

Правительство Российской Федерации
Министерство просвещения Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
«Всероссийский детский центр «Океан»

СОГЛАСОВАНО

Начальник управления общего и
дополнительного образования
ФГБОУ ВДЦ «Океан»

 М.И. Фролова
« 10 » января 2020 г.

УТВЕРЖДАЮ

Заместитель директора по
образовательной деятельности
ФГБОУ ВДЦ «Океан»

 Г.Г. Рыбкин
« 10 » января 2020 г.


Принята на заседании методического совета
Протокол № ____ от _____ г.

**Дополнительная
общеобразовательная общеразвивающая программа
технической направленности
«Intro-Coding»**

Возраст учащихся – 12 - 17 лет
Срок реализации – 1 смена (21 день)

Автор-составитель:
Аношкин Александр Николаевич,
педагог дополнительного образования

Владивосток, 2020 г.

Содержание

Раздел №1 «Комплекс основных характеристик программы»	3
1.1 Пояснительная записка.....	3
1.2 Планируемые результаты	8
1.3 Содержание программы.....	9
Раздел №2 «Комплекс организационно-педагогических условий»	12
2.1 Условия реализации программы	12
2.2 Анализ эффективности реализации программы	12
2.3 Методическое обеспечение программы	13
Глоссарий	15
Список источников	17
Раздел №3 «Приложения».....	18
3.1 Диагностические материалы	18

Раздел №1

«Комплекс основных характеристик программы»

1.1 Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Intro-coding»¹ ориентирована на развитие технического творчества, аналитических и логических способностей учащихся, формирование интереса к сфере современных информационных технологий и программирования.

Нормативная база

Программа разработана в соответствии с:

- Федеральным законом Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (ред. от 23.07.2013г.);
- Концепцией развития дополнительного образования детей (Распоряжение правительства РФ от 04.09.2014г. №1726-р);
- Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (Приказ Министерства образования и науки РФ от 29.08.2013г. №1008);
- Методическими рекомендациями по проектированию дополнительных общеразвивающих программ, включая разноуровневые программы (письмо Минобрнауки России от 18.11.2015г. № 09-3242);
- Примерными требованиями к программам дополнительного образования детей (Письмо Департамента молодежной политики, воспитания и социальной поддержки детей Минобрнауки России от 11.12.2006 № 06-1844);
- СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей»,

¹ «Intro-coding» (от англ. introduction to coding – введение в кодинг. Кодинг – от разг. кодирование – программирование)

утвержденного постановлением Главного государственного санитарного врача РФ от 04.07.2014 г. №41;

- Приказом Министерства просвещения РФ от 09 ноября 2018 г. N 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Программой развития ФГБОУ «ВДЦ Океан» на 2014-2020 гг., утвержденной распоряжением Правительства РФ от 16.12.2014г. № 2539-р;
- Уставом ФГБОУ ВДЦ «Океан»;
- Положением о дополнительной общеобразовательной общеразвивающей программе ФГБОУ ВДЦ «Океан».

Теоретико-методологическое обоснование Программы

Актуальность разработки данной программы обусловлена высокой скоростью развития в различных, не только технической и технологической, сферах деятельности общества. Модификация, эволюция, а также появление новых профессий в настоящее время и в недалёком будущем заставляет задуматься о необходимости получения навыков алгоритмизации и программирования для каждого ребенка. В независимости от выбранной сферы профессиональной деятельности в каждой из них, в той или иной степени необходимы данные навыки.

По данным Атласа новых профессий на период с 2020 до 2030 года исчезнут порядка 57 профессий, но при этом появятся около 188 новых, все они, без исключений, завязаны на работе с техникой. Понимание алгоритмов, их структуры, последовательности и взаимосвязи компонентов может оказать весомую помощь при развитии элементов системного мышления, что позволит проще осваивать инновационные, технически сложные устройства, воспринимая их, как целостную систему, зная, каким законам и правилам она подчиняется, определяя самостоятельно различные закономерности и развивая в себе «техническую интуицию». Это благотворно сказывается на способности быстро осваивать новые технологии, обдуманно и грамотно

включать их в собственную профессиональную деятельность, значительно повышая качество выполняемой работы.

Научно-технический прогресс является основным двигателем развития мировой экономики. Его результатом выступают технологические инновации, которые приводят к росту производительности труда, модернизации средств производства и трансформации действующего технологического уклада. Переход к шестому технологическому укладу открывает перед человечеством большие возможности.

Синтез достижений по основным технологическим направлениям (био- и нанотехнологии, геномная инженерия, мембранные и квантовые технологии, микромеханика, фотоника, термоядерная энергетика) может привести, например, к созданию квантового компьютера или искусственного интеллекта. Ввиду такого рода изменений ученикам, в будущем студентам, молодым специалистам, работникам данных сфер безусловно пригодятся данные навыки, а также сформированное аналитическое и логическое мышление, способности к поиску нестандартных путей решения поставленных задач, – подготовка основы для развития которых, также является одной из задач данной программы. Использование при обучении “открытого” программного обеспечения позволяет учащимся свободно использовать его на своих домашних устройствах.

Особенность образовательной программы заключается в ее реализации в условиях временного детского коллектива в сжатые сроки, ввиду чего имеет интенсивный режим.

Цель программы: Содействие освоению старшими школьниками базовых компетенций разработки программных приложений на основе языков программирования высокого уровня.

Задачи:

– познакомить с основными методами разработки программного обеспечения;

- сформировать комплекс базовых навыков программирования для их дальнейшего использования в учебной и профессиональной деятельности;
- развить аналитическое мышление, способности к поиску нестандартных путей решения поставленной задачи.

Основополагающим подходом работы педагога по данной программе является **системно-деятельностный**. Педагог в процессе образовательной деятельности при реализации программы должен опираться на следующие принципы:

Принцип деятельности заключается в том, что учащийся, получая знания не в готовом виде, а добывая их сам, осознает при этом содержание и формы своей учебной деятельности, понимает и принимает систему ее норм, активно участвует в их совершенствовании, что способствует активному успешному формированию его общекультурных и деятельностных способностей, общеучебных умений.

Принцип непрерывности означает такую организацию обучения, когда результат деятельности на каждом предыдущем этапе обеспечивает начало следующего этапа. Непрерывность процесса обеспечивается инвариативностью технологии, а также преемственностью между всеми ступенями обучения содержания и методики.

Принцип целостного представления о мире означает, что у ребенка должно быть сформировано обобщенное, целостное представление о мире (природе – обществе – самом себе), о роли и месте науки в системе наук.

Принцип психологической комфортности предполагает снятие стрессообразующих факторов учебного процесса, создание на занятиях доброжелательной атмосферы, ориентированной на реализацию идей педагогики сотрудничества.

Принцип вариативности предполагает развитие у учащихся вариативного мышления, то есть понимания возможности различных вариантов решения проблемы, формирование способности к систематическому перебору вариантов и выбору оптимального варианта.

Принцип творчества предполагает максимальную ориентацию на творческое начало в учебной деятельности школьников, приобретение ими собственного опыта творческой деятельности. Формирование способности самостоятельно находить решение нестандартных задач.

Принцип минимакса. Для реализации принципа минимакса учебное заведение должно предоставить учащийся максимальные возможности для обучения и обеспечить усвоение материала на минимальном уровне, который указан в Федеральном государственном образовательном стандарте.

Методы организации занятий.

- словесный (инструктажи, беседы, разъяснения);
- наглядный (фото и видеоматериалы);
- практический (разбор алгоритмов, блок-схем и программирование на одном из высоких языков: C++, PHP, Java, Python, Ruby);
- поисково-исследовательский, проектный, игровой.

Занятия проводятся в различных формах, таких как: дискуссия, учебная игра, лабораторно-практические занятия с использованием словесных (рассказ, беседа), графических (работа с блок-схемами), наглядных (таблицы, модели, видеоматериалы), социологических (анкетирование) материалов.

Общие сведения об условиях реализации Программы

Целевой аудиторией данной программы являются ученики 7 – 11 классов (12 – 17 лет), проявляющие интерес к сфере разработки программного обеспечения и программирования на языках высокого уровня (C++, PHP, Java, Python, Ruby).

Принцип набора – свобода выбора.

Количество учащихся в группе: от 5 до 10 человек (ориентировано на количество рабочих мест, из расчета 1 компьютер для 1 ученика), состав может быть разновозрастной и разнополый.

Уровень программы: стартовый.

Объём – 12 часов.

Срок реализации – 1 смена (21 день).

Продолжительность занятия – 2 академических часа.

Периодичность – 2-3 раза в неделю.

1.2 Планируемые результаты

Личностные:

- навыки самостоятельно и творчески реализовывать собственные замыслы;
- повышение своего образовательного уровня и уровня готовности к продолжению обучения с использованием ИКТ;
- навыки самооценки и рефлексии.

Метапредметные результаты:

- навыки поиска и структурирования информации;
- умение ориентироваться в своей системе знаний: отличать новое от уже известного;
- умение работать по предложенным инструкциям;
- знание принципов алгоритмизации, базовые компетенции для последующего самостоятельного развития аналитического и логического мышления посредством нахождения креативных решений при написании алгоритмов разработки и функционирования ПО.

Предметные результаты:

- знание принципов, особенностей и отличий разработки ПО, алгоритм и ключевые моменты процесса разработки;
- формирование базовых компетенций необходимых для работы с информационно-коммуникационными технологиями, в частности – языками программирования высокого уровня (C++, PHP, Java, Python, Ruby), средами разработки программного обеспечения, инструментами дизайна приложений, особенностями используемого ПО.

1.3 Содержание программы

Учебный план

№	Тема	Кол-во часов			Форма аттестации/контроля
		всего	теория	практика	
1	Введение в курс «Intro-coding». Фреймворк Qt Creator. Интерфейс. Возможности.	2	1,5	0,5	Опрос. Вводное анкетирование.
2	Основы алгоритмизации.	2	1	1	Результаты заданий, самодиагностика, диагностика валидными методами
3	Создание простого приложения на C++ в Qt. MessageBox, слои и отступы. Работа с окнами и приложение с авторизацией.	2	0,5	1,5	
4	Отображение изображения и статуса. Использование стилей, HTML, QCheckBox и QRadioButton. Создание меню с QAction, QMenu и QToolBar. Компонент QListWidget.	2	0,5	1,5	
5	Разработка собственного приложения в среде Qt Creator.	2	-	2	Разработанные и защищенные проекты, рефлексия, итоговое анкетирование.
6	Оценка результативности программы.	2	-	2	
	Всего	12	3,5	8,5	

Содержание занятий

Занятие № 1

Тема: «Введение в курс «Intro-coding». Фреймворк Qt Creator. Интерфейс. Возможности.»

Цель: Знакомство с содержанием курса, основными инструментами для последующей работы.

Теория: Понятия «Программирование» и «Разработка приложений». Принципы и особенности различных вариантов разработки программного обеспечения. Алгоритм процесса разработки. Ключевые моменты, ошибки и пути их устранения.

Практика: Освоение интерфейса и функциональных характеристик фреймворка Qt Creator.

Занятие № 2

Тема: «Основы алгоритмизации.»

Цель: Актуализация, обобщение и систематизация знаний по алгоритмизации.

Теория: Термины и определения: алгоритм, исполнитель алгоритма, среда исполнителя, СКИ (система команд исполнителя), способы описания алгоритма, виды алгоритмов.

Практика: Построение блок-схем. Освоение алгоритма действий элементарной программы и описание его на русском языке.

Занятие № 3

Тема: «Разработка простого приложения на C++ в Qt.»

Цель: Разработка простого приложения на C++ в Qt, изучение компонентов MessageBox, слоёв и отступов.

Теория: Функциональные характеристики и назначение компонентов MessageBox, слои и отступы.

Практика: Разработка простого приложения на C++. Окна, формы, группы объектов. Приложение с авторизацией.

Занятие № 4

Тема: «Отображение изображения и статуса. Использование стилей, HTML, QCheckBox и QRadioButton. Создание меню с QAction, QMenu и QToolBar. Компонент QListWidget.»

Цель: Освоение компоненты QCheckBox, QListWidget и QRadioButton. Меню с QAction, QMenu и QToolBar.

Теория: Функциональные характеристики и назначение компонентов QCheckBox, QAction, QMenu, QToolBar, QListWidget и QRadioButton.

Практика: Создание приложения с использованием меню с QAction, QMenu и QToolBar.

Занятие № 5 – 6

Тема: «Разработка собственного приложения в среде Qt Creator.»

Цель: Создание собственного приложения в программе Qt Creator. Оценка результативности программы.

Практика: Начало разработки: составление дизайна и алгоритма работы приложения. Основной блок разработки ПО: исправление недочетов, ошибок, тестирование готовых приложений. Анкетирование, устная рефлексия.

Раздел №2

«Комплекс организационно-педагогических условий»

2.1 Условия реализации программы

Для успешной реализации программы необходимо следующее материально-техническое обеспечение:

- помещение для постоянных занятий с хорошим освещением и доступом к сети Интернет;
- столы и стулья согласно ГОСТу и списочному составу учащихся;
- рабочие места учащихся: компьютеры с установленным ПО, соответствующим тематике курса (Qt Creator);
- рабочее место педагога: компьютер, интерактивная доска (экран), проектор.

Оснащение компьютерной техникой должно производиться из расчета: 1 рабочее место для 1 участника программы.

2.2 Анализ эффективности реализации программы

В процессе самостоятельной работы обучающихся над реализацией собственной задумки, а также этапе защиты проектов по итогу освоения программы посредством включенного наблюдения можно зафиксировать достижения участников программы, а также их внутренние приращения. Эти показатели определяются с помощью пакета диагностических методик, в который входят: анкетирование, тестирование, методика незаконченных предложений и прочие.

При успешном освоении программы учащиеся имеют сформированный набор базовых компетенций необходимых для работы с ИКТ, в частности – могут самостоятельно реализовывать собственные замыслы в процессе разработки приложений; умении декомпозировать задачи на автономные компоненты; распределять подзадачи при работе в группе; выстраивать алгоритм процесса разработки ПО; анализировать работоспособность и

самостоятельно устранять ошибки в функциональной системе собственных приложений.

Критерии и показатели уровня освоения программы:

Низкий уровень – на базе Qt Creator учащиеся умеют создавать простейшие приложения, используя подсказки или с помощью педагога. Не умеет искать и структурировать информацию.

Средний уровень – на базе Qt Creator учащиеся умеют создавать простейшие приложения без дополнительного функционала, но выполняют это самостоятельно, опираясь на полученные ранее знания, не используя дополнительные подсказки и не обращаясь за дополнительной консультацией к педагогу. Самостоятельно прорабатывают алгоритм работы программы и рисуют для нее блок-схему, могут объяснить другим участникам или посторонним людям принцип работы его ПО и функциональные особенности.

Высокий уровень – на базе Qt Creator учащиеся умеют создавать более сложные программные продукты, используя различные компоненты, изученные ранее. Самостоятельно прорабатывают алгоритм работы программы и рисуют для нее блок-схему, могут рассказать и объяснить другим участникам или посторонним людям, о чем его продукт и как он функционирует, а также какие компоненты и почему он использует.

Формы аттестации и демонстрации достижений: творческий продукт для выставки, защита проекта.

2.3 Методическое обеспечение программы

Ключевой основой методики программы является проблемно-поисковый метод.

Методика программы реализуется через:

- беседы: «Расскажи о себе», «Что такое алгоритм? Где в жизни мы встречаем примеры алгоритмов?», «Что такое программирование и чем оно отличается от разработки приложений»;
- учебная игра «Корпорация «Code».

Глоссарий

1. Qt Creator – фреймворк^[13] для создания программного обеспечения с графической оболочкой.
2. Алгоритмизация – процесс составления алгоритмов для решения поставленных прикладных задач.
3. Алгоритм – это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное число шагов.
4. Блок-схема – распространенный тип схем^[12] (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности. Правила выполнения регламентируются ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». Стандарт, в частности, регулирует способы построения схем и внешний вид их элементов.
5. Инновация – введенный в употребление новый или значительно улучшенный продукт (товар, услуга) или процесс.
6. Исполнитель алгоритма – это объект, выполняющий определенный набор действий (человек, животное, робот, компьютер)
7. Открытое программное обеспечение – распространяющееся бесплатно.
8. Рефлексия – это обращение внимания субъекта на самого себя и на своё сознание, в частности, на продукты собственной активности, а также какое-либо их переосмысление.
9. Система команд исполнителя – это все команды, которые исполнитель умеет выполнять.
10. Среда исполнителя – обстановка, в которой функционирует исполнитель.

11. Схема – графическое представление определения, анализа или метода решения задачи, в котором используются символы для отображения данных, потока, оборудования и т. д.

12. Технологический уклад – совокупность сопряжённых производств, имеющих единый технический уровень и развивающихся синхронно.

13. Фреймворк – заготовки, шаблоны для программной платформы, определяющие структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных модулей программного проекта.

14. Шестой технологический уклад – считается, что в мире пройдены 5 технологических укладов (начало Первой промышленной революции, эпоха пара, эпоха стали, эпоха нефти, эпоха компьютеров и телекоммуникаций), в настоящий момент наступает Шестой – Нанотехнологии.

Список источников

1. Гнездилов Г.Г., Абрамов С.А. и др. Задачи по программированию. М.: Наука, 2018.
2. Гейн А.Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 2013.
3. Лепехин Ю.В. Сорок пять минут с компьютером. – Волгоград: Перемена, 2016.
4. Единое окно доступа к образовательным ресурсам [Электронный ресурс] URL: <http://window.edu> (дата обращения: 14.07.2019)
5. Конструктор сайтов общеобразовательных учреждений и проектов [Электронный ресурс] URL: <http://edu.of.ru> (дата обращения: 14.07.2019)
6. Алгоритмы, методы, исходники [Электронный ресурс] URL: <http://algotist.manual.ru> (дата обращения: 07.08.2019)
7. Библиотека алгоритмов [Электронный ресурс] URL: <http://alglib.sources.ru> (дата обращения: 09.04.2019)
8. Математика и программирование [Электронный ресурс] URL: <http://www.mathprog.narod.ru> (дата обращения: 28.11.2019)
9. Виртуальный компьютерный музей [Электронный ресурс] URL: <http://www.computer-museum.ru> (дата обращения: 21.04.2019)
10. Газета «Информатика» издательского дома «Первое сентября» [Электронный ресурс] URL: <http://inf.1september.ru> (дата обращения: 21.04.2019)
11. Дискретная математика: алгоритмы (проект Computer Algorithm Tutor) [Электронный ресурс] URL: <http://rain.ifmo.ru/cat/> (дата обращения: 25.11.2019)
12. Задачи соревнований по спортивному программированию с проверяющей системой [Электронный ресурс] URL: <http://acm.timus.ru> (дата обращения: 14.07.2019)

Раздел №3 «Приложения»

3.1 Диагностические материалы

Анкета №1. Вводное анкетирование

1. Ф.И.О. _____
2. Регион _____
3. Отряд _____
4. Возраст _____
5. Класс _____
6. Который раз ты приехал в ВДЦ “Океан”? _____
7. Связаны ли твои увлечения с программированием, работой в сфере разработки компьютерных/мобильных приложений? _____
8. Напиши, чем из вышеперечисленного тебе приходилось заниматься _____
9. По какой причине ты записался в данную лабораторию?
 - а. это направление мне знакомо, хочу продолжить занятия в этой области
 - б. это направление мне знакомо, но никогда раньше не пробовал этим заниматься, хочу попробовать
 - в. мне интересно это направление, но я не имею никакого представления об этом
 - г. не было мест в других мастерских, студиях, лабораториях
 - д. свой вариант: _____
10. Чему бы ты хотел научиться в данной лаборатории?

11. Твои пожелания

Анкета №2. Промежуточное анкетирование

1. Отряд _____
2. Приходя на занятия в лабораторию, я чувствую себя:
(нужное подчеркнуть)
а). комфортно/некомфортно
б). уверенно/неуверенно
3. Педагог уделяет мне достаточно внимания/ недостаточно внимания/ не замечает моего присутствия
4. Педагог приходит мне на помощь всегда/ иногда/ крайне редко/ иногда
5. На занятия я иду с удовольствием/ потому что надо/ по-возможности не иду
6. Свою работу в лаборатории я оцениваю на 5/ 4/ 3/ 2
7. В лаборатории мне не хватает _____

8. Я бы хотел (а) изменить _____

Анкета №3. Итоговое анкетирование

1. Ф.И.О. _____
2. Отряд _____
3. Чему ты научился на занятиях в лаборатории? _____

4. Что нового ты узнал на занятиях в лаборатории? _____

5. Хотел бы ты продолжать развиваться в данном направлении? Почему?

6. Как ты оцениваешь свою работу в лаборатории? _____

7. Получил ли ты те знания и умения, которые хотел? _____
