

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: директор филиала

Дата подписания: 19.06.2025 15:33:27

Уникальный программный ключ:

23E60C8A93B1B5E5740B1

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ**

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЧЕБОКСАРСКИЙ ИНСТИТУТ ФАКУЛЬТЕТА МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

**Кафедра Информационных технологий
и систем управления**



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программирование на языке низкого уровня»

(наименование дисциплины)

Направление подготовки	09.03.01 «Информатика и вычислительная техника» (код и наименование направления подготовки)
Направленность (профиль) подготовки	«Программное обеспечение вычислительной техники и автоматизированных систем» (наименование профиля подготовки)
Квалификация выпускника	бакалавр
Форма обучения	очная, заочная
Год начала обучения	2025

Чебоксары, 2025

Рабочая программа дисциплины разработана в соответствии с:

- Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 929 от 19 сентября 2017 г. зарегистрированный в Минюсте 10 октября 2017 года, рег. номер 48489 (далее – ФГОС ВО);
- Учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

Рабочая программа дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (п.8 Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины)

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук,
доцент кафедры информационных технологий и систем управления
(указать ФИО, ученую степень, ученое звание или должность)

Программа одобрена на заседании кафедры Информационных технологий и систем управления (протокол № 8 от 12.04.2025г).

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы (Цели освоения дисциплины)

1.1. Целями освоения дисциплины «Программирование на языке низкого уровня» являются:

- ознакомление обучающихся с архитектурой виртуальных машин;
- с организацией управления и взаимодействия процессов;
- организацией управления в многопользовательских и многозадачных операционных системах;
- ознакомление обучающихся с администрированием современных операционных систем;
- ознакомление обучающихся с основами взаимодействия процессов по сети.

Задачами освоения дисциплины Программирование на языке низкого уровня являются:

- владение навыками установки современных операционных систем ;
- научиться получению основных данных о текущем состоянии операционной системы с помощью консольных команд;
- овладение обучающими базовыми навыками разработки сетевых приложений на сокетах;
- научиться получать информацию о состоянии операционной системы с выводом отчета на сценарных языках программирования встроенных в систему;
- научиться разрабатывать сценарии управления на встроенных в систему сценарных языках;
- научиться создавать группы пользователей и пользователей с различными ролями;
- приобретение обучающимися первичных навыков удаленного администрирования операционных систем.

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовалось программное обеспечение, для корректной работы которого необходимы Программирование на языке низкого уровня.

1.2. Области профессиональной деятельности и (или) сферы профессиональной деятельности, в которых выпускники, освоившие программу, могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом).

1.3. К основным задачам изучения дисциплины относится подготовка обучающихся к выполнению трудовых функций в соответствии с профессиональными стандартами:

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
06.001 Программист Профессиональный стандарт «Программист», утвержденный приказом Министерством труда и социальной защиты Российской Федерации от 18.11.2013 № 679н (зарегистрирован в Министерстве труда и социальной защиты Российской Федерации от 18 декабря 2013 г. №30635)	D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
			6	Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	
			6	Проектирование программного обеспечения	D/03.6	
06.028 Системный программист Профессиональный стандарт «Системный программист», утв. Приказом Министерством труда и социальной защиты РФ от 29 сентября 2020 года N 678н	A	Разработка компонентов системных программных продуктов	6	Разработка драйверов устройств	A/01.6	6
			6	Разработка компиляторов, загрузчиков, сборщиков	A/02.6	6
				Разработка системных утилит	A/03.6	6
				Создание инструментальных средств программирования	A/04.6	6
06.015 Специалист по информационным системам Профессиональный стандарт «Специалист по информационным системам», утв. приказом Министерством труда и социальной защиты РФ 18 ноября 2014 г. №896н	C	Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	6	Разработка модели бизнес-процессов заказчика	C/08.6	6
				Выявление требований к ИС	C/11.6	6
				Анализ требований	C/12.6	6
				Разработка архитектуры ИС	C/14.6	6
				Проектирование и дизайн ИС	C/16.6	6
				Разработка баз данных ИС	C/17.6	6

Код и наименование профессионального стандарта	Обобщенные трудовые функции			Трудовые функции		
	код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
				Организационное и технологическое обеспечение кодирования на языках программирования	C/18.6	6
				Организационное и технологическое обеспечение модульного тестирования ИС (верификации)	C/19.6	6
				Создание пользовательской документации к ИС	C/22.6	6
				Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	C/40.6	6

1.4. Компетенции обучающегося, формируемые в результате освоения дисциплины

Наименование категории (группы) компетенций	Код и наименование компетенций	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения
Разработка, отладка, проверка работоспособности, модификация программного обеспечения	ПК-2 Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению	<i>на уровне знаний:</i> знать основные результаты теории формальных языков; <i>на уровне умений:</i> уметь создавать распознаватели, интерпретаторы и трансляторы информационных потоков; уметь находить и устранять проблем взаимодействия вычислительных процессов;

			<i>на уровне навыков:</i> навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков
		ПК-2.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие	<i>на уровне знаний:</i> знать формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; <i>на уровне умений:</i> уметь моделировать работу алгоритмов взаимодействия процессов и ресурсов в работе вычислительной системы. <i>на уровне навыков:</i> навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании
		ПК-2.3 Проектирует программное обеспечение	<i>на уровне знаний:</i> теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования <i>на уровне умений:</i> уметь разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода

			для наиболее часто используемых классов формальных грамматик на уровне навыков: навыками практического анализа проблем взаимодействия вычислительных процессов.
--	--	--	---

2. Место дисциплины в структуре ОПОП

Дисциплина Б1.Д(М).В.ДВ.6.2 «Программирование на языке низкого уровня» реализуется в рамках вариативной части Блока 1 «Элективные дисциплины (модули)» программы бакалавриата.

Дисциплина «Программирование на языке низкого уровня» преподается обучающимся по очной форме обучения – в 5-м семестре, по заочной форме – в 7-м семестре.

Дисциплина «Программирование на языке низкого уровня» является промежуточным этапом формирования компетенций ПК-2 процессе освоения ОПОП.

Дисциплина «Программирование на языке низкого уровня» основывается на знаниях, умениях и навыках, приобретенных при изучении дисциплин Учебная практика: ознакомительная практика, Информационные сети и коммуникации, Проектная деятельность, Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, Базы данных, Интеллектуальные системы, Основы систем искусственного интеллекта и является предшествующей для изучения дисциплин Системное программное обеспечение, Теория вычислительных процессов и языков программирования, Эргономика, Технический дизайн, Производственная практика: технологическая (проектно-технологическая) практика, Производственная практика: научно-исследовательская работа, Цифровая обработка сигналов, Архитектура вычислительных систем, Производственная практика: преддипломная практика, Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

Формой промежуточной аттестации знаний обучаемых по очной форме обучения является экзамен в 5-м семестре, по заочной форме – в 7-м семестре.

3. Объем дисциплины

очная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 5 в часах
Общая трудоёмкость дисциплины	4 з.е. - 144 ак.час	144 ак.час
Контактная работа - Аудиторные занятия	49	49
Лекции	16	16
Лабораторные занятия	32	32
Семинары, практические занятия	-	-
Консультация	1	1
Самостоятельная работа	59	59
Курсовая работа (курсовой проект)	-	-
Вид промежуточной аттестации	Экзамен – 36 часов	Экзамен – 36 часов

заочная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 7 в часах
Общая трудоёмкость дисциплины	4 з.е. - 144 ак.час	144 ак.час
Контактная работа - Аудиторные занятия	13	13
Лекции	6	6
Лабораторные занятия	6	6
Семинары, практические занятия	-	-
Консультация	1	1
Самостоятельная работа	122	122
Курсовая работа (курсовой проект)	-	-
Вид промежуточной аттестации	Экзамен – 9 часов	Экзамен – 9 часов

4. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

4.1 Учебно-тематический план

Очная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3

Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	4	8	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 4. Виртуальная машина Си. Основные понятия языка.	2	4	-	12	ПК-2.1 ПК-2.2 ПК-2.3
Тема 5. Работа с памятью. Указатели. Массивы. Функции.	2	4	-	11	ПК-2.1 ПК-2.2 ПК-2.3
Консультации	1			-	ПК-2.1 ПК-2.2 ПК-2.3
Контроль (экзамен)	36				ПК-2.1 ПК-2.2 ПК-2.3
ИТОГО	49			59	

Заочная форма обучения

Тема (раздел)	Количество часов				Код индикатора достижений компетенции
	контактная работа			самостоятельная работа	
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	-	2	-	26	ПК-2.1 ПК-2.2 ПК-2.3
Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	-	2	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	2	-	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 4. Виртуальная машина Си. Основные понятия языка.	2	-	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Тема 5. Работа с памятью. Указатели. Массивы. Функции.	2	2	-	24	ПК-2.1 ПК-2.2 ПК-2.3
Консультации	1			-	ПК-2.1 ПК-2.2

			ПК-2.3
Контроль (экзамен)	9		ПК-2.1 ПК-2.2 ПК-2.3
ИТОГО	13	122	

4.2. Содержание дисциплины

Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T

Общее представление о виртуальной машине Ассемблер IA-32 и ее роли в системном программировании.

Сравнение синтаксиса Intel и GNU Assembler для Linux.

Применение синтаксиса AT&T и его особенности.

Основные команды и директивы, используемые в ассемблере IA-32.

Примеры написания простых программ на различных синтаксисах.

Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE

Обзор расширенной архитектуры x86-64 и ее отличия от x86.

Архитектура арифметического сопроцессора и его функции.

Применение технологий MMX и SSE в x86-64 для обработки мультимедийных данных.

Примеры использования расширений в реальных приложениях и их влияние на производительность.

Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR

Общее представление о виртуальной машине FORTH, ее архитектуре и принципах работы.

Сравнение виртуальной машины FORTH с другими виртуальными машинами.

Виртуальная машина FACTOR: основные концепции и особенности.

Использование виртуальных машин для реализации языков программирования и их применение в различных областях.

Тема 4. Виртуальная машина Си. Основные понятия языка

Обзор виртуальной машины Си и ее роли в интерпретации программ на языке Си.

Основные понятия, связанные с языком Си, такие как переменные, типы данных и функции.

Структура программы на языке Си и принципы работы виртуальной машины.

Примеры программ на Си и их выполнение в виртуальной машине.

Тема 5. Работа с памятью. Указатели. Массивы. Функции

Общее представление о работе с памятью в языках программирования.

Понятие указателей и их использование для доступа к памяти.

Массивы: объявление, инициализация и работа с ними.

Функции: передача аргументов по значению и по указателю, а также возврат значений из функций.

Примеры работы с указателями, массивами и функциями на языке Си.

5. Учебно-методическое обеспечение самостоятельной работы студентов

Самостоятельная работа проводится с целью: систематизации и закрепления полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации; формирования профессиональных компетенций; развитию исследовательских умений студентов.

Формы и виды самостоятельной работы студентов: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со словарем, справочником; поиск необходимой информации в сети Интернет; конспектирование источников; реферирование источников; составление аннотаций к прочитанным литературным источникам; составление рецензий и отзывов на прочитанный материал; составление обзора публикаций по теме; составление и разработка терминологического словаря; составление хронологической таблицы; составление библиографии (библиографической картотеки); подготовка к различным формам текущей и промежуточной аттестации (к тестированию, контрольной работе, экзамену); выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, задачи, тесты; выполнение творческих заданий).

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, компьютерные классы с возможностью работы в Интернет; аудитории (классы) для консультационной деятельности.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации.

Контроль самостоятельной работы студентов предусматривает: соотнесение содержания контроля с целями обучения; объективность контроля;

валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить); дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы: просмотр и проверка выполнения самостоятельной работы преподавателем; организация самопроверки, взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; проведение письменного опроса; проведение устного опроса; организация и проведение индивидуального собеседования; организация и проведение собеседования с группой.

Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	1. Виртуальная машина Ассемблер IA-32 и ее роль в системном программировании. 2. Синтаксис Intel и синтаксис AT&T. Отличия и преимущества. 3. Основные команды и директивы GNU Assembler для Linux. 4. Примеры написания программы на ассемблере с использованием синтаксиса Intel. 5. Основные этапы сборки программы с использованием GNU Assembler. 6. Интерпретация ассемблерного кода различными синтаксисами. - Составить схему, показывающую разницу между синтаксисом Intel и AT&T, с примерами.	Анализ документации по синтаксису ассемблера, написание простых программ на ассемблере с использованием разных синтаксисов, работа с примерами из практики.
Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	1. Основные отличия архитектуры x86-64 от x86. 2. Функции арифметического сопроцессора и его влияние на производительность программ. 3. Технологии MMX и SSE, их архитектурные особенности и применения. 4. Примеры использования инструкций MMX и SSE в ассемблерных программах. 5. Определить, как x86-64 поддерживает обратную совместимость с x86. 6. Процесс обработки мультимедийных данных с использованием MMX/SSE. - Создать таблицу, сравнивающую производительность инструкций MMX и SSE.	Изучение технической документации по архитектуре x86-64, написание примеров кода, использующего MMX и SSE, анализ производительности различных инструкций.
Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	1. Виртуальная машина FORTH и ее основные принципы работы. 2. Архитектурные особенности виртуальной машины FACTOR. 3. Виртуальные машины FORTH и FACTOR, их преимущества и недостатки. 4. Примеры использования виртуальных машин для реализации языков программирования. 5. Определить, как виртуальные машины обеспечивают переносимость программного обеспечения.	Анализ примеров программ, написанных для виртуальных машин, исследование документов по архитектуре FORTH и FACTOR, разработка простых программ на языке FORTH.

	6. Способы виртуальной машины FORTH обработки стеков и команд. - Составить схему работы виртуальной машины FORTH с указанием ключевых компонентов.	
Тема 4. Виртуальная машина Си. Основные понятия языка.	1. Виртуальная машина Си и ее роль в интерпретации программ. 2. Основные понятия языка Си, такие как переменные, типы данных и функции. 3. Структура программы на языке Си и принципы работы виртуальной машины. 4. Примеры программ на Си и их выполнение в виртуальной машине. 5. Управление памятью и ресурсами в виртуальной машине Си. 6. Взаимодействие компиляторов и интерпретаторов с виртуальной машиной Си. - Создать таблицу, показывающую различия между компилятором и интерпретатором для языка Си.	Изучение примеров программ на языке Си, разработка простых программ с использованием функций и указателей, анализ работы виртуальной машины Си.
Тема 5. Работа с памятью. Указатели. Массивы. Функции.	1. Работа с памятью в языках программирования. 2. Указатели и их использование для доступа к памяти. 3. Понятие массивов: их объявление, инициализация и работа с ними. 4. Функции и их работа с указателями и массивами в языке Си. 5. Примеры использования указателей для динамического выделения памяти. 6. Передача аргументов в функции по значению и по указателю. - Создать презентацию, иллюстрирующую работу с указателями и массивами через примеры кода.	Работа с учебными пособиями по указателям и массивам, написание программ, использующих массивы и функции, анализ работы с памятью и отладка программ.

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер
«Хорошо»	Обучающийся в целом раскрывает тему самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

6. Оценочные материалы (фонд оценочных средств) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

6.1. Паспорт фонда оценочных средств

№	Контролируемые разделы (темы) дисциплины	Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
1.	Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, экзамен
2.	Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, отчет
3.	Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, отчет
4.	Тема 4. Виртуальная машина Си. Основные понятия языка.	ПК-2. Способен разрабатывать требования и	ПК-2.1 Выполняет анализ требований к программному обеспечению	Опрос, тест, доклад, отчет

		проектировать программное обеспечение	ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	
5.	Тема 5. Работа с памятью. Указатели. Массивы. Функции.	ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	ПК-2.1 Выполняет анализ требований к программному обеспечению ПК-2.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие ПК-2.3 Проектирует программное обеспечение	Опрос, тест, доклад, отчет

Этапы формирования компетенций в процессе освоения ОПОП прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенции, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации.

Дисциплина «Программирование на языке низкого уровня» является промежуточным этапом комплекса дисциплин, в ходе изучения которых у студентов формируются компетенции ПК-2.

Формирование компетенции ПК-2 начинается с изучения дисциплины Учебная практика (ознакомительная практика), Информационные сети и коммуникации, «Проектная деятельность», Учебная практика: научно-исследовательская работа (получение первичных навыков научно-исследовательской работы), Учебная практика: технологическая (проектно-технологическая) практика, «Базы данных», «Интеллектуальные системы», «Основы систем искусственного интеллекта».

Завершается работа по формированию у студентов указанных компетенций в ходе предмета «Архитектура вычислительных систем», Производственная практика: преддипломная практика.

Итоговая оценка сформированности компетенций ПК-2 определяется в период Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

В процессе изучения дисциплины, компетенции также формируются поэтапно.

Основными этапами формирования ПК-2 при изучении дисциплины Б1.Д(М).В.ДВ.6.2 «Программирование на языке низкого уровня» является последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение студентами необходимыми дескрипторами (составляющими) компетенций. Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

6.2. Контрольные задания и материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

6.2.1. Контрольные вопросы по темам (разделам) для опроса на занятиях

Тема (раздел)	Вопросы
Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	ПК-2 1. Понятие языка программирования низкого уровня. 2. Что такое регистры процессора. 3. Структура программы на ассемблере. 4. Какие бывают флаги состояния. 5. Что такое команды и что такое данные. 6. Для чего используются директивы размещения данных. 7. Что такое косвенный операнд. 8. Порядок следования операндов в инструкции Intel. 9. Порядок следования операндов в инструкции AT&T. 10. Основы работы со строками и символами. 11. Структура памяти программы на Си. 12. Ассемблерные вставки.
Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	ПК-2 1. Расширение архитектуры x86-64. 2. Набор инструкций x86-64. 3. Архитектура арифметического сопроцессора. 4. Представление чисел с плавающей точкой. 5. Набор инструкций арифметического сопроцессора. 6. Расширение MMX (MultiMedia eXtension). 7. Набор инструкций MMX. 8. Расширение SSE (Streaming SIMD Extension).

	9. Расширение SSE2, SSE3, SSSE3, SSE4, ... 10. Использование автоматической векторизации компиляторами. 11. Указатели. Операции с указателями. 12. Связь между указателями и массивами. 13. Основные функции стандартной библиотеки Си.
Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	ПК-2 1. Машины со стековой архитектурой. 2. Шитый код. 3. Байт-код. 4. Обратная польская запись. 5. Операции со стеком данных. 6. Словарь. 7. Условный оператор. 8. Функционирование форт-системы. 9. Контекстные словари. 10. Виртуальная машина FACTOR как дальнейшее развитие FORTH. 11. Стандартная библиотека FACTOR. 12. Применение FACTOR. 13. Пользовательские типы данных с помощью typedef. 14. Перечисления. 15. Объединения.
Тема 4. Виртуальная машина Си. Основные понятия языка.	ПК-2 1. Структура программы (модуля) на языке Си. 2. Стандарты языка Си. 3. Базовые типы данных на языке Си. 4. Потоки стандартного ввода, вывода и вывода ошибки на языке Си. 5. Форматированный ввод/вывод. 6. Представление целых чисел. 7. Представление чисел с плавающей точкой. 8. Математические операции. 9. Логические и побитовые операции. 10. Управление вычислениями. Условный оператор. 11. Управление вычислениями. Циклы. 12. Функции в Си. Основные понятия. 13. Области видимости переменных. 14. Передача аргументов по значению и по ссылке. 15. Передача массивов функциям. 16. Область видимости Static. 17. Функции генераторы. 18. Указатели на функцию. 19. Функции высших порядков. 20. Элементы функционального программирования на Си. 21. Директивы препроцессора. 22. Модульное программирование на языке Си.

Тема 5. Работа с памятью. Указатели. Массивы. Функции.	ПК-2 1. Как представлены в памяти целые числа. 2. Арифметические операции над числами с фиксированной точкой. 3. Операция масштабирования. 4. Циклы со счетчиком. 5. Циклы с условием. 6. Выход из цикла (leave) и ветвление (branch). 7. Числа двойной длины. 8. Переменные, константы и массивы. 9. Массивы в Си. 10. Динамические массивы. 11. Многомерные динамические массивы. 12. Работа с файлами. 13. Сортировка и поиск в одномерных массивах. 14. Управление памятью. 15. Работа со строками. 16. Работа с графикой OpenCV.
--	---

Шкала оценивания ответов на вопросы

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает ответ на каждый теоретический вопрос, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает теоретические вопросы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает теоретические вопросы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не знает ответов на поставленные теоретические вопросы.

6.2.2. Темы для докладов

Тема (раздел)	Вопросы
Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.	ПК-2 1. Сравнительный анализ синтаксиса Intel и AT&T: преимущества и недостатки. 2. Принципы работы GNU Assembler и его использование в Linux. 3. Основные команды IA-32: как они работают и когда их применять. 4. Архитектура виртуальной машины IA-32: структура и функциональные блоки. 5. Создание и отладка программ на ассемблере: практические советы и инструменты.

Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE	ПК-2 1. Переход от x86 к x86-64: что изменилось и как это повлияло на производительность. 2. Архитектура арифметического сопроцессора: как она улучшает вычислительные возможности. 3. Расширение MMX: влияние на мультимедийные приложения. 4. SSE и его расширения: как SIMD технологии изменили обработку данных. 5. Автоматическая векторизация: что это такое и как она используется в современных компиляторах.
Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR	ПК-2 1. Основные концепции виртуальной машины FORTH: стеки, команды и взаимодействие. 2. Виртуальная машина FACTOR: особенности и преимущества по сравнению с FORTH. 3. Применение виртуальных машин в системах реального времени: примеры FORTH и FACTOR. 4. Сравнительный анализ языков программирования FORTH и FACTOR. 5. Виртуальные машины как инструмент для обучения программированию: методические аспекты.
Тема 4. Виртуальная машина Си. Основные понятия языка.	ПК-2 1. Архитектура виртуальной машины Си: как она интерпретирует код. 2. Основные понятия языка Си: переменные, типы данных и управление памятью. 3. Виртуальная машина Си и ее взаимодействие с аппаратным обеспечением. 4. Применение языка Си в системном программировании: примеры и практические задачи. 5. Сравнение виртуальной машины Си с другими виртуальными машинами: преимущества и недостатки.
Тема 5. Работа с памятью. Указатели. Массивы. Функции.	ПК-2 1. Управление памятью в языке Си: стратегии и подходы. 2. Указатели: как они работают и их роль в эффективном программировании. 3. Массивы и динамические массивы в Си: разница, преимущества и недостатки. 4. Функции в языке Си: передача аргументов и возвращаемые значения. 5. Применение указателей в современных алгоритмах: примеры и разбор.

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему доклада, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему доклада, однако

	ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему доклада и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой

6.2.3. Оценочные средства остаточных знаний (тест)

ПК-2.

1. Чем обеспечивается надежность программ, написанных на языке Си?

- 1) гибкостью языка Си
- 2) переносимостью языка Си
- 3) мобильностью языка Си
- 4) сильной типизацией языка Си

2. Как создать строковую константу?

- 1) используя кавычки
- 2) используя вопросительный знак
- 3) используя апостроф
- 4) используя восклицательный знак

3. Как обратиться к полям структуры student?

- 1) student.id
- 2) student..name
- 3) student:age
- 4) student(sex)

4. Каким типом будете пользоваться для хранения и обработки данных о количестве жителей Чебоксар?

- 1) char
- 2) int
- 3) short
- 4) long

5. Что делает функция getchar()?

- 1) читает один символ из stdin программы и возвращает его код
- 2) выводит в stdout символ, указанный в качестве аргумента
- 3) резервирует место в куче (heap) для хранения одного символа
- 4) читает один символ из указанного в качестве аргумента файла

6. Найти правильные значения переменной: int i=2; i +=5; i*=10; i-=6; i/=8; i%=3;

- 1) i=2, i=5, i=70, i= 64, i=8, i=2

- 2) i=2,i=7,i=10, i=64, i=8, i=2
- 3) i=2, i=7, i=70, i=64, i=8, i=2
- 4) i=2, i=7, i=70,i=64,i=10,i=2

7. Укажите правильный вариант функции, возвращающей сумму двух чисел:

- 1) int sum(int j; int k) { return(j+k); }
- 2) int sum(int j,int k) { return(j+k); }
- 3) int sum() { return(j+k); int j, k; }

8. Чем определяется мобильность языка Си?

- 1) тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена без изменений на другую систему
- 2) тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена с небольшими изменениями или вообще без них на другую
- 3) мобильность языка Си определяется его эффективностью

9. В чем опасность применения макроопределений?

- 1) побочный эффект
- 2) нет побочных эффектов
- 3) тяжело программировать
- 4) в создании строчного кода

10. Какая функция может объединить две строки?

- 1) strcat()
- 2) strcmp()
- 3) strcpy()
- 4) strchr()

11. Как обратиться к элементу структуры при помощи указателя?

- 1) new_student<id
- 2) new_student->id
- 3) new_student*id
- 4) new_student-id

12. Каково значение целых переменных x и y после выполнения $y=x=(2+3)/4$;

- 1) x=1, y=1
- 2) x=1.025, y=1
- 3) x=1, y=0
- 4) x=0, y=1

13. Есть ли ошибки в фрагменте программы: int age; scanf("%f", age);?

- 1) ошибок нет

- 2) одна ошибка - для переменной age целого типа, необходимо использовать %d, а не %f
- 3) одна ошибка - вместо age должно стоять &age
- 4) две ошибки, необходимо использовать %d, а не %f и &age, т.к. вторым аргументом должен быть указатель

14. Что такое EOF?

- 1) специальный символ, указывающий на начало файла
- 2) специальный символ, указывающий на конец файла
- 3) специальный символ, указывающий на то, что программа написана на Си
- 4) определяет мобильность языка Си

15. Правильна ли спецификация цикла for: for(int i=0, i<=10,i++);

- 1) Спецификация цикла for правильна
- 2) Вместо запятых должны стоять символы "точка с запятой"
- 3) Нельзя в цикле описывать переменную
- 4) Недопустимо использовать в качестве тела цикла пустой оператор

16. Как передавать функции информацию?

- 1) циклом for
- 2) вместо запятых должны стоять в формальных аргументах символы "точка с запятой"
- 3) информацию функции передать нельзя!
- 4) при помощи аргументов

17. Могут ли быть элементы массива функциями?

- 1) да, но только функции типа void
- 2) да, но только функции типа int
- 3) нет, но элементы массива могут быть указателями на функции
- 4) нет, связать массивы с функциям невозможно в принципе

18. Как сохраняются в памяти строковые константы?

- 1) каждая строковая константа, даже если она идентична другой строковой константе, сохраняется в отдельном месте памяти
- 2) строковые константы проверяются на совпадения, и в памяти хранится только один вариант
- 3) строковые константы хранятся единым блоком в памяти ЭВМ
- 4) строковые константы разбиваются на литеры и каждая литера заменяется ее адресом

19. Что будет выведено на печать в результате работы оператора printf("%d",100+65)?

- 1) 100+65
- 2) 165

- 3) 65+100
- 4) 100

20. Что называется простейшим выражением?

1) простейшим выражением называется выражение, сформированное с использованием констант типов `int`, `char`, `enum`, `sizeof`, унарных операторов - `~`, бинарных операторов `+` `~` `*` `/` `%` `&` `^` `<<` `>>` `=` `!=` `<` `>` `<=` `>=` и тернарной операции `?:`:

2) простейшим выражением называется выражение , сформированное с использованием констант типов `int`, `char`, `enum`, `sizeof`

3) простейшим выражением называется выражение , сформированное с использованием унарных операторов - `~`

4) простейшим выражением называется выражение , сформированное с использованием бинарных операторов `+` `~` `*` `/` `%` `&` `^` `<<` `>>` `=` `!=` `<` `>` `<=` `>=` и тернарной операции `?:`:

21. Будет ли работать программа и если нет - в чем ошибки? `#include <stdio.h> int main() { printf("Hello, World!\n"); }`

1) программа скомпилируется и отработает, но код завершения будет не определен

2) программа не скомпилируется из-за синтаксической ошибки

3) программа не скомпилируется, т.к. не указан возвращаемый функцией тип

22. Где имеет силу даваемое определение директивы `#define`?

1) от места появления до конца файла или до `#undef`

2) от начала файла до места появления

3) в первой описанной функции файла

4) в последней описанной функции файла

23. В какой памяти хранится строковая константа?

1) в системном стеке

2) в динамической

3) в статической

4) в циклическом списке

24. Что называется составным оператором?

1) составной оператор представляет собой два или более операторов, объединенных с помощью квадратных скобок

2) составной оператор представляет собой четное число операторов , объединенных с помощью фигурных скобок

3) составной оператор представляет собой нечетное число операторов , объединенных с помощью фигурных скобок

4) составной оператор представляет собой два или более операторов, объединенных с помощью фигурных скобок

25. Что является указателем на хранящуюся в памяти строковую константу?

- 1) первая литера строковой константы является указателем на место в памяти
- 2) вся фраза в кавычках является указателем на место в памяти
- 3) последняя литера строковой константы является указателем на место в памяти
- 4) первая четная литера строковой константы является указателем на место в памяти

26. Какие функции объявляются в заголовке <memory.h>?

- 1) набор функций, предназначенных для манипулирования символьными строками
- 2) набор функций, предназначенных для получения доступа к информации о бюджете пользователей
- 3) набор функций, предназначенных для манипулирования областями памяти
- 4) набор функций, предназначенных для вызова системных параметров времени

27. Что понимается под основным циклом?

- 1) самый внешний цикл программы
- 2) самый внутренний цикл программы
- 3) самый маленький цикл программы
- 4) самый большой цикл программы

28. Что понимается под усечением в языке Си?

- 1) в языке Си дробная часть деления дробных чисел отбрасывается
- 2) в языке Си дробная часть у результата у результата деления целых чисел отбрасывается
- 3) в языке Си дробная часть у результата деления целых чисел на данное с плавающей точкой отбрасывается
- 4) в языке Си дробная часть у результата деления данных с плавающей точкой на целое отбрасывается

29. Что такое значение ложь в языке Си?

- 1) в языке Си значение ложь равно 1
- 2) в языке Си все ненулевые значения являются ложными
- 3) в языке Си все отрицательные значения являются истинными
- 4) в языке Си значение ложь равно 0

30. Дано описание `int i;`. Верен ли синтаксически фрагмент выражения `(char)i`, и что означает запись?

- 1) фрагмент выражения записан синтаксически неверно

- 2) фрагмент выражения записан синтаксически верно. Такая запись означает, что результат вычисления переменной *i* будет приведен к типу `char`
- 3) фрагмент выражения записан верно и означает примечание
- 4) фрагмент записан синтаксически верно, и означает, что переменная целого типа неявно преобразуется к переменной символьного типа

31. Какой оператор используется для получения адреса переменной?

- 1) *
- 2) &
- 3) #
- 4) %

32. Какой из следующих типов данных позволяет хранить дробные числа?

- 1) `int`
- 2) `char`
- 3) `float`
- 4) `short`

33. Что произойдет, если в функции будет передан массив?

- 1) Будет передан копия всего массива
- 2) Будет передан указатель на первый элемент массива
- 3) Массив не может быть передан в функцию
- 4) Массив будет автоматически преобразован в строку

34. Какой оператор используется для инкремента значения переменной на единицу?

- 1) `++`
- 2) `+=`
- 3) `+1`
- 4) `--`

35. Что вернет функция `strcmp("abc", "abc")`?

- 1) 0
- 2) 1
- 3) -1
- 4) Неопределенное значение

36. Какой из следующих операторов используется для выполнения побитового И?

- 1) `&&`
- 2) `||`
- 3) `&`
- 4) `|`

37. Какой директивой можно подключить стандартную библиотеку ввода-вывода?

- 1) `#include <stdio.h>`
- 2) `#include <stdlib.h>`
- 3) `#include <string.h>`
- 4) `#include <math.h>`

38. Какой тип данных используется для хранения логических значений?

- 1) `bool`
- 2) `int`
- 3) `char`
- 4) `double`

39. Какой оператор используется для сравнения двух значений на равенство?

- 1) `=`
- 2) `==`
- 3) `!=`
- 4) `<>`

40. Какой результат будет получен при выполнении выражения 10 % 3?

- 1) 1
- 2) 2
- 3) 3
- 4) 0

41. Какой оператор позволяет выполнить условный выбор?

- 1) `if`
- 2) `switch`
- 3) `for`
- 4) both 1 and 2

42. Что будет результатом выполнения следующего кода: `int x = 5; printf("%d", x++);`?

- 1) 5
- 2) 6
- 3) Undefined behavior
- 4) 0

43. Какой символ используется для обозначения конца строки в языке Си?

- 1) `\0`
- 2) `\n`

- 3) \t
- 4) \r

44. Что такое указатель на функцию?

- 1) Переменная, которая может хранить адрес функции
- 2) Переменная, которая может хранить адрес массива
- 3) Переменная, которая может хранить адрес строки
- 4) Переменная, которая может хранить адрес структуры

45. Какой оператор используется для выполнения побитового ИЛИ?

- 1) &
- 2) |
- 3) ^
- 4) ~

Ключ к тесту:

1.4	2.1	3.1	4.2	5.1	6.3	7.2	8.2	9.1
10.1	11.2	12.1	13.4	14.2	15.2	16.4	17.3	18.2
19.2	20.1	21.1	22.1	23.3	24.4	25.1	26.3	27.1
28.2	29.4	30.2	31.2	32.3	33.2	34.1	35.1	36.3
37.1	38.1	39.2	40.2	41.4	42.1	43.1	44.1	45.2

Шкала оценивания результатов тестирования

% верных решений (ответов)	Шкала оценивания
85 - 100	отлично
70 - 84	хорошо
50- 69	удовлетворительно
0 - 49	неудовлетворительно

6.2.4. Примеры задач при разборе конкретных ситуаций

Тема 1. Виртуальная машина Ассемблер IA-32. Синтаксис Intel. GNU Assembler Linux. Синтаксис AT&T.

ПК-2.

1. Изучить различия между синтаксисом Intel и AT&T для ассемблера. Написать краткий обзор (1-2 страницы), в котором объяснить, как различные синтаксисы влияют на написание и понимание кода.

2. Написать простую программу на ассемблере IA-32, которая выводит на экран сообщение "Hello, World!" с использованием синтаксиса Intel. Прокомментировать каждую строку кода и объяснить, как она работает.

Тема 2. Расширение архитектуры x86-64. Архитектура арифметического сопроцессора. MMX.SSE

ПК-2.

1. Исследовать архитектуру x86-64 и написать отчет о том, как она улучшает производительность по сравнению с архитектурой x86. Указать ключевые изменения и их влияние на программирование.

2. Написать программу на ассемблере, использующую инструкции MMX или SSE для выполнения операций с массивами данных. Объяснить, как эти инструкции улучшают производительность по сравнению с обычными арифметическими операциями.

Тема 3. Виртуальная машина FORTH. Виртуальная машина FACTOR

ПК-2.

1. Проанализировать основные концепции виртуальной машины FORTH. Написать краткий обзор (1-2 страницы), в котором объяснить, как выполняются команды и как работает стек.

2. Написать простую программу на языке FORTH, которая принимает два числа и выводит их сумму. Прокомментировать код и объяснить, как работает взаимодействие со стеком.

Тема 4. Виртуальная машина Си. Основные понятия языка.

ПК-2.

1. Исследовать основные понятия языка Си, такие как переменные, типы данных и функции. Написать отчет (1-2 страницы), в котором объяснить, как эти концепции используются для разработки программ.

2. Написать программу на языке Си, которая считывает массив целых чисел и выводит их сумму. Прокомментировать код и объяснить, как работает управление памятью в этой программе.

Тема 5. Работа с памятью. Указатели. Массивы. Функции.

ПК-2.

1. Написать программу на языке Си, которая использует динамическое выделение памяти для создания массива целых чисел. Заполнить массив случайными значениями и вывести его на экран. Объяснить, как работает управление памятью в данном коде.

2. Разработать функцию, которая принимает указатель на массив и его размер, а затем возвращает максимальное значение в этом массиве. Прокомментировать код и объяснить, как передача указателя влияет на работу функции.

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	обучающийся ясно изложил условие задачи, решение обосновал
«Хорошо»	обучающийся ясно изложил условие задачи, но в обосновании решения имеются сомнения;
«Удовлетворительно»	обучающийся изложил решение задачи, но обосновал его формулировками обыденного мышления;
«Неудовлетворительно»	обучающийся не уяснил условие задачи, решение не обосновал либо не сдал работу на проверку (в случае проведения решения задач в письменной форме).

6.3. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Вопросы для подготовки к промежуточной аттестации по итогам освоения дисциплины Программирование на языке низкого уровня:

ПК-2.

1. Понятие языка программирования низкого уровня
2. Что такое регистры процессора
3. Структура программы на ассемблере
4. Какие бывают флаги состояния
5. Как представлены в памяти целые числа
6. Что такое команды и что такое данные
7. Для чего используются директивы размещения данных
8. Что такое косвенный операнд
9. Порядок следования операндов в инструкции Intel
10. Порядок следования операндов в инструкции AT&T
11. Расширение архитектуры x86-64
12. Набор инструкций x86-64
13. Архитектура арифметического сопроцессора
14. Представление чисел с плавающей точкой
15. Набор инструкций арифметического сопроцессора
16. Расширение MMX (MultiMedia eXtension)
17. Набор инструкций MMX
18. Расширение SSE (Streaming SIMD Extension)
19. Расширение SSE2, SSE3, SSSE3, SSE4
20. Использование автоматической векторизации компиляторами
21. Машины со стековой архитектурой
22. Шитый код
23. Байт-код
24. Обратная польская запись
25. Операции со стеком данных
26. Словарь
27. Условный оператор
28. Операции над целыми числами
29. Арифметические операции над числами с фиксированной точкой
30. Операция масштабирования
31. Циклы со счетчиком
32. Циклы с условием
33. Выход из цикла (leave) и ветвление (branch)
34. Числа двойной длины
35. Переменные, константы и массивы
36. Функционирование форт-системы
37. Контекстные словари

38. Виртуальная машина FACTOR как дальнейшее развитие FORTH.
39. Стандартная библиотека FACTOR
40. Применение FACTOR
41. Структура программы (модуля) на языке Си.
42. Стандарты языка Си.
43. Базовые типы данных на языке Си.
44. Потоки стандартного ввода, вывода и вывода ошибки на языке Си.
45. Форматированный ввод/вывод
46. Представление целых чисел
47. Представление чисел с плавающей точкой.
48. Математические операции
49. Логические и побитовые операции.
50. Управление вычислениями. Условный оператор.
51. Управление вычислениями. Циклы.
52. Основы работы со строками и символами.
53. Массивы в Си.
54. Сортировка и поиск в одномерных массивах.
55. Указатели. Операции с указателями.
56. Связь между указателями и массивами.
57. Структура памяти программы на Си.
58. Управление памятью.
59. Динамические массивы.
60. Многомерные динамические массивы.
61. Структуры в Си
62. Пользовательские типы данных с помощью typedef.
63. Перечисления.
64. Объединения.
65. Функции в Си. Основные понятия.
66. Области видимости переменных.
67. Передача аргументов по значению и по ссылке.
68. Передача массивов функциям.
69. Область видимости Static
70. Функции генераторы.
71. Указатели на функцию.
72. Функции высших порядков.
73. Элементы функционального программирования на Си
74. Директивы препроцессора
75. Модульное программирование на языке Си.
76. Основные функции стандартной библиотеки Си
77. Работа с файлами
78. Работа со строками.
79. Ассемблерные вставки
80. Работа с графикой OpenCV

6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Основной целью проведения промежуточной аттестации является определение степени достижения целей по учебной дисциплине или ее разделам. Осуществляется это проверкой и оценкой уровня теоретических знаний, полученных обучающимися, умения применять их в решении практических задач, степени овладения обучающимися практическими навыками и умениями в объеме требований рабочей программы по дисциплине, а также их умение самостоятельно работать с учебной литературой.

Организация проведения промежуточной аттестации регламентирована «Положением об организации образовательного процесса в федеральном государственном автономном образовательном учреждении «Московский политехнический университет»

6.4.1. Показатели оценивания компетенций на различных этапах их формирования, достижение обучающимися планируемых результатов обучения по дисциплине

Код и наименование компетенции ПК-2. Способен разрабатывать требования и проектировать программное обеспечение				
Этап (уровень)	Критерии оценивания			
	неудовлетворительно	удовлетворительно	хорошо	отлично
знать	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует неполное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует частичное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня	Обучающийся демонстрирует полное соответствие следующих знаний: Основные характеристики и предназначение языков программирования низкого уровня
уметь	Обучающийся не умеет или в недостаточной степени умеет Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует неполное соответствие следующих умений: Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует частичное соответствие следующих умений: Использовать понимание архитектуры виртуальной машины языка низкого	Обучающийся демонстрирует полное соответствие следующих умений: Использовать понимание архитектуры виртуальной машины языка низкого

владеть	Обучающийся не владеет или в недостаточной степени владеет: Базовыми навыками разработки ПО на языке низкого уровня	Обучающийся владеет в неполном и проявляет недостаточность владения: Базовыми навыками разработки ПО на языке низкого уровня	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет Базовыми навыками разработки ПО на языке низкого уровня	Обучающийся свободно применяет полученные навыки, в полном объеме владеет Базовыми навыками разработки ПО на языке низкого уровня
----------------	---	--	--	---

6.4.2. Методика оценивания результатов промежуточной аттестации

Показателями оценивания компетенций на этапе промежуточной аттестации по дисциплине «Программирование на языке низкого уровня» являются результаты обучения по дисциплине.

Оценочный лист результатов обучения по дисциплине

Код компетенции	Знания	Умения	Навыки	Уровень сформированности компетенции на данном этапе / оценка
ПК-2. Способен разрабатывать требования и проектировать программное обеспечение	на уровне знаний: знать основные характеристики и предназначение языков программирования низкого уровня	на уровне умений: уметь использовать понимание архитектуры виртуальной машины языка низкого уровня	на уровне навыков: базовыми навыками разработки ПО на языке низкого уровня	
Оценка по дисциплине (среднее арифметическое)				

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, навыки).

Оценка «отлично» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по дисциплине «Программирование на языке низкого уровня», при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7. Электронная информационно-образовательная среда

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде Чебоксарского института (филиала) Московского политехнического университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), как на территории филиала, так и вне ее.

Электронная информационно-образовательная среда – совокупность информационных и телекоммуникационных технологий, соответствующих технологических средств, обеспечивающих освоение обучающимися образовательных программ в полном объеме независимо от места нахождения обучающихся.

Электронная информационно-образовательная среда обеспечивает:

а) доступ к учебным планам, рабочим программам дисциплин (модулей), практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), практик;

б) формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы;

в) фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы бакалавриата;

г) проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

д) взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

Функционирование электронной информационно-образовательной среды обеспечивается соответствующими средствами информационно-коммуникационных технологий и квалификацией работников, ее использующих и поддерживающих.

Функционирование электронной информационно-образовательной среды соответствует законодательству Российской Федерации.

Основными составляющими ЭИОС филиала являются:

а) сайт института в сети Интернет, расположенный по адресу www.polytech21.ru, <https://chebpolytech.ru/> который обеспечивает:

- доступ обучающихся к учебным планам, рабочим программам дисциплин, практик, к изданиям электронных библиотечных систем, электронным информационным и образовательным ресурсам, указанных в рабочих программах (разделы сайта «Сведения об образовательной организации»);

- информирование обучающихся обо всех изменениях учебного процесса (новостная лента сайта, лента анонсов);

- взаимодействие между участниками образовательного процесса (подразделы сайта «Задать вопрос директору»);

б) официальные электронные адреса подразделений и сотрудников института с Яндекс-доменом @polytech21.ru (список контактных данных подразделений Филиала размещен на официальном сайте Филиала в разделе «Контакты», списки контактных официальных электронных данных преподавателей размещены в подразделах «Кафедры») обеспечивают взаимодействие между участниками образовательного процесса;

в) личный кабинет обучающегося (портфолио) (вход в личный кабинет размещен на официальном сайте Филиала в разделе «Студенту» подразделе

«Электронная информационно-образовательная среда») включает в себя портфолио студента, электронные ведомости, рейтинг студентов и обеспечивает:

- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися,

- формирование электронного портфолио обучающегося, в том числе с сохранение работ обучающегося, рецензий и оценок на эти работы,

- г) электронные библиотеки, включающие электронные каталоги, полнотекстовые документы и обеспечивающие доступ к учебно-методическим материалам, выпускным квалификационным работам и т.д.:

Чебоксарского института (филиала) - «ИРБИС»

- д) электронно-библиотечные системы (ЭБС), включающие электронный каталог и полнотекстовые документы:

- ЭБС «ЛАНЬ» -<https://e.lanbook.com/>

- Образовательная платформа Юрайт - <https://urait.ru>

- IPR SMART -<https://www.iprbookshop.ru/>

- е) платформа цифрового образования Политеха -<https://lms.mospolytech.ru/>

- ж) система «Антиплагиат» -<https://www.antiplagiat.ru/>

- з) система электронного документооборота DIRECTUM Standard — обеспечивает документооборот между Филиалом и Университетом;

- и) система «IC Управление ВУЗом Электронный деканат» (Московский политехнический университет) обеспечивает фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися;

- к) система «POLYTECH systems» обеспечивает информационное, документальное автоматизированное сопровождение образовательного процесса;

- л) система «Абитуриент» обеспечивает документальное автоматизированное сопровождение работы приемной комиссии.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Кудрина, Е. В. Основы алгоритмизации и программирования на языке С# : учебник для вузов / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2025. — 322 с. — (Высшее образование). — ISBN 978-5-534-09796-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565466>.

2. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Высшее образование). — ISBN 978-5-534-20430-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562040>.

Дополнительная литература:

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

2. Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563618>.

9. Профессиональные базы данных и информационно-справочные системы

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
Университетская информационная система РОССИЯ https://uisrussia.msu.ru/	Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ
научная электронная библиотека Elibrary http://elibrary.ru/	Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ
сайт Института научной информации по общественным наукам РАН. http://www.inion.ru	Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объем массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.
Федеральный портал «Российское образование» [Электронный ресурс] – http://www.edu.ru	Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки. Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Еженедельно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи.

	Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.
Информационные технологии – периодическое научно-техническое издание в области информационных технологий, автоматизированных систем и использования информатики в различных приложениях novtex.ru	Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий.
Ассоциация инженерного образования России http://www.ac-raee.ru/	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

10. Программное обеспечение (лицензионное и свободно распространяемое), используемое при осуществлении образовательного процесса

Адрес (местоположение) объекта подтверждающего наличие ПО	Программное обеспечение	Информация о праве собственности (реквизиты договора, номер лицензии и т.д.)
№2026 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Компьютерный класс Лаборатория микропроцессоров Лаборатория информационных технологий	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Гарант- справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная

	Electronic Software Delivery Academic(Microsoft Open License	лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)
№ 2196 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей)	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Google Chrome	Свободное распространяемое программное обеспечение (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
№ 1126 Помещение для самостоятельной работы обучающихся	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор № ППИ-126/2023 от 14.12.2023
	Windows 7 OLPNLAcdmс	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)

	Гарант- справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Zoom	свободно распространяемое программное обеспечение (бессрочная лицензия)
	AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)

11. Материально-техническое обеспечение дисциплины

Тип и номер помещения	Перечень основного оборудования и технических средств обучения
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры/бакалавриата/специалитета/ магистратуры,, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» № 2196 (г. Чебоксары, ул. К.Маркса, д.60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/ магистратуры, оснащенная оборудованием и	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; автоматизированные рабочие места, автоматизированное рабочее место преподавателя, проектор и экран; маркерная доска; <u>Технические средства обучения:</u> компьютерная техника (процессор Core i3, оперативная память объемом не менее 8 Гб); сервер в лаборатории (8-ядерный процессор

техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория информационных технологий №2066 (г. Чебоксары, ул. К.Маркса, д.60)	с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб
Помещение для самостоятельной работы обучающихся № 1126 (г. Чебоксары, ул. К.Маркса. 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала

12. Методические указания для обучающегося по освоению дисциплины

Методические указания для занятий лекционного типа

В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

Методические указания для занятий семинарского (практического) типа.

Практические занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Подготовка к практическому занятию включает два этапа. На первом этапе обучающийся планирует свою самостоятельную работу, которая включает: уяснение задания на самостоятельную работу; подбор основной и дополнительной литературы; составление плана работы, в котором определяются основные пункты предстоящей подготовки. Составление плана дисциплинирует и повышает организованность в работе.

Второй этап включает непосредственную подготовку к занятию, которая начинается с изучения основной и дополнительной литературы. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. Далее следует подготовить тезисы

для выступлений по всем учебным вопросам, выносимым на практическое занятие или по теме, вынесенной на дискуссию (круглый стол), продумать примеры с целью обеспечения тесной связи изучаемой темы с реальной жизнью.

Готовясь к докладу или выступлению в рамках интерактивной формы (дискуссия, круглый стол), при необходимости следует обратиться за помощью к преподавателю.

Методические указания к самостоятельной работе.

Самостоятельная работа обучающегося является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа обучающегося над усвоением учебного материала по учебной дисциплине может выполняться в библиотеке университета, учебных кабинетах, компьютерных классах, а также в домашних условиях. Содержание и количество самостоятельной работы обучающегося определяется учебной программой дисциплины, методическими материалами, практическими заданиями и указаниями преподавателя.

Самостоятельная работа в аудиторное время может включать:

- 1) конспектирование (составление тезисов) лекций;
- 2) выполнение контрольных работ;
- 3) решение задач;
- 4) работу со справочной и методической литературой;
- 5) работу с нормативными правовыми актами;
- 6) выступления с докладами, сообщениями на семинарских занятиях;
- 7) защиту выполненных работ;
- 8) участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
- 9) участие в беседах, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
- 10) участие в тестировании и др.

Самостоятельная работа во внеаудиторное время может состоять из:

- 1) повторения лекционного материала;
- 2) подготовки к практическим занятиям;
- 3) изучения учебной и научной литературы;
- 4) изучения нормативных правовых актов (в т.ч. в электронных базах данных);
- 5) решения задач, и иных практических заданий
- 6) подготовки к контрольным работам, тестированию и т.д.;
- 7) подготовки к практическим занятиям устных докладов (сообщений);
- 8) подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя;
- 9) выполнения курсовых работ, предусмотренных учебным планом;
- 10) выполнения выпускных квалификационных работ и др.
- 11) выделения наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на консультациях.

12) проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов и эссе по отдельным вопросам изучаемой темы.

Текущий контроль осуществляется в форме устных, тестовых опросов, докладов, творческих заданий.

В случае пропусков занятий, наличия индивидуального графика обучения и для закрепления практических навыков студентам могут быть выданы типовые индивидуальные задания, которые должны быть сданы в установленный преподавателем срок.

13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Обучение по дисциплине «Программирование на языке низкого уровня» инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ) осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для студентов с нарушениями опорно-двигательной функции и с ОВЗ по слуху предусматривается сопровождение лекций и практических занятий мультимедийными средствами, раздаточным материалом.

Для студентов с ОВЗ по зрению предусматривается применение технических средств усиления остаточного зрения, а также предусмотрена возможность разработки аудиоматериалов.

По дисциплине «Программирование на языке низкого уровня» обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и с использованием электронной информационно-образовательной среды, образовательного портала и электронной почты.

ЛИСТ ДОПОЛНЕНИЙ И ИЗМЕНЕНИЙ

рабочей программы дисциплины

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от «__» 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от «__» 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от «__» 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от «__» 202__ г.

Внесены дополнения и изменения _____