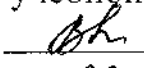


Министерство образования, науки и молодежи Республики Крым
Государственное бюджетное профессиональное образовательное учреждение
Республики Крым
«Симферопольский колледж радиоэлектроники»

УТВЕРЖДАЮ:

Заместитель директора по
учебной работе

 В.И. Полякова
« 30 » 08 2019 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП 09 «Основы алгоритмизации и программирования»

Специальность: 09.02.01 «Компьютерные системы и комплексы»

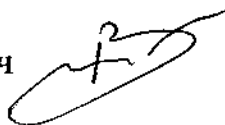
г. Симферополь
2019 г.

Рабочая программа учебной дисциплины ОП 09 Основы алгоритмизации и программирования разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования специальности **09.02.01 Компьютерные системы и комплексы**, утвержденного приказом Министерства образования и науки от 28.07.2014 года № 849.

Организация разработчик - Государственное бюджетное профессиональное образовательное учреждение Республики Крым «Симферопольский колледж радиоэлектроники»

Разработчик - преподаватель ГБПОУ РК «Симферопольский колледж радиоэлектроники»:

- Деменков Владимир Александрович



Рассмотрено и одобрено на заседании цикловой методической комиссии №_5

«17» 08 2019 г. Протокол № 1

Председатель ЦМК  Мелихова С.Г.

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	стр. 4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	15

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования *название дисциплины*

1.1. Область применения программы

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ Основы алгоритмизации и программирования

1.1. Область применения программы

Программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 09.02.01 Компьютерные системы и комплексы

Программа учебной дисциплины может быть использована в дополнительном профессиональном образовании (в программах повышения квалификации и переподготовки), а также в качестве примерной программы цикла общепрофессиональных дисциплин, вариативной части ФГОС по другим группам специальностей технического профиля.

1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы: дисциплина входит в профессиональный цикл.

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

В результате освоения учебной дисциплины обучающийся должен уметь:

- Формализовать поставленную задачу;
- Применять полученные знания к различным предметным областям;
- Составлять и оформлять программы на языках программирования;
- Тестировать и отлаживать программы

В результате освоения учебной дисциплины обучающийся должен знать:

- Общие принципы построения и использование языков программирования, их классификацию;
- Современные интегрированные среды разработки программ;
- Процесс создания программ;
- Стандарты языков программирования;
- Общую характеристику языка ассемблера: назначение, принцип построения и использования

В результате освоения дисциплины у обучающихся по базовой подготовке формируются общие компетенции (ОК):

- 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 команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), 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.

Содержание дисциплины должно быть ориентировано на подготовку обучающихся по базовой подготовке и овладению профессиональными компетенциями (ПК):

- ПК 2.1. Создавать программы на языке ассемблера для микропроцессорных систем
- ПК 2.2. Производить тестирование, определение параметров и отладку микропроцессорных систем.
- ПК 3.3. Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения

1.4. Рекомендуемое количество часов на освоение примерной программы учебной дисциплины:

максимальной учебной нагрузки обучающегося 180 часов, в том числе:

обязательной аудиторной учебной нагрузки обучающегося 120 часов
в том числе лабораторные работы: 60 часов;
самостоятельной работы обучающегося 60 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Количество часов
Максимальная учебная нагрузка (всего)	180
Обязательная аудиторная учебная нагрузка (всего)	120
в том числе:	
лабораторные занятия	60
Самостоятельная работа обучающегося (всего)	60
в том числе:	
- работа со справочной и дополнительной литературой, проработка тестовых заданий;	24
- выполнение индивидуальных заданий, создание проектов	36
<i>Промежуточная аттестация в форме экзамена</i>	

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

Наименование разделов и тем	Содержание учебного материала, практические занятия, самостоятельная работа обучающихся		Объем часов	Уровень освоения
	1	2		
Введение	Основные цели и задачи курса.		2	2
Раздел 1. Основные понятия алгоритмизации			20	
Тема 1.1. Основные принципы алгоритмизации и программирования	<p>Содержание учебного материала</p> <p><u>1 Основные понятия алгоритмизации. Понятие алгоритма и его свойства.</u></p> <p>Схема решения задач на ЭВМ. Формы записи алгоритмов. Общие принципы построения алгоритмов.</p> <p>Алгоритм и его свойства</p> <p>Для записи алгоритма решения задачи применяются следующие образные способы их представления:</p> <p>Словесно-формульное описание</p> <p>Блок-схема (схема графических символов)</p> <p>Алгоритмические языки</p> <p>Операторные схемы</p> <p>Псевдокод</p> <p><u>2 Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.</u></p> <p>Логические основы алгоритмизации. Основные базовые и структурированные типы данных, их характеристика.</p> <p>Лабораторные занятия</p> <p>Построение блок-схем линейных вычислительных процессов</p> <p>Построение блок-схем разветвляющихся вычислительных процессов</p> <p>Построение блок-схем циклических вычислительных процессов</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся</p>		16	
Тема 1.2. Языки и методы программирования	<p>Содержание учебного материала</p> <p><u>1 Понятие языков программирования. Классификация</u></p> <p>Языки программирования. Эволюция языков программирования. Классификация языков программирования. Понятие системы программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.</p> <p>Структура языка</p> <p>Процедурные и непроцедурные</p>		2	2
			2	2
			2	3
			2	3
			2	3
			6	
			4	

	<p>Макропроцессор - программа, обеспечивающая замену одной последовательности символов другой</p> <p>Языки программирования - инструменты для решения задач в разных предметных областях</p> <p>2 Методы программирования. Достоинства и недостатки</p> <p>Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования. Общие принципы разработки программного обеспечения. Жизненный цикл программного обеспечения. Типы приложений. Консольные приложения.</p> <p>Лабораторные занятия</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся</p>	2	2
<p>Раздел 2.</p> <p>Программирование на алгоритмическом языке СИ++</p>		132	
<p>Тема 2.1.</p> <p>Основные элементы языка высокого уровня</p>	<p>Содержание учебного материала</p> <p>1 Типы данных</p> <p>Переменные и константы..</p> <p>Используемые символы</p> <p>Знаки нумерации и специальные символы</p> <p>Управляющие и разделительные символы</p> <p>управляющие последовательности</p> <p>Целая константа</p> <p>Десятичная константа</p> <p>Восьмеричная константа</p> <p>Шестнадцатеричная константа</p> <p>Символьная константа</p> <p>Идентификатор</p> <p>Комбинация знаков операций и операндов, результатом которой является определенное значение, называется выражением. Константное выражение - это выражение, результатом которого является константа</p>	10	2
	<p>2 Выражения и операции</p> <p>Лабораторные занятия</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся. Решение задач – преобразовать в вид, используемый в программах</p>	2	2
<p>Тема 2.2.</p> <p>Операторы языка СИ++</p>	<p>Содержание учебного материала</p> <p>1 Стандартные функции языка СИ++.</p>	50	2

<p>Операторы языка. Синтаксис операторов.</p> <p>Объявления переменной</p> <p>Спецификатор типа</p> <p>модификатор-типа char</p> <p>Указатели</p>			
<p>2 Структура программы.</p> <p>Инициализация данных</p> <p>Операции увеличения и уменьшения</p> <p>Приоритеты операций и порядок вычислений</p> <p>Связь между функциями осуществлялась по данным посредством передачи параметров и возврата значений функций. Но компилятор языка СИ позволяет также разбить программу на несколько отдельных частей (исходных файлов), оттранслировать каждую часть отдельно, и затем объединить все части в один выполняемый файл при помощи редактора связей.</p>		2	2
<p>3 Операторы управления. Оператор if. Составной оператор.</p> <p>Вложенные условные операторы</p> <p>Синтаксис</p> <p>Выбор одного из нескольких альтернативных путей</p> <p>Полная форма оператора</p> <p>Неполная форма оператора if.</p> <p>Примеры использования оператора</p>		2	2
<p>4 Оператор switch.</p> <p>Способ изображения многопутевого ветвления</p> <p>Примеры использования оператора</p>		2	2
<p>5 Циклические конструкции. Оператор for.</p> <p>Синтаксис инструкции for</p> <p>"обоскопченный" цикл</p> <p>несколько циклов, вложенных друг в друга.</p> <p>Примеры использования оператора</p>		2	2
<p>6 Операторы while do while.</p> <p>Синтаксис</p> <p>отличие от for</p> <p>Примеры использования оператора</p>		2	2
<p>7 Операторы безусловного перехода</p> <p>Синтаксис</p> <p>Метка</p> <p>Операторы break и continue</p> <p>Примеры использования оператора</p>		2	2
<p>Лабораторное занятие Разработка программ линейной структуры.</p>		4	2
<p>Лабораторное занятие ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ.</p>		6	2
<p>Лабораторное занятие ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЦИКЛИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ</p>		6	3
<p>Практические занятия</p>			
<p>Самостоятельная работа обучающихся.</p>		20	

Тема 2.3. Структурирование типы данных	Подготовка к лабораторным работам, оформление отчетов, тестовые задания		
	Содержание учебного материала	46	
	1 <u>Массивы. Объявление массива.</u> Спецификатор-типа задает тип элементов объявляемого массива. Элементами массива не могут быть функции и элементы типа void. Константное-выражение в квадратных скобках задает количество элементов массива. Инициализация массива символов	2	2
	2 Ввод и вывод одномерных массивов. Примеры задач использование динамических переменных оператор new. Синтаксис Освобождение памяти	2	2
	3 Ввод и вывод двумерных массивов. Обработка массивов. Стандартные функции для массива целых и вещественных чисел.	2	2
	4 <u>Объявление строковых типов данных.</u> Стандартные функции и процедуры работы со строками. Ввод-вывод символьных данных. Ввод-вывод символьных данных. Строки и указателей. Массив строк. Функции для работы со строками Библиотека string.h	2	2
	5 <u>Работа в оконном режиме.</u> Работа с компонентами Plable, TButton, Свойства компонент Примеры использования	2	2
	6 Работа с компонентом TEdit.	2	2
	<u>Лабораторное занятие</u> Обработка одномерных и двумерных массивов.	6	2
	<u>Лабораторное занятие</u> Работа со строковыми переменными.	6	2
	<u>Лабораторное занятие</u> Разработка интерфейса программы с использованием основных компонент	6	2
	Практические занятия	-	
	Самостоятельная работа обучающихся. Подготовка к лабораторным работам тестовые задания	16	

Тема 2.4. Процедуры и функции	<p>Содержание учебного материала</p> <p>1 Понятие подпрограммы. Процедуры и функции, их сущность.</p> <p>Назначение и различие.</p> <p>Объявления устанавливают соответствие имени и аргументов переменной, функции или типа.</p> <p>Определение переменной вызывает выделение памяти для хранения ее значения. В языке СИ/С++ блоком считается последовательность объявлений, определений и операторов, заключенная в фигурные скобки. Существуют два вида блоков - составной оператор и определение функции, состоящее из составного оператора, являющегося телом функции, и предшествующего телу заголовка функции</p> <p>2 Объявление процедур и функций.</p> <p>Процедурные типы. Процедурные переменные</p> <p>Функция, правила их записи, вызов и передачи параметров.</p> <p>Инструкции testp</p> <p>прототип функции</p> <p>Лабораторное задание Разработка программ с использованием нестандартных функций</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся.</p> <p>Подготовка к лабораторным работам</p>	14	
Тема 2.5. Работа с файлами	<p>Содержание учебного материала</p> <p>1 Работа с файлами.</p> <p>Режимы открытия и закрытия файлов</p> <p>Общая схема работы с файлами. Текстовые файлы</p> <p>Режим открытия файлов in, out, binary, app, ate, truncate</p> <p>Классы ifstream и ofstream</p> <p>Бинарные файлы</p> <p>Лабораторное задание Обработка текстовых файлов</p> <p>Практические занятия</p> <p>Самостоятельная работа обучающихся.</p> <p>Подготовка к лабораторным работам.</p>	6 - 4 12	2 2
Раздел 3. Машинный язык и язык ассемблера	<p>6</p> <p>-</p> <p>4</p> <p>26</p> <p>4</p>	2	2
Тема 3.1 Основные элементы языка	<p>Содержание учебного материала</p> <p>1 Структура программы на языке ассемблера</p> <p>«Шанка» программы. Назначение</p> <p>Тело программы. Тело программы также можно поделить на 2 части - инициализацию и, собственно, выполнение какой-то задачи</p>	2	2

	<p>Операнды</p> <p><u>2 Методы адресации МП</u></p> <p>Режимом адресации называют процедуру нахождения операнда для выполняемой команды.</p> <p>Регистровая адресация.</p> <p>Непосредственная адресация</p> <p>Прямая адресация памяти</p> <p><u>Лабораторные занятия</u></p> <p><u>Практические занятия</u></p>	2	2
<p>Тема 3.2</p> <p>Основные команды и директивы языка ассемблера</p>	<p>Содержание учебного материала</p> <p><u>1 Арифметические команды. Логические операции.</u></p> <p>Синтаксис</p> <p>Команды арифметического сложения ADD и ADC</p> <p>Команды арифметического вычитания SUB и SBB</p> <p>Команды инкремента INC и декремента DEC</p> <p>Команды умножения MUL и IMUL</p> <p>Команды деления DIV и IDIV</p> <p>логическое умножение (I, AND), логическое сложение (ИЛИ, OR), исключающее ИЛИ (XOR) и отрицание (NOT).</p> <p><u>2 Команды пересылки данных и команды сдвига.</u></p> <p>Синтаксис. Примеры использования</p> <p>Команда MOV</p> <p>Команда обмена данных XCHG</p> <p>команды логического сдвига вправо SHR и влево SHL;</p> <p>команды арифметического сдвига вправо SAR и влево SAL;</p> <p>команды циклического сдвига вправо ROR и влево ROL;</p> <p>команды циклического сдвига вправо RCR и влево RCL с переносом</p> <p>Каждая команда содержит два операнда: первый операнд – поле данных – может быть либо регистром, либо ячейкой памяти; второй операнд – счетчик сдвигов.</p> <p><u>3 Команды передачи управления.</u></p> <p>Синтаксис</p> <p>Команды передачи управления обеспечивают переход из одной части программы в другую. По принципу действия они делятся на три группы:</p> <p>1. Команды безусловной передачи управления:</p> <ul style="list-style-type: none"> - команды безусловного перескода; - вызова процедуры и возврата из процедуры; - вызова программных прерываний и возврата из программных прерываний. <p>2. Команды условной передачи управления:</p> <ul style="list-style-type: none"> - команды переходы по результату команды сравнения cmp; 	2	2
		2	2

	- команды перехода по состоянию определенного флага; - команды перехода по содержимому регистра cx (схх). 3. Команды управления циклом: - Команда организации цикла со счетчиком cx (схх); - Команда организации цикла со счетчиком cx (схх) с возможностью досрочного выхода из цикла по дополнительному условию		
	4 Команды ввода-вывода. Программные и аппаратные прерывания Синтаксис IN - команда ввода из ВНУ; OUT - команда вывода на ВНУ	2	2
	5 Директивы (псевдооператоры) Директивы ассемблера могут быть разделены на ряд категорий: • символические определения, • резервирование пространства памяти, • инициализация данных, • управление состоянием ассемблера, • выбор сегментов, • определение макрокоманд	2	2
	Лабораторное занятие Разработка программ на ассемблере с использованием арифметических команд	4	2
	Лабораторное занятие Разработка программ на ассемблере с использованием логических операций	4	2
	Практические занятия	-	
	Самостоятельная работа обучающихся. Подготовка к лабораторным работам.	4	
	Итого	180	

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 УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Материально-техническое обеспечение

Реализация учебной дисциплины требует наличия, лаборатории программирования.

Оборудование учебного кабинета математики:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно-методической документации.

Технические средства обучения:

- персональный компьютер;
- проекционный экран;
- мультимедийный проектор;
- доска;
- колонки.

Оборудование лаборатории и рабочих мест лаборатории инструментальных средств разработки: персональные компьютеры (монитор, системный блок, клавиатура, мышь), комплект учебно-методической документации, программное обеспечение (оболочки языков программирования).

3.2. Информационное обеспечение обучения

Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники

1. Семакин И.Г. Основы алгоритмизации и программирования [Текст]: учебник для студентов учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. — 2-е изд., стер. — М.: Издательский центр «Академия», 2014. — 304 с.
2. Канцедал С.А. Алгоритмизация и программирование [Электронный ресурс]: учебное пособие / С.А. Канцедал. — М.: ИД «ФОРУМ»: ИНФРА-М, 2019. — 352 с. — (Среднее профессиональное образование). - Режим доступа: <http://znanium.com/catalog/product/987207>

Дополнительные источники

1. Семакин И.Г. Основы алгоритмизации и программирования. Практикум [Текст]: учебное пособие для студентов учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. — 2-е изд., стер. — М.: Издательский центр «Академия», 2014. — 144 с.
2. Колдаев В.Д. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2019. — 414 с. — (Среднее профессиональное образование). - Режим доступа: <http://znanium.com/catalog/product/980416>

Электронные ресурсы:

1. Информационная система "Единое окно доступа к образовательным ресурсам" [Электронный ресурс]. — ФГАУ ГНИИ ИТТ "Информика", 2005-2019. — Режим доступа: <http://window.edu.ru> , свободный. — Заглавие с экрана (дата обращения: 28.08.2018)

2. Основы алгоритмизации и языки программирования [Электронный ресурс]. — Обучение в интернет, 2007 - 2019. — Режим доступа: <https://www.lessons-tva.info/edu/e-inf1/e-inf1-4-2.html> , свободный. — Заглавие с экрана (дата обращения: 05.09.2018)

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

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

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
В результате освоения учебной дисциплины обучающийся должен уметь :	
формализовать поставленную задачу	<ul style="list-style-type: none"> - наблюдение за выполнением лабораторных работ ; - оценка выполнения заданий к самостоятельной работе (СРС).
применять полученные знания к различным предметным областям	<ul style="list-style-type: none"> - наблюдение за деятельностью студентов на учебных занятиях; - наблюдение за деятельностью студента на производственной практике по ПМ.02. «Применение микропроцессорных систем, установка и настройка периферийного оборудования».
составлять и оформлять программы на языках программирования	<ul style="list-style-type: none"> - наблюдение за выполнением лабораторных работ; - оценка выполнения заданий к самостоятельной работе (СРС).
тестировать и отлаживать программы	<ul style="list-style-type: none"> - наблюдение за выполнением лабораторных работ.
В результате освоения учебной дисциплины обучающийся должен знать :	
общие принципы построения программ и использования языков программирования, их классификацию	<ul style="list-style-type: none"> - оценка качества сформированных знаний студента при проведении устного опроса; - контроль усвоения знаний студентов в форме контрольной работы; - проверка конспектов лекций; - оценка выполнения лабораторных работ
современные интегрированные среды разработки программ	<ul style="list-style-type: none"> - оценка выполнения лабораторных работ - проверка конспектов лекций; - оценка качества сформированных знаний студента при проведении устного опроса; - оценка качества сформированных знаний студента при проведении тестирования; - контроль усвоения знаний студентов в форме контрольной работы.
процесс создания программ	<ul style="list-style-type: none"> - оценка выполнения лабораторных работ - проверка конспектов лекций; - оценка качества сформированных знаний студента при проведении устного опроса.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
стандарты языков программирования, общую характеристику языков ассемблера: назначение, принципы построения и использования	<ul style="list-style-type: none"> - оценка качества знаний осуществляется при изучении ПМ.02 Микропроцессорные системы (МДК.02.01); - оценка выполнения заданий к самостоятельной работе (СРС).
общие принципы построения алгоритмов, основные алгоритмические конструкции	<ul style="list-style-type: none"> - наблюдение за качеством работы студента на лабораторных занятиях - проверка конспектов лекций; - оценка выполнения заданий к самостоятельной работе (СРС).