

Рабочая программа учебной дисциплины **ОП.04** Основы алгоритмизации и программирования разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее - СПО) **09.02.07** Информационные системы и программирование, утвержденным приказом Министерства образования и науки Российской Федерации 09 декабря 2016 № 1547, зарегистрированным в Министерстве юстиции Российской Федерации 26 декабря 2016 года, регистрационный № 44936, входящим в укрупнённую группу 09.00.00 Информатика и вычислительная техника

Организация-разработчик: государственное бюджетное профессиональное образовательное учреждение «Пермский химико-технологический техникум» (ГБПОУ «ПХТТ»)

Разработчик:

Юшкова Евгения Владимировна, преподаватель высшей квалификационной категории

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	9
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	16

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 Основы алгоритмизации и программирования

1.1. Область применения программы

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы ГБПОУ «ПХТТ» в соответствии с ФГОС по специальности СПО: 09.02.07 Информационные системы и программирование

1.2. Место дисциплины в структуре основной профессиональной образовательной программы: общепрофессиональная дисциплина профессионального цикла.

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины:

В результате изучения дисциплины обучающийся должен уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов.
- определять сложность работы алгоритмов.
- работать в среде программирования.
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- оформлять код программы в соответствии со стандартом кодирования.
- выполнять проверку, отладку кода программы.

В результате освоения дисциплины обучающийся должен знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- эволюцию языков программирования, их классификацию, понятие системы программирования.
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- подпрограммы, составление библиотек подпрограмм.
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.

Содержание дисциплины ориентировано на подготовку обучающихся к освоению профессиональных модулей ОПОП по специальности 09.02.07 Информационные системы и программирование. В результате освоения дисциплины обучающийся осваивает элементы профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	
Умения	Знания

<p>Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификации "Программист"</i> Оценка сложности алгоритма.</p>	<p>Основные этапы разработки программного обеспечения.</p> <p>Основные принципы технологии структурного и объектно-ориентированного программирования.</p> <p><i>Дополнительно для квалификации "Программист"</i> Актуальная нормативно-правовая база в области документирования алгоритмов.</p>
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	
Умения	Знания
<p>Создавать программу по разработанному алгоритму как отдельный модуль; Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификации "Программист":</i> Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровня в том числе для мобильных платформ.</p>	<p>Основные этапы разработки программного обеспечения.</p> <p>Основные принципы технологии структурного и объектно-ориентированного программирования.</p> <p><i>Дополнительно для квалификации "Программист":</i> Знание API современных мобильных операционных систем.</p>
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств	
Умения	Знания
<p>Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификации "Программист"</i> Применять инструментальные средства отладки программного обеспечения.</p>	<p>Основные принципы отладки и тестирования программных продуктов.</p> <p>Инструментарий отладки программных продуктов.</p>
ПК 1.4. Выполнять тестирование программных модулей	
Умения	Знания
<p>Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p>	<p>Основные виды и принципы тестирования программных продуктов.</p>
ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода	
Умения	Знания
<p>Выполнять оптимизацию и рефакторинг программного кода. Работать с системой контроля версий.</p>	<p>Способы оптимизации и приемы рефакторинга.</p> <p>Инструментальные средства анализа алгоритма.</p> <p>Методы организации рефакторинга и оптимизации кода.</p>

	Принципы работы с системой контроля версий.
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.	
Умения	Знания
Осуществлять разработку кода программного модуля на современных языках программирования. Оформлять документацию на программные средства.	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования.
ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	
Умения	Знания
Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций.	Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	
Умения	Знания
Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций.	Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продук-

	тов. Методы организации работы в команде разработчиков.
--	--

В результате освоения дисциплины обучающийся осваивает элементы общих компетенций (ОК):

<i>Шифр комп.</i>	<i>Наименование компетенций</i>	<i>Дескрипторы (показатели сформированности)</i>	<i>Умения</i>	<i>Знания</i>
ОК 1.	<i>Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</i>	Распознавать сложные проблемы в знакомых ситуациях. Выделять сложные составные части проблемы и описывать её причины и ресурсы, необходимые для её решения в целом. Определять потребность в информации и предпринимать усилия для её поиска. Выделять главные и альтернативные источники нужных ресурсов. Разрабатывать детальный план действий и придерживаться его. Оценивать результат своей работы, выделять в нём сильные и слабые стороны.	Распознавать задачу и/или проблему в профессиональном и/или социальном контексте. Анализировать задачу и/или проблему и выделять её составные части. Правильно определить и найти информацию, необходимую для решения задачи и/или проблемы. Составить план действия, определить необходимые ресурсы. Владеть актуальными методами работы в профессиональной и смежных сферах. Реализовать составленный план. Оценить результат и последствия своих действий (самостоятельно или с помощью наставника).	Актуальный профессиональный и социальный контекст, в котором приходится работать и жить. Основные источники информации и ресурсов для решения задач и проблем в профессиональном и/или социальном контексте. Актуальные стандарты выполнения работ в профессиональной и смежных областях. Актуальные методы работы в профессиональной и смежных сферах.
ОК 2.	<i>Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</i>	Планировать информационный поиск из широкого набора источников, необходимого для выполнения профессиональных задач. Проводить анализ полученной информации, выделять в ней главные аспекты. Структурировать отобранную информацию в соответствии с параметрами поиска. Интерпретировать полученную информацию в контексте профессиональной деятельности.	Определять задачи поиска информации. Определять необходимые источники информации. Планировать процесс поиска. Структурировать получаемую информацию. Выделять наиболее значимое в перечне информации. Оценивать практическую значимость результатов поиска. Оформлять результаты поиска.	Номенклатура информационных источников, применяемых в профессиональной деятельности. Приемы структурирования информации. Формат оформления результатов поиска информации.

1.4. Количество часов на освоение программы дисциплины:

учебная нагрузка обучающегося 218 часов, в том числе:

во взаимодействии с преподавателем 186 часов;

самостоятельной работы обучающегося 12 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Учебная нагрузка (всего)	218
Нагрузка во взаимодействии с преподавателем	186
в том числе:	
теоретическое обучение	110
практические (лабораторные) занятия	76
консультации	8
промежуточная аттестация	12
Самостоятельная работа обучающегося (всего)	12
в том числе:	
Виды самостоятельной работы:	
– самостоятельная проработка конспектов занятий, учебной литературы, учебных пособий;	4
– подготовка к практическим занятиям с использованием методических рекомендаций преподавателя, оформление отчетов по практическим занятиям, подготовка к их защите;	4
– самостоятельное изучение отдельных вопросов (с целью углубления знаний по заданию преподавателя) с последующим оформлением реферата	4
Промежуточная аттестация в форме экзамена	

2.2. Тематический план и содержание учебной дисциплины ОП.04 Основы алгоритмизации и программирования

Наименование разделов и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся	Объем часов			Уровень освоения
		лек	пр	сам	
1	2				4
Раздел 1.	Основы алгоритмизации и программирования	76	40	8	
Тема 1.1. Основы алгоритмизации	<i>Содержание учебного материала</i>	16	8	4	
	Понятие алгоритма, его свойства и виды. Критерии «хорошего» алгоритма. Способы описания алгоритмов: псевдокоды, блок-схема, программа. Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Блок-схема: основные элементы, правила составления. Примеры.	8			2
	Различные комбинации алгоритмических конструкций. Тестовые данные. Математическая модель алгоритма. Классические алгоритмы Евклида, решения нелинейных и линейных уравнений и т.д. Примеры	8			
	<i>Практическая работа 1 Составление блок-схем алгоритмов линейной структуры (2)</i> <i>Практическая работа 2 Составление блок-схем алгоритмов разветвляющейся структуры (2)</i> <i>Практическая работа 3 Составление блок-схем алгоритмов циклической структуры (4)</i>		8		
	<i>Самостоятельная работа обучающихся по теме 1.1</i> самостоятельная проработка конспектов занятий, учебной литературы, учебных пособий;			4	
Тема 1.2. Основы программирования	<i>Содержание учебного материала</i>	60	32	4	
	История языков программирования. Генеалогическое древо C#. Преимущества и недостатки языка C#. Платформа .NET Framework: назначение, составные элементы, тонкости компиляции. Обзор интегрированной среды разработки Visual Studio .NET. Создание, сохранение и закрытие проектов и решений. Структура программы. Точка входа. Базовый синтаксис C#. Пространство имен. Классы.	4			1,2
	Переменные: определение, правила именования. Типы данных: значимые и ссылочные. Преобразование типов: явное и неявное. Объявление переменных и их инициализация. Область действия и время существования переменных. Константы: определение, виды и правила записи в программе. Суффиксы целых и вещественных констант. Escape – последовательности.	2			
	Понятие выражения. Математические операторы. Старшинство операторов. Математические функции (класс Math). Ввод – вывод данных. Операторы присваивания.	4			

Операторы отношения. Оператор if-else: назначение и правила записи. Проверка простых и сложных условий. Вложенные условные операторы.	2		
Использование поразрядных (!, &, , ^) и условных (&&,) логических операторов. Отличие в использовании операторов & и &&, и . Тернарный оператор: правила использования. Механизм получения случайного числа.	2		
Оператор выбора switch-case: назначение и правила записи. Операторы перехода: break, goto, return.	2		
Цикл с предусловием (while), цикл с постусловием (do) и цикл с параметром (for). Правила записи, отличия в применении.	4		
Стандартные операции при работе с циклическими алгоритмами: получение таблицы значений, нахождение суммы и произведения элементов ряда. Принудительный выход из цикла: операторы break и continue. Бесконечные циклы.	4		
Массивы: определение, виды. Объявление одномерного массива. Варианты инициализации. Ввод и вывод одномерных массивов.	4		
Стандартные операции для работы с массивами: заполнение случайными значениями и значениями по формуле, нахождение суммы и произведения, нахождение максимума (минимума), подсчет количества элементов, удовлетворяющих определенному условию.	4		
Обработка одномерных массивов: сортировка и поиск элементов. Цикл foreach. Двумерные массивы: объявление, ввод и вывод. Работа с двумерными массивами по строкам и по столбцам	4		
Понятие потока. Механизм буферизации. Классы библиотеки .NET для работы с потоками. Виды доступа к файлам. Объект FileStream. Классы StreamWriter и StreamReader.	4		
Основные операции при работе с файлами. Способы работы с текстовыми файлами. Обработка текстовых файлов.	6		
<i>Практическая работа 4</i> Консольные и линейные приложения (2) <i>Практическая работа 5</i> Создание проектов с использованием условного оператора if/else (2) <i>Практическая работа 6</i> Создание проектов с использованием оператора выбора switch(2) <i>Практическая работа 7</i> Создание проектов циклической структуры (for) (4) <i>Практическая работа 8</i> Создание проектов циклической структуры (while, do) (4) <i>Практическая работа 9</i> Создание проектов для работы с одномерными массивами (4) <i>Практическая работа 10</i> Создание проектов для работы с двумерными массивами (4) <i>Практическая работа 11</i> Создание проектов с использованием текстовых файлов (4) <i>Практическая работа 12</i> Использование компонентов OpenFileDialog и SaveFileDialog для работы		32	

	с файлами (б)				
	Самостоятельная работа обучающихся по теме 1.2 подготовка к практическим занятиям с использованием методических рекомендаций преподавателя, оформление отчетов по практическим занятиям, подготовка к их защите;			4	
Раздел 2.	Объектно-ориентированное программирование	34	36	4	
Тема 2.1. Объектно-ориентированная модель программирования	Содержание учебного материала	34	36	4	
	Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция.	4			1,2
	Общая форма определения класса. Модификаторы доступа к элементам класса: public, private, protected, internal. Примеры создания классов.	2			
	Переменные ссылочного типа и присваивание. Побочные эффекты множественных ссылок.	2			
	Метод: понятие, правила записи. Вызов метода. Передача параметров по значению. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования.	4			
	Создание методов, возвращающих значения. Способы размещения методов. Конструкторы.	4			
	Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи.	2			
	Наследование и полиморфизм. Иерархия классов: понятие, преимущества. Синтаксис наследования. Скрытие и перекрытие методов. Ключевые слова virtual, override.	2			
	Вызов методов базового класса («родителя»): ключевое слово base. Тонкости использования конструкторов в иерархически связанных между собой классах.	2			
	Интерфейсы: назначение, правила написания. Способы реализации интерфейсов. Работа с объектами через интерфейсы. Операторы is и as. Восходящее и нисходящее приведение.	2			
	Наследование в интерфейсах. Сходства и различия интерфейсов, абстрактных классов и обычных классов. Стандартные интерфейсы .NET: IComparable, ICloneable, IEnumerable. Примеры реализации.	2			
	Назначение и синтаксис структуры. Элементы тела структуры. Сходства и разли-	2			

чия структур и классов. Плюсы и минусы использования структур.				
Тип данных enum: объявление и тонкости использования. Операции с перечислениями.	2			
Понятие коллекции. Класс List: возможности, правила работы, встроенные методы. Работа с ArrayList.	2			
Делегаты: понятие, правила описания. Примеры и тонкости использования делегатов. Обработка события: автоматическое создание обработчиков. Решение задач на создание событий.	2			
<i>Практическая работа 13</i> Создание простейших методов (2) <i>Практическая работа 14</i> Создание перегруженных методов (4) <i>Практическая работа 15</i> Создание простейших классов (4) <i>Практическая работа 16</i> Создание классов с использованием свойств (4) <i>Практическая работа 17</i> Создание классов, иерархически связанных между собой (4) <i>Практическая работа 18</i> Решение задач с использованием классов (4) <i>Практическая работа 19</i> Создание MDI-приложений и меню (4) <i>Практическая работа 20</i> Реализация стандартных интерфейсов .NET: IComparable, ICloneable, IEnumerable. (4) <i>Практическая работа 21</i> Создание проектов с использованием структур и перечислений(4) <i>Практическая работа 22</i> Создание проектов с использованием коллекций. Работа с ArrayList (2)		36		
<i>Самостоятельная работа обучающихся по теме 2.1</i> самостоятельное изучение отдельных вопросов (с целью углубления знаний по заданию преподавателя) с последующим оформлением реферата			4	
	<i>Консультации</i>	8		
	<i>Экзамен</i>	12		
	Всего	218		

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – ознакомительный (узнавание ранее изученных объектов, свойств);
2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия лаборатории «Программирования и баз данных»

- рабочие места на 12-15 обучающихся (Процессор не ниже Core i3, оперативная память объемом не менее 8 Гб);
- рабочее место преподавателя (Процессор не ниже Core i3, оперативная память объемом не менее 8 Гб);
- сервер в лаборатории (8-х ядерный процессор с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб, программное обеспечение: Windows Server 2012 или более новая) или выделение аналогичного по характеристикам виртуального сервера из общей фермы серверов
- проектор и экран;
- маркерная доска;
- программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО: Microsoft Visio Professional, Microsoft Visual Studio, Microsoft Office.

3.2. Информационное обеспечение обучения

Основные источники:

1. Подбельский В.В. Язык C#. Базовый курс. – М: Инфра, 2015.- 384 с.
2. Васильев А.Н. Программирование на C# для начинающих. – Москва: Эксмо, 2018. – 592 с.
3. Серкова Е.Г. Основы алгоритмизации и программирования (ОП.04): практикум/Е.Г. Серкова. – Ростов н/Д : Феникс, 2019. – 188,[1] с. : - (Среднее профессиональное образование)

3.3. Общие требования к организации образовательного процесса

Освоение дисциплины ОП.04 Основы алгоритмизации и программирования производится в соответствии с учебным планом по специальности 09.02.07 Информационные системы и программирования и календарным графиком.

Образовательный процесс организуется строго по расписанию занятий, утвержденному заместителем директора.

Изучение теоретического материала может проводиться как в каждой группе, так и в нескольких группах одновременно (при наличии нескольких групп на специальности).

При проведении практических занятий проводится деление группы обучающихся на подгруппы, численностью не более 13 чел. Практические работы проводятся в специально оборудованной лаборатории «Программирования и баз данных».

В процессе освоения дисциплины предполагается проведение текущего и промежуточного контроля знаний, умений у студентов.

Текущий учет результатов освоения дисциплины производится в журнале успеваемости.

С целью оказания помощи обучающимся при освоении теоретического и практического материала, выполнения самостоятельной работы разрабатываются учебно-методические комплексы.

При освоении дисциплины, в соответствии с учебным планом и расписанием, для всех желающих проводятся консультации.

Образовательный процесс организован с использованием электронного обучения и дистанционных технологий. На сайте СДО ПХТТ размещается теоретический материал для самостоятельного изучения студентами, задания для выполнения практических работ, автоматизированные тесты и другие материалы.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических работ, тестирования, а также выполнения обучающимися индивидуальных заданий.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
умения:	
<ul style="list-style-type: none"> – разрабатывать алгоритмы для конкретных задач; – использовать программы для графического отображения алгоритмов. – определять сложность работы алгоритмов. – работать в среде программирования. – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. – оформлять код программы в соответствии со стандартом кодирования. – выполнять проверку, отладку кода программы. 	<ul style="list-style-type: none"> – Опросы устные и письменные, – практические работы, – проверочные работы, – тестирование, – экзамен.
знания:	
<ul style="list-style-type: none"> – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – эволюцию языков программирования, их классификацию, понятие системы программирования. – основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – подпрограммы, составление библиотек подпрограмм. – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения. 	<ul style="list-style-type: none"> – Опросы устные и письменные, – практические работы, – проверочные работы, – тестирование, – экзамен.
Итоговая аттестация	экзамен

Оценка результатов тестирования при текущем контроле производится в соответствии с универсальной шкалой (таблица).

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
85 ÷ 100	5	отлично
75 ÷ 84	4	хорошо
65 ÷ 74	3	удовлетворительно
менее 65	2	неудовлетворительно

Разработчик:

ГБПОУ «ПХТТ»

преподаватель

Е.В. Юшкова