

Федеральное государственное образовательное бюджетное учреждение  
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»  
(Финансовый университет)**

**Кафедра информационных технологий  
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

Проректор по учебной и  
методической работе

\_\_\_\_\_ Е.А. Каменева

25.03.2025 г.

**Абашин В.Г.**

**Низкоуровневое программирование**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки:

09.03.03 – Прикладная информатика,

ОП «Прикладные информационные системы в экономике и финансах»

*Рекомендовано Ученым советом*

*Факультета информационных технологий и анализа больших данных  
(протокол № 53 от 18 марта 2025 г.)*

*Одобрено советом Кафедры информационных технологий  
(протокол № 1 от 13 марта 2025 г.)*

**Москва 2025**

## Содержание

1. Наименование дисциплины.....	3
3	
4	
4	
5	
5	
5.2. Учебно – тематический план.....	7
9	
11	
11	
11	
12	
17	
17	
19	
20	
21	

## 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	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-3	Способен наладить и администрировать процесс построения и функционирования отказоустойчивого программного обеспечения информационных систем цифровой экономики, в том числе непрерывную поставку и развертывание программного кода	1. Применяет основные принципы организации командной работы разработчиков над проектом, процедуры, подходы и методы оптимизации и автоматизации процедур тестирования, доставки кода и запуска приложений на серверах	<b>Уметь</b> использовать средства разработки и отладки низкоуровневых программ при командной разработке. <b>Знать</b> принципы организации сегментной адресации и способы адресации, предоставляемые компьютером
		2. Разрабатывает стратегию DevOps и контейнеризированные приложения, работает с системами контроля версий, создает решения для мониторинга обратной связи	<b>Уметь</b> применять языки программирования низкого уровня для написания программного кода <b>Знать</b> особенности низкоуровневого программирования для операционных систем

		3. Владеет навыками автоматизации развертывания, масштабирования контейнеризированных приложений и управления ими, а также навыками автоматизации настройки серверов и инфраструктурных сервисов для обеспечения быстрого развертывания и сокращения времени восстановления после сбоев	<b>Уметь</b> применять выбранные технологии и языки программирования <b>Знать</b> особенности программирования и стандартные библиотеки выбранного языка программирования
--	--	---	--

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Низкоуровневое программирование» относится к Циклу профиля (элективный) по направлению подготовки 09.03.03 – Прикладная информатика, ОП «Прикладные информационные системы в экономике и финансах».

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

очная / очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 6 / 7 (в часах)
<b>Общая трудоемкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<b>Контактная работа – Аудиторные занятия</b>	<b>34</b>	<b>34</b>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<b>Самостоятельная работа</b>	<b>74</b>	<b>74</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 8 (в часах)
<b>Общая трудоемкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<i>Контактная работа – Аудиторные занятия</i>	<i>10</i>	<i>10</i>
<i>Лекции</i>	<i>2</i>	<i>2</i>
<i>Семинары, практические занятия</i>	<i>8</i>	<i>8</i>
<b>Самостоятельная работа</b>	<b>98</b>	<b>98</b>
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. Основные понятия и определения**

Введение в низкоуровневое программирование. Цели и задачи курса. Роль курса в подготовке специалиста. Сведения об истории развития проблематики дисциплины. Обзор и общая характеристика языков программирования. Стандарты языков программирования. Понятие низкоуровневого программирования. Файловые системы с точки зрения системного программиста. Конфигурационная информация операционных систем. Сборка и установка ОС Линукс (ядра операционной системы). Стек протоколов TCP/IP с точки зрения системного программиста. Текстовые редактор vim

**Тема 2. Архитектура и системы команд современных микропроцессоров**

Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана. Арифметико-логическое устройство. Основное (оперативное) запоминающее устройство. Иерархия памяти. Современные структуры микропроцессоров. Регистры общего назначения. Управление потоком команд

### **Тема 3. Введение в язык С**

Структура программы. Переменные, указатели, объявления в С. Преобразование типов данных в С. Логические, битовые, операции сдвига, сравнения в С. Квалифицирующие типы и спецификаторы функций в С. Директивы предпроцессора в С. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора. Операторы цикла. Функции. Область видимости. Операции прерывания и безусловного перехода. Стил программирования

### **Тема 4. Составные типы данных**

Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в ОЗУ ЭВМ. Составные типы данных: строки. Объявление и инициализация строк. Функции для работы со строками. Составные типы данных: многомерные массивы. Инициализация многомерных массивов. Операции с многомерными массивами. Составные типы данных: битовые поля. Составные типы данных: перечисления

### **Тема 5. Вспомогательные средства и приемы работы**

Указатели и адреса памяти. Представление памяти процесса. Управление памятью. Области данных, текста программы, стека, кучи. Модели памяти. Динамические массивы. Преобразования типов данных. Логические, битовые операции, операции сдвига, операции сравнения. Квалифицирующие типы и спецификаторы функций. Директивы предпроцессора. Отличия между стандартами языка Си разных лет. Переменные окружения. Потоки ввода вывода. Передача параметров. Этапы создания исполняемого файла. Профилирование программы. Автоматизация проверки исходных текстов и создания исполняемых файлов. Автоматическая генерация документации. Отладка без специальных программ. Отладка с использованием gdb. Поиск утечек памяти valgrind. Си и ассемблер. Технологии дезассемблирования. Ассемблерные вставки в компиляторе gcc

## Тема 6. Использование библиотечных функций в С программах

Статические библиотеки в С. Динамические библиотеки в С. Реализация односвязных и многосвязных списков. Приёмы оценки вычислительной сложности алгоритмов в системном программировании. Библиотека libusb. Получение списка доступных устройств libusb\_get\_device\_list(). Получение характеристик заданного устройства. Получение данных от USB манипулятора типа мышь. Определение libusb\_transfer\_type() устройства. Определение libusb\_bos\_type() устройства. Определение скорости передачи данных устройства libusb\_speed(). Получение номера устройства на шине libusb\_get\_bus\_number(). Получение порта устройства на шине libusb\_get\_port\_number(). Получение родительского устройства libusb\_get\_parent(). Получение максимального размера пакета устройства libusb\_get\_max\_packet\_size().

### 5.2. Учебно – тематический план

очная, очно-заочная формы обучения

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа - Аудиторная работа			Самостоя- тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Основные понятия и определения	12	4	2	2	8	Обсуждение, опрос
2	Архитектура и системы команд современных микропроцессоров	14	4	2	2	10	Обсуждение, опрос
3	Введение в язык С	22	8	4	4	14	Опрос, выполнение индивидуальных заданий
4	Составные типы данных	16	4	2	2	12	Опрос, выполнение индивидуальных заданий

5	Вспомогательные средства и приемы работы	16	6	2	4	10	Опрос, выполнение индивидуальных заданий
6	Использование библиотечных функций в С программах	28	8	4	4	20	Опрос, выполнение индивидуальных заданий, контрольная работа
	В целом по дисциплине	108	34	16	18	74	Согласно учебному плану: контрольная работа
	Итого в %		31	47	53	69	

*Институт открытого образования*

**Заочная форма обучения**

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа - Аудиторная работа			Самостоя- тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Основные понятия и определения	10	2	2	0	8	Обсуждение, опрос
2	Архитектура и системы команд современных микропроцессоров	10	0	0	0	10	Обсуждение, опрос
3	Введение в язык С	16	2	0	2	14	Опрос, выполнение индивидуальных заданий
4	Составные типы данных	20	0	0	0	20	Опрос, выполнение индивидуальных заданий



5	Вспомогательные средства и приемы работы	18	2	0	2	16	Опрос, выполнение индивидуальных заданий
6	Использование библиотечных функций в С программах	34	4	0	4	30	Опрос, выполнение индивидуальных заданий, контрольная работа
	В целом по дисциплине	108	10	2	8	98	Согласно учебному плану: контрольная работа
	Итого в %		9	20	80	91	

\* объем контактной работы в очно-заочной/заочной формах обучения и индивидуальных учебных планах определяется соответствующими учебными планами. Темы, реализуемые в виде контактной работы, определяются преподавателем самостоятельно, исходя из уровня их сложности

### 5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9	Формы проведения занятий
Основные понятия и определения	Понятие низкоуровневого программирования <b>Рекомендуемые источники:</b> 8.[1],[2],[3],[4]	Занятия в интерактивной форме в виде дискуссий.
Архитектура и системы команд современных микропроцессоров	Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана <b>Рекомендуемые источники:</b> 8.[2], [4]	Занятия в интерактивной форме в виде дискуссий.
Введение в язык С	Структура программы. Переменные, указатели, объявления в С. Преобразование типов данных в С. Логические, битовые, операции сдвига, сравнения в С. Директивы предпроцессора в С. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора. Операторы цикла. Функции. Область видимости.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение

	<b>Рекомендуемые источники:</b> 8.[1], [2]	решений
Составные типы данных	Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в ОЗУ ЭВМ. Составные типы данных: строки. Объявление и инициализация строк. Функции для работы со строками. Составные типы данных: многомерные массивы. Инициализация многомерных массивов. Операции с многомерными массивами. <b>Рекомендуемые источники:</b> 8.[2],[3]	Занятия в интерактивной форме в виде дискуссий.
Вспомогательные средства и приемы работы	Указатели и адреса памяти. Представление памяти процесса. Управление памятью. Области данных, текста программы, стека, кучи. Модели памяти. Динамические массивы. Автоматизация проверки исходных текстов и создания исполняемых файлов. Автоматическая генерация документации. Отладка без специальных программ. Отладка с использованием gdb. Поиск утечек памяти valgrind. Си и ассемблер. Технологии дезассемблирования. Ассемблерные вставки в компиляторе gcc <b>Рекомендуемые источники:</b> 8.[2],[3]	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Использование библиотечных функций в С программах	Библиотека libusb. Получение списка доступных устройств libusb_get_device_list(). Получение характеристик заданного устройства. Получение данных от USB манипулятора типа мышь. Определение libusb_transfer_type() устройства. Определение libusb_bos_type() устройства. Определение скорости передачи данных устройства libusb_speed(). Получение номера устройства на шине libusb_get_bus_number(). Получение порта устройства на шине libusb_get_port_number(). <b>Рекомендуемые источники:</b> 8.[2],[3], [4]	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
Основные понятия и определения	Стандарты языков программирования	Работа с учебной литературой. Разбор вопросов по теме занятия.
Архитектура и системы команд современных микропроцессоров	Основное (оперативное) запоминающее устройство	Работа с учебной литературой. Разбор вопросов по теме занятия. Выполнение и защита домашней контрольной работы
Введение в язык C	Операции прерывания и безусловного перехода. Стил программирования	Работа с учебной литературой. Разбор вопросов по теме занятия.
Составные типы данных	Составные типы данных: перечисления	Работа с учебной литературой. Разбор вопросов по теме занятия.
Вспомогательные средства и приемы работы	Профилирование программы. Автоматизация проверки исходных текстов и создания исполняемых файлов	Работа с учебной литературой. Разбор вопросов по теме занятия.
Использование библиотечных функций в C программах	Получение родительского устройства <code>libusb_get_parent()</code> . Получение максимального размера пакета устройства <code>libusb_get_max_packet_size()</code>	Работа с учебной литературой. Разбор вопросов по теме занятия.

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

#### Примерный вариант контрольной работы

1. Реализовать программу, получающую список всех подключенных к компьютеру USB устройств с использованием `libusb`. Для каждого найденного устройства напечатать его класс, идентификатор производителя и идентификатор изделия.

2. Изучить состав и характеристики, обнаруженных с помощью реализованной программ USB устройств.

3. Дополнить программу, реализованную в п. 2 функцией печати серийного номера USB устройства. Для написания функции рекомендуется использовать функции `libusb_open`, `libusb_close`, `libusb_get_string_descriptor_ascii` для печати поля `iSerialNumber` дескриптора устройства.

*Критерии бальной 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, знаний и умений*

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотношенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-3. Способен наладить и администрировать процесс построения и функционирования отказоустойчивого программного обеспечения информационных систем цифровой экономики, в том	1. Применяет основные принципы организации командной работы разработчиков над проектом, процедуры, подходы и методы оптимизации и автоматизации процедур тестирования, доставки кода и запуска приложений на серверах	<b>Уметь</b> использовать средства разработки и отладки низкоуровневых программ при командной разработке.  <b>Знать</b> принципы организации сегментной адресации и способы адресации, предоставляемые компьютером	<b>Пример задания</b> С помощью логических операций реализуйте запись битов в данной ячейки памяти, размером в 2 байта, в обратном порядке.  <b>Пример задания</b> Реализуйте с помощью логических операцию, при которой старший байт слова логически складывается с младшим байтом, записанным в обратном порядке.

числе непрерывную поставку и развертывание программного кода	2. Разрабатывает стратегию DevOps и контейнеризированные приложения, работает с системами контроля версий, создает решения для мониторинга обратной связи	<p><b>Уметь</b> применять языки программирования низкого уровня для написания программного кода</p> <p><b>Знать</b> особенности низкоуровневого программирования для операционных систем</p>	<p><b>Пример задания</b> Реализуйте логическую арифметику, при которой в одном байте хранилось бы 8 логических переменных.</p> <p><b>Пример задания</b> Реализуйте с помощью логических операцию, при которой старший байт слова логически умножается на младший байт, записанный в обратном порядке.</p>
	3. Владеет навыками автоматизации развертывания, масштабирования контейнеризированных приложений и управления ими, а также навыками автоматизации настройки серверов и инфраструктурных сервисов для обеспечения быстрого развертывания и сокращения времени восстановления после сбоев	<p><b>Уметь</b> применять выбранные технологии и языки программирования</p> <p><b>Знать</b> особенности программирования и стандартные библиотеки выбранного языка программирования</p>	<p><b>Пример задания</b> Используя Valgrind (cachegrind) сформировать аннотированный исходный текст программы и оценить число промахов по кеш-памяти данных и инструкций, которые произошли во внутреннем цикле функций</p> <p><b>Пример задания</b> Используя компилятор GCC, скомпилируйте подготовленную программу с включением отладочной информации (всех уровней). На сколько изменяется размер исполняемого файла в зависимости от уровня детализации отладочной информации?</p>

### Примерные вопросы для подготовки к зачету

1. Введение в низкоуровневое программирование
2. Цели и задачи курса
3. Роль курса в подготовке специалиста
4. Сведения об истории развития проблематики дисциплины
5. Обзор и общая характеристика языков программирования
6. Стандарты языков программирования

7. Понятие низкоуровневого программирования
8. Файловые системы с точки зрения системного программиста
9. Конфигурационная информация операционных систем
10. Сборка и установка ОС Линукс (ядра операционной системы)
11. Стек протоколов TCP/IP с точки зрения системного программиста
12. Текстовые редактор vim
13. Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана
14. Арифметико-логическое устройство
15. Основное (оперативное) запоминающее устройство
16. Иерархия памяти
17. Современные структуры микропроцессоров
18. Регистры общего назначения
19. Управление потоком команд
20. Структура программы на языке C
21. Переменные, указатели, объявления в C
22. Преобразование типов данных в C
23. Логические, битовые, операции сдвига, сравнения в C
24. Квалифицирующие типы и спецификаторы функций в C
25. Директивы предпроцессора в C
26. Простые типы
27. Базовые операторы
28. Выражения и операции
29. Стандартные математические функции
30. Операторы выбора
31. Операторы цикла
32. Функции
33. Область видимости
34. Операции прерывания и безусловного перехода
35. Стил программирования

36. Составные типы данных: массивы
37. Объявление и инициализация массивов
38. Обращение к элементам массива
39. Упорядочивание массивов
40. Составные типы данных: структуры и объединения
41. Расположение элементов структур в ОЗУ ЭВМ
42. Составные типы данных: строки
43. Объявление и инициализация строк
44. Функции для работы со строками
45. Составные типы данных: многомерные массивы
46. Инициализация многомерных массивов
47. Операции с многомерными массивами
48. Составные типы данных: битовые поля
49. Составные типы данных: перечисления
50. Указатели и адреса памяти
51. Представление памяти процесса
52. Управление памятью
53. Области данных, текста программы, стека, кучи
54. Модели памяти
55. Динамические массивы
56. Преобразования типов данных
57. Логические, битовые операции, операции сдвига, операции сравнения
58. Квалифицирующие типы и спецификаторы функций
59. Директивы предпроцессора
60. Отличия между стандартами языка Си разных лет
61. Переменные окружения
62. Потоки ввода вывода
63. Передача параметров
64. Этапы создания исполняемого файла

65. Профилирование программы
66. Автоматизация проверки исходных текстов и создания исполняемых файлов
67. Автоматическая генерация документации
68. Отладка без специальных программ
69. Отладка с использованием gdb
70. Поиск утечек памяти valgrind
71. Си и ассемблер
72. Технологии дезасемблирования
73. Ассемблерные вставки в компиляторе gcc
74. Статические библиотеки в С
75. Динамические библиотеки в С
76. Реализация односвязных и многосвязных списков
77. Приёмы оценки вычислительной сложности алгоритмов в системном программировании
78. Библиотека libusb
79. Получение списка доступных устройств libusb\_get\_device\_list()
80. Получение характеристик заданного устройства
81. Получение данных от USB манипулятора типа мышь
82. Определение libusb\_transfer\_type() устройства
83. Определение libusb\_bos\_type() устройства
84. Определение скорости передачи данных устройства libusb\_speed()
85. Получение номера устройства на шине libusb\_get\_bus\_number()
86. Получение порта устройства на шине libusb\_get\_port\_number()
87. Получение родительского устройства libusb\_get\_parent()
88. Получение максимального размера пакета устройства libusb\_get\_max\_packet\_size()



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

### **Основная литература:**

1. Рацеев, С. М. Программирование на языке Си: учебное пособие для вузов / С. М. Рацеев. — 2-е изд., стер. — Санкт-Петербург: Лань, 2023. — 332 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/351863> (дата обращения: 27.01.2025). — Текст: электронный.

2. Бунаков, П. Ю. Машинно-ориентированные языки программирования. Введение в ассемблер / П. Ю. Бунаков. — Санкт-Петербург: Лань, 2023. — 144 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/302627> (дата обращения: 27.01.2025). — Текст: электронный.

### **Дополнительная литература:**

3. Ларина, Т. Б. Низкоуровневые языки: учебное пособие / Т. Б. Ларина. - Москва: РУТ (МИИТ), 2018. - 147 с. - ЭБС ZNANIUM. - URL: <https://znanium.com/catalog/product/1895286> (дата обращения: 27.01.2025). — Текст: электронный.

4. Романов, С. Л. Работа в операционной среде Linux: практикум для вузов: учебное пособие / С. Л. Романов. — Санкт-Петербург: БГТУ "Военмех" им. Д. Ф. Устинова, 2017. — 74 с. - ЭБС Лань. — URL: <https://e.lanbook.com/book/121866> (дата обращения: 27.01.2025). — Текст: электронный.

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

1. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>

2. Сайт департамента анализа данных и машинного обучения.

3. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>

4. Электронно-библиотечная система BOOK.RU <http://www.book.ru>

5. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

6. Электронно-библиотечная система Znanium <http://www.znanium.com>
7. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
8. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
9. Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>
10. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
11. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>
12. Математические журналы: полнотекстовая коллекция Математического института им. В.А. Стеклова РАН <https://www.mathnet.ru/>
13. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
14. Национальная электронная библиотека <http://нэб.рф/>
15. Ресурсы информационно-аналитического агентства по финансовым рынкам Cbonds.ru <https://cbonds.ru/>
16. СПАРК <https://spark-interfax.ru/>
17. CNKI. Academic Reference <https://ar.oversea.cnki.net/>
18. Электронные продукты издательства Elsevier <http://www.sciencedirect.com>
19. Emerald: Management eJournal Portfolio <https://www.emerald.com/insight/>
20. Реферативная база данных по математике MathSciNET <https://mathscinet.ams.org/mathscinet/>
21. Коллекция научных журналов Oxford University Press <https://academic.oup.com/journals/>
22. Электронные коллекции книг и журналов издательства Springer: <http://link.springer.com/>
23. Платформа STATISTA <https://www.statista.com/>
24. База данных научных журналов издательства Wiley <https://onlinelibrary.wiley.com/>

## **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.
6. Выполнение контрольной работы.

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**

11.1. Комплект лицензионного программного обеспечения:

Astra Linux, Libre Office

Антивирус Kaspersky

11.2. Современные профессиональные базы данных и информационные справочные системы

Информационно-правовая система «Гарант»

Информационно-правовая система «Консультант Плюс»

Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Система комплексного раскрытия информации «СКРИН» -<http://www.skrin.ru/>

11.3. Сертифицированные программные и аппаратные средства защиты информации - не используются

11.4. gcc.

11.5. gdb.

11.6. make.

11.7. vim.

11.8. valgrind.

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

Лекционные занятия проводятся в мультимедийных аудиториях, а семинарские занятия – в компьютерных классах.