



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

Общий факультет (Фрязино)

УТВЕРЖДАЮ

Директор филиала РТУ МИРЭА в г.
Фрязино

_____ Макарова Л.А.

«__» _____ 2023 г.

**Рабочая программа дисциплины (модуля)
Системное программное обеспечение**

Читающее подразделение	кафедра общенаучных дисциплин
Направление	09.03.01 Информатика и вычислительная техника
Направленность	Цифровизация предприятий в области радиоэлектроники
Квалификация	бакалавр
Форма обучения	очная
Общая трудоемкость	3 з.е.

Распределение часов дисциплины и форм промежуточной аттестации по семестрам

Семестр	Зачётные единицы	Распределение часов							Формы промежуточной аттестации
		Всего	Лекции	Лабораторные	Практические	Самостоятельная работа	Контактная работа в период практики и (или) аттестации	Контроль	
6	3	108	16	0	32	24	2,35	33,65	Экзамен

Программу составил(и):

ассистент, Пасько Василий Викторович _____

старший преподаватель, Николаев Иван Вадимович _____

Рабочая программа дисциплины

Системное программное обеспечение

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

составлена на основании учебного плана:

направление: 09.03.01 Информатика и вычислительная техника

направленность: «Цифровизация предприятий в области радиоэлектроники»

Рабочая программа одобрена на заседании кафедры

кафедра общенаучных дисциплин

Протокол от 13.01.2023 № 6

Зав. кафедрой Щучкин Григорий Григорьевич _____

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина «Системное программное обеспечение» имеет своей целью способствовать формированию у обучающихся компетенций, предусмотренных данной рабочей программой в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника с учетом специфики направленности подготовки – «Цифровизация предприятий в области радиоэлектроники».

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Направление:	09.03.01 Информатика и вычислительная техника
Направленность:	Цифровизация предприятий в области радиоэлектроники
Блок:	Дисциплины (модули)
Часть:	Часть, формируемая участниками образовательных отношений
Общая трудоемкость:	3 з.е. (108 акад. час.).

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В результате освоения дисциплины обучающийся должен овладеть компетенциями:

УК-1 - Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

ПК-2 - Способен выполнять работы по созданию (модификации) информационной системы и ее частей, автоматизирующей задачи организационного управления, согласно установленному проекту

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), ХАРАКТЕРИЗУЮЩИЕ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

УК-1 : Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

УК-1.1 : Анализирует задачу, выделяя её базовые составляющие и осуществляет поиск информации для её решения

Знать:

- Устройство современных операционных систем, основы программирования системного программного обеспечения на языках высокого уровня, системы программирования для разработки системного программного обеспечения, основы теории формальных языков и грамматик.

Уметь:

- Использовать утилиты, программы, эмуляторы для установки, настройки операционных систем, разрабатывать системное программное обеспечение на объектно-ориентированных языках, применять различные среды программирования и способы отладки разрабатываемых программ, умение формально представлять грамматики различных языков.

УК-1.2 : Определяет, интерпретирует и ранжирует информацию, требуемую для решения поставленной задачи

Знать:

- Необходимые для осуществления профессиональной деятельности операционные системы и системы программирования.

Уметь:

- Анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять план и целевые этапы разрабатываемых

программ.

В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ) ОБУЧАЮЩИЙСЯ ДОЛЖЕН

Знать:

- Необходимые для осуществления профессиональной деятельности операционные системы и системы программирования.
- Устройство современных операционных систем, основы программирования системного программного обеспечения на языках высокого уровня, системы программирования для разработки системного программного обеспечения, основы теории формальных языков и грамматик.

Уметь:

- Анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять план и целевые этапы разрабатываемых программ.
- Использовать утилиты, программы, эмуляторы для установки, настройки операционных систем, разрабатывать системное программное обеспечение на объектно-ориентированных языках, применять различные среды программирования и способы отладки разрабатываемых программ, умение формально представлять грамматики различных языков.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

При проведении учебных занятий организация обеспечивает развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений и лидерских качеств.

Код занятия	Наименование разделов и тем /вид занятия/	Сем.	Часов	Компетенции
1. Операционные системы				
1.1	Введение. Термины и определения. Ядро ОС. Процессы, программные потоки. (Лек). Ядро операционной системы. Архитектура ядер. Типы архитектур ядер. Уровень системного программного обеспечения.	6	2	УК-1.1, УК-1.2
1.2	Выполнение практических заданий (Пр). Установка и настройка ОС на базе Linux.	6	2	УК-1.1, УК-1.2
1.3	Выполнение практических заданий (Пр). GNU/Linux. Работа с базовыми утилитами командной строки.	6	2	УК-1.1
1.4	Выполнение практических заданий (Пр). Grep. Работа с регулярными выражениями в Linux.	6	2	УК-1.1, УК-1.2
1.5	Процессы, программные потоки. (Лек). Процессы: программа, контекст данные. Планировщик процессов. Межпроцессорное взаимодействие. POSIX. Файловая система. GNU/Linux. Каналы.	6	2	УК-1.1
1.6	Выполнение практических заданий (Пр). Работа с файлами и каталогами. Терминал. технологий	6	2	УК-1.1, УК-1.2

1.7	GNU/Linux. Команды и каталоги. (Лек). Открытие файла. Команды навигации по файлам. Команды для работы с файлами. Структура каталогов. тем "	6	2	УК-1.1, УК-1.2
1.8	Выполнение практических заданий (Пр). Практические задания представлены в виде задач, в которых дается постановка задачи, исходные данные и что необходимо получить в качестве результата.	6	2	
1.9	Подготовка к аудиторным занятиям (Ср). Подготовки к аудиторным занятиям проводились устным опросом, контрольной работой.	6	12	УК-1.1
1.10	Выполнение домашнего задания (Ср). В качестве домашних заданий были даны задачи на тему изучения языка программирования Python, разработка регулярных выражений на данном языке программирования.	6	6	УК-1.2
1.11	Написание домашней письменной работы (эссе, реферата) (Ср). Разработка регулярных выражений на языке программирования Python.	6	6	
2. Язык программирования Python				
2.1	Системы программирования. (Лек). Знакомство с Python. Определение языка. Типы данных и переменные в Python. Списки. Сетевые приложения на Python. Работа с сокетами. Работа с данными в различных форматах. Обработка csv-файлов. Программные средства для программирования интерфейса RS-232.	6	2	УК-1.1, УК-1.2
2.2	Выполнение контрольной работы (Пр). Контрольные работы, представлены в виде вопросов к практическим задачам и в виде дополнительных заданий связанных с ними.	6	2	
2.3	Выполнение практических заданий (Пр). Программирование последовательных интерфейсов с помощью современных программных средств. Современные средства для разработки клиентского ПО в ОС Linux. Современные средства для разработки серверного ПО в ОС Linux. Обзор программных средств для работы с данными в различных форматах.	6	2	УК-1.1, УК-1.2
2.4	Выполнение практических заданий (Пр). Python. Работа с файлами, списками и с данными в различных форматах. Работа с регулярными выражениями в Python.	6	2	УК-1.1, УК-1.2

2.5	Составление документации и тестирование разрабатываемых программ. (Лек). Составление документации для клиент-серверного приложения. Составление документации для приложения по работе с интерфейсом RS232. Тестирование разрабатываемого сетевого приложения на виртуальной машине VirtualBox. Тестирование разрабатываемой программы по работе с интерфейсом RS-232 на эмуляторе VSPE.	6	2	УК-1.1, УК-1.2
2.6	Выполнение практических заданий (Пр). Выполнение практических заданий на тему: составление документации для клиент-серверного приложения. Составление документации для приложения по работе с интерфейсом RS232. Тестирование разрабатываемого сетевого приложения на виртуальной машине VirtualBox. Тестирование разрабатываемой программы по работе с интерфейсом RS-232 на эмуляторе VSPE.	6	2	
2.7	Написание эссе (Пр). Современные подходы к разработке кроссплатформенного программного обеспечения для различным интерфейсов.	6	2	УК-1.1, УК-1.2
3. Теория формальных языков и грамматик				
3.1	Введение в теорию трансляции. Грамматики. Регулярные выражения. (Лек). Классификация языков. Алфавит. Классификация грамматик. Нотация Бекуса Наура. Терминалы, нетерминалы. Лексемы. Продукции. Регулярные выражения. Назначение, применение. Типы регулярных выражений. Символы, группы, квантификаторы. Псевдонимы групп символов. Синтаксис регулярных выражения для разных программных средств.	6	2	УК-1.1
3.2	Устный опрос (Пр). Устные опросы проводились в виде блиц-опросов по лекционным занятиям и по результатам практических работ.	6	2	
3.3	Синтаксический анализ. Парсер. (Лек). Дерево разбора. Токен. Метод рекурсивного спуска. Обобщение нисходящий анализа. Восходящий анализ. LL, LR парсеры. Таблицы символов. Проектирование и разработка синтаксического анализатора.	6	2	УК-1.1, УК-1.2
3.4	Выполнение практических заданий (Пр). Выполнение практических заданий на тему: дерево разбора. Токен. Метод рекурсивного спуска. Обобщение нисходящий анализа. Восходящий анализ. LL, LR парсеры. Таблицы символов. Проектирование и разработка синтаксического анализатора.	6	2	УК-1.2

3.5	Выполнение практических заданий (Пр). Выполнение практических заданий на тему: конечный автомат, построенный по регулярным выражениям. Терминал. Лексема. Поточная обработка символов. Проектирование и разработка лексического анализатора.	6	2	УК-1.1, УК-1.2
3.6	Лексический анализ. Лексер. (Лек). Конечный автомат, построенный по регулярным выражениям. Терминал. Лексема. Поточная обработка символов. Проектирование и разработка лексического анализатора.	6	2	УК-1.1, УК-1.2
3.7	Выполнение практических заданий (Пр). Особенности разработки лексического анализатора.	6	2	УК-1.1, УК-1.2
3.8	Выполнение практических заданий (Пр). Построение цепочек вывода для различных грамматик.	6	2	УК-1.1, УК-1.2
3.9	Защита рефератов (Пр). Рефераты представлены в виде отчета с текстовыми и графическими материалами.	6	2	
3.10	Выполнение домашнего задания (Ср). Выполнение домашнего задания по выданным преподавателем вариантам.	6	0	УК-1.1
3.11	Подготовка к аудиторным занятиям (Ср). Подготовка к аудиторным занятиям и повторение изученного материала.	6	0	УК-1.1, УК-1.2
4. Промежуточная аттестация (экзамен)				
4.1	Подготовка к сдаче промежуточной аттестации (Экзамен).	6	33,65	УК-1.1, УК-1.2
4.2	Контактная работа с преподавателем в период промежуточной аттестации (КрПА).	6	2,35	УК-1.1, УК-1.2

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Перечень компетенций

Перечень компетенций, на освоение которых направлено изучение дисциплины «Системное программное обеспечение», с указанием результатов их формирования в процессе освоения образовательной программы, представлен в п.3 настоящей рабочей программы

5.2. Типовые контрольные вопросы и задания

5.2.1. Экзаменационные вопросы

1. Дать определение и привести примеры системного программного обеспечения?
2. Определение GNU/Linux. На каких PC-совместимых системах семейства процессоров работает? 3. Кем и когда был разработан проект GNU?
4. Структурная схема работы ядра Linux?
5. Организация вычислительной среды Linux?
6. Команды и каталоги GNU/Linux?
7. Регулярные выражения. Операторы регулярных выражений. Grep?
8. Системы программирования. Типы данных, переменные, модули, списки и функции Python?
9. Сокеты. Уровни модели OSI. Какой уровень модели OSI используют сокеты?
10. Python модуль socket. серверные и клиентские методы для передачи и приема данных? Привести примеры.

11. Python модуль csv. Методы для передачи и приема данных CSV-файла? Привести примеры.
12. Python модуль PySerial. Основные команды и методы? Привести примеры.
13. Языки программирования. Классификация и определения? Объектно-ориентированные языки?
14. Какие операции можно выполнять над цепочками символов? Что такое грамматика языка?
15. Как и чем формально определяется грамматика?
16. Как выглядит описание грамматики в форме Бэкуса-Наура? Какие ещё формы описания грамматик существуют?
17. На основе какого принципа классифицируются грамматики в классификации Н. Хомского?
18. Какие типы грамматик выделяют по классификации Н. Хомского?
19. Что такое сентенциальная форма грамматики? Что такое левосторонний и правосторонний выводы? Привести примеры.
20. Какие грамматики относятся к регулярным? Назовите два класса регулярных грамматик.
21. Диаграмма состояния?
22. Алгоритм удаления бесплодных символов?
23. Детерминированный и недетерминированный конечный автомат?
24. Процессы: программа, контекст данные. Планировщик процессов.
25. Синтаксический анализ. Нисходящий и восходящий анализ.
26. Проектирование и разработка лексического анализатора.
27. Дерево разбора. Токен. Метод рекурсивного спуска.
28. Нотация Бекуса Наура. Терминалы, нетерминалы.
29. Регулярные выражения в Python.
30. Схема взаимодействия работы синтаксического и лексического анализатора?
31. Каналы в GNU/Linux?
32. Проектирование и разработка транслятора?
33. Межпроцессорное взаимодействие?
34. Проектирование и разработка интерпретатора?
35. POSIX?
36. Общая схема трансляции?
37. Алгоритм построения детерминированного КА по НКА. Привести пример.
38. Алгоритм построения и разбора диаграммы состояния?
39. Алгоритм удаления недостижимых символов?
40. Грамматики с фразовой структурой?
41. Контекстно-зависимые и неукорачивающие грамматики?
42. Контекстно-свободные грамматики?
43. Регулярные грамматики?
44. Дерево вывода?

5.2.2. Задания

1. Создание, редактирование каталогов и файлов в ОС Linux.

- 1.1. Создать каталог, создать 2 текстовых файла подсчитать количество слов в каждом файле.
- 1.2. Создать каталог, создать текстовый файл подсчитать количество слов в файле.
- 1.3. Создать каталог, создать текстовый файл подсчитать количество символов в файле.
- 1.4. Создать каталог, создать текстовый файл изменить права доступа только для чтения.
- 1.5. Создать каталог, создать текстовый файл изменить права доступа только для записи.
- 1.6. Создать каталог, создать текстовый файл изменить права доступа только для записи и чтения.
- 1.7. Подсчитать в подкаталоге число файлов, начинающихся с точки.
- 1.8. Подсчитать в подкаталоге число файлов, не содержащих в имени точек.
- 1.9. Подсчитать в подкаталоге число файлов, содержащих в имени хотя бы одну точку.

- 1.10. Подсчитать в подкаталоге число файлов, не являющихся подкаталогами.
- 1.11. Подсчитать в подкаталоге число файлов, являющихся подкаталогами.
- 1.12. Найти в подкаталоге файл, содержащий наибольшее количество строк.
- 1.13. Найти в подкаталоге файл, содержащий наименьшее количество строк.
- 1.14. Найти в подкаталоге файл, содержащий наибольшее количество слов.
- 1.15. Дан текстовый файл text.txt, содержащий некоторый текст. Необходимо вычислить длину его 36-й строки.
- 1.16. Необходимо подсчитать, в скольких строках заданного файла содержится слово «UNIX».
- 1.17. Привести все имена файлов в подкаталоге к нижнему регистру.
- 1.18. Привести все имена файлов в подкаталоге к верхнему регистру.
- 1.19. Подсчитать в подкаталоге число файлов, начинающихся с прописной буквы.
- 1.20. Подсчитать в подкаталоге число файлов, начинающихся со строчной буквы.
- 1.21. Подсчитать в подкаталоге число файлов, начинающихся с точки.
- 1.22. Подсчитать в подкаталоге число файлов, не содержащих в имени точек.
- 1.23. Подсчитать в подкаталоге число файлов, содержащих в имени хотя бы одну точку.
- 1.24. Подсчитать в подкаталоге число файлов, не являющихся подкаталогами.
- 1.25. Подсчитать в подкаталоге число файлов, являющихся подкаталогами.
- 1.26. Найти в подкаталоге файл, содержащий наибольшее количество строк.
- 1.27. Найти в подкаталоге файл, содержащий наименьшее количество строк.
- 1.28. Найти в подкаталоге файл, содержащий наибольшее количество слов.

2. Работа с регулярными выражениями в ОС Linux.

Дан текстовый файл heroes.txt со следующим содержимым:

“Catwoman Batman The Tick Spider Man Black Cat Batgirl Danger Girl Wonder Woman Luke Cage The Punisher Ant Man Dead Girl Aquaman SCUD Spider Woman Blackbolt Martian Manhunter Google Factory Gooogle Goooogle Makswell Computer Dark Yandex Mister Yaanndex Bot Yandexx Wolf Face Score Victory Max Sem Vicictory Mistake Somebody Vicictory”.

- 2.1. Составить регулярное выражение с игнорированием регистра которое выведет все строки за исключением тех, которые содержат подстроку “spider”.
- 2.2. Составить регулярное выражение, которое выведет все строки, начинающиеся с “Bat”.
- 2.3. Составить регулярное выражение, которое выведет все строки, заканчивающиеся на “man”.
- 2.4. Составить регулярное выражение, которое выведет все строки, начинающиеся с bat, Bat, cat или Cat.
- 2.5. Составить регулярное выражение, которое выведет все строки повторения слова “Google”.
- 2.6. Составить регулярное выражение, которое выведет все строки повторения слова “Yandex ”.
- 2.7. Составить регулярное выражение, которое выведет все строки начинающиеся с “Wo”.
- 2.8. Составить регулярное выражение, которое выведет все строки начинающиеся с “Ma”.
- 2.9. Составить регулярное выражение, которое выведет все строки, начинающиеся с bat, Bat или Wo.
- 2.10. Составить регулярное выражение, которое выведет все строки, начинающиеся с bat и заканчивающиеся “man”.
- 2.11. Составить регулярное выражение, которое выведет все строки, заканчивающиеся “man” и строки повторения “Google”.
- 2.12. Составить регулярное выражение, которое выведет все строки, заканчивающиеся “man” и строки повторения “Yandex ”.
- 2.13. Составить регулярное выражение, которое выведет все строки, заканчивающиеся “man”

и строки повторения “Yandex” и “Google” .

2.14. Составить регулярное выражение, которое выведет все строки, начинающиеся с “ bat ” и строки повторения “Yandex” или “Google”.

2.15. Составить регулярное выражение, которое выведет все строки повторения слова “Victory”.

2.16. Составить регулярное выражение, которое выведет все строки повторения слова “Victory” и “Google”.

2.17. Составить регулярное выражение, которое выведет все строки повторения слова “Victory” и все строки, начинающиеся с “bat”, “Bat”.

2.18. Составить регулярное выражение, которое выведет все строки повторения слова “Victory” и “Yandex” а так же все строки, начинающиеся с “bat”, “Bat”.

2.19. Составить регулярное выражение, которое выведет все строки, начинающиеся с “ bat ” и заканчивающиеся на “ry” .

2.20. Составить регулярное выражение, которое выведет две строки повторения слова “Victory”.

2.21. Составить регулярное выражение, которое выведет две строки повторения слова “Google”.

2.22. Составить регулярное выражение, которое выведет две строки повторения слова “ Yandex ”.

2.23. Составить регулярное выражение, которое выведет две строки повторения слова “ Yandex ”.

2.24. Составить регулярное выражение, которое выведет две строки повторения слова “ Yandex ” и “Google”.

2.25. Составить регулярное выражение, которое выведет две строки повторения слова “ Victory” и “Google”.

2.26. Составить регулярное выражение, которое выведет две строки повторения слова “ Victory ” и “ Yandex”.

2.27. Составить регулярное выражение, которое выведет две строки повторения слова “ Victory ” и все строки повторения слова “Google” а так же все слова заканчивающиеся “man”.

2.28. Составить регулярное выражение, которое выведет все строки повторения слова “ Victory ” и все строки повторения слова “Yandex” а так же все слова заканчивающиеся “man” и “Wo”.

3. Написание программ на языке программирования Python.

3.1. Написать программу калькулятор. Калькулятор должен выполнять следующие операции: сложение, вычитание, умножение, деление, возведение в степень.

3.2. Написать программу, которая позволит реализовать различные функции над списками (одномерный список). Длина одномерного списка – 20 элементов. Элементы должны генерироваться случайным образом. Код программы писать в виде функций. Операции: Поиск максимального элемента в списке, поиск минимального элемента в списке, сортировка элементов списка по возрастанию и сортировка элементов списка по убыванию. Данные операции реализовать в отдельных функциях.

3.3. Написать программу, которая позволит реализовать работу с текстовым файлом. Запись и чтение в текстовый файл, редактирование, и поиск по считанным данным из файла.

4. Написание клиент-серверного приложения на языке Python.

4.1. Реализовать серверную часть в операционной системе Windows. Сервер должен возвращать посланные сообщения от клиента.

4.2. Реализовать клиентскую часть в операционной системе Linux. Клиент должен послать сообщение на сервер и ждать от него ответ.

5. Написание программы по работе с CSV-файлами на языке Python.

5.1. На языке программирования Python написать программу, которая будет считывать данные из файла “products_in.csv”, сохранять их в список с именем “csv_list”, отображать содержимое этого списка в консоли. Затем программа должна создать файл “products_out.csv”, сохранить данные из csv_list в файл products_out.csv со следующими заголовками “Номер” для первого столбца, “Вид продукта” для второго столбца, “Количество товара” для третьего столбца.

6. Написание программы записи и чтения команд по СОМ-порту на языке Python.

6.1. Написать программу записи команд по СОМ-порту.

Требования:

- а) Посылаемые команды вводятся пользователем.
- б) Отправить введенную пользователем команду программе чтения.
- в) Программа записи должна завершаться при вводе команды “exit”.

6.2. Написать программу чтения команд с СОМ-порта.

Требования:

- а) Считать команду, посланную от программы записи и отобразить ее в консоли.
- б) Сохранить считанные команды в текстовый файл.
- в) Программа чтения должна завершаться при вводе команды “exit”.

7. Построение цепочки вывода для заданной грамматики.

7.1. Дана грамматика. Построить вывод цепочки $a-b*a+b$.

$$S \rightarrow T \mid T+S \mid T-S$$

$$T \rightarrow F \mid F*T$$

$$F \rightarrow a \mid b$$

8. Построение всех сентенциальных форм для грамматики с правилами.

8.1. Построить все сентенциальные формы для грамматики с правилами:

$$S \rightarrow A+B \mid B+A$$

$$A \rightarrow b$$

$$B \rightarrow a$$

5.3. Фонд оценочных материалов

Полный перечень оценочных материалов представлен в приложении 1.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

5.3. Фонд оценочных материалов

Полный перечень оценочных материалов представлен в приложении 1.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование помещения	Перечень основного оборудования
Учебная аудитория для проведения занятий	Мультимедийное оборудование,

лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	специализированная мебель, наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации.
Помещение для самостоятельной работы обучающихся	Компьютерная техника с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду организации.

6.2. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Р7-Офис.

6.3. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

6.3.1. Основная литература

1. Бражникова Е. В. Аппаратно-программное обеспечение ЭВМ: Методические указания по выполнению лабораторных работ [Электронный ресурс]:. - Москва: РТУ МИРЭА, 2020. - 21 с. – Режим доступа: <https://e.lanbook.com/book/163858>
2. Волкова Е. М. Информационное и программное обеспечение архитектурно-строительной деятельности [Электронный ресурс]: учебное пособие. - Нижний Новгород: ННГАСУ, 2020. - 81 с. – Режим доступа: <https://e.lanbook.com/book/164862>
3. Орещенков И. С. Инструментальные средства разработки программного обеспечения. Система Fossil [Электронный ресурс]: учебное пособие для вузов. - Санкт-Петербург: Лань, 2021. - 284 с. – Режим доступа: <https://e.lanbook.com/book/159492>
4. Иванько А. Ф., Иванько М. А. Системное программное обеспечение информационных мультимедиа систем [Электронный ресурс]:. - Санкт-Петербург: Лань, 2020. - 80 с. – Режим доступа: <https://e.lanbook.com/book/139325>
5. Операционные системы. Программное обеспечение [Электронный ресурс]: учебник. - Санкт-Петербург: Лань, 2020. - 248 с. – Режим доступа: <https://e.lanbook.com/book/131045>
6. Журавлев А. Е., Макшанов А. В., Иванищев А. В. Инфокоммуникационные системы. Программное обеспечение [Электронный ресурс]: учебник для вузов. - Санкт-Петербург: Лань, 2020. - 376 с. – Режим доступа: <https://e.lanbook.com/book/147335>

6.4. РЕКОМЕНДУЕМЫЙ ПЕРЕЧЕНЬ СОВРЕМЕННЫХ ПРОФЕССИОНАЛЬНЫХ БАЗ ДАННЫХ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

1. База данных Web of Science <http://www.webofknowledge.com>
2. iXBT — интернет-издание о компьютерной технике <https://www.ixbt.com>
3. Естественно-научный образовательный портал <http://www.en.edu.ru>
4. Информационный портал Российского научного фонда <http://www.rscf.ru>
5. Научная электронная библиотека <http://www.elibrary.ru>

6.5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Самостоятельная работа студента направлена на подготовку к учебным занятиям и на развитие знаний, умений и навыков, предусмотренных программой дисциплины.

В соответствии с учебным планом дисциплина может предусматривать лекции, практические занятия и лабораторные работы, а также выполнение и защиту курсового проекта (работы). Успешное изучение дисциплины требует посещения всех видов занятий, выполнение заданий преподавателя и ознакомления с основной и дополнительной литературой. В зависимости от мероприятий, предусмотренных учебным планом и разделом 4, данной программы, студент выбирает методические указания для самостоятельной работы из приведенных ниже.

При подготовке к лекционным занятиям студентам необходимо:

перед очередной лекцией необходимо просмотреть конспект материала предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях.

Практические занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности студентов по изучаемой дисциплине.

При подготовке к практическому занятию студенты имеют возможность воспользоваться консультациями преподавателя.

При подготовке к практическим занятиям студентам необходимо: приносить с собой рекомендованную преподавателем литературу к конкретному занятию; до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия; в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения; в ходе семинара давать конкретные, четкие ответы по существу вопросов; на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин), не имеющим письменного решения задач или не подготовившихся к данному практическому занятию, рекомендуется не позже чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, изученную на занятии.

Методические указания, необходимые для изучения и прохождения дисциплины приведены в составе образовательной программы.

6.6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОБУЧЕНИЮ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ограниченными возможностями здоровья.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. № АК-44/05вн) в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Медиа материалы также следует использовать и адаптировать с учетом индивидуальных особенностей обучения лиц с ОВЗ.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается

доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);

- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);

- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);

- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);

- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

