

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И МОЛОДЁЖНОЙ ПОЛИТИКИ
СВЕРДЛОВСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ
«АСБЕСТОВСКИЙ ПОЛИТЕХНИКУМ»**

УТВЕРЖДАЮ

Директор ГАПОУ СО

«Асбестовский политехникум»

В.А. Суслопаров

«29» июня 2020 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.05 ОСНОВЫ ПРОГРАММИРОВАНИЯ

для специальности СПО

09.02.03 «Программирование в компьютерных системах»

Форма обучения – очная

Срок обучения 3 года 10 месяцев

Асбест
2020

Рабочая программа учебной дисциплины **«ОСНОВЫ ПРОГРАММИРОВАНИЯ»** разработана на основе Федерального государственного образовательного стандарта (далее ФГОС) по специальности среднего профессионального образования (далее СПО) **09.02.03 «Программирование в компьютерных системах»**, приказ Минобрнауки №804 от 28 июля 2014 года.

Организация-разработчик: ГАПОУ СО «Асбестовский политехникум»

Разработчики:

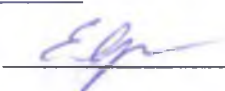
Копина Марина Геннадьевна, преподаватель ГАПОУ СО «Асбестовский политехникум», г. Асбест

Савина Ольга Николаевна, преподаватель, высшая квалификационная категория, ГАПОУ СО «Асбестовский политехникум», г. Асбест

РАССМОТРЕНО

цикловой комиссией информационных и экономических дисциплин,
протокол № 6

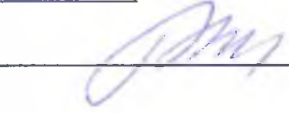
«23» июня 2020 г.

Председатель  Е.А. Ярышева

СОГЛАСОВАНО

Методическим советом, протокол № 3

«25» июня 2020 г.

Председатель  Н.Р. Караваева

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	8
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	10

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОСНОВЫ ПРОГРАММИРОВАНИЯ

1.1. Область применения программы

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 09. 02.03 «Программирование в компьютерных системах»

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании (в программах повышения квалификации и переподготовки), и профессиональной подготовке по специальности 09. 02.03 «Программирование в компьютерных системах»

Общие компетенции

- ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
- ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
- ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
- ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
- ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.
- ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
- ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
- ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Профессиональные компетенции:

- ПК 1.1 Выполнять разработку спецификаций отдельных компонент.
- ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
- ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.
- ПК 1.4. Выполнять тестирование программных модулей.
- ПК 1.5. Осуществлять оптимизацию программного кода модуля.
- ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

1.2. Место дисциплины в структуре основной профессиональной образовательной программы: блок общепрофессиональных дисциплин

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

В результате освоения дисциплины студент должен **уметь:**

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

В результате освоения дисциплины студент должен **знать:**

- этапы решения задач на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования.

1.4. Количество часов на освоение учебной дисциплины:

максимальная учебная нагрузка обучающегося 264 часа, в том числе:

обязательная аудиторная учебная нагрузка обучающегося 176 часа;

самостоятельной работы обучающегося 88 часа.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Количество часов
Максимальная учебная нагрузка (всего)	264
Обязательная аудиторная учебная нагрузка (всего)	176
в том числе:	
практические занятия	106
контрольные работы	
Самостоятельная работа обучающегося (всего)	88
в том числе:	
домашняя контрольная работа	60
Домашняя контрольная работа	28
<i>Промежуточная аттестация в форме экзамена</i>	

2.2. Тематический план и содержание учебной дисциплины «Основы программирования»

Наименование разделов и тем	Содержание учебного материала, практические занятия и самостоятельная работа обучающихся	Количество часов	Уровень освоения
1	2	3	4
Раздел 1 Структурное программирование			
Тема 1.1. Начальные сведения о программировании и языке C++	Содержание учебного материала	2	
	1. Начальные сведения о программировании и языке C++ (сферы применения языка, пример простой программы, использование компилятора, транслятора, запуск программы)	2	1
Тема 1.2. Лексические основы языка C++	Содержание учебного материала	2	
	1. Лексические основы языка C++ (правила именования переменных, констант и функций языка, понятие ключевого или зарезервированного слова, список ключевых слов языка C++)	2	1
Тема 1.3. Типы данных. Преобразование типов. Структура программы.	Содержание учебного материала	14	
	1. Типы данных языка C++.	8	1
	2. Преобразование типов данных		
	3. Синтаксис объявления переменных и констант.		
	5. Понятие препроцессора, директивы. Структура программы.		
	4. Разработка программ линейной структуры.	6	2
	Практические занятия		
	Разработка программ линейной структуры		
Тема 1.4. Управляющие операторы языка C++	Содержание учебного материала	14	
	1. Синтаксис и использование оператора выбора: if-else	6	1
	2. Синтаксис оператора множественного выбора switch		
	3. Примеры решения задач с операторами выбора		
	Практические занятия	8	2
	Разработка программ разветвляющейся структуры		
Тема 1.5. Циклические конструкции. Оператор цикла: for, while, do..while. Операторы break, continue, goto.	Содержание учебного материала	16	
	1. Синтаксис цикла for. Правила использования.	8	1
	2. Операторы цикла while, do..while.		
	3. Управляющие операторы в циклах: break, continue, goto		
	4. Решение задач с циклами for, while, do..while		
	Практические занятия	10	2
	Разработка программ циклической структуры		
Тема 1.6. Одномерные массивы на языке C++	Содержание учебного материала	14	
	1. Структурированные данные. Понятие массива. Синтаксис объявления одномерного массива.	6	1
	2. Основные приемы работы с массивами.		
	3. Решение задач с массивами		
	Практические занятия	6	2
	Обработка данных организованных в форме одномерного массива		
Тема 1.7. Двумерные массивы на языке C++	Содержание учебного материала	16	
	1. Двумерные массивы. Особенности. Синтаксис объявления двумерного массива.	6	1
	2. Основные приемы работы с двумерными массивами		
	3. Вставка и удаление строк, столбцов в массиве.		
	Практические занятия	10	2
	Обработка данных в двумерном массиве		
Тема 1.8. Понятие функции. Прототипы функций	Содержание учебного материала	18	
	1. Синтаксис объявления функции на языке C++	8	1
	2. Синтаксис вызова функции и передачи параметров. Понятие формальных и фактических параметров		
	3. Прототипы функций.. Встраиваемые функции		
	4. Рекурсивные функции.		
	Практические занятия	10	2
	Разработка пользовательских функций для решения задач		
Тема 1.9. Массивы в качестве параметров функции	Содержание учебного материала	18	
	1. Синтаксис объявления и вызова функции, содержащей массив в качестве параметра функции	6	1
	2. Вставка и удаление элементов массива через функцию.		
	3. Сортировка элементов массива.		

	4. Примеры программ, использующих функции		
	Практические занятия		
	Использование функции, содержащей массив в качестве параметра, для решения задач (удаление, вставка, сортировка)	12	2
Тема 1.10. Указатели на объекты и массивы	Содержание учебного материала	8	
	1. Создание и использование указателей.	4	1
	2. Адресная арифметика		
	3. Связь между массивами и указателями		
	Практические занятия		
	Указатели. Решение задач	4	2
Тема 1.12. Символьная информация в C++	Содержание учебного материала	16	
	1. Строки и литералы	4	1
	2. Стандартные библиотеки и функции для работы со строками		
	Практические занятия		
	Строки. Решение задач	12	2
Тема 1.13. Типы данных, определяемые пользователем	Содержание учебного материала	10	
	1. Синтаксис объявления структур и объединений	4	1
	2. Примеры использования структур и объединений		
	Практические занятия		
	Решение задач с использованием структур и объединений	6	2
Тема 1.Объектно-ориентированное программирование	Содержание учебного материала	28	
	Основные понятия объектно-ориентированного программирования (ООП). Концепции ООП. Объекты. Классы	8	1
	Объекты и классы в ООП. Конструкторы. Деструкторы		
	Операции над объектами. Дружественные функции		
	Обработка массивов через объекты		
	Практические занятия	20	
Самостоятельная работа Домашняя контрольная работа по теме «Массивы»		60	3
Самостоятельная работа Домашняя контрольная работа по теме «Обработка строк»		28	3
ИТОГО: теории - 70часов, практических занятий – 106часов, самостоятельной работы – 88часа			
ИТОГО		264	

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 ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к материально-техническому обеспечению

Реализация программы учебной дисциплины требует наличия лаборатории «Системного и прикладного программирования».

Наименование оборудования, приспособлений, инструментов, оснастки, наглядных пособий и документации определено в соответствии с требованиями к охране труда и техники безопасности на рабочем месте.

Оборудование лаборатории «Системного и прикладного программирования»:

1. Персональные компьютеры.
2. Видеопроектор
3. Интерактивная доска
4. Принтер

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов

1. Герберт Шилдт. С++: базовый курс. 3-е издание, изд. дом «Вильямс», 2010
2. Глушаков С.В., Дуравкина Т.В. Программирование на С++, – изд. «Аст», 2008
3. Ишкова Э.А. С++. Начала программирования. Издание 3 – Бином-Пресс, 2009
4. Лаптев В.В., Морозов А.В., Бокова А.В. С++ объектно-ориентированное программирование. Задачи и упражнения – Питер, 2007 г.

Перечень дополнительной литературы

5. Литвиненко Н.А. Технология программирования на С++. Начальный курс, – изд. «ВНУ–Санкт-Петербург», 2006
6. Павловская Т.А. С/С++. Программирование на языке высокого уровня. — СПб.: Питер, 2010. — 461 с.
7. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: Практикум. — СПб: ПИТЕР, 2007. — 239 с.

8. Стенли Б. Липпман, Барбара Э. Му, Жози Лажойе. Язык программирования C++. Вводный курс. – 4-е издание, изд. дом «Вильямс», 2006
9. Фридман А.Л. Язык программирования C++. М.: Интуит.ру «Интернет-университет Информационных Технологий», 2009

Перечень рекомендуемых Интернет-ресурсов

10. Дейтел Х., Дейтел П. – Электронный учебник – Как программировать на C++
http://reslib.com/book/Kak_programmirovatj_na_C_Dejtel_H_/130
11. Сайт «Основы программирования на C++»
<http://kufas.ru/programming104.htm>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

ГАПОУ СО «Асбестовский политехникум», реализующее подготовку по учебной дисциплине «Основы программирования», обеспечивает организацию и проведение промежуточной аттестации и текущего контроля, демонстрируемых обучающимися знаний, умений и навыков. Текущий контроль проводится преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий.

Формы и методы текущего контроля по учебной дисциплине самостоятельно разрабатываются преподавателем, рассматриваются на заседании цикловой комиссии информационных технологий, согласуются с работодателями, методическим советом и доводятся до сведения обучающихся в начале обучения.

Обучение по учебной дисциплине завершается проведением экзамена по билетам.

Такая форма аттестации позволяет охватить весь пройденный теоретический материал по дисциплине, проверить системность знаний, а также умение применять полученные знания на практике.

Для текущего контроля преподавателем создаются фонды оценочных средств (ФОС).

ФОС включают в себя педагогические контрольно-измерительные материалы, предназначенные для определения соответствия (или несоответствия) индивидуальных образовательных достижений основным показателям результатов подготовки (таблицы).

Результаты обучения (освоенные умения, усвоенные знания)	Показатели освоения результата	Формы и методы контроля и оценки результатов обучения
Усвоенные знания: • этапы решения задач на компьютере; • типы данных; • базовые конструкции изучаемых языков программирования; • принципы структурного и модульного программирования; • принципы объектно-ориентированного программирования.	– правильно и корректно отвечает на поставленные вопросы; – демонстрирует модели алгоритмов и методы построения; – объясняет структуру программы и правила работы операторов в программе, синтаксис операторов;	– Опрос, – собеседование – решение профессиональных и нетиповых задач (использование материала для самостоятельного изучения), – тестовый контроль, – домашняя контрольная работа
Освоенные умения: • работать в среде программирования • реализовывать построенные алгоритмы в виде программ на языке программирования C++.	– составляет листинг программы по разработанному алгоритму в заданной среде программирования; – оформляет структуру программы; – проверяет программу на правильность результата; – исправляет синтаксические и логические ошибки программы; – применяет для нахождения ошибок в программе ручную трассировку.	– пошаговый разбор выполнения программы; – нахождение ошибок при выполнении программ с использованием трассировки; – анализ и оценивание результата программы.