ГАОУ ВО «ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НАРОДНОГО ХОЗЯЙСТВА»

Утверждены решением
Ученого совета ДГУНХ,
протокол № 9
от 24 марта 2025 г.

КАФЕДРА «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

ПО ДИСЦИПЛИНЕ

«СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ И БЕЗОПАСНОСТЬ ПРИЛОЖЕНИЙ»

НАПРАВЛЕНИЕ ПОДГОТОВКИ – 10.03.01
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, ПРОФИЛЬ
«БЕЗОПАСНОСТЬ АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

Уровень высшего образования – бакалавриат

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00C5A9FA0FAD6D6153888A98846B419FD3
Владелец: Бучаев Ахмед Гамидович
Действителен: с 11.07.2024 до 04.10.2025

Махачкала – 2025

УДК 681.518(075.8)
ББК 32.81.73

Составитель – Кобзаренко Дмитрий Николаевич, доктор технических наук, профессор кафедры «Информационные технологии и информационная безопасность» ДГУНХ.

Внутренний рецензент – Галяев Владимир Сергеевич, кандидат физико-математических наук, доцент кафедры «Информационные технологии и информационная безопасность» ДГУНХ.

Внешний рецензент – Абдуллаев Ших-Саид Омаржанович, доктор технических наук, главный научный сотрудник Отдела математики и информатики Дагестанского научного центра Российской академии наук.

Представитель работодателя – Зайналов Джабраил Тажутдинович, директор регионального экспертно-аттестационного центра «Экспертиза», эксперт-представитель работодателя.

Оценочные материалы по дисциплине «Системное программирование и безопасность приложений» разработаны в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 10.03.01 Информационная безопасность, утвержденного приказом Министерства образования и науки Российской Федерации от 17 ноября 2020 г., № 1427, в соответствии с приказом Министерства науки и высшего образования Российской Федерации от 6.04.2021 г. № 245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры»

Оценочные материалы по дисциплине «Системное программирование и безопасность приложений» размещены на официальном сайте www.dgunh.ru

Кобзаренко Д.Н. Оценочные материалы по дисциплине «Системное программирование и безопасность приложений» для направления подготовки 10.03.01 Информационная безопасность, профиль «Безопасность автоматизированных систем». – Махачкала: ДГУНХ, 2025 г. – 37 с.

Рекомендованы к утверждению Учебно-методическим советом ДГУНХ 10 марта 2025 г.

Рекомендованы к утверждению руководителем основной профессиональной образовательной программы высшего образования – программы бакалавриата по направлению подготовки 10.03.01 Информационная безопасность, профиль «Безопасность автоматизированных систем», к.пед.н., Гасановой З.А.

Одобрены на заседании кафедры «Информационные технологии и информационная безопасность» 24 февраля 2025 г., протокол № 7.

.

СОДЕРЖАНИЕ

НАЗНАЧЕНИЕ ОЦЕНОЧНЫХ МАТЕРИАЛОВ.....4

РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины.....5

1.1 Перечень формируемых компетенций.....5

1.2 Перечень компетенций с указанием видов оценочных средств.....5

РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине.....8

РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....27

РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций31

Лист актуализации оценочных материалов по дисциплине.....37

НАЗНАЧЕНИЕ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

Оценочные материалы для текущего контроля успеваемости (оценивания хода освоения дисциплин), для проведения промежуточной аттестации (оценивания промежуточных и окончательных результатов обучения по дисциплине) обучающихся по дисциплине «Системное программирование и безопасность приложений» на соответствие их учебных достижений поэтапным требованиям образовательной программы высшего образования 10.03.01 Информационная безопасность, профиль «Безопасность автоматизированных систем»

Оценочные материалы по дисциплине «Системное программирование и безопасность приложений» включают в себя: перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины; описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания; типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения ОПОП; методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Оценочные материалы сформированы на основе ключевых принципов оценивания:

- валидности: объекты оценки должны соответствовать поставленным целям обучения;
- надежности: использование единообразных стандартов и критериев для оценивания достижений;
- объективности: разные обучающиеся должны иметь равные возможности для достижения успеха.
- Основными параметрами и свойствами оценочных материалов являются:
- предметная направленность (соответствие предмету изучения конкретной дисциплины);
- содержание (состав и взаимосвязь структурных единиц, образующих содержание теоретической и практической составляющих дисциплины);
- объем (количественный состав оценочных материалов);
- качество оценочных материалов в целом, обеспечивающее получение объективных и достоверных результатов при проведении контроля с различными целями.

РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины

1.1 Перечень формируемых компетенций

код компетенции	формулировка компетенции
ПК	**ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ**
ПК-2	Способен учитывать и использовать особенности информационных технологий, применяемых в автоматизированных системах, при организации защиты обрабатываемой в них информации

1.2 Перечень компетенций с указанием видов оценочных средств

Формируемые компетенции	***Код и наименование индикатора достижения компетенции***	***Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций***	***Уровни освоения компетенций***	***Критерии оценивания сформированности компетенций***	***Виды оценочных средств***
ПК-2. Способен учитывать и использовать особенности информационных технологий, применяемых в автоматизированных системах, при организации защиты обрабатываемой в них информации	ИПК-2.2. Обеспечивает безопасность информационных технологий, применяемых в автоматизированных системах	**<u>Знать</u>:** Основные понятия системного программирования, принципы применения системных функций и механизмов обеспечения безопасности приложений.	Пороговый уровень	Обучающийся слабо (частично) знает основные понятия системного программирования, принципы применения системных функций и механизмов обеспечения безопасности приложений.	**Блок А** – задания репродуктивного уровня - тестовые задания; - вопросы для обсуждения (устный опрос).
			Базовый уровень	Обучающийся с незначительными ошибками и отдельными пробелами знает основные понятия системного программирования, принципы	

				применения системных функций и механизмов обеспечения безопасности приложений.	
			Продвинутый уровень	Обучающийся с требуемой степенью полноты и точности знает основные понятия системного программирования, принципы применения системных функций и механизмов обеспечения безопасности приложений.	
		<u>Уметь</u>: Применять знания системных функций и механизмов обеспечения безопасности приложений на практике в различных средах программирования при решении прикладных, и системных задач.	Пороговый уровень	Обучающийся слабо (частично) умеет применять знания системных функций и механизмов обеспечения безопасности приложений на практике в различных средах программирования при решении прикладных, и системных задач.	**Блок В** – задания для реконструктивного уровня - тестовые задания; - лабораторные задания.
			Базовый уровень	Обучающийся с незначительными затруднениями умеет применять знания системных функций и механизмов обеспечения безопасности приложений на практике в различных средах программирования при решении прикладных, и	

				системных задач.	
			Продвинутый уровень	Обучающийся умеет применять знания системных функций и механизмов обеспечения безопасности приложений на практике в различных средах программирования при решении прикладных, и системных задач.	
		<u>Владеть</u>: Навыками программирования системных функций и применения механизмов обеспечения безопасности при решении прикладных, и системных задач.	Пороговый уровень	Обучающийся слабо (частично) владеет навыками программирования системных функций и применения механизмов обеспечения безопасности при решении прикладных, и системных задач.	**Блок С** – задания для практико-ориентированного уровня -индивидуальные проекты.
			Базовый уровень	Обучающийся с незначительными затруднениями владеет навыками программирования системных функций и применения механизмов обеспечения безопасности при решении прикладных, и системных задач.	
			Продвинутый уровень	Обучающийся свободно владеет навыками программирования системных функций и применения механизмов обеспечения безопасности при	

				решении прикладных, и системных задач.	

РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине

Для проверки сформированной компетенции
ПК-2. Способен учитывать и использовать особенности информационных технологий, применяемых в автоматизированных системах, при организации защиты обрабатываемой в них информации.
ИПК-2.2. Обеспечивает безопасность информационных технологий, применяемых в автоматизированных системах

Блок А. Задания репродуктивного уровня («знать»)

А.1 Фонд тестовых заданий по дисциплине

1. В какую категорию WinApi входят объекты, которые используются для создания окон или курсора?
А. Users
Б. Graphics Device Interface
В. Kernel
Г. Windows

2. Что делает функция CloseHandle в случае, если на указанный в ее параметрах дескриптор ссылаются другие дескрипторы
А. Удаляет дескриптор и связанный с ним объект
Б. Удаляет дескриптор и не удаляет связанный с ним объект
В. Удаляет только объект, связанный с дескриптором
Г. Возвращает NULL

3. В каком из нижеперечисленных примеров функцию myfunc можно считать безопасной для применения в многопоточном приложении:

А.
```
int myfunct(int n) {
  if (n > 0) --n;
  if (n < 0) ++n;
  return n
}
```

Б.
```
int n;
int myfunct() {
  if (n > 0) --n;
  if (n < 0) ++n;
  return n
}
```

В.
```
int myfunct () {
  static int n = 0;
  --n;
  return n
}
```

Г.
```
int myfunct () {
  static int k;
  if (k > 0) --k;
  return k
}
```

4. Согласно модели состояний потоков Windows, в какое из состояний не может перейти поток, находящийся в состоянии выполняется?
А. Выбран
Б. Ждет
В. Готов
Г. Завершен

5. Что можно ожидать в случае, если в параметре lpStartAddress при вызове функции CreateThread указать NULL?
А. Создание рабочего потока
Б. Создание системного потока
В. Поток не будет создан
Г. Поток будет создан с защитой по умолчанию

6. Один поток может завершить другой поток с помощью функции:
А. ExitThread
Б. TerminateThread
В. ResumeThread
Г. SuspendThread

7. Если мы с помощью функции CreateProcess создаем консольный процесс, то что мы должны передать в параметр lpStartInfo?
А. NULL.
Б. Ссылку на запись типа STARTUPINFO, заполненную нулями.
В. CREATE_NEW_CONSOLE.
Г. 0.

8. Вызывать функцию TerminateProcess имеет смысл:
А. Если в текущем процессе запущены больше одного потока и их требуется завершить вместе с процессом.
Б. Если мы хотим аварийно завершить другой запущенный процесс.
В. Если мы хотим завершить текущий процесс с одним текущим потоком.
Г. Если мы хотим приостановить выполнение процесса.

9. Чтобы установить высокий приоритет процесса нужно вызвать следующее:
А. SetPriorityClass(hProcess, HIGH_PRIORITY_CLASS)
Б. SetPriorityClass(0)
В. SetPriorityClass(hProcess, THREAD_PRIORITY_HIGHEST)
Г. SetPriorityClass(HIGH_PRIORITY_CLASS)

10. Что из нижеперечисленного нельзя сделать с помощью псевдодескриптора текущего потока:
А. Получить информацию о текущем потоке.

Б. Изменить приоритет текущего потока.
В. Использовать в другом потоке.
Г. Получить настоящий дескриптор текущего потока.

11. Функция AllocConsole возвращает:
А. Дескриптор консоли
Б. Дескриптор ввода-вывода.
В. Логическое значение
Г. Ничего (VOID)

12. Какие флаги в структуре STARTUPINFO задаются в поле dwFillAttributes чтобы установить яркий красный цвет текста?
А. BACKGROUND_RED + BACKGROUND_INTENSITY
Б. FOREGROUND_RED + FOREGROUND_INTENSITY
В. TEXT_RED + TEXT_INTENSITY
Г. FONT_RED + FONT_INTENSITY

13. Сколько способов существует для создания новой консоли в Windows?
А. 1 **Б. 2** В. 3 Г. 4

14. Что возвращает функция GetConsoleWindow в случае успешного завершения?
А. Дескриптор окна консоли
Б. TRUE
В. Дескриптор процесса консоли
Г. Заголовок окна консоли

15. Что возвращает функция GetConsoleTitle?
А. Адрес буфера для получения заголовка окна консоли
Б. Длину строки заголовка в символах
В. Ничего (VOID)
Г. Логическую переменную

16. Что вернет функция GetStdHandle(0)?
А. Дескриптор стандартного ввода
Б. Дескриптор стандартного вывода
В. Значение INVALID_HANDLE_VALUE
Г. Значение FALSE

17. Какой параметр отсутствует в структуре типа type _CONSOLE_SCREEN_BUFFER_INFO?
А. Координаты курсора в символах
Б. Цвет фона и цвет текста
В. Максимальный размер окна
Г. Минимальный размер окна

18. Какой формат функций передачи сообщений при косвенной адресации?
А. send(Process P, сообщение) / receive(Process Q, сообщение)
Б. send(сообщение) / receive (сообщение)
В. send(Connection S, сообщение) / receive(Connection R, сообщение)
Г. send(Process Q) / receive (Connection R)

19. Какие параметры функции CreatePipe являются возвращаемыми? (в данном вопросе множественный выбор)
А. hReadHandle
Б. hWriteHandle
В. lpPipeAttributes
Г. dwSize

20. Для записи данных в анонимный канал используется функция:
А. WriteFile
Б. WritePipe
В. WriteChannel
Г. WriteToPipe

21. Какую константу вернет функция GetExceptionCode, если произошла попытка чтения или записи в виртуальную память без права доступа?
А. EXCEPTION_ACCESS_VIOLATION
Б. EXCEPTION_SINGLE_STEP
В. EXCEPTION_MEMORY_ERROR
Г. EXCEPTION_MEMORY_FAILED

22. Какая функция Windows API позволяет генерировать программное исключение?
А. CreateException
Б. GenerateException
В. GetException
Г. RaiseException

23. Алгоритм замещения станиц, который заключается в том, что первой выталкивается на диск первая из загруженных в реальную память виртуальных страниц, имеет аббревиатуру:
А. FIFO Б. LRU В. LFU Г. NUR

24. Какая API функция запрещает системе выгружать виртуальные страницы из реальной памяти?
А. VirtualAlloc
Б. VirlualLock
В. VirtualProtect
Г. VirtualBlock

25. Пусть в виртуальной памяти со страничной организацией адрес ячейки памяти формируется двухбайтовым словом (WORD). Старшие 6 бит адресуют номер страницы, а остальные – смещение в странице. Каков размер страницы в килобайтах?
Ответ: 1

26. Пусть на диске C присутствует файл «1.dat». Какое значение вернет в переменную hFile строчка программного кода hFile = CreateFile("C:\\1.dat", GENERIC_WRITE, 0, NULL, CREATE_NEW, FILE_ATTRIBUTE_NORMAL, NULL); ?
А. Дескриптор открытого файла
Б. INVALID_HANDLE_VALUE
В. 0
Г. NULL

27. Перечислите в правильной последовательности функции API из следующего списка: CloseHandle, ReadFile, CreateFile, WriteFile, SetFilePionter, GetFileSize, LockFile, GetFileType, GetFileInformationByHandle, которые должны быть использованы для получения размера файла.
Ответ: CreateFile, GetFileSize, CloseHandle,

28. Изменить размер файла можно с помощью API функции:
А. SetEndOfFile
Б. SetFileSize
В. ResizeFile
Г. SetFilePointer

29. Какие из перечисленных ниже API функций не предусматривают предварительное использование функции CreateFile?
А. MoveFile
Б. CopyFile
В. FlushFileBuffers
Г. SetFileAttributes

30. Отметьте API функции, которые могут выполнять действия над каталогами:
А. RemoveDirectory
Б. MoveFile
В. CreateDirectory
Г. DeleteFile

31. Напишите последовательность из 3-х API функций, которые необходимо применить при поиске файлов в текущем каталоге:
Ответ: FindFirstFile, FindNextFile, FindClose

32. Чего не хватает в приведенной последовательности действий для отображения файла в память? Укажите, после какого шага нужно вставить недостающее действие.
- Получение дескриптора объекта, отображающего файл в память (CreateFileMapping)
- Создание вида файла (MapViewOfFile)
- Работа с данными вида файла (чтение/запись)
- Отмена вида файла (UnmapViewOfFile)
- Закрытие дескриптора объекта, отображающего файл в память (CloseHandle)
- Закрытие дескриптора файла (CloseHandle)

Ответ: Получение дескриптора файла (CreateFile)

33. Выберите верные утверждения:
А. Программный модуль, оформленный в виде DLL, хранится на диске в виде файла и может содержать данные, функции и визуальные компоненты.
Б. Для загрузки DLL в память используется механизм отображения файлов в память.
В. DLL позволяют уменьшить объем используемой физической памяти при одновременной работе нескольких приложений, использующих одну и ту же библиотеку.
Г. Процессы совместно разделяют исполняемый код и переменные в DLL.

А.2 Вопросы для обсуждения (устный опрос)

1. Перечислите основные ресурсы операционной системы и коротко охарактеризуйте их.
2. Что представляет собой масштабируемость?
3. Что представляет собой интеграция?
4. Что представляет собой API?
5. Охарактеризуйте историческую линейку ОС Windows.
6. Что представляет собой процесс?
7. Что представляет собой поток?
8. Что такое контекст потока?
9. Какие существуют состояния потока?
10. Чем занимается планировщик потока?
11. Что представляет собой поток Windows?
12. Как происходит управление потоками в Windows?
13. Какие существуют приоритеты в выполнении потоков в Windows?
14. Какие существуют состояния потоков Windows?
15. Что представляют собой процессы в Windows?
16. Как создавать и идентифицировать процесс в Windows?
17. Что представляет собой ожидание завершение процесса в Windows?
18. Какие существуют приоритеты в выполнении процессов в Windows?
19. Какова структура консольного приложения?

20. Какие функции существуют для работы с окном консоли?
21. Какие функции существуют для работы с буфером экрана консоли?
22. Как осуществлять ввод-вывод на консоль на низком уровне?
23. Какие существуют способы передачи данных между процессами?
24. Какие типы связей существуют между процессами?
25. Что такой синхронный и асинхронный обмен данными?
26. Что представляет собой анонимный канал и как его создать в Windows?
27. Как выполнять обмен данными по анонимному каналу?
28. Что такое исключение и для чего его обрабатывать?
29. Каким образом выполняется обработка исключений?
30. Как обрабатываются исключения при выполнении операций с плавающей точкой?
31. Как связаны ошибки исключения?
32. Как работает обработчик завершения?
33. Каковы концепция и организация виртуальной памяти?
34. Какие существуют алгоритмы замещения страниц?
35. Как организована виртуальная память в Windows?
36. Как работать с виртуальной памятью в Windows?
37. Каковы функции файловой системы?
38. Как организована работа с файлами в Windows?
39. Как организована работа с каталогами в Windows?
40. Что представляют собой атрибуты файла и как их изменить?
41. Как получить информацию о файле?
42. Как организовывается поиск в каталоге?
43. Что такое отображение файлов в память?
44. Чем отличаются статические и динамические библиотеки?
45. Чем отличается явное и неявное связывание во время загрузки?
46. Что представляет собой точка входа библиотеки DLL?
47. Как в Windows контролируется доступ к ресурсам?
48. В чем заключаются политика и модель безопасности?
49. В чем заключаются понятия дискреционная модель и дискреционная политика безопасности?
50. Что представляет собой учетная запись Windows?
51. Что представляют собой домены?
52. Что представляют собой группы?
53. Что представляют собой идентификаторы безопасности в Windows?
54. Что представляют собой дескрипторы безопасности в Windows?
55. Что представляют собой маркеры доступа?

Блок В. Задания реконструктивного уровня («уметь»)

В.1 Тестовые задания

1. Какое число будет напечатано на экране консоли при выполнении программного кода на C?

```
int main(int argc, char *argv) {
  HANDLE HThread;
  DWORD ExCode;
  int n, i;
  n = 10; i = 0;
  HThread = GetCurrentThread();
  GetExitCodeThread(HThread, &ExCode);
  while (n > 2) {
    n -= 2;
    i++;
    if (n < 6) {
    printf("%d\n", i);
    Sleep(5000);
    ExitThread(ExCode);
    }
  }
  return 0;
}
```

Ответ: 3

2. В каком виде запустит данный программный код на C программу блокнот?

```
int main(int argc, char *argv) {
  char lpszAppName[] = "c:\\Windows\\notepad.exe";
  STARTUPINFO si;
  PROCESS_INFORMATION piApp;
  ZeroMemory(&si, sizeof(STARTUPINFO));
  si.cb = sizeof(STARTUPINFO);
  si.dwFlags = STARTF_USESHOWWINDOW;
  ZeroMemory(&piApp, sizeof(PROCESS_INFORMATION));
  CreateProcess(lpszAppName, NULL, NULL,
                NULL, FALSE, 0, NULL,
                       NULL, &si, &piApp);
  CloseHandle(piApp.hThread);
  CloseHandle(piApp.hProcess);
  return 0;
}
```

А. В режиме распахнутого во весь экран окна
Б. В режиме окна по умолчанию
В. В режиме минимизированного окна
Г. В режиме скрытого окна

3. Какое число будет напечатано на экране консоли при выполнении программного кода на C?

```
int n;
DWORD D() {
  int i;
  i = 0;
  do {
    n++;
    i++;
    Sleep(500);
  } while (n <= 10);
  printf("%d\n", i);
}
int main(int argc, char *argv) {
  HANDLE HThread;
  DWORD ThreadID;
  n = 0;
  HThread = CreateThread(NULL, 0, D, NULL, 0, &ThreadID);
  Sleep(3750);
  SuspendThread(HThread);
  n += 5;
  ResumeThread(HThread);
  CloseHandle(HThread);
  getchar();
  return 0;
}
```

Ответ: 8

4. Какое число будет напечатано на экране консоли последним при выполнении программного кода на C?

```
const int n = 1000;
DWORD D() {
  int i;
  i = 0;
  while (i < 13) {
    i++;
    printf("%d\n", i);
    Sleep(2*n);
  }
}
int main(int argc, char *argv) {
  HANDLE hThread;
  DWORD ThreadID;
  DWORD ExCode;
  hThread = CreateThread(NULL, 0, D,
```

```
                                NULL, 0, &ThreadID);
  Sleep(3*n);
  GetExitCodeThread(hThread, &ExCode);
  TerminateThread(hThread, ExCode);
  CloseHandle(hThread);
  return 0;
}
```
Ответ: 2

5. Чему будет равно значение переменной s перед выполнением инструкции return в С программе?

```
int main(int argc, char** argv) {
    WORD *a;
    int n = 5;
    a = (WORD*)VirtualAlloc(NULL, sizeof(WORD) * n,
                            MEM_COMMIT, PAGE_READWRITE);
    a[0] = 1;    a[1] = 2;
    a[2] = 3;    a[3] = 4;    a[4] = 5;
    ZeroMemory(a + 1, sizeof(WORD) * 2);
    int s = 0;
    for (int i = 0; i < 5; i++)
        s += a[i];
    VirtualFree(a, sizeof(WORD) * n, MEM_RELEASE);
    printf("%d", s);
    getchar();
    return 0;
}
```
Ответ: 10

6. Чему будет равно значение переменной s перед выполнением инструкции return в С программе?

```
int main(int argc, char** argv) {
    WORD *a;
    a = (WORD*)VirtualAlloc(NULL, sizeof(WORD),
                            MEM_COMMIT, PAGE_READWRITE);
    FillMemory(a, sizeof(WORD), 1);
    int s = a[0];
    printf("%d", s);
    VirtualFree(a, sizeof(WORD), MEM_RELEASE);
    getchar();
    return 0;
}
```
Ответ: 257

7. Чему будет равно значение переменной s перед выполнением инструкции return в C программе?

```
int main(int argc, char** argv) {
  BYTE *a, *b;
  int n = 5;
  a = (BYTE*)VirtualAlloc(
               NULL,
               sizeof(WORD) * n,
               MEM_COMMIT,
               PAGE_READWRITE
               );
  b = (BYTE*)VirtualAlloc(
               NULL,
               sizeof(WORD) * n,
               MEM_COMMIT,
               PAGE_READWRITE
               );
  a[0] = 1; a[1] = 2; a[2] = 3;
  a[3] = 4; a[4] = 5;
  ZeroMemory(b, n);
  CopyMemory(b,  a + 1, 2);
  int s = 0;
  for (int i = 0; i < 5; i++)
    s += b[i];
  VirtualFree(
            a,
            n,
            MEM_RELEASE
            );
  VirtualFree(
            b,
            n,
            MEM_RELEASE
            );
  return 0;
}
```

Ответ: 5

8. Чему будет равно значение переменной s перед выполнением инструкции return в C программе?

```
int main(int argc, char** argv) {
  BYTE *a;
  int n = 5;
  a = (BYTE*)VirtualAlloc(
                NULL,
```

```
                sizeof(WORD) * n,
                MEM_COMMIT,
                PAGE_READWRITE
                );
  a[0] = 1; a[1] = 2; a[2] = 3;
  a[3] = 4; a[4] = 5;
  MoveMemory(a + 1, a, 2);
  int s = 0;
  for (int i = 0; i < 5; i++)
    s += a[i];
  VirtualFree(
           a,
           n,
           MEM_RELEASE
           );
  return 0;
}
```

Ответ: 13

9. Что высветится на экране консоли после запуска следующего программного кода на C?

```
int main(int argc, char** argv) {
  HANDLE hFile;
  DWORD n, s;
  int k = 5, m = 3;
  BYTE b;

  hFile = CreateFile("1.dat", GENERIC_WRITE,
                     0, NULL, CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL,0);
  for (int i = 1; i <= k; i++) {
    b = i + 5;
    WriteFile(hFile, &b, 1, &n, NULL);
  }
  CloseHandle(hFile);
  hFile = CreateFile("1.dat", GENERIC_READ,
                     0, NULL,OPEN_EXISTING,
                     FILE_ATTRIBUTE_NORMAL,0);
  SetFilePointer(hFile, m, NULL, FILE_BEGIN);
  s = 0;
  do {
    ReadFile(hFile, &b, 1, &n, NULL);
    if (n > 0)
     s += b;
  } while (n > 0);
```

```
  CloseHandle(hFile);

  printf("%d", s);
  return 0;
}
```
Ответ: 19

10. Что высветится на экране консоли после запуска следующего программного кода на C?
```
int main(int argc, char** argv) {
  HANDLE hFile;
  DWORD n, s;
  WORD w = 1;
  int k = 2, m = 2;
  BYTE b;

  hFile = CreateFile("1.dat", GENERIC_WRITE,
                     0, NULL, CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL,0);
  for (int i = 1; i <= k; i++)
    WriteFile(hFile, &w, sizeof(WORD), &n, NULL);
  s = GetFileSize(hFile, NULL);
  CloseHandle(hFile);
  CopyFile("1.dat", "2.dat", FALSE);
  hFile = CreateFile("2.dat", GENERIC_WRITE,
                    0, NULL,OPEN_EXISTING,
                    FILE_ATTRIBUTE_NORMAL,0);
  SetFilePointer(hFile, m, NULL, FILE_END);
  SetEndOfFile(hFile);
  s += GetFileSize(hFile, NULL);
  CloseHandle(hFile);

  printf("%d", s);
  return 0;
}
```
Ответ: 10

11. Что высветится на экране консоли после запуска следующего программного кода на C?
```
int main(int argc, char** argv) {
  HANDLE hFile, hMapping;
  BYTE *a, *p;
  DWORD n, s;
  int k = 6, m = 4;
```

```
  a = new BYTE[k];
  for (int i = 0; i < k; i++)
    a[i] = k - i;

  hFile = CreateFile("1.dat", GENERIC_WRITE,
                     0, NULL, CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL, 0);
  WriteFile(hFile, a, k, &n, NULL);
  CloseHandle(hFile);

  ZeroMemory(a, k);

  hFile = CreateFile("1.dat", GENERIC_READ |
                     GENERIC_WRITE,
                     0, NULL,OPEN_EXISTING,
                     FILE_ATTRIBUTE_NORMAL,0);
  hMapping = CreateFileMapping(hFile, NULL,
                    PAGE_READWRITE, 0, 0,NULL);
  p = (BYTE*)MapViewOfFile(hMapping,
                           FILE_MAP_ALL_ACCESS,
                                     0, 0, 0);
  CopyMemory(a, p, m);

  UnmapViewOfFile(p);
  CloseHandle(hMapping);
  CloseHandle(hFile);

  s = 0;
  for (int i = 0; i < k; i++)
    s += a[i];
  printf("%d", s);

  delete [] a;
  return 0;
}
```

Ответ: 18

12. Что высветится на экране консоли после запуска следующего программного кода на C?

```
int main(int argc, char** argv) {
  HANDLE hFile, hFindFile;
  WIN32_FIND_DATA fd;
  int k1 = 10, k2 = 2;
  DWORD n, m = 9;
  WORD w = 1;
```

```
  hFile = CreateFile("2.dat", GENERIC_WRITE,
                     0, NULL, CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL,0);
  for (int i = 1; i <= k1; i++)
    WriteFile(hFile, &w, sizeof(w), &n, NULL);
  CloseHandle(hFile);
  hFile = CreateFile("3.dat", GENERIC_WRITE,
                     0, NULL, CREATE_ALWAYS,
                     FILE_ATTRIBUTE_NORMAL,0);
  for (int i = 1; i <= k2; i++)
    WriteFile(hFile, &w, sizeof(w), &n, NULL);
  CloseHandle(hFile);
  hFindFile = FindFirstFile("*.dat", &fd);

  if (!(hFindFile == INVALID_HANDLE_VALUE)) {
    if (fd.nFileSizeLow > m)
      printf("%s_", fd.cFileName);
    while (FindNextFile(hFindFile,&fd)==TRUE) {
      if (fd.nFileSizeLow > m)
        printf("%s_", fd.cFileName);
    }
  }
  FindClose(hFindFile);

  return 0;
}
```

Ответ: «2.dat»

B.2 Лабораторные задания

Лабораторная работа №1 «Создание многопоточного приложения»

Задание 1:

Изучить, каким образом создавать многопоточное приложение Windows.

Задание 2:

Создать многопоточное приложение Windows, в котором кроме основного потока, существует второй дополнительный поток. В рамках дополнительного потока выполняется сканирование всех папок диска D на предмет подсчета общего объема скрытых файлов и их количества.

Лабораторная работа №2 «Взаимодействие процессов»

Задание 1:

Изучить возможности взаимодействия процессов в Windows.

Задание 2:

Написать два приложения (пусть это будут два процесса), которые способны перехватывать сообщения друг друга и выдавать их на экране монитора. Пользователь может в любой момент послать широковещательное сообщение, которое должно быть тут же перехвачено другим приложением и выведено на экран в виде окна сообщения Windows.

Лабораторная работа №3 «Исключительные ситуации»

Задание 1:

Изучить конструкции языка программирования, которые позволяют обрабатывать исключительные ситуации.

Задание 2:

Написать программу, которая решает систему двух линейных уравнений 1-го порядка и записывает результат в файл. При написании программы учесть обработку следующих исключительных ситуаций: деление на ноль, переполнение регистров при умножении и невозможность записи в файл. При возникновении данных исключительных ситуаций программа должны выдавать сообщения о соответствующих ошибках на русском языке.

Лабораторная работа №4 «Управление памятью»

Задание 1:

Изучить функции управления памятью и функции отображения файлов.

Задание 2:

Написать программу, которая выделяет дополнительную память для хранения 100 мегабайт информации, создает динамический двумерный массив, который адресуется в выделенный участок памяти. С массивом выполняется некоторое действие, например, сортировка данных. Далее выделенный участок памяти освобождается.

Задание 3:

Написать программу, которая выполняет отображение заданного файла в оперативную память. И позволяет в памяти посмотреть его данные.

Лабораторная работа №5 «Работа с файлами с помощью системных функций»

Задание 1:

Получить навыки работы с API-функциями по созданию файла, его удалению и изменению атрибутов. Изучить понятие дескриптора файла.

Задание 2:

Получить навыки работы с API-функциями по чтению данных из файла, записи данных в файл, добавлению данных в конец файла.

Задание 3:

Разработать программу, которая на основе только возможностей API-функций создает файл текстового формата, записывает в него три слова на трех строках и затем считывает построчно данные строки.

Лабораторная работа №6 «Создание динамической библиотеки»

Задание 1:
Изучить, каким образом создавать файл динамической библиотеки (dll) и подключаться к функциям библиотеки из основного программного кода.
Задание 2:
Создать динамическую библиотеку, в которую включить функции: вычисления факториала, решения квадратного уравнения, решение системы двух линейных уравнений 1 порядка. Написать программу, которая позволяет решать задачи, выполняемые функциями написанной dll библиотеки.

Блок С. Задания практикоориентированного уровня для диагностирования сформированности компетенций («владеть»)

С.1 Индивидуальные проекты

Условия выполнения задания по проектам:
1. Программа пишется в любой среде программирования, которая позволяет использовать библиотеку системных функций Windows API, желательно Delphi или Dev C++.
2. Приложение должно быть консольным.
3. В программе допускается использовать функции высокого уровня ТОЛЬКО для ввода-вывода информации. Все остальные функции должны быть системными функциями интерфейса API. Если программа сделана без API функций – она не принимается.

1. Программа для автоматизированного добавления в заголовок файла, его порядкового номера в каталоге.
Дан текущий каталог, в котором находится группа файлов. Требуется добавить ко всем файлам в заголовок порядковый номер и символ «_».
Например,

..		..
Readme.txt	→→→→→→→→→→	01_Readme.txt
Game.dat		02_Game.dat
Докумен		03_Документ1.docx
т1.docx		

2. Программа для автоматической сортировки файлов разных форматов по папкам.
Составить программу, которая в указанном каталоге, на основе расширений найденных файлов создает папки, названные, данным расширением и переписывает в эти папки соответствующие файлы. Например, если имеются файлы: 1.docx, 2.docx, 3.txt. То автоматически создаются две папки DOCX и TXT, первые два файла перемещаются в первую папку, третий файл – во вторую.

3. Программа для автоматического снятия атрибута «скрытый файл».

Составить программу, которая в указанном каталоге находит все скрытые файлы и снимает у них атрибут «скрытый файл». Таким образом, все файлы становятся видимыми.

4. Программа автоматического удаления файлов определенного типа.
Разработать программу, которая на основе задания определенной папки (или целого диска), выполняет поиск файлов определенного типа (по расширению) и автоматически удаляет найденные файлы. При этом на экране высвечивается перечень удаленных файлов

5. Программа поиска наиболее часто встречающегося слова в текстовом файле.
Составить программу, которая в указанном текстовом файле ищет наиболее часто встречающееся слово, если таких слов несколько, то выводится – первое из них. Местоимения, предлоги и союзы из поиска исключаются.

6. Программа поиска текстовых файлов по ключевому слову.
Составить программу, которая по указанному ключевому слову в указанном каталоге (включая подкаталоги), просматривает текстовые файлы и ищет в них наличие ключевого слова. При нахождении ключевого слова в файле его имя добавляется в результирующий список файлов. Цель программы: выдать список найденных файлов.

7. Программа отрисовки однострочного рекламного табло.
С помощью стандартных средств GDI интерфейса Windows написать программу, которая представляет экран монитора как однострочное табло (типа как рекламные табло на уличных магазинах). В программу заносится предложение S, которое циклично визуализируется в виде бегущей строки

8. Программа – светофор.
С помощью стандартных средств GDI интерфейса Windows написать программу, которая на весь экран визуализирует светофор из трех «лампочек» красная, желтая, зеленая. Каждая «лампочка» может быть в состоянии «горит» или «не горит». После запуска программы, все – не горят. Потом по нажатию кнопки «Пуск» светофор активизируется с красного цвета. Далее он работает по временным интервалам, который можно регулировать в программе. Например, для красного 20 секунд, для желтого – 4 секунды, для зеленого – 15 секунд. Светофор из красного в зеленый и обратно переходит только через желтый свет.

9. Программа для форматирования текста в файле.
Дано целое число *K* и текстовый файл, содержащий текст, выровненный по левому краю. Абзацы выделяются в нем с помощью *красной строки* (5 начальных пробелов), а пустых строк нет. Отформатировать текст так, чтобы его ширина не превосходила *K* позиций, и выровнять текст по левому краю, сохранив деление на абзацы. Пробелы в конце строк удалить. Сохранить отформатированный текст в новом текстовом файле.

10. Программа для анализа информации, содержащейся в каталоге.
Разработать программу, которая выполняет анализ указанного каталога Dir (включая подкаталоги) в следующем виде и записывает информацию о нем в текстовый файл в следующем виде:
Общий размер каталога в байтах
Общее количество файлов
Количество файлов без расширения и их объем
Количество файлов с расширением *.* и их объем
Количество скрытых файлов и их объем
Количество файлов только для чтения и их объем

11. Программа для замены символов в группе текстовых файлов.
Разработать программу, которая для всех текстовых файлов заданного каталога (включая подкаталоги) вставляет заданную подстроку StrA вместо указанной подстроки StrB. Если подстрока StrA пустая, то получается удаление подстроки StrB.

12. Программа, объединяющая информацию нескольких текстовых файлов в один файл.
Разработать программу, которая объединяет все текстовые файлы заданного каталога N в единый файл K с сохранением его в этом же каталоге.

Блок Д. Задания для использования в рамках промежуточной аттестации

Д.1 Перечень вопросов на зачет

1. Операционные системы и их интерфейсы. Windows API.
2. Организация потоков и процессов в операционной системе.
3. Управление потоками в Windows.
4. Управление процессами в Windows.
5. Работа с консолью.
6. Передача данных между процессами. Работа с анонимными каналами в Windows.
7. Структурная обработка исключений.
8. Виртуальная память. Работа с виртуальной памятью в Windows.
9. Управление файлами. Работа с файлами и каталогами в Windows.
10. Динамически подключаемые библиотеки.
11. Управление безопасностью в Windows.
12. Управление пользователями и группами в Windows.
13. Работа с идентификаторами и дескрипторами безопасности в Windows.
14. Аспекты безопасности при программировании на языке Python.

РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Балльно-рейтинговая система является базовой системой оценивания сформированности компетенций обучающихся очной формы обучения.

Итоговая оценка сформированности компетенци обучающихся в рамках балльно-рейтинговой системы осуществляется в ходе текущего контроля успеваемости, промежуточной аттестации и определяется как сумма баллов, полученных обучающимися в результате прохождения всех форм контроля.

Оценка сформированности компетенции по дисциплине складывается из двух составляющих:

✓ первая составляющая – оценка преподавателем сформированности компетенции в течение семестра в ходе текущего контроля успеваемости (максимум 100 баллов). Структура первой составляющей определяется технологической картой дисциплины, которая в начале семестра доводится до сведения обучающихся;

✓ вторая составляющая – оценка сформированности компетенции обучающихся на зачете (максимум – 20 баллов).

Для студентов очно-заочной формы обучения применяется 4-балльная и бинарная шкалы оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся.

уровни освоения компетенций	продвинутый уровень	базовый уровень	пороговый уровень	допороговый уровень
100-балльная шкала	85 и ≥	70-84	51-69	0-50
Бинарная шкала	Зачтено			Не зачтено

Шкала оценок при текущем контроле успеваемости по различным показателям

Показатели оценивания сформированности компетенций	**Баллы**	**Оценка**
Устный опрос	0-15	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»
Выполнение лабораторной работы	0-15	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»
Тестирование / контрольная работа	0-30	«неудовлетворительно»

		«удовлетворительно» «хорошо» «отлично»
Выполнение индивидуального проекта	0-10	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»

Соответствие критериев оценивания уровню освоения компетенций по текущему контролю успеваемости

Баллы	Оценка	Уровень освоения компетенций	Критерии оценивания
0-50	«неудовлетворительно»	допороговый уровень	Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины
51-69	«удовлетворительно»	пороговый уровень	Не менее 50% заданий, подлежащих текущему контролю успеваемости, выполнены без существенных ошибок
70-84	«хорошо»	базовый уровень	Обучающимся выполнено не менее 75% заданий, подлежащих текущему контролю успеваемости, или при выполнении всех заданий допущены незначительные ошибки; обучающийся показал владение навыками систематизации материала и применения его при решении практических заданий; задания выполнены без ошибок
85-100	«отлично»	продвинутый уровень	100% заданий, подлежащих текущему контролю успеваемости, выполнены самостоятельно и в требуемом объеме; обучающийся проявляет умение обобщать, систематизировать материал и применять его при решении практических заданий; задания выполнены с подробными пояснениями и аргументированными выводами

Шкала оценок по промежуточной аттестации

Наименование формы промежуточной аттестации	Баллы	Оценка
Зачет	0-20	«не зачтено» «зачтено»

Соответствие критериев оценивания уровню освоения компетенций по промежуточной аттестации обучающихся

Баллы	Оценка	Уровень освоения компетенций	Критерии оценивания
0-9	«не зачтено»	допороговый уровень	Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины; обучающийся не смог ответить на вопросы
10-14	«зачтено»	пороговый уровень	Обучающийся дал неполные ответы на вопросы, с недостаточной аргументацией, практические задания выполнены не полностью, компетенции, осваиваемые в процессе изучения дисциплины сформированы не в полном объеме.
15-17	«зачтено»	базовый уровень	Обучающийся в целом приобрел знания и умения в рамках осваиваемых в процессе обучения по дисциплине компетенций; обучающийся ответил на все вопросы, точно дал определения и понятия, но затрудняется подтвердить теоретические положения практическими примерами; обучающийся показал хорошие знания по предмету, владение навыками систематизации материала и полностью выполнил практические задания
18-20	«зачтено»	продвинутый уровень	Обучающийся приобрел знания, умения и навыки в полном объеме, закрепленном рабочей программой

			дисциплины; терминологический аппарат использован правильно; ответы полные, обстоятельные, аргументированные, подтверждены конкретными примерами; обучающийся проявляет умение обобщать, систематизировать материал и выполняет практические задания с подробными пояснениями и аргументированными выводами

РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций

Тестирование (контрольная работа) выполняется на подготовленных бланков, в которых имеется теоретическая часть – контрольные вопросы с вариантами ответов и практическая часть – задачи с полем ответа для записи результата. Для выполнения тестирования студенту отводится 90 минут времени. За правильные ответы на теоретические вопросы студент получает по 1-2 балла, за решенные задачи 3-5 баллов. Весь тест составляется из расчета максимального получения 30 баллов.

Методика оценивания выполнения тестов

Баллы	Оценка	Показатели	Критерии
25-30	«Отлично»	1. Полнота выполнения тестовых заданий; 2. Своевременность выполнения; 3. Правильность ответов на вопросы; *4.* Самостоятельность тестирования.	Выполнено более 85 % заданий предложенного теста.
19-24	«Хорошо»		Выполнено более 70 % заданий предложенного теста.
13-18	«Удовлетворительно»		Выполнено более 54 % заданий предложенного теста.
0-11	«Неудовлетворительно»		Выполнено не более 53 % заданий предложенного теста

Устный опрос проводится на практических занятиях по пройденному лекционному материалу. Студенту дается вопрос, на который предлагается дать развернутый ответ. Максимально за ответ на вопрос студент может получить 15 баллов.

Методика оценивания ответов на устные вопросы

Баллы	Оценка	Показатели	Критерии
13-15	«Отлично»	1. Полнота данных ответов; *2.* Аргументированность данных ответов; *3.* Правильность отве-	Полно и аргументировано даны ответы по содержанию задания. Обнаружено понимание материала, может обосновать свои суждения, применить знания на практике, при-

		тов на вопросы.	вести необходимые примеры не только по учебнику, но и самостоятельно составленные. Изложение материала последовательно и правильно.
10-12	«Хорошо»		Студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1-2 ошибки, которые сам же исправляет.
6-9	«Удовлетворительно»		Студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.
0-5	«Неудовлетворительно»		Студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

Лабораторные задания выполняются на лабораторных занятиях на основе методического руководства к выполнению лабораторной работы. Студент должен выполнить задание к лабораторной работы, а затем ответить на контрольные вопросы, связанные с выполнением лабораторной работы. Максимально за выполнение лабораторной работы студент может получить 15 баллов.

Методика оценивания выполнения лабораторных заданий

Баллы	Оценка	Показатели	Критерии
13-15	«Отлично»	1. Полнота выполнения лабораторной работы; 2. Своевременность выполнения лабораторной работы; *3.* Правильность выполнения лабораторной работы.	Выполнены все требования к лабораторной работе; разработана, отлажена и протестирована программа; даны правильные ответы на дополнительные вопросы.
10-12	«Хорошо»		Выполнены основные требования к лабораторной работе, имеются недочеты в разработке и тестировании программы; на дополнительные вопросы при защите даны неполные ответы.
6-9	«Удовлетворительно»		Требования к лабораторной работе выполнены не полностью, программа разработана, но в ней имеются существенные недостатки; допущены фактические ошибки при ответе на дополнительные вопросы.
0-5	«Неудовлетворительно»		Лабораторная работа не выполнена; обнаруживается существенное непонимание в ее выполнении.

Индивидуальный проект выполняется студентом в течении всего срока аттестационного периода. Выполнив проект, студент сдает его на проверку преподавателю. Если проект соответствует требованиям задания, то назначается его защита. На защите студент излагает суть проекта и отвечает на вопросы преподавателя относительно выполненной работы. Максимально за выполнение индивидуального проекта студент может получить 10 баллов.

Методика оценивания выполнения индивидуальных проектов

Баллы	Оценка	Показатели	Критерии
9-10	«Отлично»	1. Полнота выполнения проекта; 2. Своевременность выполнения проекта;	Выполнены все требования к выполнения проекта; разработана, отлажена и протестирована программа; даны правильные ответы на дополнительные вопро-

		3. Правильность выполнения проекта.	сы.
7-8	«Хорошо»		Выполнены основные требования к проекту, имеются недочеты в разработке и тестировании программы; на дополнительные вопросы при защите даны неполные ответы.
5-6	«Удовлетворительно»		Требования к проекту выполнены не полностью, программа разработана, но в ней имеются существенные недостатки; допущены фактические ошибки при ответе на дополнительные вопросы.
0-4	«Неудовлетворительно»		Проект не выполнен; обнаруживается существенное непонимание в том, как его выполнять.

Зачет принимается в письменной форме на последнем практическом занятии. В задании к зачету даются два теоретических вопроса и одна задача. Студенту предоставляется 90 минут для письменного изложения ответов на вопросы и решению задачи. За ответы на теоретические вопросы студент максимально может получить 14 баллов. За решение задачи – максимально 6 баллов. Проходной балл на зачете – 12 баллов.

Методика оценивания ответов на зачете

Баллы	**Оценка**	**Показатели**	**Критерии**
12-20	«Зачтено»	*1.* Полнота изложения теоретического материала; *2.* Полнота и правильность решения практического задания; *3.* Правильность и/или аргументированность изложения	Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где обучающийся продемонстрировал знание дисциплины в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно и исчерпывающе отвечает на дополнительные вопросы,

		(последовательность действий); *4.* Самостоятельность ответа т.д.	приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок. Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями. Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры. Допускается несколько ошибок в содержании ответа и решении практических заданий.
0-11	«Не зачте-но»		Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание

			процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы. Выводы поверхностны. Решение практических заданий не выполнено, т.е. студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.

Лист актуализации оценочных материалов по дисциплине «Системное программирование и безопасность приложений»

Оценочные материалы пересмотрены,
обсуждены и одобрены на заседании кафедры

Протокол от «_____»_____ 20___ г. №_____

Зав. кафедрой _____

Оценочные материалы пересмотрены,
обсуждены и одобрены на заседании кафедры

Протокол от «_____»_____ 20___ г. №_____

Зав. кафедрой _____

Оценочные материалы пересмотрены,
обсуждены и одобрены на заседании кафедры

Протокол от «_____»_____ 20___ г. №_____

Зав. кафедрой _____

Оценочные материалы пересмотрены,
обсуждены и одобрены на заседании кафедры

Протокол от «_____»_____ 20___ г. №_____

Зав. кафедрой _____