

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью

Организация: Финансовый университет при Правительстве РФ

Сертификат: 317IOS0B3hbH3tTgltezO2fnDO+4QmCM

Утверждено: Проректор по учебной и методической работе Е.А. Каменева

Дата: 11.02.2026 г.

А.В. Антипов, А.М. Ерофеев

Низкоуровневое программирование

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике и
финансах»

Профиль:

«Прикладные информационные системы в экономике и финансах»

Рекомендовано

Факультет информационных технологий и анализа больших данных

(протокол № 06 от 17.03.2026 г.)

Одобрено

Кафедра информационных технологий

(протокол № 1 от 27.02.2026 г.)

Содержание

1. Наименование дисциплины	4
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	4
3. Место дисциплины в структуре образовательной программы	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий . . .	6
5.1. Содержание дисциплины	6
5.2. Учебно-тематический план	8
5.3. Содержание семинаров	9
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	11
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	11
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю . . .	13
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	14
8. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины	18
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	19
10. Методические указания для обучающихся по освоению дисциплины	20
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	22

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	23
--	----

1. Наименование дисциплины

«Низкоуровневое программирование».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	
ПКП-1	Способен разрабатывать и отлаживать программный код	
Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	
Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода	знать: основные методы формализации задач, типы алгоритмов и способы их описания, принципы структурного программирования уметь: формулировать математическую постановку задачи, строить блок-схемы алгоритмов, выбирать оптимальные алгоритмические решения для низкоуровневых задач	
Оформляет программный код в соответствии с установленными требованиями	знать: стандарты оформления кода, требования к документированию, правила именования и структурирования программ уметь: писать читаемый и документированный код на языке ассемблера, применять отладочные инструменты, оформлять техническую документацию к программам	

3. Место дисциплины в структуре образовательной программы

Дисциплина «Низкоуровневое программирование» относится к «Модулю ”Системное программирование”».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 6 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа – Аудиторные занятия	34	34
Лекции	16	16
Семинары, практические занятия	18	18
Самостоятельная работа	74	74
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

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

5.1. Содержание дисциплины

Тема 1. Основы архитектуры вычислительных систем и языка ассемблера

Понятие низкоуровневого программирования. Архитектура фон Неймана. Регистры процессора. Системы счисления: двоичная, шестнадцатеричная, восьмеричная. Представление данных в памяти: целые числа со знаком и без знака, символы, строки. Организация памяти: сегментная модель, стек, куча. Структура программы на ассемблере. Директивы сегментации. Команды передачи данных (MOV, PUSH, POP, LEA). Арифметические команды (ADD, SUB, INC, DEC, MUL, DIV, IMUL, IDIV). Логические команды (AND, OR, XOR, NOT, SHL, SHR). Команды сравнения и перехода (CMP, JMP, условные переходы).

Тема 2. Управление выполнением программы

Организация циклов (LOOP, LOOPE, LOOPNE). Вложенные циклы. Процедуры и функции: директивы PROC и ENDP, передача параметров через регистры и стек. Команды CALL и RET. Локальные и глобальные метки. Макроопределения и макроподстановки.

Тема 3. Работа с массивами и строками

Индексация элементов массива. Косвенная адресация. Команды работы со строками (MOVS, LODS, STOS, CMPS, SCAS). Префиксы повторения (REP, REPE, REPNE). Обработка одномерных и двумерных массивов. Поиск и сортировка на уровне ассемблера.

Тема 4. Ввод-вывод и прерывания

Порты ввода-вывода. Команды IN и OUT. Программные прерывания (INT). Функции DOS (INT 21h): ввод-вывод символов и строк, работа с файлами, завершение программы. Базовые функции BIOS. Обработка аппаратных прерываний. Векторы прерываний.

Тема 5. Связь ассемблера с языками высокого уровня

Встроенный ассемблер (inline assembly). Соглашения о вызовах (cdecl, stdcall, fastcall). Передача параметров между модулями. Использование ассемблерных вставок в C/C++. Создание библиотек на ассемблере. Оптимизация критических участков программ.

Тема 6. Отладка и оптимизация программ

Инструменты отладки (DEBUG, TD, GDB). Трассировка выполнения. Точки останова. Просмотр регистров и памяти. Профилирование кода. Методы оптимизации: развертывание циклов, инструкции сдвига, использование регистров. Анализ машинного кода.

Тема 7. Специфические возможности архитектуры x86

Расширенные регистры (EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP). 32-битная адресация. Режимы работы процессора. Защищенный режим. Сегментные дескрипторы. Базовые понятия многозадачности. SIMD-инструкции (MMX, SSE) —вводный обзор.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя- тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ- ие занятия	
1	Основы архитектуры вычис- лительных систем и языка ассемблера	22	8	4	4	14
2	Управление выполнением программы	14	4	2	2	10
3	Работа с массивами и строками	14	4	2	2	10
4	Ввод-вывод и прерывания	14	4	2	2	10
5	Связь ассемблера с языками высокого уровня	14	4	2	2	10
6	Отладка и оптимизация программ	16	6	2	4	10
7	Специфические возможности архитектуры x86	14	4	2	2	10
	Итого	108	34	16	18	74

Формы текущего контроля успеваемости согласно учебному плану: **Контрольная работа.**

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Основы архитектуры вычислительных систем и языка ассемблера	Преимущества и недостатки низкоуровневого программирования. Сравнение языков высокого и низкого уровня. Перевод чисел между системами счисления. Диапазоны представления целых чисел. Переполнение и его последствия. Синтаксис языка ассемблера NASM и MASM. Различия в директивах. Ручная трассировка простых программ. Влияние флагов на выполнение команд. Особенности команд умножения и деления.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Управление выполнением программы	Сравнение циклов на ассемблере и в языках высокого уровня. Организация рекурсивных процедур. Проблемы переполнения стека. Сравнение макросов и процедур. Области видимости меток.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Работа с массивами и строками	Эффективность команд строковой обработки. Сравнение программной обработки и строковых команд. Алгоритмы поиска подстроки. Оптимизация обработки массивов.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Ввод-вывод и прерывания	Сравнение программного и аппаратного ввода-вывода. Буферизация данных. Обработка ошибок при файловых операциях. Цепочка прерываний. Резидентные программы — концепция.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Связь ассемблера с языками высокого уровня	Когда целесообразно использовать ассемблерные вставки. Проблемы переносимости кода. Взаимодействие с компилятором. Оптимизация компилятора vs ручная оптимизация.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Отладка и оптимизация программ	Методология отладки сложных программ. Анализ дампов памяти. Поиск утечек ресурсов. Измерение производительности. Профилирование реального времени.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений

Специфические возможности архитектуры x86	Эволюция архитектуры x86. Современные тенденции развития процессоров. Проблемы обратной совместимости. Векторные инструкции в современных задачах.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
--	---	--

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. 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
Основы архитектуры вычислительных систем и языка ассемблера	История развития архитектур процессоров. Микропроцессоры Intel: от 4004 до современности. Альтернативные архитектуры (ARM, RISC-V). Представление чисел с плавающей точкой (IEEE 754). Коды Хэмминга и контроль четности. Директивы определения данных (DB, DW, DD, DQ). Метки и переменные. Константы и выражения. Таблицы перекодировки. Особенности синтаксиса различных ассемблеров (NASM, MASM, FASM, GAS).	Работа с учебной литературой. Разбор вопросов по теме занятия.
Управление выполнением программы	Рекурсивные алгоритмы на ассемблере. Фреймы стека. Соглашение о вызовах cdecl. Локальные переменные в процедурах. Вложенные и рекурсивные макросы. Условная компиляция.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Работа с массивами и строками	Многомерные массивы и их адресация. Динамическое выделение памяти. Работа с указателями. Строки с различными кодировками (ASCII, UTF-8). Алгоритмы сортировки: пузырьковая, быстрая.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Ввод-вывод и прерывания	Аппаратные прерывания и контроллер прерываний (PIC). Обработка исключений процессора. Прямой доступ к памяти (DMA). Порты COM и LPT. Программирование таймера.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Связь ассемблера с языками высокого уровня	Создание DLL на ассемблере. Связывание модулей. Внешние символы (EXTERN, PUBLIC). Использование ассемблера с языками Pascal, Delphi. Встроенный ассемблер в Visual C++ и GCC.	Работа с учебной литературой. Разбор вопросов по теме занятия.

Отладка и оптимизация программ	Статический и динамический анализ кода. Дизассемблирование. Отладка на уровне машинного кода. Анализ производительности кэш-памяти. Оптимизация под конвейер процессора.	Работа с учебной литературой. Разбор вопросов по теме занятия.
Специфические возможности архитектуры x86	Механизмы защиты памяти. Deskрипторные таблицы. Переключение задач. Расширения x86-64. SIMD-инструкции в обработке изображений и звука. Многопоточность на уровне ОС.	Работа с учебной литературой. Разбор вопросов по теме занятия.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания для контрольной работы

1. Перевести число из десятичной системы в двоичную, восьмеричную и шестнадцатеричную системы счисления.
2. Определить диапазон представления 8-битного числа со знаком и без знака.
3. Написать фрагмент кода для обмена значений двух регистров без использования дополнительной памяти.
4. Объяснить различие между командами MOV и LEA.
5. Выполнить трассировку фрагмента программы с арифметическими операциями, определить значения регистров и флагов.
6. Написать программу вычисления выражения с использованием команд умножения и деления.
7. Объяснить назначение флагов ZF, SF, CF, OF и привести примеры их установки.
8. Составить блок-схему алгоритма и реализовать его на ассемблере: нахождение максимума из трех чисел.
9. Описать механизм работы стека при выполнении команд PUSH и POP.
10. Написать программу подсчета единичных битов в байте с использованием логических операций.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКП-1 Способен разрабатывать и отлаживать программный код

1) Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные методы формализации задач, типы алгоритмов и способы их описания, принципы структурного программирования

Уметь: формулировать математическую постановку задачи, строить блок-схемы алгоритмов, выбирать оптимальные алгоритмические решения для низкоуровневых задач

Типовые контрольные задания

Задание 1.Задана задача: вычислить сумму элементов массива, расположенных между первым и последним нулевыми элементами. Выполнить формализацию: определить входные и выходные данные, ограничения, математическую постановку задачи. Оценить сложность алгоритма.

Задание 2.Разработать алгоритм поиска подстроки в строке методом Бойера-Мура. Построить блок-схему алгоритма. Описать таблицу сдвигов для примера строки «АВАСАВА». Оценить временную сложность в лучшем и худшем случаях.

Задание 3.Дана задача сортировки массива из 1000 элементов, частично упорядоченного (80% элементов на своих местах). Сравнить применимость сортировки вставками, быстрой сортировки и пирамидальной сортировки. Обосновать выбор оптимального алгоритма с учетом особенностей низкоуровневой реализации.

Задание 4. Разработать алгоритм вычисления значения многочлена по схеме Горнера для процессора без команды умножения. Представить алгоритм в виде псевдокода и блок-схемы. Определить минимально необходимый набор команд процессора.

Задание 5. Дан рекурсивный алгоритм вычисления чисел Фибоначчи. Построить дерево вызовов для $n=5$. Определить сложность по времени и памяти. Предложить оптимизацию с использованием итеративного подхода и динамического программирования. Сравнить эффективность подходов. 2) **Оформляет программный код в соответствии с установленными требованиями**

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: стандарты оформления кода, требования к документированию, правила именования и структурирования программ

Уметь: писать читаемый и документированный код на языке ассемблера, применять отладочные инструменты, оформлять техническую документацию к программам

Типовые контрольные задания

Задание 1. Написать процедуру на языке ассемблера, выполняющую побитовое реверсирование байта. Код должен содержать: заголовок с описанием назначения, входные/выходные параметры, комментарии к ключевым инструкциям, соблюдение отступов и именования. Предоставить пример вызова.

Задание 2. Предоставлен фрагмент кода с синтаксической и логической ошибками. Найти и исправить ошибки с использованием отладчика. Оформить отчет об отладке: описание обнаруженных ошибок, методы их выявления, исправления. Привести корректный вариант кода с комментариями.

Задание 3. Дан неструктурированный фрагмент кассовой программы на ассемблере без комментариев с магическими константами. Привести код в соответствие со стандартами оформления: выделить константы, добавить комментарии, структурировать логику, обеспечить читаемость. Обосновать внесенные изменения.

Задание 4. Разработать модульную библиотеку строковых функций (`strlen`, `strcpy`, `strcmp`) на ассемблере. Оформить как отдельный файл с заголовком. Обеспечить единый интерфейс вызова. Написать тестовую программу с демонстрацией использования. Документировать требования к вызывающему коду (сохраняемые регистры, флаги).

Задание 5. Написать процедуру на встроенном ассемблере для интеграции с программой на C. Обеспечить корректное соглашение о вызовах (`cdecl`). Предусмотреть сохранение

регистров, которые необходимо сохранить. Написать Makefile для сборки проекта. Провести тестирование корректности работы.

Примеры практико-ориентированных заданий

Задание 1. Разработать программу на ассемблере, выполняющую сложение, вычитание, умножение и деление двух 16-битных целых чисел. Программа должна запрашивать ввод чисел в десятичной форме, выполнять операцию и выводить результат. Предусмотреть проверку деления на ноль и переполнения.

Задание 2. Дан массив из 20 байтов, содержащий знаковые числа. Найти сумму положительных элементов, количество отрицательных элементов и максимальное значение. Реализовать два варианта: с использованием индексной адресации и с использованием строковых команд.

Задание 3. Разработать программу, выполняющую поиск подстроки в строке. Реализовать алгоритм Кнута-Морриса-Практа на ассемблере. Сравнить производительность с наивным алгоритмом поиска.

Задание 4. Реализовать на ассемблере рекурсивную функцию вычисления факториала числа. Проанализировать использование стека при различных значениях аргумента. Определить максимальное значение аргумента до переполнения стека.

Задание 5. Создать программу, копирующую текстовый файл. Реализовать посимвольное и поблочное копирование. Сравнить производительность. Обработать возможные ошибки (отсутствие файла, недостаток места).

Задание 6. Разработать библиотеку на ассемблере, содержащую функцию быстрого копирования блока памяти. Подключить библиотеку к программе на C и сравнить производительность со стандартной функцией memcpy.

Задание 7. Дан фрагмент программы на ассемблере, выполняющий обработку изображения (инверсия цветов). Провести анализ и оптимизировать код по скорости выполнения. Измерить время выполнения до и после оптимизации.

Задание 8. Предоставлена программа с логической ошибкой (некорректная работа с флагами при сравнении). Используя отладчик, найти и исправить ошибку. Оформить отчет о проведенной отладке.

Примерные вопросы для подготовки к зачету

1. Архитектура фон Неймана и её основные принципы.
2. Представление целых чисел в дополнительном коде. Диапазоны представления.
3. Системы счисления и алгоритмы перевода между ними.

4. Структура программы на ассемблере: сегменты, директивы, метки.
5. Классификация команд процессора x86: передача данных, арифметические, логические, управления.
6. Регистры процессора x86 и их специализация.
7. Организация циклов в программах на ассемблере.
8. Механизм вызова процедур: передача параметров, сохранение контекста.
9. Строковые команды процессора и их эффективное применение.
10. Программные и аппаратные прерывания: механизм обработки.
11. Функции DOS для ввода-вывода и файловых операций.
12. Встроенный ассемблер: синтаксис и особенности использования.
13. Соглашения о вызовах и совместимость с языками высокого уровня.
14. Методы отладки программ на ассемблере.
15. Основы оптимизации низкоуровневого кода.
16. Защищенный режим процессора: базовые понятия.
17. Расширения архитектуры x86 (MMX, SSE): назначение и применение.
18. Сравнительный анализ различных ассемблеров (NASM, MASM, FASM).
19. Практические случаи применения низкоуровневого программирования.
20. Тенденции развития архитектур процессоров и роль ассемблера в современной разработке.

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

Основная литература:

1. Рацеев, С. М. Программирование на языке Си [Электронный ресурс] : учебное пособие для вузов / Рацеев С. М. 2-е изд., стер. Санкт-Петербург : Лань, 2023 332 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/351863> ISBN 978-5-507-47236-9. [БИК ID: RU-LAN-BOOK-351863]
2. Бунаков, П. Ю. Машинно-ориентированные языки программирования. Введение в ассемблер [Электронный ресурс] : учебное пособие для вузов / Бунаков П. Ю. Санкт-Петербург : Лань, 2023 144 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/302627> ISBN 978-5-507-45490-7. [БИК ID: RU-LAN-BOOK-302627]

Дополнительная литература:

1. Золкин, А. Л. Машинно-ориентированные языки программирования в сфере искусственного интеллекта [Электронный ресурс] : учебное пособие для вузов / Золкин А. Л. Санкт-Петербург : Лань, 2025 168 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/494978> ISBN 978-5-507-52465-5. [БИК ID: RU-LAN-BOOK-494978]
2. Пичугин, В. Н. Теория трансляций и низкоуровневое программирование [Электронный ресурс] : учеб. пособие / Пичугин В. Н., Солдатов А. А. Чебоксары : ЧГУ им. И. Н. Ульянова, 2024 92 с. Книга из коллекции ЧГУ им. И. Н. Ульянова - Информатика <https://e.lanbook.com/book/482207> ISBN 975-8-7677-3830-4. [БИК ID: RU-LAN-BOOK-482207]
3. Ларина, Татьяна Борисовна Низкоуровневые языки : Учебное пособие / Российский университет транспорта (МИИТ) Москва : Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет транспорта», 2018 147 с. ВО - Бакалавриат <https://znanium.com/catalog/document?id=415617>. [БИК ID: RU\{\{infra-m\}\}znanium\{\}bibl\{\}1895286]

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

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
(<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
3. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
4. • Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
5. • Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
6. • Электронно-библиотечная система Znanium <http://www.znaniy.ru/>
7. • Электронно-библиотечная система BOOK. RU <http://www.book.ru>

10. Методические указания для обучающихся по освоению дисциплины

Основные этапы работы студента по дисциплине «Низкоуровневое программирование»:

1. Предварительная ориентировка в подлежащем изучению учебном материале по программе.
2. Ознакомление с рекомендованной учебной литературой.
3. Слушание и конспектирование лекций, а также выполнение других видов учебной работы.
4. Планирование самостоятельной работы.
5. Обобщение и систематизация информации, почерпнутой из лекций и прочитанной литературы.

Выполнение контрольной работы:

Студентам при подготовке к контрольной работе следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

Рекомендации по работе с учебным материалом:

1. Осознавайте наличный уровень полученных вами знаний.
2. В ситуации непонимания нужно выявить тот первичный уровень и факторы непонимания, которые стали препятствием понимания последующего.
3. Задавайте сами себе вопросы и пытайтесь ответить на них.

Рекомендации по работе на лекции и с лекционным материалом:

1. Основная задача на лекции – осмысление излагаемого в ней материала. Для этого необходимо слушать лекцию с самого начала, не упуская общих, ориентирующих в материале рассуждений и установок лектора.
2. Ведение записей на лекции важно и полезно для лучшего осмысливания материала, для сохранения информации, с целью ее дальнейшего использования.

3. Для облегчения записи рекомендуется применять сокращения повторяющихся терминов или хорошо известных понятий.

Рекомендации по работе с литературой:

1. Если возникли затруднения при разыскивании материала, по какому-либо конкретному вопросу, следует обратиться к предметному указателю, напечатанному, как правило, в конце каждого литературного источника.
2. Предметный указатель – это алфавитный список основных научных понятий (терминов), содержание которых раскрыто в книге, рядом с термином стоят числа, обозначающие номера страниц, на которых изложен материал, относящийся к данному понятию.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Текстовый редактор
2. Редактор кода VS CODE
3. Операционная система
4. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)