

Хазиахметов Максим Шамилиевич

Вождаев Анатолий Юрьевич

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++»**

Срок реализации – 2 года

Количество часов в год – 84 часа

Общее количество часов – 168 часов

Возраст обучающихся – 12-18 лет

Раздел 1. Пояснительная записка

Программа реализуется в рамках **технической направленности**.

Уровень программы – **базовый**.

Актуальность программы. Язык C является универсальным языком программирования и позволяет решить практически любую задачу в данной области. Семейство языков программирования C очень обширно, существуют различные реализации языка. Поэтому здесь и далее под языком C будут подразумеваться две его реализации, а именно – C++ и C#, которые изучаются в настоящей программе,

Поскольку системное программное обеспечение часто бывает написано на языке C, то и программные интерфейсы к подсистемам ОС также часто пишут на C. Соответственно, те программы, которые взаимодействуют с операционными системами, включая и прикладные, написаны на языке C.

C – современный язык. Он включает управляющие конструкции, которые рекомендуются теоретическим и практическим программированием. Его структура побуждает программиста использовать в своей работе нисходящее проектирование, структурное программирование и пошаговую разработку модулей. Результатом такого подхода является надежная и читаемая программа.

C – эффективный язык. Его структура позволяет наилучшим образом использовать возможности современных ЭВМ. Написанные на языке C программы обычно отличаются компактностью и быстротой исполнения.

C – переносимый (или мобильный) язык. Это означает, что программа,

написанная на С для одной вычислительной системы, может быть перенесена с небольшими изменениями или вообще без них на другую.

С – мощный и гибкий язык.

С – удобный язык. Он достаточно структурирован, чтобы поддерживать хороший стиль программирования, и вместе с тем не связывает ограничениями. Обработка сложных структур данных – текста, бизнес-информации, Internet-страниц и т.п. – одна из наиболее распространенных возможностей применения языка. В прикладном программировании, наверное, проще назвать те области, где язык С применяется мало.

Программа, в отличие от других типовых программ, предусматривает полное обучение языку С и в консольном варианте, и в варианте 32-разрядных приложений Windows.

Цель программы – обучение программированию на языке С, развитие у обучающихся навыков в области программирования и компьютерной графики для формирования у них основ информационной культуры.

Задачи:

Обучающие:

- познакомить обучающихся с основами программирования;
- научить их решать основные типовые задачи;
- научить написанию простейших программ с использованием структур и элементов языка С;
- научить написанию программ в среде разработки Visual Studio .NET;
- познакомить с особенностями программирования под Windows;
- научить написанию программ со сложной структурой;
- научить использовать полученные знания в описании и оформлении итогового продукта деятельности в рамках программирования.

Развивающие:

- развить навыки работы с компьютером;
- развивать любознательность, наблюдательность, память, пространственное мышление, воображение;
- развивать интерес к творческой деятельности;
- развить навыки самостоятельного использования знаний и умений в

области программирования;

- сформировать и развивать творческий подход к решению творческо-продуктивных задач в области программирования;
- развивать способность самостоятельно действовать, выбирать способ решения задач;
- сформировать устойчивую мотивацию, направленную на решение творческих задач в рамках создания итогового продукта (программы).

Воспитательные:

- воспитывать чувство коллективизма, чувство ответственности за свою деятельность, уважительное отношение к сверстникам и взрослым;
- воспитывать потребность в самообразовании, целеустремленность, усидчивость и терпение в достижении творческих результатов.

Возраст детей – 12-18 лет. Количество детей в группе 8-12 человек.

Формы и режим занятий. Занятия по программе проводятся в форме лекций и практических занятий за компьютерами. На лекциях дается материал очередной темы. После каждой лекции проходят практические занятия, на которых обучающиеся пишут программы по изученной теме. Именно в ходе таких занятий обучающиеся получают основные навыки.

Задания каждому обучающемуся даются индивидуально. Это помогает развить в каждом из них навыки самостоятельной работы.

Занятия проходят 1 раз в неделю по 2 часа. Во время занятий предусмотрен перерыв 15 минут (каждый час) на отдых, физкультминутки, проветривание помещения.

Сроки реализации программы: программа рассчитана на 2 года, 84 часа в год. Общее количество часов, запланированных на реализацию программы – 168 часов.

Планируемые результаты. По итогам реализации программы обучающиеся **будут знать:**

- основы программирования на языке C++;
- среду разработки Visual Studio .NET;
- основы программирования на языке C#;
- особенности программирования под Windows.

Обучающиеся *будут уметь*:

- решать основные типовые задачи;
- писать простейшие программы с использованием структур и элементов языка С;
- писать программы под Windows со сложной структурой.

Раздел 2. Формы аттестации и оценочные материалы ***Формы контроля и подведения итогов реализации дополнительной общеразвивающей программы***

Для каждого года обучения выделены основные параметры. В соответствии с данными параметрами производится оценка знаний и умений обучающихся по указанным критериям.

Таблица с измеряемыми параметрами и соответствующими им оценкам знаний и умений приведена ниже.

Проверка результатов изучения каждой темы осуществляется посредством написания программы, решающей задачу по изученной теме. После изучения каждого подраздела программы обучающемуся выдается контрольно-экзаменационный материал, позволяющий оценить, насколько усвоено содержание подраздела темы. В качестве итоговой работы обучающийся по согласованию с преподавателем сам разрабатывает тему для итоговой программы, которая будет представлять собой проект, охватывающий большую часть содержания, изученного в течение года.

Теоретические знания проверяются посредством опросов в начале каждого занятия и итогового тестирования после изучения очередной темы.

**Механизм оценивания результативности освоения программы
Параметры оценивания знаний, умений и навыков обучающихся**

| Измеряемые параметры | Критерии оценки (1-й год обучения) | | | |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| | Минимальный уровень знаний (1 балл) | Допустимый уровень знаний (2 балла) | Приемлемый уровень знаний (3 балла) | Оптимальный уровень знаний (4 балла) |
| 1. Знания в области техники безопасности | | | | |
| 1.1. Знание правил внутреннего распорядка и техники безопасности при работе с вычислительной техникой. | Обучающийся не знает правила ТБ при эксплуатации вычислительной техники | Обучающийся имеет слабые знания по правилам ТБ при эксплуатации вычислительной техники | Обучающийся хорошо разбирается в правилах ТБ при эксплуатации вычислительной техники | Обучающийся отлично знает и соблюдает правила ТБ при эксплуатации вычислительной техники |
| 2. Теоретические знания в области программирования | | | | |
| 2.1. Основы синтаксиса языка С++. Основные структурные элементы языка С++, встроенные типы данных языка С++. | Обучающийся не знает основ языка С++ и не владеет основными структурными элементами языка и встроенными типами данных | Обучающийся на низком уровне знает основы синтаксиса языка С++ и слабо ориентируется в структурных элементах языка и встроенных типах данных | Обучающийся уверенно ориентируется в основах синтаксиса языка С++, его структурных элементах и встроенных типах данных | Обучающийся отлично знает основы синтаксиса языка С++, а также уверенно использует основные структурные элементы языка и его встроенные типы данных |
| 2.2. Управляющие элементы языка С++. Производные типы данных. | Обучающийся не знает управляющие элементы и производные типы данных языка С++ | Обучающийся слабо знает управляющие элементы и производные типы данных языка С++ | Обучающийся на хорошем уровне знает и использует управляющие элементы и производные типы данных языка С++ | Обучающийся отлично знает, понимает и использует управляющие элементы и производные типы данных языка С++ |
| 2.3. Структуры. Классы и объекты. Виды отношений между классами. | Обучающийся не имеет понятия об объектно ориентированных типах языка С++ (структуры и классы) | Обучающийся имеет слабое представление о взаимоотношении объектов в структурах и классах | Обучающийся на хорошем уровне знает структуры и классы языка С++, уверенно использует взаимоотношения между объектами структур и классов | Обучающийся отлично разбирается в основах объектно ориентированного программирования и в основных его парадигмах |
| 3. Практические навыки в области программирования | | | | |

| | | | | |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 3.1. Решение простейших задач путем написания программ на языке C++. | Обучающийся не может решать элементарные задачи при помощи написания простейших программ на языке C++ | Обучающийся не всегда может написать программы на языке C++, решающие простейшие задачи | Обучающийся почти всегда самостоятельно пишет простейшие программы на языке C++ | Обучающийся самостоятельно решает элементарные задачи путем написания простейших программ на языке C++ |
| 3.2. Разработка нескольких алгоритмов для решения одной задачи. | Обучающийся не может решить задачу даже с помощью хотя бы одного алгоритма | Обучающийся решает поставленную задачу только при помощи одного алгоритма | Обучающийся пытается искать и предлагает новые алгоритмы (не всегда правильные) для решения поставленной задачи | Обучающийся самостоятельно придумывает и реализовывает различные алгоритмы для решения поставленной задачи |
| 3.3. Решение задач, ориентированных на составление математических алгоритмов для их решения. | Обучающийся не решает задачи, ориентированные на математические алгоритмы | Обучающийся не всегда может решить задачи на математические алгоритмы ввиду нехватки соответствующих знаний по математике | Обучающийся решает задачи на составление математических алгоритмов | Обучающийся решает задачи на составление математических алгоритмов в том числе предлагая свои необычные варианты их составления |
| 4. Личностные качества ребенка | | | | |
| 4.1. Самостоятельность | Обучающийся не может работать самостоятельно | Обучающийся плохо работает самостоятельно | Обучающийся стремится работать самостоятельно, но это не всегда удается | Обучающийся всегда работает самостоятельно |
| 4.2. Коммуникабельность | Обучающийся замкнут, не идет на контакт с товарищами и педагогом | Обучающийся редко обращается за помощью и плохо контактирует с товарищами | Обучающийся общается с товарищами, обращается за помощью, но при этом сам не всегда помогает товарищам | Обучающийся общается с товарищами и всегда им при необходимости помогает |

| | | | | |
|-----------------|-----------------------------------------------------------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 4.3. Трудолюбие | Обучающийся не хочет писать программы, не проявляет стремление к получению знаний | Обучающийся неохотно пишет программы и исправляет ошибки | Обучающийся охотно принимается за решение поставленной задачи, но не всегда исправляет ошибки после первого трудового порыва | Обучающийся всегда охотно пишет программы, кропотливо их проверяет на наличие ошибок и исправляет последние при их выявлении |
|-----------------|-----------------------------------------------------------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|

Приведенные в данной таблице измеряемые параметры и критерии оценки используются и на втором году обучения с тем отличием, что обучающиеся осваивают язык программирования С#.

Обязательным для каждого обучающегося является участие в семинарах, конкурсах, научно-практических конференциях, где оценивается степень овладения программным материалом и проектными навыками. Участие в них будет развивать у обучающегося потребность творческого подхода к решению той или иной поставленной задачи, способность самостоятельно концентрировать и выражать основные идеи и подходы к созданию продукта собственной творческой деятельности.

Аттестация обучающихся проводится в соответствии с Положением об аттестации. По итогам освоения программы выдается свидетельство.

Раздел 3. Содержание программы Учебно-тематический план Первый год обучения

| № п/п | Название раздела, темы | Количество часов | | | Формы аттестации (контроля) по разделам |
|-------|--------------------------------------------------------------------|------------------|---------------|--------------|-----------------------------------------|
| | | Всего | Теоретических | Практических | |
| 1. | Вводное занятие | 2 | 2 | – | Опрос |
| 2. | Имена, переменные и константы. Операции и выражения. Операторы. | 8 | 4 | 4 | Анализ проб и ошибок |
| 3. | Встроенные типы данных. Преобразование типов. | 8 | 4 | 4 | Анализ проб и ошибок |
| 4. | Циклы и другие управляющие средства. Структурное программирование. | 12 | 6 | 6 | Анализ проб и ошибок |

| | | | | | |
|----|----------------------------------------------------------|-----------|-----------|-----------|----------------------|
| 5. | Функции. | 8 | 2 | 6 | Анализ проб и ошибок |
| 6. | Производные типы данных. Массивы и указатели. Структуры. | 14 | 6 | 8 | Анализ проб и ошибок |
| 7. | Распределение памяти. | 12 | 6 | 6 | Анализ проб и ошибок |
| 8. | Классы и объекты. Производные классы, наследование. | 14 | 6 | 8 | Анализ проб и ошибок |
| 9. | Итоговое занятие | 6 | 2 | 4 | Зачет |
| | Всего часов: | 84 | 38 | 46 | |

Содержание учебно-тематического плана Первый год обучения

1. *Вводное занятие*

Основные правила и требования техники безопасности и противопожарной безопасности при работе в помещении компьютерного класса. Инструктаж по технике безопасности.

История создания языка и его эволюция. Международный стандарт языка. Сферы применения языка C++. Достоинства языка C++. Будущее языка C++. Использование языка C++. Пример простой программы на языке C++. Структура простой программы.

2. *Имена, переменные и константы. Операции и выражения. Операторы*

Правила именования переменных и функций языка, правила записи констант. Понятие ключевого или зарезервированного слова, список ключевых слов C++. Основные операции. Дополнительные операции. Перечень операций языка C++. Выражения. Простейшие выражения. Операторы. Уставные операторы.

Практическая часть. Написание первой программы на определение переменных, применение операций над переменными.

3. *Встроенные типы данных. Преобразование типов*

Рассмотрение всех встроенных типов языка C++: целые числа разной разрядности, вещественные числа, логические величины, перечисляемые значения, символы и их кодировка. Эквивалентность типов. Преобразование типов. Неявное преобразование типа. Арифметические преобразования. Явные преобразования типов. Синтаксис типов.

Практическая часть. Написание программы, работающей со встроенными типами данных. Переопределение целочисленных и вещественных типов.

4. Циклы и другие управляющие средства. Структурное программирование

Цикл с предусловием. Цикл со счетчиком. Цикл с постусловием. Другие управляющие операторы. Структурное программирование.

Практическая часть. Написание программ, работающих со встроенными типами циклов.

5. Функции

Функции с переменным количеством аргументов. Функции – это основные единицы построения программ при процедурном программировании на языке C++. Правила их записи, вызова и передачи параметров. Издание и использование функций. Аргументы функции. Возвращение значений. Локальные переменные. Нахождение адресов. Указатели, первое знакомство.

Практическая часть. Написание программ, оперирующих как встроенными, так и пользовательскими функциями.

6. Производные типы данных. Массивы и указатели. Структуры

Создание и использование массивов, структур, объединений, указателей. Адресная арифметика. Строки и литералы. Массивы. Указатели. Динамические объекты. Создание динамических объектов. Доступ к динамическим объектам. Строки – дополнительные сведения о связи между указателями и массивами. Инициализация массивов и классы памяти. Функции. Операции с указателями. Определение структурных переменных. Доступ к компонентам структуры. Поля битов в структурах. Объединения. Перечисления. Переменные структуры. Указатели структуры. Массив структур. Переименования типов.

Практическая часть. Написание программ работы со статическими и динамическими массивами, различные виды сортировки массивов, использование структур и массивов структур.

7. Распределение памяти

Проблемы при явном распределении памяти в C++, способы их решения. Ссылки и указатели. Распределение памяти под переменные, управление памятью с помощью переопределения операторов `new` и `delete`.

Практическая часть. Написание программы, использующей динамически созданные переменные. Примеры написания программ, некорректно работающих с памятью.

8. *Классы и объекты. Производные классы, наследование*

Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Наследование, виды наследования. Виртуальные методы. Абстрактные классы. Множественное наследование.

Практическая часть. Написание программ, построенных на объектно ориентированных принципах программирования. Изучение различных видов взаимоотношений между классами.

9. *Итоговое занятие*

Планирование занятий на следующий год.

Практическая часть. Проверка итоговых работ за учебный год.

Учебно-тематический план Второй год обучения

| № п/п | Название раздела, темы | Количество часов | | | Формы аттестации (контроля) по разделам |
|-------|----------------------------------------------------------------------------|------------------|----------------|--------------|-----------------------------------------|
| | | Всего | Т еоретических | Практических | |
| 1 | Вводное занятие | 2 | 2 | – | Опрос |
| 2 | Первое знакомство с Visual Studio .Net. Visual Studio .Net, Framework .Net | 6 | 2 | 4 | Анализ ошибок и успехов |
| 3 | Язык С# и первые проекты | 6 | 2 | 4 | Анализ ошибок и успехов |
| 4 | Система типов языка С# | 8 | 4 | 4 | Анализ ошибок и успехов |
| 5 | Присваивание и встроенные функции. | 10 | 3 | 7 | Анализ ошибок и успехов |
| 6 | Процедуры и функции – методы класса | 7 | 2 | 5 | Анализ ошибок и успехов |
| 7 | Корректность методов. Рекурсия | 18 | 8 | 10 | Анализ ошибок и успехов |

| | | | | | |
|---------------------|----------------------------------------------------------------------------------|-----------|-----------|-----------|-------------------------|
| 8 | Массивы языка С#. Класс Array и новые возможности массивов | 10 | 3 | 7 | Анализ ошибок и успехов |
| 9 | Символы и строки постоянной длины в С#. Строки С#. Классы String и StringBuilder | 15 | 4 | 11 | Анализ ошибок и успехов |
| 10 | Итоговое занятие | 2 | 1 | 1 | Защита проекта зачет |
| Всего часов: | | 84 | 31 | 53 | |

Содержание учебно-тематического плана *Второй год обучения*

1. *Вводное занятие*

Повторение основных правил и требований техники безопасности и противопожарной безопасности при работе в помещении компьютерного класса. Инструктаж по технике безопасности. Бренд .Net. Visual Studio .Net – открытая среда разработки. Общая структура и элементы.

Практическая часть. Краткое повторение пройденного за первый учебный год материала.

2. *Первое знакомство с Visual Studio .Net. Visual Studio .Net, Framework .Net*

Каркас Framework .Net. Библиотека классов FCL – статический компонент каркаса. Общеязыковая исполнительная среда CLR – динамический компонент каркаса. Управляемый код. Общеязыковые спецификации CLS и совместимые модули.

Практическая часть. Создание первого проекта в среде Microsoft Visual Studio. Подключение среды разработки Microsoft .Net.

3. *Язык С# и первые проекты*

Создание языка. Его особенности. Решения, проекты, пространства имен. Консольные и Windows-приложения С#, построенные по умолчанию.

Практическая часть. Изучение встроенных типов проектов в среде Microsoft Visual Studio. Разделение структуры программы на пространства имен.

4. *Система типов языка С#*

Общий взгляд. Система типов. Типы-значения и ссылочные типы. Встроенные типы. Сравнение с типами С++. Типы или классы? И типы, и классы!

Преобразования переменных в объекты и vice versa. Операции «упаковать» и «распаковать». Преобразования типов. Преобразования внутри арифметического типа. Преобразования строкового типа. Класс Convert и его методы. Проверяемые преобразования. Управление проверкой арифметических преобразований.

Практическая часть. Написание программ, работающих со встроенными типами данных C#. Изучение методов преобразования типов в C#. Использование встроенных методов класса Convert для совместной работы с типами всех категорий данных.

5. Присваивание и встроенные функции

Присваивание. Новинка C# – определенное присваивание. Классы Math, Random и встроенные функции.

Практическая часть. Изучение новых конструкций в операции присваивания в C# по сравнению с C++. Написание программы, использующей объекты и встроенные функции классов Math и Random.

6. Процедуры и функции – методы класса

Процедуры и функции – две формы функционального модуля. Чем отличаются эти формы? Процедуры и функции – это методы класса. Описание методов (процедур и функций). Синтаксис. Атрибуты доступа. Статические и динамические методы. Формальные аргументы. Статус аргументов. Тело методов. Вызов процедур и функций. Фактические аргументы. Семантика вызова. Поля класса или аргументы метода? Поля класса или функции без аргументов? Проектирование класса Account. Функции с побочным эффектом. Перегрузка методов.

Практическая часть. Практика в написании функций и процедур. Разработка структуры и написание класса Account.

7. Корректность методов. Рекурсия

Корректность метода. Спецификации. Триады Хоара. Предусловие метода. Постусловие метода. Корректность метода по отношению к предусловию и постусловию. Частичная корректность. Завершаемость. Полная корректность. Инвариант цикла. Вариант цикла. Подходящий инвариант. Корректность циклов. Рекурсия. Прямая и косвенная рекурсия. Стратегия «разделяй и властвуй». Сложность рекурсивных алгоритмов. Задача «Ханойские башни». Быстрая

сортировка Хоара.

Практическая часть. Написание программ и их анализ на соответствие корректности методов. Практическое рассмотрение механизма рекурсии. Задача «Ханойские башни». Быстрая сортировка Хоара.

8. **Массивы языка C#. Класс Array и новые возможности массивов**

Общий взгляд на массивы. Сравнение с массивами C++. Почему массивы C# лучше, чем массивы C++. Виды массивов: одномерные, многомерные и изрезанные. Динамические массивы. Семейство классов-массивов. Родительский класс Array и наследуемые им интерфейсы. Новые возможности массивов в C#. Как корректно работать с массивами объектов?

Практическая часть. Написание программы, использующей и наглядно показывающей преимущества массивов в C# по сравнению с C++.

9. **Символы и строки постоянной длины в C#. Строки C#. Классы String и StringBuilder.**

Общий взгляд. Строки в C++. Строки в C#. Строки постоянной и переменной длины. Классы char, char[]. Строки C#. Класс String. Изменяемые и неизменяемые строковые классы. Классы Net Framework, расширяющие строковый тип. Класс StringBuilder.

Практическая часть. Сравнение при написании программ возможностей, предоставляемых встроенными строковыми типами в C# по сравнению с C++. Написание программ, работающих со встроенными строковыми классами среды разработки Microsoft .Net.

10. **Итоговое занятие**

Практическая часть. Защита творческих проектов. Проверка итоговых работ за учебный год. Аттестация обучающихся. Выдача свидетельств.

Раздел 4. Организационно-педагогические условия реализации программы Учебно-методическое обеспечение программы

Программа занятий ориентирована на работу с обучающимися в компьютерных классах, оборудованных современной вычислительной техникой с возможностью выхода в глобальную сеть Интернет, с обеспечением каждого обучающегося отдельным рабочим местом.

В ходе реализации программы изучаются классические теоретические

основы языка программирования С, дается формальное изложение всех конструкций языка программирования С. Материал наполнен примерами и тематическими задачами. Все конструкции языка С, независимо от частоты использования, синтаксической и семантической сложности, описаны одинаково кратко, но понятно и исчерпывающе. Предусмотрено обучение всем основным возможностям языка С и их применению при разработке объектно-ориентированных программ. Также дается краткое описание библиотек языка С, необходимых для создания типичных программ.

После изучения большого раздела программы обучающиеся будут писать программу по изученной тематике. Причем придумывать и реализовывать программу обучающиеся будут сами. Это поможет развивать у них фантазию и творческое мышление.

Вначале обучения проводится собеседование и тестирование. Тестирование поможет выявить уровень знаний по информационным технологиям.

В процессе обучения педагог действует в такой последовательности:

1. Лекция.

На лекции раскрывается весь теоретический материал данного занятия. Педагог объясняет текущую тему и отвечает на возникающие вопросы.

2. Проветривание помещения.

Необходимая для комфорта обучающихся процедура. На проветривание отводится 15 минут. В это время педагог будет отвечать на индивидуальные вопросы обучающихся.

3. Практическая часть.

На практическом занятии обучающиеся выполняют задания педагога непосредственно за компьютером. Педагог будет отвечать индивидуально на вопросы обучающихся и помогать в разрешении возникающих трудностей.

Для успешной реализации программы необходимы: класс – помещение для проведения лекционных занятий со всем необходимым для этого оборудованием (столы, стулья, доска, мел, хорошая освещенность, наличие открывающихся окон и т.д.).

Материально-технические условия реализации программы.

Для проведения занятий необходимо достаточно просторное помещение, которое должно быть хорошо освещено и оборудовано необходимой мебелью: столы, стулья, шкафы-витрины для хранения материалов, специального инструмента, приспособлений, чертежей, моделей. Для работы необходимо иметь достаточное количество наглядного и учебного материала и ТСО.

Для реализации программы необходимы:

1. Программное обеспечение для проектной деятельности (Microsoft Office).
2. Компьютеры.
3. Системное программное обеспечение (Windows).
4. Принтер.
5. Картридж, бумага.
6. Проектор мультимедийный.
7. Экран для проектора.
8. Сканер.
9. Тренировочные поля (основание поля – ламинированная ДСП, препятствия: горка, коробки, стенки, изолента черная, изолента красная, оргстекло, сетка, банки и др.).

Учебно-информационное обеспечение программы

Нормативно-правовые акты и документы

1. Федеральный закон от 29.12.2012 N273-ФЗ «Об образовании в Российской Федерации».
2. Концепция развития дополнительного образования детей (утверждена распоряжением Правительства Российской Федерации от 04 сентября 2014 г. №1726-р.).
3. Приказ Министерства образования и науки РФ от 29 августа 2013 г. №1008 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
4. Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы): приложение к письму Министерства образования и науки Российской Федерации от 18.11.15 №09-3242.

5. «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы дошкольных образовательных организаций» (СанПиН 2.4.1.3049-13).

6. «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей». (СанПиН 2.4.4.3172-14).

7. Приказ Департамента образования города Москвы от 17.12.2014 г. №922 «О мерах по развитию дополнительного образования детей в 2014-2015 году».

8. Приказ Департамента образования города Москвы от 7.08.2015 г. №1308 «О внесении изменений в приказ Департамента образования города Москвы от 17 декабря 2014 г. № 922».

9. Приказ Департамента образования города Москвы от 08.09.2015 г. № 2074 «О внесении изменений в приказ № 922 от 17.12.2014 г..

10. Устав Государственного бюджетного профессионального образовательного учреждения города Москвы «Воробьевы горы».

11. Приказ Государственного бюджетного профессионального образовательного учреждения города Москвы «Воробьевы горы» от 25.04.2016 №851 «Об утверждении Положения о порядке разработки и реализации дополнительной общеразвивающей программы Государственного бюджетного профессионального образовательного учреждения города Москвы «Воробьевы горы».

Списки рекомендуемой литературы

Литература для педагога

1. Страуструп Б. Язык программирования C++. Специальное издание. – М.: Бином, Невский Диалект, 2004. – 1104 с.

Литература для обучающихся

2. Страуструп Б. Язык программирования C++. Специальное издание. – М.: Бином, Невский Диалект, 2004. – 1104 с.

3. Страуструп Б. Дизайн и эволюция C++. Классика CS. – СПб. Питер, 2006. – 448 с.

- Фленов М. Программирование на С++ глазами хакера. – СПб: ВHV-СПб, 2004. – 336 с.
4. Хенкеманс Д., Ли М. Программирование на С++. – СПб. Символ-Плюс, 2004. – 416 с.
 5. Эккель Б., Эллисон Ч. Философия С++. Практическое программирование. – СПб. Питер, 2004. – 608 с.
 6. Бланшет Ж., Саммерфилд М. Программирование GUI на С++. – СПб. Кудиц-Пресс 2005. – 448 с.
 7. Глушаков С. Программирование на Visual С++. – М.: АСТ, Сталкер, 2002. – 130 с.
 8. Рихтер Д. Программирование на платформе MS .NET Framework. 3-е изд. – СПб. Питер, 2006. – 512 с.
 9. Страуструп Б. Язык программирования С++. Специальное издание. – М.: Бином, Невский Диалект, 2004. – 1104 с.
 10. Страуструп Б. Дизайн и эволюция С++. Классика СS. – СПб. Питер, Сталкер, 2006. – 448 с.
 11. Шеферд Джордж. Программирование на Microsoft Visual С# .NET. / Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 2003. – 928 с.

Кадровое обеспечение программы

Для успешной реализации данной программы необходим педагог дополнительного образования, имеющий опыт преподавания в области робототехники, электроники и программирования, а также технической направленности.

Календарный учебный график¹
Первый год обучения

| № п/п | Месяц | Дата | Форма занятия | Кол-во часов | Тема занятия | Место проведения | Форма контроля |
|-------|-------|------|---------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------|
| 1. | | | Практическая работа | 2 | Вводное занятие | | Опрос |
| 2. | | | Практическая работа | 2 | Понятие алгоритм. Блок-схемы. Основные структуры: следование, условие, цикл. Системы счисления. Двоичное кодирование. | | Анализ ошибок и успехов |
| 3. | | | Практическая работа | 2 | Понятие программирования. Обзор языков программирования. Язык C/C++. Средства разработки. Представление программы в памяти компьютера. | | Анализ ошибок и успехов |
| 4. | | | Практическая работа | 2 | Основные типы данных языка C. Операции и операторы. Синтаксис программ и основные конструкции языка. | | Анализ ошибок и успехов |
| 5. | | | Практическая работа | 2 | Решение задач объединяющих в себе темы алгоритмизации и арифметических операций. Функции посимвольного ввода\вывода. Написание простейшей программы копирования файла. | | Анализ ошибок и успехов |
| 6. | | | Практическая работа | 2 | Директивы препроцессора. Решение задач, связанных с посимвольной обработкой информации. | | Анализ ошибок и успехов |
| 7. | | | Практическая работа | 2 | Преобразование типов. Функции printf(), scanf() | | Анализ ошибок и успехов |
| 8. | | | Практическая работа | 2 | Решение задач на тему «Ввод/вывод» | | Анализ ошибок и успехов |
| 9. | | | Практическая работа | 2 | Функции. Основные сведения о функциях. Области видимости переменных. Блочная структура. | | Анализ ошибок и успехов |
| 10. | | | Практическая работа | 2 | Решение задач на тему «Массивы» | | Анализ ошибок и успехов |
| 12. | | | Практическая работа | 2 | Массивы. Реализация массивов в языке C. | | Анализ ошибок и успехов |
| 13. | | | Практическая работа | 2 | Рекурсия. Препроцессоры языка. | | Анализ ошибок и успехов |
| 14. | | | Практическая работа | 2 | Особенности рекурсивных алгоритмов. | | Анализ ошибок и успехов |
| 15. | | | Практическая работа | 2 | Указатели. Адресная арифметика. Файлы. Типы. Ввод-вывод | | Анализ ошибок и успехов |
| 16. | | | Практическая работа | 2 | Массивы указателей. Многомерные массивы. | | Анализ ошибок и успехов |
| 17. | | | Практическая работа | 2 | Решение задач с использованием многомерных массивов | | Анализ ошибок и успехов |
| 18. | | | Практическая работа | 2 | Способы отладки программ. Практическая работа «Debugging» | | Анализ ошибок и успехов |
| 19. | | | Практическая работа | 2 | Обобщающее занятие по языку программирования C. | | Анализ ошибок и успехов |
| 20. | | | Практическая работа | 2 | Обзор IDE для языков C/C++ | | Анализ ошибок и успехов |
| 21. | | | Практическая работа | 2 | Парадигмы программирования. Основные отличия C от C++. | | Анализ ошибок и успехов |
| 22. | | | Практическая работа | 2 | Абстрактные контейнерные типы. | | Анализ ошибок и успехов |
| 23. | | | Практическая работа | 2 | Решение задач на тему «Классы» | | Анализ ошибок и успехов |
| 24. | | | Практическая работа | 2 | Перегрузка функций и операторов. | | Анализ ошибок и успехов |
| 25. | | | Практическая работа | 2 | Шаблоны функций. Написание собственного шаблона. | | Анализ ошибок и успехов |
| 26. | | | Практическая работа | 2 | Исключения. Ошибки выполнения программы. Обработка исключений. | | Анализ ошибок и успехов |
| 27. | | | Практическая работа | 2 | Определение класса. Объекты классов. Функции-члены класса. | | Анализ ошибок и успехов |
| 28. | | | Практическая работа | 2 | Конструкторы и деструкторы класса. | | Анализ ошибок и успехов |
| 29. | | | Практическая работа | 2 | Шаблоны классов. Наследование и подтипизация классов. | | Анализ ошибок и успехов |
| 30. | | | Практическая работа | 2 | Множественное наследование. Виртуальное наследование. Применение наследования в C++. | | Анализ ошибок и успехов |

¹ Календарный учебный график приводится как пример. Календарный учебный график второго года обучения составляется согласно учебно-тематическому плану второго года обучения.

| | | | | | | | |
|-----|--|--|---------------------|---|-----------------------------------------------------------------|--|-------------------------|
| 31. | | | Практическая работа | 2 | Фреймворки. .NET Различия версий | | Анализ ошибок и успехов |
| 32. | | | Практическая работа | 2 | Интерфейсы. Основные элементы и их свойства. Событийная модель. | | Анализ ошибок и успехов |
| 33. | | | Практическая работа | 2 | Основы проектирования user-friendly интерфейсов | | Анализ ошибок и успехов |
| 34. | | | Практическая работа | 2 | Создание собственных элементов управления. | | Анализ ошибок и успехов |
| 35. | | | Практическая работа | 2 | Решение и разбор алгоритмических задач. | | Анализ ошибок и успехов |
| 36. | | | Практическая работа | 2 | Практическая работа по проектированию ПО. | | Анализ ошибок и успехов |
| 37. | | | Практическая работа | 2 | Проектирование ПО творческие работы | | Анализ ошибок и успехов |
| 38. | | | Практическая работа | 2 | Обсуждение индивидуальных особенностей творческих работ. | | Анализ ошибок и успехов |
| 39. | | | Практическая работа | 2 | Выполнение итоговой творческой работы. | | Анализ ошибок и успехов |
| 40. | | | Практическая работа | 2 | Представление творческой работы. | | Анализ ошибок и успехов |
| 41. | | | Практическая работа | 2 | Творческие проекты | | Защита творческих |
| 42. | | | Практическая работа | 2 | Итоговое занятие. | | Зачет |