ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.В.ДВ.03.01 «Программирование сетевых устройств»

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) подготовки

Искусственный интеллект и информационные технологии

Квалификация (степень) выпускника - бакалавр

Форма обучения — очная

Рязань, 2025 г.

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной профессиональной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретенных обучающимися в ходе выполнения индивидуальных заданий на лабораторных работах. При оценивании результатов освоения лабораторных работ применяется шкала оценки «зачтено – не зачтено». Количество лабораторных и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

Промежуточный контроль по дисциплине осуществляется проведением теоретического зачета.

Форма проведения зачета – устный ответ по утвержденным вопросам, сформулированным с учетом содержания учебной дисциплины.

Паспорт фонда оценочных средств по дисциплине

№ п/п	**Контролируемые разделы (темы) дисциплины**	**Код контролируемой компетенции (или её части)**	**Вид, метод, форма оценочного мероприятия**
1	Инструментальные средства разработки ПО для встроенных систем	ПК-5.1, ПК-5.2	текущий контроль, зачет
2	Основы сетевого программирования	ПК-5.1, ПК-5.2	текущий контроль, зачет
3	Разработка систем с элементами искусственного интеллекта на языках программирования C/C++	ПК-19.1, ПК-19.2	текущий контроль, зачет

Шкала оценки сформированности компетенций

В процессе оценки сформированности знаний, умений и навыков обучающегося по дисциплине, производимой на этапе промежуточной аттестации в форме теоретического зачета, используется оценочная шкала «зачтено – не зачтено»:

Оценка **«зачтено»** выставляется обучающемуся, который прочно усвоил предусмотренный программный материал; правильно, аргументировано ответил на все вопросы, с приведением примеров; показал глубокие систематизированные знания, владеет приемами рассуждения и сопоставляет материал из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов; без ошибок выполнил практическое задание.

Обязательным условием выставленной оценки является правильная речь в быстром или умеренном темпе. Дополнительным условием получения оценки «зачтено» могут стать хорошие успехи при выполнении самостоятельной и лабораторной работы, систематическая активная работа на практических занятиях.

Оценка **«не зачтено»** выставляется обучающемуся, который не справился с 50% вопросов и заданий при прохождении тестирования, в ответах на другие вопросы допустил существенные ошибки. Не может ответить на дополнительные вопросы, предложенные преподавателем. Целостного представления о взаимосвязях элементов курса и использования предметной терминологии у обучающегося нет. Оценивается качество устной и письменной речи, как и при выставлении положительной оценки.

Типовые контрольные задания или иные материалы

Вопросы к лабораторным занятиям по дисциплине

1. Что собой представляет программа на языке Shell? В каких случаях лучше не использовать язык Shell?
2. В чем особенность переменных в языке Shell? Каким образом переменные определяются в программе? Как осуществляется присвоение значения переменной? Как выполняются операции ввода-вывода?
3. Что собой представляют позиционные и специальные параметры, переменные окружения?
4. В чем особенности использования различных скобок и кавычек в языке Shell? Как можно вычислить значение арифметического выражения в языке Shell?
5. Для чего предназначены команды if и test языка Shell? Перечислите основные параметры команды test.
6. Для чего предназначены команды for, while и until языка Shell? В чем особенности циклов языка Shell по сравнению с другими языками программирования?
7. Что собой представляет компилятор GCC? Какие основные аппаратные платформы и языки программирования им поддерживаются?
8. Что собой представляет процесс компиляции программы на языке C компилятором
9. GCC? Каковы основные этапы данного процесса? Какие получаются промежуточные и конечные результаты?
10. С помощью какой команды выполняется компиляция простейшей программы на языке C? Как выполняется компиляция программы, состоящей из нескольких файлов? Как выполняется компиляция программы, использующей статическую библиотеку?
11. Как выполняется компиляция статических и динамических библиотек?
12. Как при компиляции задать используемый стандарт языка C? Как включить в исполняемый файл отладочную информацию? Какие существуют режимы оптимизации кода в компиляторе GCC? Какие существуют параметры выдачи предупреждений компилятора?

13. Что собой представляет система сборки? Какие основные задачи она решает?

14. Откуда берет исходные данные система сборки Make? Какой используется формат данных? Какой командой запускается сборка? Что означают понятия «цель» и «зависимости» в терминологии утилиты Make? Что такое «абстрактная цель»?

15. В соответствии с каким алгоритмом работает утилита Make?

16. Перечислите основные стандартные имена целей в Make. Для чего предназначена каждая из указанных целей?

17. Что собой представляют переменные в Make? Какие существуют ограничения на их объявление и использование?

18. Что такое сокет? Какие существуют виды сокетов?

19. Перечислите основные функции Linux, которые используются для программирования клиентских потоковых сокетов.

20. Перечислите основные функции Linux, которые используются для программирования серверных потоковых сокетов.

21. Перечислите основные функции Linux, которые используются для программирования клиентских дейтаграммных сокетов.

22. Перечислите основные функции Linux, которые используются для программирования серверных дейтаграммных сокетов.

23. Чем потоковые сокеты отличаются от дейтаграммных?

24. Перечислите основные возможности библиотеки LibTorch/Pytorch.

25. Перечислите основные основные функции библиотеки LibTorch/Pytorch для работы с моделями машинного обучения.

26. Перечислите основные возможности библиотеки TensorFlow.

27. Перечислите основные основные функции библиотеки TensorFlow для работы с моделями машинного обучения.

28. Перечислите основные возможности библиотеки OpenCV.

Типовые задания для самостоятельной работы

Изучение теоретического материала по разделам

1. Понятие встроенной системы. Примеры встроенных систем. Понятие системы реального времени. Встроенные системы на основе Linux.

2. Командные оболочки Linux. Командная оболочка Bash. Язык программирования Shell.

3. Основы работы с компилятором GCC. Компиляция простейшей программы в компиляторе GCC. Структура компилятора GCC: препроцессор, компилятор, ассемблер, компоновщик. Компиляция программ на языке C. Структура компилятора GCC.

4. Понятие о системах сборки. Утилита Make. Введение в написание Make-файлов. Алгоритм работы утилиты Make. Стандартные имена целей.

5. Основные команды CMake. Структура файла CMakeLists.txt. Основные синтаксические конструкции CMake. Подпроекты в CMake.

6. Сетевые сокеты. Понятие сетевого сокета. Потоковые и дейтаграммные сокеты. Клиентские и серверные сокеты. Интернет-сокеты и другие виды сокетов. Пример программы с сетевыми сокетами.

Потоковые сокеты. Основные системные вызовы клиентских и серверных потоковых сокетов. Системный вызов socket(). Системный вызов connect(). Кодирование сетевых адресов. Системный вызов bind(). Системный вызов listen(). Системный вызов accept(). Системные вызовы send() и recv(). Системный вызов close(). Пример программирования потоковых клиента и сервера.

7. Дейтаграммные сокеты. Основные системные вызовы дейтаграммных сокетов. Системные вызовы sendto() и recvfrom(). Пример программирования дейтаграммных клиента и сервера. Параметры сокетов. Системные вызовы getsockopt() и setsockopt(). Пример передачи широковещательных сообщений.

8. Неблокирующий ввод-вывод в сетевых сокетах. Мультиплексирование ввода-вывода в сетевых сокетах.

9. Параметры сокетов. Системные вызовы getsockopt() и setsockopt(). Пример передачи широковещательных сообщений.

10. Использование библиотеки Libtorch/Pytorch в программах на C++. Установка и компиляция библиотеки. Создание модели. Загрузка данных. Обучение и тестирование модели. Экспорт и использование обученной модели.

11. Использование библиотеки TensorFlow в программах на C++. Установка и компиляция библиотеки. Создание модели. Загрузка данных. Обучение и тестирование модели. Экспорт и использование обученной модели.

12. Использование библиотеки OpenCV в программах на C++. Установка и компиляция библиотеки. Основные возможности библиотеки. Пример использования библиотеки.

Вопросы к зачету по дисциплине

1. Командные оболочки Linux. Командная оболочка Bash.
2. Язык программирования Shell. Переменные и выражения. Вввод и вывод данных.
3. Язык программирования Shell. Позиционные переменные. Специальные переменные. Переменные окружения.
4. Язык программирования Shell. Скобки и кавычки в Shell.
5. Язык программирования Shell. Управляющие оперторы. Оператор if.
6. Язык программирования Shell. Управляющие оперторы. Команда test.
7. Язык программирования Shell. Управляющие оперторы. Оператор case.
8. Язык программирования Shell. Управляющие оперторы. Оператор цикла for.
9. Язык программирования Shell. Управляющие оперторы. Операторы цикла while и until.
10. Язык программирования Shell. Функции.
11. Основы работы с компилятором GCC. Компиляция простейшей программы в компиляторе GCC. Компиляция программ на языке C.
12. Структура процесса компиляции программы на языке C в GCC: препроцессор, компилятор, ассемблер, компоновщик.
13. Структура компилятора GCC.
14. Понятие о системах сборки. Утилита Make.
15. Введение в написание Make-файлов. Структура правила.
16. Алгоритм обработки правил в утилите Make.
17. Стандартные имена целей в Make.
18. Понятие сокета. Типы сокетов. Unix-сокеты, Интернет-сокеты, NETLINK-сокеты.
19. Понятие сокета. Типы сокетов. Клиентские и серверные сокеты.
20. Понятие сокета. Типы сокетов. Потоковые и дейтаграммные сокеты.
21. Основные системные вызовы для клиентских потоковых сокетов.
22. Основные системные вызовы для серверных потоковых сокетов.
23. Основные системные вызовы для дейтаграммных сокетов.
24. Блокирующий и неблокирующий ввод/вывод в сокетах.
25. Мультиплексирование ввода/вывода. Функция poll().
26. Мультиплексирование ввода/вывода. Программный интерфейс epoll().
27. Параметры сокетов. Широковещательная передача данных.
28. Основные возможности библиотеки машинного обучения Libtorch/Pytorch.

29. Использование библиотеки машинного обучения Libtorch/Pytorch в программах на языке программирования C/C++.

30. Основные возможности библиотеки машинного обучения Libtorch/Pytorch.

31. Использование библиотеки машинного обучения Libtorch/Pytorch в программах на языке программирования C/C++.

32. Использование библиотеки OpenCV в программах на языке программирования C/C++.

Составил
доцент кафедры САПР ВС
к.т.н., доцент В.А. Шибанов

Заведующий кафедрой САПР ВС,
д.т.н., профессор В.П. Корячко

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**19.10.26** 18:15 (MSK)	Простая подпись
ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**19.10.26** 18:15 (MSK)	Простая подпись