



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
филиал федерального государственного бюджетного образовательного учреждения
высшего образования Кубанский государственный университет»
в г.Геленджике



УТВЕРЖДАЮ

Проректор по работе с филиалами

А.А. Евдокимов

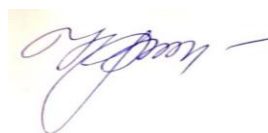
2020 г.

Рабочая программа дисциплины
МДК.01.01. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ
специальность 09.02.03 Программирование в компьютерных системах

2020

Рабочая программа учебной дисциплины МДК.01.01. Системное программирование разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки РФ от 28.07.2014 №804 (зарегистрирован в Минюсте России 21.08.2014 № 33733)

Дисциплина	МДК.01.01. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ		
Форма обучения	очная		
Учебный год	2020-2021		
2 курс		4 семестр	
лекции		36 час.	
практические занятия		36 час.	
самостоятельные занятия		36 час.	
форма итогового контроля		диффер.зачет	



Составитель: преподаватель

Т.П. Кривошеенко

Рабочая программа утверждена на заседании предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах

Протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах



Л.А. Благова

подпись

Рецензент,
Директор ООО «Современные
информационные технологии»



А.В.Сметанин

Программист ГБУЗ
"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников

ЛИСТ
согласования рабочей учебной программы по дисциплине
МДК.01.01 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ
Специальность среднего профессионального образования:
09.02.03 Программирование в компьютерных системах

СОГЛАСОВАНО:

Зам. директора по УР филиала



Т. А. Резуненко

«27» мая 2020 г.

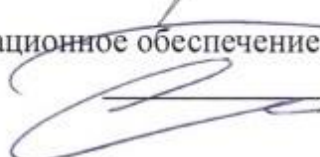
Заведующая сектором библиотеки



Л. Г. Соколова

«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение
образовательной программы)



А. В. Сметанин

«27» мая 2020 г.

СОДЕРЖАНИЕ

1.1. Область применения программы.....	5
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:.....	5
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины:.....	7
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций).....	7
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
2.1. Объем учебной дисциплины и виды учебной работы	14
2.2. Структура дисциплины.....	14
2.3. Тематический план и содержание учебной дисциплины.....	15
МДК.01.01 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	15
2.4. Содержание разделов дисциплины	19
2.4.1. Занятия лекционного типа.....	19
2.4.2. Занятия семинарского типа	20
2.4.3. Практические занятия	20
2.4.4. Содержание самостоятельной работы	21
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	26
4.2. Перечень необходимого программного обеспечения	26
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	27
5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	28
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ	29
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ «СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»	29
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ	33
7.1. Паспорт фонда оценочных средств	33
7.2. Критерии оценки знаний	33
7.3. Оценочные средств для проведения текущей аттестации	34
7.4. Оценочные средств для проведения промежуточной аттестации.....	42
7.4.1. Примерные вопросы для проведения промежуточной аттестации (экзамен, зачет)	42
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	44

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.01.01 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

1.1. Область применения программы

Рабочая программа учебной дисциплины МДК.01.01. Системное программирование является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах.

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Учебная дисциплина относится к общепрофессиональным дисциплинам профессионального цикла ПМ.

Для освоения дисциплины обучающиеся используют знания, умения и навыки, сформированные при изучении дисциплины ПД.02 «Информатика», ОП.02 «Архитектура компьютерных систем» (), ОП.04 «Информационные технологии», «Основы программирования», «Ввод и обработка данных с помощью прикладного программного обеспечения»

№ п. п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	практический опыт (владеть)
1	ПК 1.6	Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификации.	Знать графические языки спецификаций, применяемые для разработки проектной и технической документации	Использовать средства и методы разработки требований и спецификаций.	Владеть методами абстрагирования спецификаций до уровня требований; уметь разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.
2	ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.	Методы разработки технических документов, методические и нормативные документы, техническую документацию, а также предложения и мероприятия по реализации разработанных проектов и программ	Разработка технических заданий: способность осуществлять разработку технических заданий и проектов, технико-экономического обоснования	Проверять соответствие выполняемых работ требованиям проектной документации, анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения

3	ПК 3.2	Выполнять интеграцию модулей в программную систему.	Знать методы и технологии разработки программной системы, способы генерации исполняемого кода	Уметь выполнять интеграцию модулей в программную систему	Владеть методами и инструментами анализа и проектирования программного обеспечения, технологией интеграции проектных компонент и подсистем и поддерживающим инструментарием
4	ПК 1.5	Осуществлять оптимизацию программного кода модуля.	Организацию и принцип работы основных логических блоков компьютерных систем	Подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы	Владеть подключением дополнительного оборудования и настраиванием связи между элементами компьютерной системы
5	ПК 2.4	Реализовывать методы и технологии защиты информации в базах данных.	Базовые понятия и основные принципы построения архитектур вычислительных систем	Производить установку и настройку программного обеспечения компьютерных систем	Владеть навыком настройки программного обеспечения компьютерных систем
6	ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.	Процессы обработки информации на всех уровнях компьютерных архитектур, основные компоненты программного обеспечения компьютерных систем	Получать информацию о параметрах компьютерной системы, Производить установку и настройку программного обеспечения компьютерных систем	Владеть навыками анализа проектной и технической документации на уровне взаимодействия компонент программного обеспечения
7	ПК 3.2	Выполнять интеграцию модулей в программную систему.	Принципы управления ресурсами в операционной системе, Особенности построения и функционирования операционных систем Unix и Windows	Управлять дисками и файловыми системами	Иметь практический опыт интеграции модулей в программную систему

Изучение дисциплины «Системное программирование» предваряет изучение дисциплин ОП.01 «Операционные системы», ОП.03 «Технические средства информатизации», ОП.11 «Информационная безопасность», ОП.12 «Интернет программирование. Web-программирование», МДК.01.02 «Прикладное программирование», МДК.02.02 «Технология разработки и защиты баз данных»,

МДК 03.01 «Технология разработки программного обеспечения», МДК 03.02 «Инструментальные средства программного обеспечения», МДК 03.03 «Документирование и сертификация».

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

В результате изучения обязательной части учебного цикла дисциплины обучающийся должен:

уметь:

- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- оформлять документацию на программные средства;
- использовать инструментальные средства для автоматизации оформления документации.

знать:

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные принципы отладки и тестирования программных продуктов;
- методы и средства разработки технической документации.

Максимальная учебная нагрузка обучающегося 108 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 72 часа;
- самостоятельная работа обучающегося 36 часов.

1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)

Обучающийся должен обладать общими компетенциями, включающими в себя способность:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, определять 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, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

ОК 6. Работать в коллективе и команде, обеспечивать ее сплочение, 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.

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 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
1.	ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	Свободное владение профессиональными знаниями вводом и обработки информации, использование современных компьютерных программных продуктов для решения задач профессиональной деятельности и за ее пределами	умения использовать знания при вводе и обработки информации для развития профессиональных навыков, способность использовать современные инструментальные программные средства при решении социальных и профессиональных задач	Практическое использование профессиональных знаний: способность самостоятельно использовать на практике интегрированных знаний вводом и обработки информации, умение выдвигать и применять идеи, вносить оригинальный вклад в будущую профессию
2.	ОК 2	Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	знание основных тенденций развития прикладного программного обеспечения, способность использовать базовые положения при решении социальных и	умение использовать прикладное программное обеспечение для решения различных социальных, производственных, управленческих и других	способностью критически переосмысливать накопленный опыт, вносить изменения в рабочие процессы с учетом инноваций, совершенствовать навыки самостоятельной разработки методов и средств

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
			профессиональных задач, развивать способность к приращению знаний и внедрению передового российского и мирового опыта в своей профессиональной и междисциплинарной областях	профессиональных задач, критически оценить освоенные технологии, границы их применимости	автоматизации информационных процессов; делать свой вклад в оптимизацию рабочих процессов с учетом развития науки и технологий
3.	ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Знать методы принятия решений в рамках компетентности специалиста, знать о кризис-менеджменте	Анализ информации: способность осуществлять сбор, обработку, анализ и систематизацию информации, обобщать и критически оценивать результаты	Выработка и принятие управленческих решений: способность разрабатывать варианты управленческих решений и обосновывать их выбор в рамках компетентности специалиста. Кризис-менеджмент: способностью управлять в кризисных ситуациях и применять технологии кризис-менеджмента.
4.	ОК 4	Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и	Возможности сетевых технологий работы с информацией.	Осуществлять поиск информации используя ресурсы локальной и глобальной информационной сетей.	способность осуществлять сбор, обработку, анализ и систематизацию необходимой информации, выбор подходящей технологии, инструментальных средств решения профессиональных

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
		личностного развития			задач, используя обзоры научной литературы и электронные информационно-образовательные ресурсы, информационно-коммуникационные технологии
5.	ОК 5.	Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности	владение теоретическими основами ввода и обработки данных с помощью прикладного программного обеспечения, готовность применять основные их методы в своей профессиональной деятельности, знание методов самостоятельного поиска и использования различных источников информации по проблеме	осуществлять сбор, обработку, анализ и систематизацию научно-технической информации по профессии, выбирать методику и средства решения задач, используя научную литературу и электронные информационно-образовательные ресурсы, информационно-коммуникационные технологии	способность учитывать современные тенденции развития прикладной математики, информатики и вычислительной техники, компьютерных технологий в профессиональной деятельности, способность проводить научные, в том числе маркетинговые, исследования; готовность разрабатывать необходимое программное обеспечение для совершенствования профессиональной деятельности
6.	ОК 6	Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями	знание методов работы в коллективе и команде; понимание психологических особенностей функционирования личности, группы, общества, мирового	способность включаться в работу профессиональных групп; способность организовывать сотрудничество обучающихся, поддерживать активность и инициативность,	способность использовать современные информационно-коммуникативные технологии в работе с текстами, информацией, результатами исследований и для реализации профессиональной

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
			сообщества	самостоятельность и творческие способности, готовность к взаимодействию с учениками, родителями, коллегами, социальными партнерами	деятельности; способность делать вклад в личностный рост и повышение эффективности других участников профессиональной деятельности
7.	ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	Владеть психологическими знаниями методов работы в коллективе и команде; понимать психологические особенности функционирования личности, группы, общества, мирового сообщества	Уметь развивать аналитическое мышление, ответственность, коммуникабельность, креативность, инициативность, эмоциональную сдержанность, лидерские и организаторские качества	Развивать в себе аналитическое мышление, методичность, дисциплинированность, коммуникабельность, креативность, организованность, инициативность стрессоустойчивость, толерантность, ответственность, требовательность, коммуникабельность, умение убеждать
8.	ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Следить за отечественными и зарубежными разработками в области информационных и коммуникационных технологий	Осваивать новые методы и технологии в области инфокоммуникационных систем и сетей	Развивать способность к освоению новых технологий; организовывать самообразование и повышение своей квалификации
9.	ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Работа со знаниями: Способность к приращению знаний и внедрению передового российского и мирового опыта	Уметь выполнять разнообразные профессиональные задачи с использованием на практике своих знаний и навыков в различных	Способностью и готовностью к творческой адаптации к конкретным условиям выполняемых задач и их инновационным

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
			в своей профессиональной и междисциплинарной областях.	меняющихся и нестандартных условиях	решениям, способность порождать новые идеи в условиях смены технологий.
	ПК 1.1	Выполнять разработку спецификаций отдельных компонент.	основные этапы разработки программного обеспечения	создавать программу по разработанному алгоритму как отдельный модуль	Способностью создавать программу по разработанному алгоритму как отдельный модуль
	ПК 1.2	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	основные этапы разработки программного обеспечения	осуществлять разработку кода программного модуля на современных языках программирования	Способностью осуществлять разработку кода программного модуля на современных языках программирования
	ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.	основные принципы технологии структурного и объектно-ориентированного программирования	выполнять отладку и тестирование программы на уровне модуля	Иметь практический опыт выполнения отладки и тестирования программы на уровне модуля
	ПК 1.4	Выполнять тестирование программных модулей.	основные принципы отладки и тестирования программных продуктов	выполнять отладку и тестирование программы на уровне модуля	Иметь практический опыт выполнения отладки и тестирования программы на уровне модуля
	ПК 1.5	Осуществлять оптимизацию программного кода модуля.	основные принципы технологии структурного и объектно-ориентированного программирования	создавать программу по разработанному алгоритму как отдельный модуль	Способностью создавать программу по разработанному алгоритму как отдельный модуль
	ПК 1.6	Разрабатывать компоненты проектной и технической документации с	методы и средства разработки технической документации	оформлять документацию на программные средства; использовать	Способностью оформлять документацию на программные средства;

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	Уметь	Практический опыт (владеть)
		использованием графических языков спецификаций.		инструментальные средства для автоматизации оформления документации	Иметь практический опыт использовать инструментальные средства для автоматизации оформления документации

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	108
Обязательная аудиторная учебная нагрузка (всего)	72
в том числе:	
занятия лекционного типа	36
практические занятия	36
Самостоятельная работа обучающегося (всего)	36
в том числе:	
самостоятельная внеаудиторная работа в виде домашних практических заданий, индивидуальных заданий, самостоятельного подбора и изучения дополнительного теоретического материала	36
Промежуточная аттестация в форме зачета/экзамена/дифзачета	Диф.зачет

2.2. Структура дисциплины

Наименование разделов и тем	Количество аудиторных часов			Самостоятельная работа обучающегося (час)
	Всего	Теоретическое обучение	Практические и лабораторные занятия	
Раздел 1. Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования.	12	4	4	4
Тема 1.1. Языки и эволюция технологий программирования.	6	2	2	2
Тема 1.2. Состав и схема функционирования классической системы программирования.	6	2	2	2
Раздел 2. Языки и их представление. Грамматика	18	6	6	6
Тема 2.1. Основные понятия теории формальных грамматик и языков.	6	2	2	2
Тема 2.2. Классификация формальных грамматик и языков по Хомскому	6	2	2	2
Тема 2.3. Общая схема работы компилятора.	6	2	2	2
Раздел 3. Лексический анализ	12	4	4	4
Тема 3.1. Лексический анализ.	6	2	2	2
Тема 3.2. Объектная модель лексического анализатора. Схемы его работы.	6	2	2	2
Раздел 4. Синтаксический анализ	18	4	6	6
Тема 4.1. Метод рекурсивного спуска(РС):О применимости метода РС.	6	2	2	2
Тема 4.2. достаточные условия применимости метода рекурсивного спуска.	6	2	2	2
Раздел 5. Семантический анализ	12	4	4	4
Тема 5.1. Задачи семантического анализа.	6	2	2	2
Тема 5.2. Семантический анализатор для	6	2	2	2

модельного языка.				
Раздел 6. Промежуточное представление программы	18	6	6	6
Тема 6.1. Генерация внутреннего представления программы.	6	2	2	2
Тема 6.2. Синтаксически управляемый перевод.	6	2	2	2
Тема 6.3. Генератор внутреннего представления программы на М-языке.	6	2	2	2
Раздел 7. Отладчики. Загрузчики. Библиотеки.	24	8	8	8
Тема 7.1. Интегрированная среда разработки программного обеспечения	6	2	2	2
Тема 7.2. Основные функции редакторов текстов в рамках ИСР.	6	2	2	2
Тема 7.3. Загрузчик. Библиотеки. Основные типы библиотек.	12	4	4	4
Всего по дисциплине	108	36	36	36

2.3. Тематический план и содержание учебной дисциплины МДК.01.01 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа обучающихся, курсовая работа (если предусмотрена)	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. Понятия системного и прикладного программного обеспечения. Основные понятия системы программирования.		12	
Тема 1.1. Языки и эволюция технологий программирования.	Содержание учебного материала	6	1
	Лекции		
	1 Описание языков программирования.	2	
	2 Эволюция языков программирования.		
	Практические занятия	2	
	1 Построение алгоритма для работы анализатора текста для выделения заданной последовательности символов.		
Самостоятельная работа обучающихся	2		
1 Построение алгоритма для работы анализатора текста для выделения заданной последовательности символов.			
Тема 1.2. Состав и схема функционирования классической системы программирования.	Содержание учебного материала	6	1
	Лекции		
	1 Состав классической системы программирования.	2	
	2 Схема функционирования классической системы программирования.		
	Практические занятия	2	
	1 Реализация алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на С++.		
Самостоятельная работа обучающихся	2		
1 Реализация алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на С++.			
Раздел 2. Языки и их представление. Грамматики.		18	2,3
Тема 2.1. Основные	Содержание учебного материала	6	
	Лекции		

понятия теории формальных грамматик и языков.	1	Формальные грамматики.	2	
	2	Формальные языки.		
	Практические занятия		2	
	1	Тестирование алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на языке С++ и исследование возможностей его модификации.		
	Самостоятельная работа обучающихся			
1	Описание процесса выполнения программ.	2		
Тема 2.2. Классификация формальных грамматик и языков по Хомскому.	Содержание учебного материала		6	2,3
	Лекции			
	1	Классификация формальных грамматик по Хомскому.	2	
	2	Классификация формальных языков по Хомскому.		
	Практические занятия			
	1	Решение задач на построение различных видов цепочек символов для заданной грамматики.	2	
	Самостоятельная работа обучающихся			
1	Решение задач на построение различных видов цепочек символов для заданной грамматики.	2		
Тема 2.3. Общая схема работы компилятора	Содержание учебного материала		6	2,3
	Лекции			
	1	Этапы компилирования.	2	
	2	Схема работы компилятора.		
	Практические занятия			
	1	Решение задач на построение сентенциальной формы для заданной грамматики.	2	
	Самостоятельная работа обучающихся			
1	Решение задач на построение сентенциальной формы для заданной грамматики.	2		
Раздел 3. Лексический анализ			12	
Тема 3.1. Лексический анализ.	Содержание учебного материала		6	2,3
	Лекции			
	1	Задачи лексического анализа.	2	
	2	Модельный язык.		
	3	Лексический анализатор для М-языка на основе регулярных грамматик.		
	Практические занятия			
	1	Решение задач на классификацию грамматик и на построение грамматик.	2	
Самостоятельная работа обучающихся				
1	Решение задач на классификацию грамматик и на построение грамматик.	2		
Тема 3.2. Объектная модель лексического анализатора. Схемы его работы.	Содержание учебного материала		6	2,3
	Лекции			
	1	Объектная модель лексического анализатора.	2	
	2	Схема работы объектной модели лексического анализатора.		
	Практические занятия			
	1	Решение задач на построение регулярных грамматик и эквивалентных грамматик с заданными правилами.	2	
Самостоятельная работа обучающихся				
1	Решение задач на построение регулярных грамматик и эквивалентных грамматик с заданными правилами.	2		
Раздел 4. Синтаксический анализ			18	
Тема 4.1. Метод рекурсивног	Содержание учебного материала		12	2,3
	Лекции			
	1	Метод рекурсивного спуска.	4	

о спуска (РС): О применимости метода РС.	2	Исследование применимости метода РС.		
	Практические занятия			
	1	Решение задач на построение дерева разбора грамматик.	4	
	Самостоятельная работа			
	1	Решение задач на построение дерева разбора грамматик.	4	
Тема 4.2. Достаточные условия применимости метода рекурсивного спуска.	Содержание учебного материала		6	2,3
	Лекции			
	1	Синтаксический анализатор для модельного языка.		
	2	РС-метода в случае наличия ε-альтернативы и итерационных правил.	2	
	Практические занятия			
	1	Решение задач по пройденному материалу.	2	
	Самостоятельная работа			
1	Подготовка к контрольной работе на решение задач по пройденному материалу.	2		
Раздел 5. Семантический анализ.			12	
Тема 5.1. Задачи семантического анализа.	Содержание учебного материала		6	2,3
	Лекции			
	1	Граматики с действиями.	2	
	2	Семантические процедуры.		
	Практические занятия			
	1	Контрольная работа на решение задач по пройденному материалу.	2	
Самостоятельная работа				
1	Ответить на теоретические вопросы по контрольной работе.	2		
Тема 5.2. Семантический анализатор для модельного языка.	Содержание учебного материала		6	2,3
	Лекции			
	1	Контроль контекстных условий в выражениях.	2	
	2	Контроль контекстных условий в операторах.		
	Практические занятия			
	1	Разбор результатов контрольной работы и работы над ошибками.	2	
Самостоятельная работа				
1	Написать реферат. Темы даны индивидуально.	2		
Раздел 6. Промежуточное представление программы.			18	
Тема 6.1 Генерация внутреннего представления программы.	Содержание учебного материала		6	2,3
	Лекции			
	1	Язык внутреннего представления программы.	2	
	2	Некоторые общепринятые способы внутреннего представления программы.		
	Практические занятия			
	1	Создание объектной модели лексического анализатора в виде работающей программы.	2	
Самостоятельная работа				
1	Создание объектной модели лексического анализатора в виде работающей программы.	2		
Тема 6.2 Синтаксически управляемый перевод.	Содержание учебного материала		6	2,3
	Лекции			
	1	Понятие синтаксически управляемого перевода.	2	
	2	Схемы синтаксически управляемого перевода.		
Практические занятия				
			2	

	1	Создание объектной модели синтаксического анализатора в виде работающей модели.		
	Самостоятельная работа			
	1	Создание объектной модели синтаксического анализатора в виде работающей модели.	2	
Тема 6.3 Генератор внутреннего представлен ия программы на М-языке.	Содержание учебного материала		6	2,3
	Лекции			
	1	Интерпретатор ПОЛИЗ.	2	
	2	Принципы реализации виртуальных функций.		
	Практические занятия			
	1	Создание объектной модели семантического анализатора в виде работающей программы.	2	
	Самостоятельная работа			
	1	Создание объектной модели семантического анализатора в виде работающей программы.	2	
Раздел 7. Отладчики. Загрузчики. Библиотеки.			24	
Тема 7.1 Интегрирова нная среда разработки программно го обеспечения	Содержание учебного материала		6	
	Лекции			
	1	История развития ИСР.	2	
	2	Достоинства и недостатки программ.		
	Практические занятия			
	1	Разработка для данной грамматики анализатора, действующего методом рекурсивного спуска.	2	
	Самостоятельная работа			
	1	Подготовка к контрольной работе на решение задач по пройденному материалу.	2	
Тема 7.2 Основные функции редакторов текстов в рамках ИСР.	Содержание учебного материала		6	
	Лекции			
	1	Функции редакторов текстов.	2	
	2	Отладчики и их возможности.		
	Практические занятия			
	1	Контрольная работа на решение задач по пройденному материалу.	2	
	Самостоятельная работа			
	1	Ответить на теоретические вопросы по контрольной работе	2	
Тема 7.3 Загрузчик. Библиотеки. Основные типы библиотек.	Содержание учебного материала		12	
	Лекции			
	1	Понятие загрузчик, библиотека.	4	
	2	Основные типы библиотек.		
	Практические занятия			
	1	Разбор результатов контрольной работы работа над ошибками.	4	
	2	Решение задач для подготовки к экзамену.		
Самостоятельная работа				
	1	Подготовка к экзамену.	4	
Всего:			108	

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
<i>4 семестр</i>			
1	Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования.	Понятие системного программного обеспечения, понятие прикладного программного обеспечения, описание языков программирования, этапы жизненного цикла программного продукта, состав и схема функционирования классической системы программирования, типы трансляторов, особенности интерпретаторов и компиляторов, смешанная стратегия трансляции, языки и эволюция технологий программирования	У
2	Языки и их представление. Грамматики.	Основные понятия теорий формальных грамматик и языков, разбор цепочек, эквивалентные грамматики, классификация формальных грамматик и языков по Хомскому, соотношение между типами грамматик, соотношение между типами языков, примеры грамматик языков, неоднозначность грамматик и языков, недостижимые и бесполезные (бесплодные) символы грамматики, алгоритмы удаления недостижимых и бесполезных (бесплодных) символов, приведенная грамматика, алгоритм устранения правил с пустой правой частью, определение недетерминированного конечного автомата по НКА, алгоритм моделирования работы ДКА, конечные автоматы с действиями, общая схема работы компилятора	У
3	Лексический анализ.	Лексический анализ, задачи лексического анализа, модельный язык, лексический анализатор для М-языка на основе регулярных грамматик, объектная модель лексического анализатора, схема его работы.	У
4	Синтаксический анализ	Задачи синтаксического анализа, метод рекурсивного спуска (РС-метод): О применимости метода РС, достаточные условия применимости метода рекурсивного спуска, Исследование применимости метода РС, РС-метода в случае наличия ϵ -альтернативы и итерационных правил, синтаксический анализатор для модельного языка	У
5	Семантический анализ	Задачи семантического анализа, грамматики с действиями, семантические процедуры, семантический анализатор для модельного языка, обработка описаний, контроль контекстных условий в выражениях, контроль контекстных условий в операторах, создание функции для описания переменных для М-языка (нетерминал D и М-языке) и операторов	У
6	Промежуточное представление программы	Генерация внутреннего представления программы, язык внутреннего представления программы, некоторые общепринятые способы внутреннего представления программы, ПОЛИЗ выражений, синтаксически управляемый перевод, генератор внутреннего представления программы на М-языке, интерпретатор ПОЛИЗ, основные стратегии	У

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
		распределения памяти, принципы реализации виртуальных функций	
7	Отладчики. Загрузки. Библиотеки.	Интегрированная среда разработки программного обеспечения, основные функции редакторов текстов в рамках интегрированной среды разработки, отладчики, и их возможности, примеры интегрированности отладчика с другими компонентами интегрированной среды разработки, редактор внешних связей, его назначение принципы работы, загрузчик, библиотеки, основные типы библиотек	У,Т
Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа			

2.4.2. Занятия семинарского типа

не предусмотрены

2.4.3. Практические занятия

№	Наименование раздела	Наименование практических (лабораторных) работ	Форма текущего контроля
<i>4 семестр</i>			
1	2	3	4
1.	Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования.	Построение алгоритма для работы анализатора текста для выделения заданной последовательности символов.	У, ПР
		Реализация алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на языке C++.	
2.	Языки и их представление. Грамматики.	Тестирование алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на языке C++ и исследование возможностей его модификации.	ПР
		Решение задач на построение различных видов цепочек символов для заданной грамматики.	
		Решение задач на построение сентенциальной формы для заданной грамматики.	
3.	Лексический Анализ	Решение задач на классификацию грамматик и на построение грамматик	ПР
		Решение задач на построение регулярных грамматик и эквивалентных грамматик с заданными правилами.	
4.	Синтаксический анализ	Решение задач на построение дерева разбора грамматик.	ПР
		Решение задач по пройденному материалу.	
5.	Семантический анализ	Контрольная работа на решение задач по пройденному материалу.	ПР, КР
		Разбор результатов контрольной работы и работы над ошибками	

6.	Промежуточное представление программы	Создание объектной модели лексического анализатора в виде работающей программы.	ПР
		Создание объектной модели синтаксического анализатора в виде работающей программы.	
		Создание объектной модели семантического анализатора в виде работающей модели.	
7.	Отладчики. Загрузчики. Библиотеки.	Подготовка к контрольной работе на решение задач по пройденному материалу.	ПР, КР
		Контрольная работа на решение задач по пройденному материалу.	
		Разбор результатов контрольной работы и работа над ошибками.	
		Решение задач для подготовки к экзамену.	

Примечание: ПР- практическая работа, ЛР- лабораторная работа; Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа

2.4.4. Содержание самостоятельной работы

Учебно-методическое обеспечение для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа обучающихся является важнейшей формой учебно-познавательного процесса.

Основная цель самостоятельной работы обучающегося при изучении дисциплины – закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области системного программирования.

Самостоятельная работа обучающегося в процессе освоения дисциплины включает:

- изучение основной и дополнительной литературы по курсу;
- самостоятельное изучение некоторых вопросов (конспектирование);
- работу с электронными учебными ресурсами;
- изучение материалов периодической печати, интернет ресурсов;
- подготовку к тестированию;
- подготовку к практическим занятиям,
- самостоятельное выполнение домашних заданий.

На самостоятельную работу обучающихся отводится 36 часов учебного времени.

№	Наименование раздела	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1.	Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования.	Партыка, Т.Л. Операционные системы, среды и оболочки: учеб. пособие для СПО /Т.Л. Партыка, И.И. Попов.- М.:ФОРУМ,2013.- 559 с. Гвоздева В.А. Введение в специальность программиста: учебник для студентов СПО. - М.: Форум, 2014. - 256с. Черпаков, И. В. Основы программирования: учебник и практикум для СПО / И. В. Черпаков. — М.: Издательство Юрайт, 2016. — 219 с.
2.	Языки и их	Царев, Р.Ю. Программные и аппаратные средства информатики

	представления. Грамматики	[Электронный ресурс]: учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков. - Красноярск: Сибирский федеральный университет, 2015. - 160 с. - URL: http://biblioclub.ru/index.php?page=book_red&id=435670&sr=1 Малявко, А. А. Параллельное программирование на основе технологий openmp, mpi, cuda [Электронный ресурс]: учебное пособие / А. А. Малявко. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2017. — 115 с. – URL: https://www.biblio-online.ru/book/46BBEB77-8697-4FF5-BE49-711BB1388D50
3.	Лексический анализ	Операционная система Microsoft Windows XP [Электронный ресурс]. – 2-е изд., испр. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 375 с. – URL: http://biblioclub.ru/index.php?page=book&id=429091 Эффективное программирование современных микропроцессоров [Электронный ресурс]: учебное пособие / В.П. Маркова, С.Е. Киреев, М.Б. Остапкевич, В.А. Перепелкин. – Новосибирск: НГТУ, 2014. – 148 с. – URL: http://biblioclub.ru/index.php?page=book&id=435972
4.	Синтаксический анализ	Лоскутов В.И. Разработка информационных систем для Windows Store [Электронный ресурс] / В.И. Лоскутов, И.Л. Коробова. – 2-е изд., исправ. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 180 с. URL: http://biblioclub.ru/index.php?page=book&id=428809 Карпов В. Основы операционных систем [Электронный ресурс]: практикум / В. Карпов, К. Коньков. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 301 с. – URL: http://biblioclub.ru/index.php?page=book&id=429022
5.	Семантический анализ	Макаров, А.В. Common Intermediate Language и системное программирование в Microsoft.NET [Электронный ресурс]: учебное пособие / А.В. Макаров и др.- М.: НОУ Интуит, 2016.- 399 с. - URL: http://biblioclub.ru/index.php?page=book&id=429250&sr=1 Кулямин В. Компонентный подход в программировании [Электронный ресурс]: учебное пособие / В. Кулямин. – 2-е изд., исправ. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 591 с. – URL: http://biblioclub.ru/index.php?page=book&id=429086
6.	Промежуточное представление программы.	Информатика и программирование [Электронный ресурс]: учебное пособие / Р.Ю. Царев, А.Н. Пупков, В.В. Самарин, Е.В. Мыльникова. – Красноярск: Сибирский федеральный университет, 2014. – 132 с. – http://biblioclub.ru/index.php?page=book_red&id=364538&sr=1 Макаров, А.В. Common Intermediate Language и системное программирование в Microsoft.NET [Электронный ресурс]: учебное пособие / А.В. Макаров и др.- М.: НОУ Интуит, 2016.- 399 с. - URL: http://biblioclub.ru/index.php?page=book&id=429250&sr=1
7.	Отладчики. Загрузчики. Библиотеки.	Кулямин В. Компонентный подход в программировании [Электронный ресурс]: учебное пособие / В. Кулямин. – 2-е изд., исправ. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 591 с. – URL: http://biblioclub.ru/index.php?page=book&id=429086 Зыков С.В. Введение в теорию программирования. Функциональный подход [Электронный ресурс]: учебное пособие / С.В. Зыков. – 2-е изд., испр. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 153 с. – URL: http://biblioclub.ru/index.php?page=book&id=429119

Кроме перечисленных источников по темам самостоятельной работы, студент может воспользоваться Электронно-библиотечными системами (ЭБС), профессиональными базами данных, электронными базами периодических изданий,

другими информационными ресурсами, указанными в разделе 5.4 «Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины», включающий ресурсы, доступ к которым обеспечен по договорам с правообладателями, и образовательные, научные, справочные ресурсы открытого доступа, имеющие статус официальных (федеральные, отраслевые, учреждений, организаций и т.п.), а также поисковыми системами сети Интернет для поиска и работы с необходимой информацией.

Для освоения данной дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе обучающийся может использовать учебно-методическое обеспечение:

-методические рекомендации преподавателя к практическим занятиям;

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Обучающийся должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Обучающиеся для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (лабораторным) занятиям. Желательно, чтобы конспекты лекций записывались в логической последовательности изучения курса и содержались в одной тетради.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для реализации компетентностного подхода предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе преподавания применяются образовательные технологии развития критического мышления. Обязательны компьютерные лабораторные практикумы по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

№	Раздел	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
1	Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования	Лекция - дискуссия	4
2	Языки и их представления. Грамматики	Аудиовизуальная технология*, лекция-дискуссия	4*
3	Лексический анализ	Аудиовизуальная технология*, лекция - дискуссия	6*
4	Синтаксический анализ	Аудиовизуальная технология*, лекция - дискуссия	6*
5	Семантический анализ	Аудиовизуальная технология*, лекция - дискуссия	6*
6	Промежуточное представление программы.	Аудиовизуальная технология*, лекция - дискуссия	6*
7	Отладчики. Загрузчики. Библиотеки.	Лекция - дискуссия	4
Итого по курсу			36
в том числе интерактивное обучение*			14*

3.2. Образовательные технологии при проведении практических занятий

№	Тема занятия	Виды применяемых образовательных технологий	Кол. час
1	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 по теме 1 раздела 1 «Построение алгоритма для работы анализатора текста для выделения заданной последовательности символов»	Дискуссия по теоретическим вопросам.	2
2	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 по теме 2 раздела 1 «Реализация алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на языке C++»	Дискуссия по теоретическим вопросам.	2

3	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3 по теме 1 раздела 2 «Тестирование алгоритма для работы анализатора текста для выделения заданной последовательности символов в виде проекта на языке C++ и исследование возможностей его модификации»	Дискуссия по теоретическим вопросам.	2
4	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 по теме 2 раздела 2 «Решение задач на построение различных видов цепочек символов для заданной грамматики»	Выполнение практической работы.	2*
5	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 по теме 1 раздела 3 «Работа с процессами и потоками»	Выполнение практической работы.	4*
6	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6 по теме 2 раздела 3 «Работа с процессами и потоками»	Выполнение практической работы.	2*
7	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7 по теме 1 раздела 4 «Синхронизация потоков»	Дискуссия по теоретическим вопросам.	2
8	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8 по теме 2 раздела 4 «Взаимодействие процессов»	Выполнение практической работы.	4*
9	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9 по теме 1 раздела 5 «Взаимодействие процессов»	Выполнение практической работы.	2*
10	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10 по теме 2 раздела 5 «Взаимоблокировка»	Выполнение практической работы.	4*
11	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 11 по теме 1 раздела 6 «Взаимоблокировка»	Выполнение практической работы	2*
12	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12 по теме 2 раздела 6 «Создание драйвера»	Выполнение практической работы	4*
13	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 13 по теме 1 раздела 7 «Создание драйвера»	Выполнение практической работы	2*
14	ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 14 по теме 2 раздела 7 «Создание драйвера»	Выполнение практической работы.	2*
		Итого по курсу	32
		в том числе интерактивное обучение*	12

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Реализация учебной дисциплины осуществляется в специально оборудованной лаборатории системного и прикладного программирования.

Оборудование учебной лаборатории прикладного и системного программирования:

- мультимедийный проектор, экран;
- персональные компьютеры;
- локальная компьютерная сеть с выходом в Интернет;
- учебная мебель;
- доска аудиторная

Наглядные пособия:

Комплект плакатов

4.2. Перечень необходимого программного обеспечения

1. Microsoft Office Professional
2. PascalABC - визуальная среда программирования;
3. PascalABC.NET - визуальная среда программирования ;
4. WEB-среда разработки PascalABC.NET.
5. Среда разработки ПО PortableDelphi;
6. Разработчик инсталляторов InnoSetup;
7. 7-zip архиватор;
8. Adobe Acrobat Reader просмотрщик файлов ;
9. Adobe Flash Player –графический редактор;
10. Apache OpenOffice – офисный пакет;
11. FreeCommander - проводник;
12. Google Chrome - браузер;
13. LibreOffice – офисный пакет
14. Mozilla Firefox – браузер
15. nanoCAD версия 5.1 локальная
16. ЭкзамL – Система компьютерного тестирования;
17. Программный комплекс "Универсальный тест 4.0.0.1";
18. Lazarus – визуальная среда программирования;

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

5.1 Основная литература

1. Черпаков, И. В. Основы программирования: учебник и практикум для СПО / И. В. Черпаков. — М.: Издательство Юрайт, 2016. — 219 с.
2. Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М.: Издательство Юрайт, 2018. — 219 с. - URL: <https://www.biblio-online.ru/viewer/F79BE55A-C6F1-439D-9ED5-0D78A50B403F#page/1>
3. Гостев, И. М. Операционные системы [Электронный ресурс]: учебник и практикум для СПО / И. М. Гостев. — М.: Издательство Юрайт, 2018. — 158 с. – URL: <https://www.biblio-online.ru/viewer/F4160D29-926C-4141-8B87-DF34DDB19B99#page/1>

5.2 Дополнительная литература

1. Гвоздева В.А. Введение в специальность программиста: учебник для студентов СПО. - М.: Форум, 2014. - 256с.
2. Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М.: Издательство Юрайт, 2018. — 213 с. – URL: <https://www.biblio-online.ru/viewer/290801FB-F8CF-47B3-9559-6BADEC310243#page/1>
3. Малявко, А. А. Параллельное программирование на основе технологий openmp, mpi, cuda [Электронный ресурс]: учебное пособие / А. А. Малявко. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2018. — 115 с. – URL: <https://www.biblio-online.ru/viewer/46BBEB77-8697-4FF5-BE49-711BB1388D50#page/1>

5.3 Периодические издания

1. Среднее и профессиональное образование
2. Компьютер Пресс
3. Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
4. Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
5. Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>
6. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
7. Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
8. Computerword Россия.- URL: <http://dlib.eastview.com/browse/publication/64081>
9. Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
10. Информационно-управляющие системы.- URL: <http://dlib.eastview.com/browse/publication/71235>
11. Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
12. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
13. Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>
14. Прикладная информатика.- URL: http://elibrary.ru/title_about.asp?id=25599

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «Университетская библиотека ONLINE». – URL: www.biblioclub.ru
2. ЭБС издательства «Лань». – URL: <https://e.lanbook.com>
3. ЭБС «Юрайт». – URL: <http://www.biblio-online.ru/>
4. ЭБС «BOOK.ru» <https://www.book.ru>
5. ЭБС «ZNANIUM.COM» www.znanium.com
6. Электронный каталог Научной библиотеки КубГУ. – URL:<http://212.192.134.46/MegaPro/Catalog/Home/Index>
7. Электронная библиотека «Издательского дома «Гребенников» - URL:www.grebennikon.ru
8. Научная электронная библиотека (НЭБ) «eLibrary.ru». - URL:<http://www.elibrary.ru>
9. Базы данных компании «Ист Вью». - URL:<http://dlib.eastview.com>
10. Лекториум ТВ». - URL: <http://www.lektorium.tv/>
11. Национальная электронная библиотека «НЭБ». - URL:<http://нэб.рф/>
12. КиберЛенинка: научная электронная библиотека. – URL: <http://cyberleninka.ru/>
13. Единое окно доступа к образовательным ресурсам : федеральная ИС свободного доступа. – URL: <http://window.edu.ru>.
14. Справочно-правовая система «Консультант Плюс» - URL <http://www.consultant.ru>

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ «СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»

Учащиеся для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа учащихся является важнейшей формой учебно-познавательного процесса. Цель заданий для самостоятельной работы – закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Учащийся должен изучить список нормативно-правовых актов и экономической литературы, рекомендуемый по учебной дисциплине; уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Обучение студентов осуществляется по традиционной технологии (лекции, практики) с включением инновационных элементов.

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;

- запись ведётся очень быстро, чётко, по возможности короткими выражениями;

– не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить. Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;

– имена, даты, названия, выводы, определения записываются точно;

– следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине «Информационные технологии» проводятся в основном по схеме:

-устный опрос по теории в начале занятия (обсуждение теоретических проблемных вопросов по теме);

-работа в группах по разрешению различных ситуаций по теме занятия;

-решение практических задач индивидуально;

-подведение итогов занятия (или рефлексия);

-индивидуальные задания для подготовки к следующим практическим занятиям.

Цель практического занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

-вербальные (преобладающим методом должно быть объяснение);

-практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

– библиотечные фонды филиала КубГУ в г. Геленджике;

– электронная библиотечная система «Университетская библиотека онлайн»;

– электронная библиотечная система Издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения.

Просмотр книги начинается с титульного листа, следующего после обложки.

На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание.

Прочитав предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно

относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

- пометки, замечания, выделение главного;
- план, тезисы, выписки, цитаты;
- конспект, рабочая записка, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике.

Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка.

Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи- записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7-15 раз;

– конспект может быть, как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:

– прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;

– на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;

– записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;

– конспектирование ведётся не с целью иметь определённый записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;

– после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обращаться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

- конспектируя книгу большого объёма, запись следует вести в общей тетради;
- на каждой странице слева оставляют поля шириной 25-30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;

– каждая страница тетради нумеруется;

– для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;

– при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.

– не следует сокращать имена и названия, кроме очень часто повторяющихся;

– в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Для написания реферата необходимо выбрать тему, согласовать ее с преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

7.1. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Понятие системного и прикладного программного обеспечения. Основные понятия системы программирования.	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9 ,ПК 1.1	Практическая работа, устный опрос
2.	Языки и их представления. Грамматики	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.2, П.К 1.3,ПК 1.5	Практическая работа
3.	Лексический анализ	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.4,ПК 1.3	Практическая работа, устный опрос
4.	Синтаксический анализ	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.3, ПК 1.4	Практическая работа
5.	Семантический анализ	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.4, ПК 1.3,ПК 1.6	Практическая работа
6.	Промежуточное представление программы.	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.5, ПК 1.6,ПК 1.3	Практическая работа
7.	Отладчики. Загрузчики. Библиотеки.	ОК 1, ОК2, ОК3, ОК4, ОК5, ОК6, ОК7, ОК8, ОК9, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4	Практическая работа

7.2. Критерии оценки знаний

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

Тест. Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов (не менее 50%).

Критерии оценки знаний обучающихся в целом по дисциплине:

«отлично» - выставляется обучающемуся, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

«хорошо» - выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется обучающемуся, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется обучающемуся, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

7.3. Оценочные средств для проведения текущей аттестации

Текущий контроль проводится в форме:

- фронтальный опрос
- индивидуальный устный опрос
- тестирование по теоретическому материалу
- практическая работа
- защита реферата
- защита выполненного задания,

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества обучающегося	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературным и источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении	Оценка навыков работы с вычислительной техникой, прикладными программным и средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются

		практических задач.			
Тестирование	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

Примерные тестовые задания:

Вариант 1

1. Системная программа – это
 - a. программа, предназначенная для поддержания работоспособности СОИ или повышения эффективности ее использования.
 - b. программа, предназначенная для решения задачи или класса задач в определенной области применения СОИ
 - c. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОИ, восстановление работы системы после проявления неисправностей в технических средствах.

2. Исходный модуль – это
 - a. программа или функционально заверченный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
 - b. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
 - c. программный модуль, получаемый в результате трансляции исходного модуля.

3. Автокод – это
 - a. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
 - b. язык программирования, понятия и структура которого удобны для восприятия человеком.
 - c. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

4. ... - реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.
 - a. Интерпретация
 - b. Трансляция
 - c. Компиляция

5. Программное обеспечение – это
 - a. совокупность программ СОИ и программных документов, необходимых для их эксплуатации
 - b. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
 - c. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

6. Регистр - это
 - a. Ячейка в оперативной памяти
 - b. Ячейка памяти процессора
 - c. Адресуемая ячейка памяти

7. Какие регистры относятся к сегментным:
 - a. Es
 - b. Dх
 - c. Ip
 - d. Cs
 - e. Al

8. Команда mov –
 - a. Команда помещения в стек
 - b. Команда перемещения данных
 - c. Команда пересылки байта

9. Какая команда относится к логическим командам:
 - a. Sub
 - b. Or

c. Inc

10. Команда ... начинает выполнение с новой ветки в любом случае

- a. Безусловного перехода
- b. Условного перехода
- c. Цикла

11. Объявляется непосредственно внутри ассемблера - ...

- a. Локальная метка
- b. Глобальная метка
- c. Системная метка

12. Прерывание, которое возникает при изменении какого-либо устройства:

- a. Внутренние
- b. Аппаратное
- c. Программное

13. Какой командой вызывается процедура:

- a. Proc
- b. Call
- c. Ret

14. Директива – это

- a. Символическое имя для определения места в памяти, где размещены данные
- b. Блок команд, который может быть вызван из любого места основной программы
- c. Процесс приостановке программы

15. Какой командой вызывается прерывания:

- a. In
- b. Inc
- c. Int

16. Определить ошибки в программе:

1. Var X: integer;
2. Begin
3. X:=100;
4. asm
5. Mov bl, 25
6. Mov ax, X
7. Div ax,bl
8. Mov X,bl
9. end;
10. writeln ('X=',X);
11. end.

- a. Строка 5,7
- b. Строка 6,7
- c. Строка 7,8

17. Определить результат выполнения программы:

```
Begin  
asm  
Mov Ah,02h  
Mov Dl,'*'  
int 21h  
end;  
end.
```

- a. Вывести символ *
- b. Вывести символ с кодом *
- c. Вывести символ * 2 раза

Вариант 2

1. Управляющая программа - это
 - a. программа, предназначенная для поддержания работоспособности СОИ или повышения эффективности ее использования.
 - b. программа, предназначенная для решения задачи или класса задач в определенной области применения СОИ

- c. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОО, восстановление работы системы после проявления неисправностей в технических средствах.

2. Объектный модуль – это

- a. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- b. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
- c. программный модуль, получаемый в результате трансляции исходного модуля.

3. Язык высокого уровня – это

- a. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
- b. язык программирования, понятия и структура которого удобны для восприятия человеком.
- c. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

4. ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

- a. Интерпретация
- b. Трансляция
- c. Компиляция

5. Прикладное программное обеспечение – это

- a. совокупность программ СОО и программных документов, необходимых для их эксплуатации
- b. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- c. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

6. Какие регистры относятся к регистрам данных:

- a. Es
- b. Dх
- c. Ip
- d. Cs
- e. Al

8. Команда add –

- a. Команда сложения
- b. Команда умножения
- c. Команда вычитания

9. Какая команда относится к арифметическим командам:

- a. In
- b. Or
- c. Inc

10. Команда ... повторяет действия несколько раз

- a. Безусловного перехода
- b. Условного перехода
- c. Цикла

11. Ссылка на определенное место в программе - ...

- a. Адресация
- b. Метка
- c. Переход

12. Прерывание, которое возникает при существовании ошибок в программе:

- a. Внутренние
- b. Аппаратное
- c. Программное

13. Какой командой осуществляется начало процедуры:

- a. Proc
- b. Call
- c. Ret

14. Прерывания – это

- a. Символическое имя для определения места в памяти, где размещены данные
- b. Блок команд, который может быть вызван из любого места основной программы
- c. Процесс приостановке программы

15. Какие команды относятся к директивам:

- a. DD
- b. DE
- c. DB
- d. DC
- e. DW

16. Определить ошибки в программе:

- 1. Var a:boolean;
- 2. begin
- 3. asm
- 4. Mov AX,00111011b
- 5. Or Al,00001111b
- 6. Mov a,AL
- 7. end;
- 8. Writeln('A=',A);
- 9. end.

- a. Строка 1,4
- b. Строка 1,5
- c. Строка 4,6

17. Определить результат выполнения программы:

```
Var x:integer;  
Begin  
X:=50;  
asm
```

```

mov ax,x
sub ax,13
sub ax,4
sub ax,20
mov x,ax
end;
writeln('x=',x);
end.

```

- a. 13
- b. 14
- c. 15

7.4. Оценочные средств для проведения промежуточной аттестации

Промежуточная аттестация

Форма аттестации	Знания	Умения	Практический опыт (владеть)	Личные качества обучающегося	Примеры оценочных средств
Экзамен/дифзачет	Контроль знания базовых положений в области информатики	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области профессиональной деятельности	Оценка навыков логического мышления при решении задач в области профессиональной деятельности	Оценка способности грамотно и четко излагать ход решения задач в области профессиональной деятельности и аргументировать результаты	Задачи прилагаются

7.4.1. Примерные вопросы для проведения промежуточной аттестации (экзамен, зачет)

1. Назначение, основные этапы развития операционных систем. Принципы построения ОС.
2. Понятие процесса, потока, ресурса, свойства, классификация. Концепция виртуализации. Концепция прерывания.

3. Состояние процессов. Описание процессов. Взаимодействие процессов. Задача взаимного исключения. Решение задачи взаимного исключения. Задача «производители-потребители» и её решения.
4. Распределение ресурсов, проблема тупиков. Алгоритм банкира. Применение алгоритма банкира.
5. Требования к управлению памятью. Схемы распределения памяти. Страничная организация памяти. Сегментация памяти.
6. Структуризация адресного пространства виртуальной памяти. Задачи управления виртуальной памятью: задача размещения, задача перемещения, задача преобразования адресов, задача замещения.
7. Типы планирования. Алгоритмы планирования. Примеры реализации алгоритмов планирования в современных операционных системах.
8. Организация функций ввода-вывода. Буферизация операций ввода-вывода. Дисковое планирование. Система управление файлами. Организация файлов, доступ к файлам. Управление внешней памятью.
9. Управление памятью в реальном и защищённом режимах. Deskрипторные таблицы и дескрипторы сегментов
10. Понятие процесса, потока, ресурса, свойства, классификация. Концепция виртуализации. Концепция прерывания.
11. Типы файлов. Владельцы файлов. Управление правами доступа в файловой системе. Атрибуты файлов. Управление свойствами файлов. Работа с файлами. Структура файловой системы.
12. Сигналы. Обработка сигналов. Неименованные каналы. Именованные каналы. Дополнительные средства взаимодействия между процессами. Сообщества, семафоры, разделяемая память.
13. Архитектура и основные подсистемы ОС Windows. Системный реестр ОС Windows, его назначение и использование.
14. Основные элементы программ с оконным пользовательским интерфейсом. Понятие оконного сообщения. Источники сообщений. Очереди сообщений. Обработка сообщений мыши, клавиатуры.
15. Понятие ресурсов программ. Виды ресурсов.

Другие оценочные средства по дисциплине не предусмотрены.

8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложение 1. Краткий конспект лекционных занятий

Лекция №1.

Тема. «Языки и эволюция технологий программирования.»

Если рассматривать историю языков программирования с точки зрения историка это будет огромное количество дат, плотно расположенных начиная с 1950-х годов и по настоящее время. Нас интересует история языков как процесс, растянутый во времени и зависящий от развития компьютерной техники и задач, стоящих перед программистами конкретного периода.

Итак, на заре развития языков программирования 1954-1958 гг. компьютерная техника представляла из себя огромные машины, занимающие целые залы в вычислительных центрах, перед ними ставились в основном задачи на численные вычисления. Заказчик либо ставил задачу перед программистом ВЦ и получал только ленту с результатом вычислений, либо сам вынужден был становиться разработчиком ПО, что бы получить нужные результаты (то есть он одновременно был экспертом в предметной области). Для решения этих задач было достаточно так называемой хаотической концепции программирования (Fortran, Algol-58). Для программ того времени была характерна неконтролируемая последовательность передач управления (оператор goto), отсутствие подпрограмм и требований к обязательному объявлению данных. Все эти свойства характерны для языков первого поколения.

Языки второго поколения появились в 1959-1961гг. (Fortran2, Algol-60, Lisp, Cobol). В них уже реализуются описание данных, разделение программы на подпрограммы. В то же время, нет передачи данных через параметры функций, реализованы только глобальные переменные. Эти изменения потребовались потому, что задачи, стоящие перед программистами, усложнились. Стало сложно решить их при помощи единого блока, к тому же к этому времени накопились написанные вычислительные программы, которые хотелось использовать, а не писать программу с нуля. Из-за использования перекрестных ссылок и глобальных переменных подобные программы часто вызывали путаницу и были сложно структурированными. Используя языки первого и второго поколений, программисты могли писать программы до нескольких тысяч строк.

Языки третьего поколения возникли для преодоления этого барьера и позволили писать программы до 50 000 строк, причём программу уже писал не один программист, как раньше, а группа программистов. Эти языки возникли в 1962-1970 гг. (Algol-68, PL/1, Pascal, Simula, C). В них используются: структурное программирование (отдельные функции со своим набором данных), модульное программирование (отдельно-компилируемые файлы, возможно разных авторов, впоследствии собираемые в один программный продукт), передача данных в подпрограмму при помощи параметров, к тому же добавилась возможность описать новые типы данных. Последнее связано с тем, что вычислительные задачи уступают место задачам обработки данных. Языки второго и третьего поколений называют процедурно-ориентированными языками.

Однако и они оказались неэффективными, когда программы стали ещё больше. К тому же на данном этапе развития вычислительной техники жизненный цикл программ изменился.

На предыдущих этапах развития вычислительной техники заказчику, в основном, требовались результаты вычислений, возможно неоднократные с изменяемыми начальными значениями. Программный код мог вообще не интересоваться заказчика, либо он создавался и эксплуатировался всю свою жизнь на одной ЭВМ. На описываемом этапе на первое место выходит последний этап – поддержка функционирования программы на компьютере заказчика, причём работать с программой приходится не разработчику, а совсем другому человеку. К тому же заказчику требуется возможно модернизировать ПО, без больших дополнительных затрат и достаточно быстро. Всё это привело к тому, что в 1970-1980 гг. появились языки четвертого поколения (Object Pascal, C++, ADA). Они получили название объектно-ориентированных языков. Основным элементом в объектно-ориентированных языках служит модуль, состоящий из логически связанных классов и объектов, а не подпрограмма. Класс содержит описание данных и методы обработки этих данных. Этим достигается легкость модернизации кода и гарантия всегда корректной обработки данных. К тому же принцип наследования свойств и данных уже написанных классов даёт возможность программисту писать Windows-ориентированные приложения не тратя время на прописывание свойств окон.

Объектно-ориентированное программирование позволяет программисту оперировать с гораздо большими по объёму программами и пользоваться гораздо большим числом уже написанных библиотек для решения стандартных задач программирования. Так же снижается стоимость разработки и обслуживания надёжной программы.

На современном этапе задачи перед автоматизированной системой ставятся самые разнообразные. ПК используются во всех науках и отраслях хозяйства. Таким образом перед программистом ставится достаточно широкий круг задач. Именно поэтому перед решением задачи программисту требуется квалифицированная помощь эксперта по предметной области, так как программист всё чаще не является специалистом узкого профиля, как этого требуют задачи. Современную ситуацию можно рассматривать как частичный возврат к 50-м годам 20 века, когда специалист сам разрабатывал для себя задачу для вычисления на ЭВМ. Некоторые современные языки (логическое программирование Dragon) ориентированы не на программиста, а на специалиста. А так же многие распространенные языки приобретают черты программирования для «чайников», то есть имеют более простой и понятный синтаксис, не имеют функций для работы с битами и указателями, не позволяют разработчику самому выделять и удалять динамическую память, то есть, рассчитаны на программиста не высокой квалификации, или предполагают, что программирование это не профессия, а навык необходимый каждому. Это совершенно не означает, что программист как профессия скоро прекратит своё существование. Это совершенно не так. В мире достаточно много (что-то современной ситуации, у меня нет)

язык C

Среди современных языков программирования язык С является одним из наиболее распространенных. С одной стороны, язык С универсален, как с точки зрения решаемых задач, так и с точки зрения компьютеров и ОС, на которых может работать программа (существуют огромное количество компиляторов языка С для любых существующих ОС, и программа написанная на языке С обладает свойством переносимости на различные ПК). С другой стороны, язык С особенно эффективен для задач системного программирования (разработка трансляторов, ОС, инструментальных средств и оконных интерфейсов). Поскольку язык С хорошо зарекомендовал себя эффективностью, логической стройностью, лаконичностью, скоростью работы программ (во многих случаях её можно сравнить со скоростью выполнения программ, написанных на языке ассемблера), то естественно, что язык С со времени появления, до современного уровня вынужден приспосабливаться к изменившимся задачам, стоящим перед программистами. Поэтому сейчас мы говорим о современных модификациях языка С – С++ и С#. Перед авторами учебного пособия не стоит задача рассмотрения С#, поэтому в дальнейшем словосочетания язык С, будет рассматриваться как С++.

Язык С обладает рядом особенностей, присущих только ему. Это связано с тем, что изначально перед разработчиками языка стояла задача обеспечить системных программистов языком высокого уровня позволяющим заменить собой ассемблер. Поэтому язык С и по сей день не утратил необычайной лёгкости доступа к аппаратным средствам компьютера. С одной стороны, как ЯВУ С поддерживает полный набор конструкций структурного и объектно-ориентированного программирования, модульность и отдельную компиляцию. С другой стороны язык С имеет ряд низкоуровневых свойств. Так например, в языке С реализованы побитовые операции, существуют типы данных соответствующие типам данных в ассемблере (байты, машинные слова). Работа с указателями позволяет оперировать адресами в памяти компьютера с той же лёгкостью, что и в ассемблере, но поскольку это требует высокой квалификации программиста, то в современных системах программирования (например: Microsoft Visual Studio 2008) встроены защиты от неверного обращения к адресам памяти. Таким образом современные среды обработки не поддерживают заложенный изначально в язык С принцип доверия к программисту.

Язык С был разработан в США в 70-х годах 20 века. Тогда же он был использован для написания ОС UNIX (модификация её LINUX всем известна). Из-за хорошей переносимости язык С стал очень распространён, это распространённость привела к возникновению огромного количества «диалектов» языка С, что создавало некоторую путаницу в работе программ и уменьшило переносимость. Поэтому в 1989 г. в Американском Национально Институте Стандартов (ANSI) была принята окончательная редакция стандарта языка С. Впоследствии к этому стандарту была добавлена объектно-ориентированная версия языка С – С++ Стандарт ISO/ANSI определен в документе ISO/IEC 14882, опубликованном Американским национальным институтом стандартизации (ANSI).

Стандарт ISO/ANSI С++ описывает устойчивую версию С++, которая существует с 1998г. и поддерживается компиляторами большинства аппаратных компьютерных платформ и операционных систем. Программы, написанные на ISO/ANSI С++, относительно легко могут быть перенесены с одной платформы на

другую, хотя используемые ими библиотечные функции, в частности, связанные с построением графического интерфейса, являются главным фактором, определяющим, насколько легко или трудно такой перенос осуществить. Стандарт C++ ISO/ANSI — это главный инструмент, который выбирают профессиональные разработчики программ, поскольку он широко поддерживается и потому, что на сегодняшний день он является одним из наиболее мощных доступных языков программирования.

C++/CLI — версия C++, расширяющая стандарт C++ ISO/ANSI в целях лучшей поддержки общей инфраструктуры языка (CLI), определенной в стандарте ECMA355. Первый набросок этого стандарта появился в 2003 г. и был разработан на основе технических спецификаций, представленных Microsoft для поддержки программ C++ в среде .NET Framework. То есть, как CLI вообще, так и C++/CLI в частности, родились в Microsoft и предназначены для поддержки .NET Framework. Конечно, стандартизация CLI и C++/CLI значительно повысила вероятность появления реализаций в средах, отличных от Windows. Важно оценить то, что хотя C++/CLI — это расширение ISO/ANSI C++, существуют такие средства ISO/ANSI C++, которые не следует использовать в программах, предназначенных полностью для выполнения под управлением CLR.

Кстати C# случайно не он?

. В начале 21 века появляется ещё одна версия языка -- C#, эта версия ориентирована на создание сетевых Windows приложений. Но, как и любая версия языка содержит в себе определённые в стандарте языка правила работы с языком C, и при малейшем усилии позволяет писать консольные приложения на стандарте языка C.

ЛИСТ
изменений рабочей учебной программы по дисциплине
МДК.01.01 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Дополнения и изменения, вносимые в рабочую программу дисциплины

Основания внесения дополнений и изменений	Раздел РПД, в который вносятся изменения	Содержание вносимых дополнений, изменений
Предложение работодателя	нет	нет
Предложение составителя программы	нет	нет
Приобретение, издание литературы, обновление перечня и содержания ЭБС, баз данных	Разделы №2.4.5 и №5 Перечня основной и дополнительной учебной литературы	Обновлен список рекомендуемой литературы

Составитель: преподаватель



Т.П. Кривошеенко

Утвержден на заседании предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах протокол № 10 от «27» мая 2020 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности Программирование в компьютерных системах



Л.А. Благова

подпись

«27» мая 2020 г.

Зам. директора по УР филиала



Т. А. Резуненко

«27» мая 2020 г.

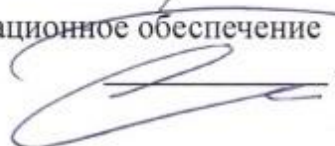
Заведующая сектором библиотеки



Л. Г. Соколова

«27» мая 2020 г.

Инженер-электроник (программно-информационное обеспечение образовательной программы)



А. В. Сметанин

«27» мая 2020 г.

Рецензия

на рабочую программу учебной дисциплине
МДК 01.01 «Системное программирование»
по специальности 09.02.03 Программирование в компьютерных системах

Рабочая программа учебной дисциплины МДК 01.01 «Системное программирование» разработана на основе требований ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины МДК 01.01. «Системное программирование», в соответствии с Рекомендациями по организации получения среднего общего образования в пределах освоения образовательных программ среднего профессионального образования на базе основного общего образования с учетом требований федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки Российской Федерации от 28.07.2014 № 804 зарегистрирован в Минюсте России 21.08.2014 (рег. № 33733).

Рабочая программа детализирует и раскрывает содержание стандарта, определяет общую стратегию обучения, воспитания и развития, обучающихся средствами учебного предмета в соответствии с целями изучения системного программирования, которые определены стандартом.

При освоении программы у обучающихся формируется знания, умения и навыки по системному программированию, необходимые для изучения других общеобразовательных предметов.

Содержание рабочей программы учебной дисциплины полностью соответствует требованиям Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах. Изучение данной дисциплины способствует эффективной и качественной подготовке молодых специалистов.

Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные в последнее время (не позднее 5 лет). Перечисленные Интернет-ресурсы актуальны и достоверны.

Рабочая программа дисциплины «Системное программирование» может быть рекомендована для реализации в среднем профессиональном образовании при подготовке специалистов филиала «Кубанского государственного университета» в г. Геленджике по специальности 09.02.03 Программирование в компьютерной системе.

Рецензент:

Директор ООО «Современные
информационные технологии»

А.В.Сметанин

Рецензия

на рабочую программу по учебной дисциплине
МДК 01.01 «Системное программирование»
по специальности 09.02.03 Программирование в компьютерных системах

Рабочая программа учебной дисциплины МДК.01.01 «Системное программирование» разработана на основе требований ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины МДК 01.01 «Системное программирование», по специальности 09.02.03 Программирование в компьютерных системах, утвержденного приказом Минобрнауки Российской Федерации от 28.07.2014 № 804 зарегистрирован в Минюсте России 21.08.2014 (рег. № 33733).

Программа предусматривает освоение профессиональных компетенций и видов деятельности, согласно ФГОС № 804 от «28» июля 2014 г.

В программу включены необходимые для понимания общих закономерностей и принципиальных положений, определяющих способность баз данных сохранять свою работоспособность в различных условиях их функционирования. Программа имеет достаточную степень полноты и законченности изучения предмета в условиях СПО.

Дисциплина «Системное программирование» предусматривает ознакомление обучающихся с языками программирования.

В программе нашли отражение основные теоретические и практические направления.

Структура программы соответствует современным требованиям. Содержание каждого её элемента разработано с достаточной степенью полноты и законченности.

Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные в последнее время (не позднее 5 лет). Перечисленные Интернет-ресурсы актуальны и достоверны.

В целом рецензируемая программа учебной дисциплины заслуживает высокой оценки, она хорошо продумана и ориентирована на подготовку обучающихся к использованию полученных навыков в своей профессиональной деятельности.

Следовательно, рабочая программа содержит все необходимые элементы рекомендуемой структуры, обладает достаточной полнотой и законченностью, является ценным практическим документом при преподавании дисциплины МДК 01.01 «Системное программирование».

Рецензент:

Программист ГБУЗ

"Геленджикский психоневрологический диспансер"
министерства здравоохранения Краснодарского края

Е.В.Мельников