

1. РАБОЧАЯ ПРОГРАММА.

1.1. Задачи образовательной программы.

Программный материал объединен в целостную систему подготовки ИТ специалистов и предполагает решение следующих **основных задач**:

Обучающие:

- познакомить с основными технологиями, которые применяются совместно с программированием и смогут пригодиться в реальной жизни;
- анализ существующей системы и представление идей по усовершенствованию, включая анализ затрат-выгод;
- анализ и уточнение требований пользователя;
- составление детальных спецификаций для новых систем или для модификаций существующих систем;
- разработку систем программного обеспечения и тщательное тестирование программных решений;
- подготовку обучающих материалов для пользователей, обучение пользователей и представление программного решения пользователям;
- установку, развертывание и обслуживание программной системы.

Развивающие задачи:

- развить исследовательский интерес у обучающихся;
- развить творческое воображение, математическое и образное мышление обучающихся;
- развить умение работать с компьютерными программами и дополнительными источниками информации;
- развить навык планирования проекта, умение работать в группе.

Воспитательные задачи:

- формировать интерес к программированию;
- формировать коммуникативные навыки;
- формировать культуру безопасного труда при работе с компьютером.

Основной показатель работы кружка по программированию – является проектирование, разработка, тестирование программного продукта. Оформление сопутствующей рабочей документации.

Режим теоретической подготовки и учебно-тренировочной работы рассчитан на 72 часа занятий.

1.2. Содержание программы обучения.

В основу отбора и систематизации материала содержания программы положены принципы комплексности, преемственности и вариативности. Принцип комплексности ДОП выражен в теснейшей взаимосвязи всех сторон учебно - практического процесса: теоретической, практической, психологической подготовки, педагогического контроля. Принцип преемственности прослеживается в последовательности изложения теоретического материала по этапам обучения, в углублении и расширении знаний по вопросам теории в соответствии с требованиями возрастающего мастерства программистов, постепенном, от этапа к этапу усложнении содержания заданий. Принцип вариативности дает определенную свободу выбора средств и методов, в определении времени для подготовки. Исходя из конкретных обстоятельств, при решении той или иной педагогической задачи педагог может вносить свои коррективы в построении учебно-практических занятий, не нарушая общих подходов. Образовательный компонент программы предполагает обучение обучающихся, параллельно которому идёт включение воспитательного процесса, организованного через работу в команде. Данный процесс происходит не стихийно, а в результате целенаправленного педагогического воздействия.

Программа предполагает: индивидуальное консультирование в течение учебного года и предсоревновательную подготовку обучающихся.

В конце обучения по программе обучающиеся должны знать основные принципы программирования и принимать участие в чемпионатах, форсайтах по программированию.

Содержание самостоятельной работы включает в себя выполнение комплексов упражнений для повышения общей и специальной физической подготовки.

Примечание: В зависимости от индивидуальных возможностей учебной группы или при невозможности провести 72 часа занятий по объективным причинам (болезни, каникулы или др. уважительные причины) педагог может по своему усмотрению изменять порядок различных тем внутри учебного плана, или добавлять занятия в другие дни взамен пропущенных. Можно также исключать некоторые уроки или заменить на другие, которые хуже усваиваются.

1.3. Планируемые результаты обучения (личностные, метапредметные, предметные).

По результатам обучения обучающиеся должны:

Знать:

- принципы и методы, обеспечивающие продуктивную работу в команде;
- как взять на себя инициативу и быть предприимчивыми с целью выявления, анализа и оценки информации из различных источников;
- как подготовить соответствующую документацию об использовании разрабатываемой системы;
- диагностические подходы и подходящие к решению проблем системы или программные решения;
- тенденции и разработки в отрасли, включая новые платформы, языки, условные обозначения и технические навыки;
- как использовать новейшие технологии, которые будут применяться в сценарии программного решения, которое требуется для наглядного сложного бизнес-решения проблемы;
- как настроить, разработать и интегрировать в разработанное решение новейшие технологии и оборудование, которые будут способствовать лучшему бизнес-решению;
- важность использования системного анализа и методологий проектирования (например, унифицированного языка моделирования (Unified Modelling Language), программной платформы MVC (Model-View-Control), фреймворков, шаблонов проектирования);
- принципы построения хранилищ данных, необходимых для бизнес-аналитики / отчетов о состоянии выполненных работ;
- важность рассмотрения всех нормальных и ненормальных сценариев и обработки исключений;
- важность соблюдения стандартов (например, соглашения по формату кода, руководства по стилю, дизайна пользовательского интерфейса, управления каталогами и файлами);
- важность точного и постоянного контроля версий;
- важность использования существующего кода в качестве основы для анализа и модификации;
- принципы устранения распространенных проблем программных решений;
- важность отладки программных решений;
- важность тщательного тестирования программных решений.

Уметь:

- планировать производственный график на каждый день в соответствии с доступным временем и принимать во внимание временные ограничения и сроки сдачи работы;
- применять исследовательские навыки и методики, чтобы поддерживать уровень собственной осведомлённости в актуальных отраслевых руководствах;
- анализировать результаты собственной деятельности в сравнении с ожиданиями и потребностями клиента и организации;
- создавать корректную последовательность операций разрабатываемой системы, с необходимыми уведомлениями;
- готовить необходимую системную документацию по использованию, установке и запуску системы;
- использовать навыки грамотности для интерпретации и понимания системных спецификаций;
- поддержания уровня собственной осведомлённости в актуальных отраслевых руководствах.
- использовать аналитические навыки для синтезирования сложной или неоднородной информации и определения функциональных, нефункциональных требований спецификации
- анализировать системы с помощью моделирования и анализа вариантов
- использования (например, диаграммы прецедентов, описания прецедентов, описания действующих субъектов (актеров), диаграммы пакетов вариантов использования), структурного моделирования и анализа (например, объекты, классы, диаграммы классов предметной области), динамического моделирования и анализа (например, диаграммы последовательностей, диаграммы взаимодействия, диаграммы состояний, диаграммы деятельности), инструментов и методов моделирования (например, диаграмма сущностей и связей, нормализация, словарь данных).
- проектировать системы на основе диаграммы классов, диаграммы последовательностей, диаграммы состояний, диаграммы деятельности, описания объектов и пакетов, схемы реляционной или объектной базы данных и диаграмм потоков данных, структуры человеко-машинного интерфейса /механизма взаимодействия с пользователем;
- применять средства безопасности и контроля, структуры многозвенного приложения использовать системы управления базами данных для построения, хранения и управления структурами и наборами данных для требуемой системы;
- использовать подходящие версии программного обеспечения, среды разработки и инструменты, предназначенные для изменения существующего
- и написания нового исходного кода клиент-серверного программного обеспечения;
- использовать подходящие версии программного обеспечения, среды разработки и инструменты, предназначенные для изменения существующего
- и написания нового исходного кода для системной интеграции с использованием веб-решений, веб-сервисов или единой подписки (например, с использованием службы каталогов) или API;
- определять и интегрировать соответствующие библиотеки и фреймворки в программные решения;
- строить и обслуживать многоуровневые приложения;
- осуществлять отладку программных решений;
- разрабатывать тест-кейсы и проверять результаты тест-кейсов;
- устранять и исправлять ошибки в программных решениях.