

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Гарант Максим Алексеевич  
Должность: Ректор  
Дата подписания: 03.11.2023 13:48:45  
Уникальный программный ключ:  
7708e7a47e66a8ee02711b298d7e78bd1e40bf88

Приложение  
к рабочей программе дисциплины

## **ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

### **Системное программирование**

---

*(наименование дисциплины(модуля))*

Направление подготовки / специальность

**09.03.02 Информационные системы и технологии**

---

*(код и наименование)*

Направленность (профиль)/специализация

**Информационные системы и технологии на транспорте**

---

*(наименование)*

**Форма обучения**

Очное

Семестр 4 (зачет)

## Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

## 1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: **зачет в 4 семестре.**

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1 Способен создавать инструментальные средства программирования	ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня
	ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.	Вопросы (1-10)
	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач	Вопросы (8-19)
	Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ	Вопросы (14-25)
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства	Вопросы (20-27)
	Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.	Вопросы (25-38)
	Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.	Вопросы (38-48)

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий;
- 2) выполнение заданий в ЭИОС СамГУПС.

## 2. Типовые<sup>1</sup> контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

### 2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся знает: основные методы и алгоритмы управления ресурсами вычислительных систем; современные инструментальные средства разработки приложений для ПЭВМ; архитектуру и особенности работы современных микропроцессоров в реальном и защищенном режимах; особенности работы с памятью; особенности файловых систем.
<i>Примеры вопросов/заданий</i> <ol style="list-style-type: none"><li>1. Что относится к ресурсам ЭВМ.</li><li>2. Организация памяти ЭВМ.</li><li>3. Оперативная память как ресурс.</li><li>4. Система регистров микропроцессора и их использование в качестве ресурсов.</li><li>5. Система адресации в современных компьютерах.</li><li>6. Процессы.</li><li>7. Потоки.</li><li>8. Алгоритмы работы с памятью.</li><li>9. Алгоритмы планирования потоков.</li><li>10. Диспетчеризация.</li><li>11. Многопоточные приложения.</li></ol>	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся знает: способы разработки кроссплатформенных приложений; методы портирования приложений с одной платформы на другую; базовые принципы и современные методы алгоритмизации, написания программ и автономной отладки при программировании последовательных, параллельных, распределенных приложений, приложений реального времени; современные языки и средства
<i>Примеры вопросов/заданий</i> <ol style="list-style-type: none"><li>1. Понятие о библиотеках функций.</li><li>2. Статические библиотеки.</li><li>3. Динамические библиотеки.</li><li>4. Способы компиляции, сборки и использования статических библиотек.</li><li>5. Способы компиляции, сборки и использования динамических библиотек.</li><li>6. Сборка многофайловых программ.</li><li>7. Основы кроссплатформенной разработки приложений.</li><li>8. Переносимость приложений на уровне исходных кодов.</li><li>9. Использование для разработки приложений кроссплатформенных библиотек.</li><li>10. Способы условной компиляции. Использование препроцессора</li><li>11. Стандартная библиотека функций.</li><li>12. Использование стандартной библиотеки для программирования многопоточных приложений.</li></ol>	

<sup>1</sup> Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

## 2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся умеет: использовать методы и алгоритмы управления ресурсами вычислительных систем; использовать современные инструментальные средства разработки приложений для ПЭВМ; использовать языки и системы программирования для решения профессиональных задач

### *Примеры заданий*

1. Системная программа – это:

- a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации
- b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.
- c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.
- d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

2. Прикладная программа – это

- a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.
- b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.
- c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.
- d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

3. Программный модуль — это

- a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.
- b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.
- c. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.
- d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

4. Объектный модуль — это

- a. программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения.
- b. программный модуль, получаемый в результате трансляции исходного модуля.
- c. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- d. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

5. Программа обслуживания (утилита) — это

- a. программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации.
- b. программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования.

<p>с. программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации.</p> <p>d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.</p>	
ПК-1.1 Разрабатывать программный код на языках программирования низкого уровня	Обучающийся владеет: навыками использования методов и алгоритмов управления ресурсами вычислительных систем; навыками использования современных инструментальных средства разработки приложений для ПЭВМ
<ol style="list-style-type: none"> <li>1. Методы и алгоритмы управления ресурсами вычислительной системы.</li> <li>2. Использование многопоточности в приложениях.</li> <li>3. Управление процессами.</li> <li>4. Управление потоками.</li> <li>5. Компиляция многопоточных приложений.</li> <li>6. Использование командной строки для создания многопоточных приложений.</li> </ol> <p style="text-align: center;"><b>Задание</b></p> <ol style="list-style-type: none"> <li>1. Выполнить проектирование многопоточного приложения используя нотацию UML.</li> <li>2. Произвести компиляцию приложения, используя командную строку.</li> <li>3. При необходимости исправить ошибки и повторить компиляцию.</li> <li>4. Выполнить запуск программы и убедиться в ее работоспособности.</li> </ol>	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся умеет: отлаживать и тестировать системные программы; использовать способы кроссплатформенной разработки приложений.
<ol style="list-style-type: none"> <li>1. Особенности работы с памятью программ. Операторы выделения блоков памяти и удаления выделенных блоков памяти.</li> <li>2. Особенности работы с процессами. Создание процесса. Удаление процесса.</li> <li>3. Особенности работы с потоками. Создание потоков и управление ими.</li> <li>4. Использование средств управления потоками.</li> <li>5. Семафоры и мьютексы.</li> <li>6. Использование API функций операционной системы Windows.</li> <li>7. Использование системных вызовов операционной системы Linux.</li> <li>8. Особенности тестирования системных программ.</li> <li>9. Особенности компиляции программ на уровне ядра операционной системы.</li> </ol>	
ПК-1.2 Осуществлять отладку программ, написанных на языке низкого уровня	Обучающийся владеет: навыками создания кроссплатформенных приложений; инструментальными средствами создания программных библиотек.
<ol style="list-style-type: none"> <li>1. Использование стандартной библиотеки для создания кроссплатформенных многопоточных приложений.</li> <li>2. Ключевые режимы компилятора для создания системных приложений.</li> <li>3. Особенности программирования программных библиотек в операционной среде Windows.</li> <li>4. Особенности программирования библиотек в операционной системе Linux.</li> <li>5. Особенности создания сервисных программ в операционных системах Windows и Linux.</li> <li>6. Вызов библиотечных функций и их использование в программах.</li> <li>7. Компиляция программы с включением в нее отладочной информации.</li> <li>8. Межпроцессное взаимодействие программ.</li> <li>9. Взаимодействие программных потоков.</li> </ol>	

### 2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

1. Назначение, состав и функции ОС.
2. Вычислительные процессы и ресурсы. Виды ресурсов.
3. Диаграмма состояний вычислительного процесса в ОС общего назначения и в ОС реального времени.
4. Понятие процесса Win32. Операции, выполняемые с процессами.
5. Мониторинг процессов и потоков Win32
6. Диспетчеризация потоков в Win32.
7. Потоки в Win32. Операции выполняемые с потоками.
8. Использование потоков при разработке приложений.
9. Квантование времени в MS Windows. Определение величины кванта времени.
10. Независимые и взаимодействующие потоки.
11. Понятие критического участка и критической секции.
12. Средства синхронизации потоков Win32.
13. Задача поставщик-потребитель и ее решение с помощью операций P(S) и V(S).
14. Объекты ядра MS Windows, их создание и использование.
15. Мьютексы, как средство синхронизации потоков.
16. Семафоры и их использование.
17. События и их использование.
18. Функции ожидания одного и нескольких событий
19. Сегментная и страничная адресация памяти. Сравнительный анализ
20. Дескрипторные таблицы. Схема вычисления физического адреса в режиме 286.
21. Формат дескрипторов в защищенных режимах 286 и 386.
22. Байт доступа. Поле доступа. Назначение битов.
23. Условие доступа программы к сегменту данных.
24. Реализация защиты ОС от прикладных программ.
25. Кольца защиты. Шлюзы вызова и задачи.
26. Аппаратная поддержка многозадачности. Сегмент состояния задачи.
27. Обработка прерываний в защищенном режиме.
28. Модель Холта.
29. Понятие тупика. Пример тупика процессов, использующих семафоры.
30. Необходимые условия возникновения тупика.
31. Методы борьбы с тупиками.
32. Управление страничной памятью. Стратегии подкачки и рабочие наборы страниц.
33. Разделы в виртуальном адресном пространстве процесса. Адресное пространство процесса MS Windows.
34. Управление памятью с помощью AWE.
35. Получение информации о состоянии виртуальной памяти
36. Резервирование регионов в адресном пространстве и передача физической памяти региону.  
Освобождение регионов
37. Кучи. Необходимость создания дополнительных куч. Операции с кучами.
38. Файлы, проецируемые в память
39. Динамически загружаемые библиотеки. Области применения.
40. Основные DLL MS Windows. Достоинства и недостатки DLL.
41. Использование DLL - импорт функций из DLL.
42. Явная и неявная загрузка DLL.
43. Эволюция архитектуры файловых систем.
44. Файловые системы Win32.
45. Структура магнитного диска.
46. Файловая система FAT.
47. Файловая система NTFS.
48. Основные отличия FAT и NTFS.

### 3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

#### Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90 % от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76 % от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60 % от общего объёма заданных вопросов.

#### Критерии формирования оценок по результатам выполнения заданий

**«Отлично/зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов.

**«Хорошо/зачтено»** – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

**«Удовлетворительно/зачтено»** – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

**«Неудовлетворительно/не зачтено»** – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

*Виды ошибок:*

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*
- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*
- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

#### Критерии формирования оценок по результатам выполнения лабораторных работ

**«Зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием, выданным для выполнения лабораторной работы. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

**«Не зачтено»** - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при расчетах, сформулировал неверные выводы по результатам работы.

#### Критерии формирования оценок по результатам выполнения практических работ

**«Зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

**«Не зачтено»** - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при расчетах, сформулировал неверные выводы по результатам работы.

#### Критерии формирования оценок по зачету

**«Зачтено»** - обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при



ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности.

**«Не зачтено»** - выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.