, что должен сделать твой искусственный интеллект - спросить имя человека, а затем поздороваться с человеком по его имени и представится самому.

Второе задание — узнать сколько человеку лет и вывести на экран год его рождения.

Третье задание — попросить пользователя ввести с клавиатуры любое десятичное число и умножить возраст человека на это число. Вывести на экран результат.

Четвертое задание — вывести на экране с помощью стандартных функций ввода и вывода и различных символов какую-то простую картинку на прощание.

Для преподавателя, пример кода:

```
char name[100];
int age;
double num;
cout << "Привет! Как тебя зовут?" << endl;
cin >> name;

cout << name << ", очень приятно! А меня зовут Компьютер 1.0." << endl;
cout << name << ", а сколько тебе лет?" << endl;
cin >> age;
cout << name << ", ты родился " << 2019 - age << " или в " << 2019 - 1 - age << " году"  << endl;
cout << name << ", введи с клавиатуры любое десятичное число" << endl;
cin >> num;
cout << name << ", если твой возраст умножить на это число, то получится " << num * age << endl;
cout << "    /\\" << endl;
cout << "   / \\" << endl;
cout << "  /   \\" << endl;
```

```
cout << "  /      \\" << endl;
cout << " /_________\\" << endl;
cout << " |   _     |" << endl;
cout << " |  |_|    |" << endl;
cout << " |         | " << endl;
cout << " |_________| " << endl;
```

Кейс для раздела «Простейшие конструкции языка С++» базового модуля

Цель: определить качество усвоения материала обучающимися в игровой форме, в частности:
- знание логических операторов языка С++;
- знание циклических операторов языка С++;
- знание и умение работать с массивами в языке С++;
- умение применить имеющиеся знания к решению абстрактной задачи;
- творческий подход в решении поставленной задачи.

Тема: прохождение квеста «Ты — капитан космического корабля».

Длительность: два занятия по 45 минут с перерывом в десять минут.

Организация: обучающиеся получают задания в распечатанном виде и выполняют их индивидуально. Следует учитывать, что каждый ребенок может творчески подходить к выполнению задания и выполнять его с помощью различных алгоритмов и операторов.

Оценка результативности: в задании представлено всего 6 заданий, каждое задание оценивается в один балл. Таким образом максимальный балл за выполнение итогового задания — 6 баллов. Все оценки заносятся в форму из Приложения 3. Дополнительным баллом можно оценить творческий подход ребенка к поставленной задаче.

Задача преподавателя: преподаватель должен не только проверить итог работы кода, но и посмотреть сам код, насколько он написан правильно. Сомневающимся ребятам можно предоставить подсказку — какой оператор или логическую конструкцию нужно использовать.

Необходимое оборудование и материалы: среда программирования Eclipse.

Предполагаемый результат:
- закрепление полученных знаний с помощью самостоятельной деятельности;
- умение самостоятельно построить алгоритм решения поставленной задачи;
- развитие творческого подхода к решению поставленной задачи.

Артефакт: готовая программа на языке С++.

Текст квеста «Ты — капитан космического корабля»

Приветствуем тебя, капитан корабля «Восход Альфа»! Сегодня ты должен отправить первый межгалактический корабль «Восход Альфа» в первый полет к далекой неизведанной планете под названием Альфа Сибериуса. Когда-то давно ученые получили с этой планеты зашифрованный сигнал, но так и не смогли понять, что же он означает.

Было принято решение всеми силами как можно скорее создать корабль, который сможет осуществить перелет до этой планеты. Это долгий и сложный полет, к которому люди готовились несколько десятилетий, но честь быть капитаном этого корабля выпала именно тебе!!!

Но помимо того, что это большая честь, это еще и большая ответственность, поэтому тебе предстоит сложная и серьезная работа, так что давай скорее начнем… Занимай ГЛАВНОЕ место за командным пультом управления и начнем проверку всех систем, чтобы быть уверенным, что наш корабль не подведет нас во время полета.

На старт!!! Первое, что ты должен сделать, это проверить, как функционирует система оповещения. Для этого ты должен проверить ввод и вывод информации на экран главного монитора. Нужно ввести число с консоли и вывести его обратно на консоль.

Для преподавателя, пример кода программы:

```
int x;
cout << "Input your number" << endl;
cin >> x;
cout << x;
```

Система оповещения работает? Отлично!!!

Теперь нужно разослать всем членам экипажа сообщения. Наш экипаж делится на две команды – команда капитана и команда обслуживающего персонала. Команда капитана помогает тебе вести корабль по бескрайним просторам космоса и всегда должна находится на капитанском мостике, а остальные работают на корабле и следят за тем, что на корабле все в порядке и не было никаких проблем. У команды капитана номер позывного 1, а у остальных членов экипажа номер позывного – 2. Тебе нужно вызвать свою команду на мостик. Для этого ты должен проверить номер позывного, который человек введет с консоли своего коммуникатора, и вывести срочный приказ прибыть на капитанский мостик ТОЛЬКО тем, кто входит в состав команды капитана. Остальным членам экипажа пожелай доброго утра и сообщи, что скорой готовности отправится в полет.

Для преподавателя, пример кода программы:

```
cout << "Input your number" << endl;
cin >> x;
if(x == 1)
    cout << "Go to the captain's bridge";
else if(x == 2)
    cout << "Good morning";
else
    cout << "Invalid number";
```

Ну вот твоя команда наконец-то собралась на мостике. Начинаем обратный отсчет. Чтобы все члены экипажа увидели его и заняли свои места, нужно вывести на консоль коммуникатора цифры обратного отсчета от 10 до 1 в текстовом виде. А после строки «ОДИН» вывести такое долгожданное слово «ПОЕХАЛИ!!!».

Для преподавателя, пример кода программы:

```
setlocale(0,"");
int x = 10;
while(x >= 0){
    switch (x) {
    case 10: cout << "Десять" << endl; break;
    case 9: cout << "Девять" << endl; break;
    case 8: cout << "Восемь" << endl; break;
    case 7: cout << "Семь" << endl; break;
    case 6: cout << "Шесть" << endl; break;
    case 5: cout << "Пять" << endl; break;
    case 4: cout << "Четыре" << endl; break;
    case 3: cout << "Три" << endl; break;
    case 2: cout << "Два" << endl; break;
    case 1: cout << "Один" << endl; break;
    case 0: cout << "Поехали!" << endl; break;
    }
    x--;
}
```

Ура!!! Благодаря твоим стараниям и помощи всего экипажа корабль стартовал успешно! Вы начали свой длинный полет, который наполнен не только восхищением и радостью, но и сложной работой.

Теперь нужно проверить 10 систем контроля состояния корабля. Требуется вывести на консоль сообщения, что:

«Система контроля корабля № N готова к работе».

Для преподавателя, пример кода программы:

```
for(int i = 1; i < 11; i++){
cout << "Система контроля корабля № " << i << " N готова к работе" << endl;
}
```

Отлично, все работает как часы! Летим дальше…

Внезапно вы попадаете в метеоритный дождь, который частично повредил ваши термозащитные панели, и они требуют срочной обработки сульфидом тория.

Известно, что на обработку одного квадратного метра панели требуется 1 нанограмм сульфида. Всего необходимо обработать N прямоугольных панелей размером 7 на 9 метров.

Вы должны написать программу, на вход которой подается количество панелей, которые требуют обработки и количество сульфида тория, находящегося на вашем корабле. В программе производится

подсчет нужного количество сульфида тория, необходимого для обработки введенного количество панелей и это количество сравнивается с тем количеством, которое есть на корабле. Если сульфида тория хватит, то необходимо как можно скорее вывести сообщение о том, что все в порядке и панели будут обработаны в скором времени. Если же количества сульфида тория не хватит, то нужно вывести предупреждение об этом.

Все в порядке! Запасов сульфида тория хватило на обработку всех панелей. Полет продолжается…

Для преподавателя, пример кода программы:

```
int n;
int count;
cout << "Input count of panels" << endl;
cin >> n;
cout << "Input count of sulfid" << endl;
cin >> count;
int all = 7 * 9 * n;
if(all > count)
   cout << "sulfid is not enough";
else
   cout << "sulfid is enough!";
```

Пока Вы и ваша команда была занята проверкой и починкой корабля, команда ученых корабля получила новый сигнал с планеты Альфа Сибериуса. Сигнал состоял из 10 одинаковых по типу импульсов, но каждый импульс имеет свою код. Команда ученых определила код каждого импульса и предоставила Вам с командой:

104, 101, 108, 108, 111, 32, 102, 114, 105, 101, 110, 100. Теперь Вам нужно написать программу, которая бы смогла расшифровать этот сигнал, используя массивы.

Для преподавателя, пример кода программы:

```
char arr[12];
arr[0] = 104;
arr[1] = 101;
arr[2] = 108;
arr[3] = 108;
arr[4] = 111;
arr[5] = 32;
arr[6] = 102;
arr[7] = 114;
arr[8] = 105;
arr[9] = 101;
arr[10] = 110;
arr[11] = 100;
for(int i = 0; i < 12; i++)
cout << arr[i];
```

Кейс для раздела «Составные типы данных языка С++. Введение в робототехнику» базового модуля

Цель: определить качество усвоения материала обучающимися в игровой форме, в частности:

- умение собирать робота из базовых деталей конструктора «Стажер А»;
- знание основных принципов использования базовых устройств ввода-вывода;
- умение работать с непрограммируемой платой конструктора «Стажер А»;
- умение работать в команде;
- творческий подход в решении поставленной задачи.

Тема: создание машины, управляющейся с пульта управления.

Требования:

- команда должна собрать двигающегося на колесах робота;
- робот не должен падать при остановке или движении;
- робот должен управляться с помощью пульта управления;
- робот должен уметь двигаться в двух направлениях вперед, назад, а также уметь совершать разворот вправо и влево;
- робот должен использовать хотя бы одно устройство вывода — светодиод.

Длительность: два занятия по 45 минут с перерывом в десять минут.

Организация: обучающиеся работают в команде по два человека.

Оценка результативности: команда из двух человек должна представить робота-автомобиль, который управляется с пульта управления. Оценивается завершенность робота, количество использованных деталей, творческий подход к решению задачи, взаимодействие ребят в команде. Максимально — 4 балла. Все оценки заносятся в форму из Приложения 3.

Необходимое оборудование и материалы: конструктор «Стажер А», в частности обязательно использование:

- электродвигатели;
- светодиоды;
- пульт управления;
- непрограммируемая плата конструктора «Стажер А».

Предполагаемый результат:

- закрепление полученных знаний с помощью самостоятельной деятельности;
- умение в команде прийти к единому результату;
- развитие творческого подхода к поставленной технической задачи.

Артефакт: готовый робот.

Кейс для раздела «Функции в языке С++. Работа с программируемыми контроллерами» базового модуля

Цель: определить качество усвоения материала обучающимися в игровой форме, в частности:

- умение работать в программной среде MRT;
- знание синтаксиса команд основных устройств ввода-вывода конструктора «Стажер А»;
- умение построить алгоритм программы для робота;
- умение работать с программируемой платой конструктора «Стажер А»;
- умение работать в команде;
- творческий подход в решении поставленной задачи.

Тема: написание программы для роботизированной системы.

Требования:

- команда должна собрать роботизированную систему, которая будет решать поставленную перед ней конкретную задачу;
- роботизированная система должна использовать хотя бы одно устройство ввода — пульт управления или любой датчик;
- роботизированная система должна использовать хотя бы одно устройство вывода — светодиод, сервопривод и т.д.;
- для роботизированной системы необходимо использовать программируемую плату конструктора «Стажер А», то есть программа для роботизированной системы должна быть написана командой;
- программа должна быть написана в среде программирования MRT вручную, с использованием функций и процедур и без использования визуальной среды программирования.

Длительность: два занятия по 45 минут с перерывом в десять минут.

Организация: обучающиеся работают в команде по два человека.

Оценка результативности: команда из двух человек должна представить роботизированную систему и продемонстрировать ее работу. Оценивается завершенность робота, количество использованных деталей, правильность написания кода программы, творческий подход к решению задачи, взаимодействие ребят в команде. Максимально — 5 баллов. Все оценки заносятся в форму из Приложения 3.

Необходимое оборудование и материалы: конструктор «Стажер А», в частности обязательно использование:

- минимально одно устройство ввода;
- минимально одно устройство вывода;
- программируемая плата конструктора «Стажер А».

Предполагаемый результат:

- закрепление полученных знаний с помощью самостоятельной деятельности;
- умение в команде прийти к единому результату;
- развитие творческого подхода к поставленной технической задачи.

Артефакт: готовая роботизированная система.

Варианты возможных роботизированных систем:

1 Создание машины на пульте управления с возможностью движения «Вперед» и «Назад». При

движении назад на машине сзади должен загораться красный светодиод. При движении вперед загораются два светодиода впереди.

2 Создание светофора, у которого красный и зеленый свет горят по 10 секунд, а желтый свет 5 секунд. К светофору должна быть подключена кнопка пешехода. Если нажимается кнопка пешехода, то загорается зеленый свет на 10 секунд. После чего светофор переходит в работу по штатному режиму.

3 Создание турникета. Если к турникету подходит человек, то турникет открывается. Как только человек уходит, турникет закрывается. При открытом турникете горит зеленый светодиод. При закрытом турникете горит красный светодиод. Эмуляция человека производится путем поднесения руки к инфракрасному датчику.

Конспект преподавателя для раздела «Знакомство с C++» базового модуля
Занятие №1.1

Цель:
- сформировать у обучающихся понятия «программист» и «программа»;
- познакомить обучающихся с понятием «переменная»;
- продемонстрировать принципы работы с переменными;
- заинтересовать обучающихся с помощью написания своей первой программы.

Тема: первая программа на языке C++.
Длительность: два занятия по 45 минут с перерывом в десять минут.
Необходимое оборудование и материалы: среда программирования Eclipse, .
Предполагаемый результат:
- формирование заинтересованности;
- закрепление полученных знаний с помощью самостоятельной деятельности;

Артефакт: готовая программа на языке C++.

План занятия

1 Обсуждение и дискуссия на тему: кто такой программист и что он делает.

Организация: обучающимся предлагается для обсуждения вопрос, на который они должны дать свой ответ. С помощью наводящих вопросов и уточнений наставник должен подвести детей к конкретному итогу, чтобы дети «сами» пришли к правильной формулировке ответа.

Цель дискуссии: сформировать у детей правильное представление о работе программиста.

Вопрос: кто такой программист и что он делает.

Итог обсуждения вопроса: программист пишет код программы, составленный на каком-то конкретном языке программирования. То есть он получает задание, описание проблемы, которую должен решить. Затем он продумывает внешний вид программы, структуры данных, потом составляет алгоритмы для работы с этими структурами и в итоге представляет все это в виде кода.

Вопрос: что такое компьютер с точки зрения программиста.

Итог обсуждения вопроса: это машина, которая может выполнить только те инструкции, которые ей задал в программе программист. Компьютер не может додумать что-то сам или выполнить какое-то действие без инструкции.

Для программиста компьютер — это помощник для отладки кода. Сам код можно написать хоть на листке бумаги и отдать этот код на исполнение, например, роботу, не используя компьютер вообще.

Вопрос: что такое программа.

Итог обсуждения вопроса: программист пишет программу на своем языке программирования, в которой он в понятном для человека виде объясняет компьютеру что нужно делать. Специальная программа «компилятор» (переводчик) преобразовывает написанный код в код из 0 и 1 (двоичный код), который компьютер понимает и может исполнить.

2 Работа с презентацией на тему занятия «Что такое переменная?»

Любая компьютерная программа производит какие-то математические или логические операции и отображает результат своей работы на экране. Многие программы используют графические и звуковые эффекты. Современные игры почти все работают с объемными 3D-эффектами. Все это, вычисления,

эффекты, графика - результат вычислений, который появляется благодаря размещению большого объема чисел в ячейках памяти компьютера и работе с ними. Задача программиста — написать программу, а это значит заполнить память компьютера нужными данными, произвести с ними необходимые действия, получить результат и вывести его пользователю в нужном виде. Соответственно, для того чтобы написать полезную программу, нужно знать, как организовать ячейки памяти для хранения данных, как работать с этими данными и, в конечном итоге, вывести эти данные на экран.

Для того, чтобы записать что-то в память компьютера нужно создать так называемую переменную. Переменная — это «ячейка» оперативной памяти компьютера, в которой может храниться какая-либо информация. И, конечно, логично, что если мы хотим поработать с какой-то ячейкой памяти, то нужно дать ей имя.

Самой простой и независимой частью программы на C++ является инструкция или стейтмент. Это такой аналог предложения в русском языке. Фраза, которая несет в себе смысловую нагрузку. И как в русском языке предложение завершается точкой, в программировании стейтмент обязательно завершается точкой с запятой.

```
//так оформляется комментарий
int a;  // объявление переменной
a = 4;  // инициализация переменной
int b = 5;  // объявление и инициализация вместе
```

Простейшая необходимая инструкция в программе — это объявление и инициализация переменной.

Помимо стейтментов в языке программирования существуют комментарии, которые не несут в себе никакой смысловой нагрузки для компьютера, но очень помогают программисту ориентироваться в коде программы. Комментарии пишутся после двух знаков слеша «/».

Первая инструкция — это объявление переменной. Объявить переменную – это значит сказать компьютеру, чтобы он выделил ячейку памяти и дал этой ячейке тип и имя, которое мы указали, в данном случае наша переменная имеет тип int, а имя - «a»

Что такое тип переменной. Мы привыкли, что в математике есть целые числа и дробные. В русском языке мы пользуемся буквами и составляем слова из них. Мы можем рассуждать логическими понятиями – «правда» или «ложь», «да» или «нет». Поэтому компьютеру мы тоже хотим дать такую возможность. Для этого и созданы различные типы, про которые мы подробно поговорим на следующих занятиях, а пока попробуем воспользоваться одним из таких типов данных – int, который означает целые числа.

Так вот с помощью строки объявления переменной мы сказали компьютеру, что хотим завести в его памяти ячейку, у которой будет имя «a». Какую конкретно ячейку выделит для нас компьютер и где она будет расположена в его памяти, нам не важно. Важно, что, компьютер будет знать, что в его памяти есть ячейка с именем «a» и когда мы захотим с ней поработать, он будет работать именно с этой конкретной ячейкой.

Второй стейтмент на слайде — это инициализация переменной. Инициализировать переменную – это значит положить в ячейку с ее именем какое-то значение.

Таким образом мы уже используем нашу ячейку, которую завели в предыдущем стейтменте. Компьютер знает, что у него в памяти есть ячейка с именем «a», поэтому он обращается именно к ней и помещает в нее значение «4».

Нужно запомнить, что переменная связывает между собой три понятия – это тип, имя и значение. Без связки этих трех свойств невозможно производить какие-то действия с переменными и памятью компьютера.

То есть первое, что вы должны сделать – это сказать компьютеру, что вам нужна ячейка, в которой вы будете хранить данные определенного типа, в нашем случае целое число, и дадите этой ячейке имя. А затем эту ячейку нужно заполнить какими-то данными.

Задание обучающимся – объяснить третий стейтмент на слайде.

Теперь попробуем составить более сложные инструкции, чтобы связать между собой несколько переменных.

```
int a = 4;
int b = 2;
int c = a * b + 10;
a = a + c;
```

С помощью арифметических операций можно изменять значения переменных в ячейках. Как только вы присваиваете переменной новое значение, старое значение затирается и в ячейку помещается это новое значение.

Таким образом, изначально в ячейке с именем «a» находится значение 4, а в ячейке с именем «b» находится значение 2. Чтобы заполнить значением ячейку с именем «c», компьютер сначала выполняет арифметические операции a * b + 10, и полученное значение 18 помещает в ячейку с именем «c».

Задание обучающимся – объяснить четвертый стейтмент на слайде.

В языке программирования C++ есть правила, которые определяют какие имена переменным можно использовать, а какие нельзя.

1 Идентификатор не может быть ключевым словом. В языке C++ есть список слов, которые зарезервированы, то есть мы не можем использовать их для своих целей, поскольку каждое из этих ключевых слов уже определяет какую-то инструкцию.

Список ключевых слов представлен на следующем слайде:

alignas (C++11)	decltype (C++11)	namespace	struct
alignof (C++11)	default	new	switch
and	delete	noexcept (C++11)	template
and_eq	do	not	this
asm	double	not_eq	thread_local (C++11)
auto	dynamic_cast	nullptr (C++11)	throw
bitand	else	operator	true
bitor	enum	or	try
bool	explicit	or_eq	typedef
break	export	private	typeid
case	extern	protected	typename
catch	false	public	union
char	float	register	unsigned
char16_t (C++11)	for	reinterpret_cast	using
char32_t (C++11)	friend	return	virtual
class	goto	short	void
compl	if	signed	volatile
const	inline	sizeof	wchar_t
constexpr (C++11)	int	static	while
const_cast	long	static_assert (C++11)	xor
continue	mutable	static_cast	xor_eq

2 Идентификатор может состоять только из букв (нижнего или верхнего регистра), цифр или символов подчёркивания. Это означает, что все другие символы и пробелы — запрещены.

3 Идентификатор должен начинаться с буквы (нижнего или верхнего регистра). Он не может начинаться с цифры или с другого какого-то символа.

Задание обучающимся - работа со слайдом - определить, где правильные, а где неправильные имена переменных.

```
int value 2;
int value_2;
int value*;
int 2value;
int value2;
```

4 C++ различает нижний регистр от верхнего myvalue отличается от myValue и отличается от MYVALUE.

```
int value;
int Value;
int ValuE;
```

5 Правила хорошего кода. В C++ принято имена переменных писать с маленькой буквы.

Если имя переменной состоит из одного слова, то это слово должно быть записано в нижнем регистре. А если имя переменной или функции состоит из нескольких слов, то используют стиль CamelCase – это когда несколько слов пишутся слитно, без пробелов, и каждое новое слово пишется с заглавной буквы

6 И самое важное правило из всех - используйте в качестве идентификаторов только те имена, которые реально описывают то, чем является объект. Особенно это важно для больших программ. Но привыкать нужно сразу.

Если, например, вам нужно в программе посчитать количество яблок, то назовите переменную appleCount и вы всегда сможете по имени переменной понять, что хранится в ячейке с этим именем.

Любая программа должна как-то взаимодействовать с пользователем, ведь она работает для него — это либо введение в программу каких-то входных данных, либо вывод результата или процесса выполнения программы на экране. Работа с экраном или консолью — это неотъемлемая часть практически любой программы. Чтобы упростить жизнь программисту, разработчики языка программирования пишут так называемые библиотеки, в которых содержатся функции для выполнения каких-то стандартных действий. Например, самые часто используемые функции – это функции ввода данных с клавиатуры и функция вывода данных на экран. И вместо того, чтобы думать о том, как нам вывести на экран или считать с него информацию, мы просто можем использовать в нашей программе библиотеку и готовую функцию из него. Каждая функция имеет свое название. Так, например, функция вывода на экран называется cout, а функция считывания с экрана cin.

```
cout  - функция вывода на экран
cin - функция считывания с экрана.
```

Чтобы использовать библиотеку и описанные в ней функции, мы должны использовать так называемые заголовочные файлы. Это файлы, в которых описаны имена и правила использования функций библиотеки. Заголовочные файлы включаются в текст программы с помощью специальной команды #include. Такие команды называются директивы препроцессора.

```
#include <iostream>
```

Директивы препроцессора начинаются со знака "решетка" (#), который должен быть самым первым символом строки. Таким образом, всё содержимое iostream становится доступным для использования. Теперь нам нет необходимости в своей программе создавать функции работы с консолью, а достаточно просто написать директиву include и использовать уже написанные функции из библиотеки iostream.

По сути библиотека – это такая же программа, в которой описаны нужные и полезные функции, используя которые программист может облегчить свою работу.

Представим ситуацию, что у нас есть две библиотеки, которые содержат функции с одинаковыми названиями. Нам нужно подключить обе эти библиотеки в свою программу. Но как компьютер поймет, какую из функций нужно использовать в конкретный момент? Для того, чтобы не было конфликтов с одинаковыми именами, существует понятие пространства имен. Допустим у нас есть в классе два мальчика по имени Петя. Как мы их различим - по фамилии. Так и здесь, пространство имен гарантирует нам то, что мы сможем различить две функции, которые названы одинаково.

В библиотеке iostream используется пространство std, поэтому использовать функции мы будем таким образом

Теперь давайте попробуем объединить все полученные знания и написать программу, которая считывает с экрана число и выводи его обратно на экран.

Создаем пустой консольный проект.

Программа на языке С++ состоит из функций, описаний и директив препроцессора. Отправной точкой выполнения любой С++ программы является функция, которая называется main(). Без этой функции программа просто не запустится. Функция содержит четыре элемента:
- возвращаемый тип (в нашем случае int);
- имя функции (main);
- список входных параметров, заключенный в круглые скобки (в данном случае список пуст);
- заключенное в фигурные скобки, тело функции, представляющее собой блок инструкций.

Таким образом весь наш код программы будет помещаться в фигурных скобках.
Первое, что мы сделаем – подключим библиотеку.

```
#include <iostream >
```

И теперь напишем код программы в функции main().
Создадим переменную типа int с именем age:

```
int age;
```

Считаем с консоли количество лет пользователя:

```
cin >> age;
```

Выведем на экран считанное количество лет:

```
cout << "Мне " << age  << " лет" << endl;
```

Наша программа готова.

Конспект преподавателя для раздела «Составные типы данных языка С++. Введение в робототехнику» базового модуля
Занятие № 3.4

Цель:
- познакомить обучающихся с видами деталей и принципами их соединения;
- познакомить обучающихся с непрограммируемой платой HUNAROBO и принципами ее работы;
- создать робота на колесах, который управляется с пульта управления.

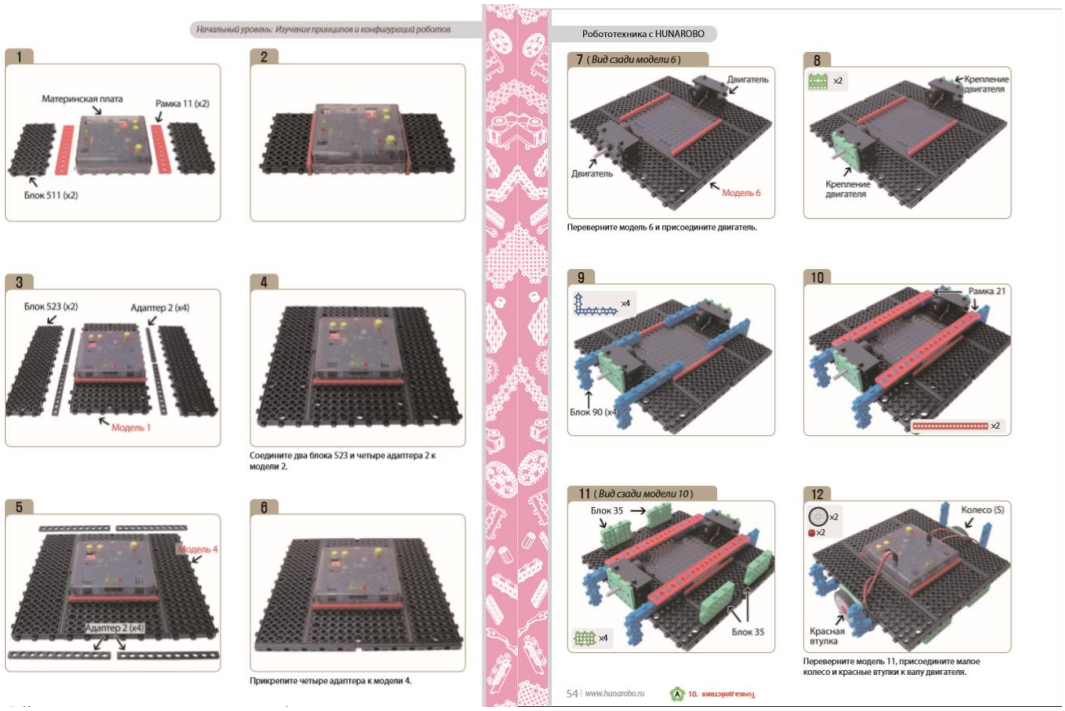
Тема: создание робота.
Длительность: два занятия по 45 минут с перерывом в десять минут.
Необходимое оборудование и материалы: конструктор «Стажер А», инструкция по сборке.
Предполагаемый результат:
- умение работать с деталями конструктора «Стажер А»;
- понимание принципов работа непрограммируемой платы HUNAROBO;
- формирование заинтересованности.

Артефакт: готовый робот.
Для презентации использованы иллюстрации брошюр «Робототехника с Хунаробо» от ООО «Брейн Девелопмент».

План занятия

1 Собираем каркас робота.

Первая страница инструкции демонстрирует общий вид робота, которого нужно собрать и список необходимых деталей для этого.



Следующие 4 страницы инструкции посвящены последовательной сборке. С помощью них, дети пошагово собирают каркас робота. Непрограммируемая плата является частью конструкции, поэтому используется на данном этапе. Провода элементов, которые требуют присоединения к плате по инструкции, обучающимся предлагается пока не присоединять, а оставить свободно лежащими. Провода будут подсоединяться на втором этапе занятия.

2 Соединяем каркас с непрограммируемой платой

Провода элементов, которые требуют присоединения к плате, подключаются согласно следующей страницы инструкции.

После подключения проводов, требуется включить робота и выбрать режим работы непрограммируемой платы. В данном случае нам требуется режим дистанционного управления роботом. Выбор этого режима делается согласно инструкции:

И последнее, что нужно сделать – это настроить пульт управления на нужный канал. Каждый пульт имеет 8 возможных каналов, то есть сигнал пульта может транслироваться с восемью разными частотами. Соответственно каждая плата имеет переключатели, для того чтобы настроить ее на прием восьми возможных частот. Для того, чтобы наш робот работал от пульта управления, нам нужно пульт настроить на номер определенного канала, а переключатели платы выставить на прием этого же канала. Если переключатели платы другого робота будут установлены на прием такого же канала, как и на первого робота, то оба робота будут управляться одним пультом. Поэтому, чтобы каждый робот управлялся с одного пульта управления, необходимо создать связку «канал ДПУ» - «канал платы».
Канал пульта управления и переключатели каналов платы настраиваются согласно следующей инструкции:

После настройки каналов можно проверять работоспособность робота. Можно поделить детей на команды и провести «битву роботов».

Приложение 8
к дополнительной общеобразовательной
общеразвивающей программе

Оценочный лист освоения программы «С++=Я++»

№ п/п	Ф.И. обучающегося	Критерии оценки				Итог
		Работа на уроке (1 — 5 баллов)	Самостоятельная работа (1 — 5 баллов)	Выполнение практических заданий (1 — 5 баллов)	Выполнение контрольных тестов и заданий (1 — 5 баллов)	Итоговая оценка
1	Раздел Программы					

Приложение 9
к дополнительной общеобразовательной
общеразвивающей программе

Оценка творческих проектов

№ п/п	Ф.И. обучающегося	Критерии оценки					Итог
		Соответствие проекта заданию (1-5 баллов)	Оригинальность идеи и содержания проекта (1-5 баллов)	Сложность проекта (1-5 баллов)	Качество исполнения (1-5 баллов)	Качество презентации (1-5 баллов)	Итоговая оценка
1							
2							

_____ _____(_____)

(дата) (подпись) (расшифровка)