

**ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«РОСТОВСКИЙ ЦЕНТР ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
В ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СВЯЗИ»**

**«УТВЕРЖДАЮ»
Директор
ЧОУ ДПО «РЦПК ИТС»**

_____ **С.Д. Мармоленко**

«05» мая 2022 г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2d6385008cae5e94492ef0ae9a16f647

Владелец: ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
"РОСТОВСКИЙ ЦЕНТР ПОВЫШЕНИЯ КВАЛИФИКАЦИИ В ОБЛАСТИ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СВЯЗИ"

Дата подписи: 05.05.22 15:15

Действителен: с 2022-05-05 до 2023-08-05

Программа дополнительного образования

Основы программирования и алгоритмизации.

Ростов-на-Дону

СОДЕРЖАНИЕ

1 ОБЩИЕ ПОЛОЖЕНИЯ.....	3
2 ЦЕЛЬ РЕАЛИЗАЦИИ ПРОГРАММЫ.....	3
3 ТРЕБОВАНИЯ К КВАЛИФИКАЦИИ ПОСТУПАЮЩЕГО НА ОБУЧЕНИЕ.....	4
4 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ.....	4
5 ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ.....	5
6 ФОРМЫ АТТЕСТАЦИИ И ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ.....	6
7 УЧЕБНЫЙ ПЛАН ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ.....	10
8 КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК.....	12
9 РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА.....	12

1 ОБЩИЕ ПОЛОЖЕНИЯ

Настоящая программа дополнительного образования «Основы программирования и алгоритмизации» (далее - программа) разработана Частным образовательным учреждением дополнительного профессионального образования «Ростовский центр повышения квалификации в области информационных технологий и связи» (ЧОУ ДПО «РЦПК») с учётом имеющейся потребности на рынке ИТ-компаний региона и в целях подготовки кадрового резерва специалистов, адаптированных под бизнес-процессы ИТ-компаний и запросы рынка информационных технологий.

2 ЦЕЛЬ РЕАЛИЗАЦИИ ПРОГРАММЫ

Углубление и расширение базовых и профессиональных компетенций, получение нового набора компетенций.

Программа курса «Основы программирования и алгоритмизации» несёт для слушателей курса фундаментальные знания в области разработки и проектирования программного обеспечения, развивая, по ходу изучения курса, системное мышление, прививая культуру и общепринятую стилистику написания программного кода. В целом курс строится по принципу «от простого к сложному», когда сложная понятийная база и сложный, для понимания не подготовленного человека, материал максимально упрощается и с течением времени многократно повторяется, но с каждым разом в более усложнённой форме. Например, такие фундаментальные объекты, используемые при разработке программного обеспечения, как переменная, блок памяти, структура и многие другие проходят сквозь весь курс приобретая новые с каждым разом более сложные формы. То же касается и понятийной базы. В итоге, успешное завершение курса позволяет перейти слушателям на более высокий уровень цифровой грамотности, разобраться в фундаментальных понятиях современных языков программирования высокого уровня, приобрести навыки проектирования и разработки ПО.

Слушатели познакомятся с основными тенденциями развития одного из наиболее распространенных и интенсивно развивающихся языков программирования C++.

В ходе курса слушатели расширяют и дополняют набор профессиональных компетенций за счёт изучения и использования дополнительных инструментальных средств, связанных с разработкой программного обеспечения, формирования системного подхода в анализе предметной области и конструировании алгоритмов решения поставленных задач по обработке данных, а также приобретают навыки автоматизации решения задач обработки данных за счёт реализации алгоритмов на языке программирования высокого уровня.

Программа повышения квалификации рассчитана на 70 академических часов. Рекомендуемая форма обучения: очная.

Закрепление теоретического материала осуществляется через выполнение практических заданий.

3 ТРЕБОВАНИЯ К КВАЛИФИКАЦИИ ПОСТУПАЮЩЕГО НА ОБУЧЕНИЕ

Студенты и выпускники ВУЗов и ССУЗов; граждане, желающие повысить уровень цифровой грамотности, получить профессию в сфере информационных технологий и переквалифицироваться на IT-специалиста.

Уровень сложности - начало профессиональной деятельности в данном направлении, базовые знания математики и информатики.

4 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Программа курса «Основы программирования и алгоритмизации» дает слушателям курса фундаментальные знания в области разработки и проектирования программного обеспечения, развивает системное мышление, культуру и общепринятую стилистику написания программного кода.

В частности, в курсе будет рассмотрено:

В первом модуле: понятие алгоритма, свойства алгоритмов; структурное построение алгоритмов; способы записи алгоритма; базисные алгоритмические элементы, их запись; схемы алгоритмов и псевдокод; трассировка алгоритмов; метод поэтапного уточнения (нисходящее проектирование); практические приемы построения алгоритмов; структурограммы; представление основных управляющих структур программирования; практическая реализация базовых алгоритмов.

Слушатели научатся: Проектировать алгоритмы; Использовать разные формы представления алгоритмов; Выполнять трассировку алгоритмов; Реализовывать алгоритмы на языке программирования.

Во втором модуле будут рассмотрены: основы Git; современные парадигмы программирования; синтаксис и семантика языка программирования; типы данных; основные операторы; структурные типы данных; процедуры и функции; алгоритмы вычисления сумм и произведений; задача табулирования функций; вычисление многочленов; вычисление рекуррентных соотношений; сортировка и поиск; понятие вычислительной сложности алгоритма; преобразование символьных последовательностей; реализация операций над множествами; работа с динамической памятью; потоковое взаимодействие с внешними устройствами.

Слушатели научатся: пользоваться системой контроля версий Git; писать программный код с использованием процедурной парадигмы; синтаксису языка программирования C++; выполнять структуризацию программного кода; алгоритмам сортировки и поиска и их реализации; выполнять декомпозицию программного кода; организовывать программный код с использованием принципа модульности; работать с динамической

памятью; организовывать потоковое взаимодействие с внешними устройствами; писать код, используя объектно-ориентированную парадигму.

Перечень основных новых компетенций, формируемых в результате освоения программы:

- использование средств разработки;
- использование систем контроля версий;
- анализ предметной области;
- конструирование алгоритмов;
- автоматизированная обработка данных;
- структуризация программного кода;
- организация потокового взаимодействия с внешними устройствами;
- программирование на языке C++.

Курс является базовым, формирует набор новых компетенций, необходимых для последующего обучения слушателей по углубленным курсам подготовки. Приобретенные на курсе знания и компетенции будут использоваться при выполнении следующего функционала:

- анализ предметной области;
- формирование требований к программному средству;
- алгоритмическое конструирование;
- программное проектирование;
- разработка программного обеспечения;
- организация процесса разработки.

5 ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

Компьютерные классы ЧОУ ДПО «РЦПК» оснащены современным оборудованием и программным обеспечением, позволяющими изучать и получать необходимые навыки по реализуемой программе.

Формирование профессиональных компетенций обеспечивается широким использованием в учебном процессе активных и интерактивных форм проведения занятий (компьютерных симуляций, деловых и ролевых игр, разбора конкретных ситуаций) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Программа повышения квалификации предусматривает проведение занятий в соответствии с целевыми установками программы, которые обеспечивают требуемый уровень усвоения учебного материала. Знания приобретаются путем прослушивания лекций и практической работы.

Для обучающихся обеспечивается доступ к современным профессиональным базам данных, информационным справочным и поисковым системам по тематике курса.

Изменения и дополнения вносятся в программу по мере необходимости в целях ее актуализации в случае изменений законодательной базы и осуществляются по распоряжению руководителя ЧОУ ДПО «РЦПК».

На лекционных занятиях излагаются наиболее важные и сложные вопросы. Часть лекций может излагаться проблемным методом с привлечением слушателей для решения сформулированной преподавателем проблемы. С целью текущего контроля знаний в ходе лекций могут использоваться различные приёмы тестирования.

С целью текущего контроля знаний в ходе практических занятий должны проводиться выборочные опросы и (или) использоваться различные приёмы тестирования.

Закрепление теоретического материала осуществляется через выполнение практических заданий.

В работе используется программное обеспечение Git, Eclipse.

Научно-педагогические работники, осуществляющие преподавание данной программы, имеют образование, соответствующее профилю преподаваемой дисциплины (отдельных модулей), конкретный опыт практической разработки в действующих ИТ-компаниях.

6 ФОРМЫ АТТЕСТАЦИИ И ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Контрольно-проверочные занятия включают входной контроль. Освоение обучающимися программы повышения квалификации завершается итоговой аттестацией в форме теста.

Перечень вопросов (тестов), используемых для проведения итоговой аттестации, полностью соответствует и отражает содержание лекционных и практических занятий по всем темам программы.

Для проведения итоговой аттестации создается аттестационная комиссия, состав которой утверждается директором ЧОУ ДПО «РЦПК».

Входной контроль охватывает всех обучаемых и проводится в форме тестирования. Целью его является определение уровня знаний обучаемых для корректировки и адаптации учебного процесса под конкретные потребности обучаемых, с учётом уровня освоения учебного материала, изученного ими ранее в рамках получения базового образования или на курсах повышения квалификации.

Оценочные средства, включают тесты, позволяющие оценить знания, умения и уровень приобретенных компетенций. В ходе тестирования используются современные способы и формы оценивания обучающихся.

Основными критериями оценки усвоения слушателями учебного материала при проведении текущего контроля в ходе практических занятий являются: полнота ответов на поставленные вопросы; правильность выполнения действий при отработке практических.

Конкретные формы и процедуры входного контроля знаний по каждой теме разрабатываются учебным заведением самостоятельно и доводятся до сведения обучающихся в течение первого дня обучения.

Для проведения контрольно-проверочных занятий образовательным учреждением разработаны тестовые задания, включающие вопросы для тестирования (не менее 30 вопросов для итогового теста).

Для успешного прохождения тестирования и получения оценки «зачтено» необходимо набрать не менее 70 баллов.

Ответ на вопрос считается правильным, если он является полным.

Тест включает в себя вопросы, направленные как на контроль знаний, так и на проверку полученных навыков работы. Во время тестирования запрещается пользоваться какой-либо литературой.

При проведении тестирования с использованием электронных форм контроля и оценки у каждого слушателя есть три попытки напрохождение тестирования. Время на одну попытку - 60 минут. По окончании попытки слушатель может видеть результаты теста и полученные баллы. Также имеется возможность просмотра отчета, показывающего ошибки при прохождении теста. Оценка выставляется по последней попытке.

Для аттестации обучающихся на соответствие их персональных достижений поэтапным требованиям соответствующей образовательной программы создаются фонды оценочных средств, включающие типовые задания, контрольные работы, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций. Фонды оценочных средств разрабатываются и утверждаются образовательным учреждением самостоятельно.

Лицам, успешно освоившим программу и прошедшим итоговую аттестацию, выдаются удостоверения о повышении квалификации.

Лицам, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть программы повышения квалификации и (или) отчисленным из организации, выдается справка об обучении или о периоде обучения, по установленному образцу.

Примерный список вопросов итоговой аттестации:

1. Что такое алгоритм?

- Четкая последовательность действий
- Не четкая последовательность действий
- Строго определенный набор машинных команд
- Минимально необходимый набор предписаний для разработчика
- Ничего из выше перечисленного

2. Что из перечисленного является основной алгоритмической конструкцией (возможно несколько вариантов)?

- Ветвление
- Дробление
- Следование
- Разбиение
- Объединение

3. Какими свойствами обладает алгоритм?

- дискретность
- детерминированность
- направленность
- массовость
- элементарность

4. По каким критериям осуществляется выбор алгоритма для решения одной и той же задачи?

- Временная сложность
- Сложность реализации
- Сложность описания
- Количество шагов алгоритма
- Ничего из выше перечисленного

5. Каковы принципы структурного подхода, применяемого при разработке ПО?

- Переход от сложного к простому
- Переход от простого к сложному
- Последовательная декомпозиция
- Использование структурного кодирования
- Разработка модулей со слабой связностью
- Ничего из выше перечисленного

6. Какова цель структурного программирования?

- Повышение надежности программ
- Обеспечение возможности дальнейшего сопровождения и развития ПО
- Облегчение и ускорение разработки
- Разделение кода на части, обладающие слабой логической связностью
- Повышение производительности программ

7. Укажите способы представления алгоритма:

- Словесное описание
- Псевдокод
- Графическое представление
- Венгерская нотация

Изоморфное представление

8. Парадигма программирования это:

комплекс концепций, принципов и абстракций, определяющих фундаментальный стиль программирования

семантическая составляющая языка программирования, определяющая его смысл

синтаксическая составляющая языка программирования, определяющая набор допустимых синтаксических конструкций

стиль программирования, обеспечивающий наиболее эффективный подход к разработке ПО

концептуальное представление некоторого языка программирования, определяющее его особенности

7 УЧЕБНЫЙ ПЛАН ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

7.1 Студенты и выпускники ВУЗов и ССУЗов; граждане, желающие повысить уровень цифровой грамотности, получить профессию в сфере информационных технологий и переквалифицироваться на IT-специалиста.

Уровень сложности - начало профессиональной деятельности в данном направлении, базовые знания математики и информатики.

7.2 Форма обучения: очная

7.3 Продолжительность обучения: 70 часов

7.4 Режим занятий: 2 часа в вечернее время

7.5 План учебного процесса.

№ п/п	Наименование разделов и тем	Всего, час.	В том числе	
			лекции	практич. и лаборат. занятия
1	2	3	4	5
	Входное тестирование	-	-	-
1	Учебный модуль № 1. Основы алгоритмизации	6	2	4
1.1	Понятие алгоритма, свойства алгоритмов.		0,2	
1.2	Структурное построение алгоритмов.		0,3	
1.3	Способы записи алгоритма.		0,5	
1.4	Базисные алгоритмические элементы, их запись.		0,2	
1.5	Схемы алгоритмов и псевдокод.		0,3	
1.6	Трассировка алгоритмов.		0,3	
1.7	Метод поэтапного уточнения (нисходящее проектирование).		0,2	
1.8	Практические приемы построения алгоритмов. Структурограммы.			1
1.9	Представление основных управляющих структур программирования.			1
1.10	Практическая реализация базовых алгоритмов.			2
2	Учебный модуль № 2. Основы программирования	64	14	50
2.1	Основы Git		1	3
2.2	Введение в программирование		3	5
2.3	Программирование базовых алгоритмов и обработка статических структур данных		4	12
2.4	Построение программ на основе принципа модульности		1	11
2.5	Использование динамической памяти. Динамические структуры данных		2	8
2.6	Работа с внешними ресурсами		1	5
2.7	Введение в ООП		2	6

	Итоговая аттестация по курсу (тестирование)			-
	Итого	70	16	54

7.6 Сводные данные по бюджету времени

Общий объем времени, отводимого на освоение программы (календарных дней/часов)			Распределение учебного времени (количество часов)					
Всего	Из них		Всего часов учебных занятий	В том числе		Время на самостоятельную работу	Итоговая аттестация	Резерв учебного времени
	Выходные, праздничные дни	Учебное время		Учебные занятия по расписанию	Практики			
		70	70	16	54	-	Зачет	-

8 КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Сроки проведения занятий	Количество рабочих недель	Количество занятий в неделю	Количество занятий в курсе	Продолжительность занятия (в часах)	Сроки итоговой аттестации
Устанавливаются решением руководителя учреждения по мере формирования групп	9	4	35	2	в конце 9-й недели

9 РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА

9.1 Введение в алгоритмизацию

- 9.1.1 Понятие алгоритма, свойства алгоритмов.
- 9.1.2 Структурное построение алгоритмов.
- 9.1.3 Способы записи алгоритма.
- 9.1.4 Базисные алгоритмические элементы, их запись.
- 9.1.5 Схемы алгоритмов и псевдокод.
- 9.1.6 Трассировка алгоритмов.
- 9.1.7 Метод поэтапного уточнения (нисходящее проектирование).
- 9.1.8 Практические приемы построения алгоритмов. Структурограммы.
- 9.1.9 Представление основных управляющих структур программирования.
- 9.1.10 Практическая реализация базовых алгоритмов.

9.2 Основы Git

9.3 Основы программирования

- 9.3.1 Современные парадигмы программирования.
- 9.3.2 Синтаксис и семантика языка программирования.
- 9.3.3 Типы данных.
- 9.3.4 Основные операторы.
- 9.3.5 Структурные типы данных.
- 9.3.6 Процедуры и функции.

9.4 Программирование базовых алгоритмов и обработка

статических структур данных

- 9.4.1 Алгоритмы вычисления сумм и произведений.
- 9.4.2 Задача табулирования функций.
- 9.4.3 Вычисление многочленов
- 9.4.4 Вычисление рекуррентных соотношений.
- 9.4.5 Сортировка и поиск. Понятие вычислительной сложности алгоритма.
- 9.4.6 Преобразование символьных последовательностей.
- 9.4.7 Реализация операций над множествами.
- 9.4.8 Решение задач с применением сложно структурированных типов данных.

9.5 Построение программ на основе принципа модульности

- 9.5.1 Правила конструирования модульных программ.
- 9.5.2 Пространства имен и области видимости.
- 9.5.3 Принцип сокрытия информации.
- 9.5.4 Организация взаимодействия модулей.
- 9.5.5 Практическая реализация модульной архитектуры приложения.
- 9.5.6 Итоговый контроль по блоку.

9.6 Использование динамической памяти. Динамические структуры данных

- 9.6.1 Понятие динамической памяти.
- 9.6.2 Принципы работы с динамической памятью.
- 9.6.3 Назначение указателей, типы указателей, операция разадресации, нулевой указатель, бестиповые указатели.
- 9.6.4 Ссылки. Отличия ссылок и указателей.
- 9.6.5 Понятие динамических структур данных.
- 9.6.6 Основные динамические структуры: стек, очередь, линейный список.
- 9.6.7 Итоговый контроль по блоку.

9.7 Работа с внешними ресурсами

- 9.7.1 Понятие файла, классификация, понятие файловой переменной.
- 9.7.2 Текстовые файлы. Методы для работы с текстовым файлом.
- 9.7.3 Особенности типизированного файла, подпрограммы для работы с типизированным файлом.
- 9.7.4 Использование типизированных файлов для организации простейшего хранилища данных.
- 9.7.5 Итоговый контроль по блоку

9.8 Введение в ООП

- 9.8.1 Основные понятия ООП.
- 9.8.2 Отличие классов от записей, видимость членов класса, поля и методы класса, создание/уничтожение экземпляров класса.
- 9.8.3 Базовый и производный классы, переопределение методов.
- 9.8.4 Включение экземпляров одних классов в состав других
- 9.8.5 Итоговый контроль по блоку