

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаранн Максим Алексеевич
Должность: Ректор
Дата подписания: 03.11.2023 13:43:07
Уникальный программный ключ:
7708e7a47e66a8ee02711b298d7e78bd1e40bf88

Приложение
к рабочей программе дисциплины

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Компонентное 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

Очная

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 - 4 семестр.**

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1: Способен создавать инструментальные средства программирования	ПК-1.3, ПК-1.4

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

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы (семестр _)
ПК-1.3: Разрабатывает программный код на языках программирования высокого уровня	Обучающийся знает: основные принципы компонентно-ориентированного программирования и их место в программной инженерии в целом; методы определения и использования программных компонент;	Вопросы (1 - 10)
	Обучающийся умеет: применять методы компонентно-ориентированного программирования при проектировании программного обеспечения; выделять компоненты при проектировании программных приложений и систем; реализовывать программные компоненты в соответствии с общими компонентными моделями;	Задания
	Обучающийся владеет: навыками использования и композиции различных программных компонент при проектировании современных программных систем; навыками использования инструментальных средств, применяемых при компонентно-ориентированном программировании;	Задания
ПК-1.4: Осуществляет отладку программ, написанных на языке высокого уровня	Обучающийся знает: распространенные технологии компонентно-ориентированного программирования и соответствующие инструментальные средства;	Вопросы (11 - 20)
	Обучающийся умеет: оценивать преимущества и недостатки конкретных компонентных моделей и соответствующих им компонент при решении различных практических задач; использовать существующие компонентные модели и имеющиеся для них наборы программных компонент при решении задач программной инженерии; ориентироваться в потоке научной информации для изучения и использования новых технологий программирования;	Задания
	Обучающийся владеет: навыками самостоятельного научного исследования в области разработки современных программных систем и инженерных методов их построения; навыками практической реализации результатов научного исследования. навыками выступлений с научными докладами.	Задания

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

- 1) проводится в форме устного ответа на вопросы из перечня для подготовки обучающихся к промежуточной аттестации;
- 2) выполнение заданий в ЭИОС СамГУПС.

2. Типовые¹ 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-1.3: Разрабатывает программный код на языках программирования высокого уровня	Обучающийся знает: основные принципы компонентно-ориентированного программирования и их место в программной инженерии в целом; методы определения и использования программных компонент;
<p>Алгоритмический подход к программированию концентрирует внимание на:</p> <p>A. Последовательности действий</p> <p>B. Интерфейсе</p> <p>C. Наследовании свойстве</p> <p>D. Файловой системе.</p> <p>Взаимодействие программных объектов при объектно – ориентированном программировании (ООП) осуществляется:</p> <p>A. На базе рабочей группы предприятия.</p> <p>B. В иерархических сетях.</p> <p>C. В составе СУБД Microsoft Access.</p> <p>D. Путем передачи сообщений.</p> <p>В ООП используются виды иерархии:</p> <p>A. Часть -- целое.</p> <p>B. Общее -- частное.</p> <p>C. Справедливы оба случая.</p> <p>D. В составе территориально разнесенных узлов и сетей.</p> <p>Основным недостатком ООП является :</p> <p>A. Снижение быстродействия(*)</p> <p>B. Невозможность применения при большом объеме программы</p> <p>C. Невозможность применения при малом объеме программы</p> <p>D. Необходимость разработки развитых средств интерфейса</p> <p>Позднее связывание объектов реализовано в интегрированной среде:</p> <p>A. Delphi всех версий</p> <p>B. C Builder всех версий</p> <p>C. Pascal 7.0</p> <p>D. Реализовано во всех случаях</p> <p>Этапы логического и физического проектирования отличаются учетом:</p> <p>A. типа операционной системы и используемого оборудования</p> <p>B. механизма сокрытия деталей реализации</p> <p>C. разделения времени</p> <p>D. позднего связывания</p> <p>Разработка программного обеспечения с учетом требований ООП НЕ включает этап:</p> <p>A. анализ и уточнение спецификаций</p> <p>B. реализации</p> <p>C. типизации</p> <p>D. модернизации</p> <p>Объектной декомпозицией называют:</p> <p>A. анализ и уточнение спецификаций</p> <p>B. представление предметной области в виде объектов</p> <p>C. моделирование предметной области</p> <p>D. определение приоритетов сообщений между объектами</p> <p>Сложная предметная область декомпозируется с использованием:</p> <p>A. анализа и уточнения спецификаций</p> <p>B. представления предметной области в виде объектов</p> <p>C. контекстной диаграммы классов</p> <p>D. абстрагирования и установления ассоциаций</p>	

¹ 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.

Над объектом НЕ совершается действие: А. переработки В. создания С. итерации D. селекции	
ПК-1.4: Осуществляет отладку программ, написанных на языке высокого уровня	Обучающийся знает: распространенные технологии компонентно-ориентированного программирования и соответствующие инструментальные средства;
<p>11. Конструированием объекта называется: А. Верификация и разбиение полей В. Проверка и добавление записей С. Тестирование и завершение моделирования D. Создание и инициализация полей</p> <p>12. Деструкцией объекта называется: А. формирование на каждом этапе полного комплекта документации В. уничтожение объекта С. обсуждение вариантов использования объекта</p> <p>13. Механизм наследования позволяет: А. конструировать новые классы из имеющихся В. задавать поведение объекта С. интерпретировать коды функций</p> <p>14. Позднее связывание реализуется при помощи методов: А. динамических и виртуальных В. статических С. наследуемых от класса -- родителя</p> <p>15. При объявлении класса в среде Delphi скрытые элементы класса определяются через ключевое слово: А. published В. protected С. private</p> <p>16. Абстрактными называют классы, имеющие в своем составе: А. Статические методы В. Методы с отложенной реализацией С. Динамические и виртуальные методы</p> <p>17. В C# для работы с метаклассами используются: А. Механизм ссылок В. Структуры данных С. Динамические массивы</p> <p>18. Создание нового сообщения в C# НЕ требует выполнения: А. Описание типа сообщения В. Объявление индекса (номера) сообщения С. Объявление класса объекта – получателя сообщения</p> <p>19. Основными концепциями объектно – ориентированного программирования являются: А. Инкапсуляция, наследование и полиморфизм В. Алгоритмы, иерархии объектов и IP – адресация С. Исходный, объектный и исполняемый модули</p> <p>20. Инкапсуляция – это: А. Декомпозиция данных с методами В. Комбинирование данных с процедурами и функциями С. Объявление переменных внутри класса</p>	

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.3: Разрабатывает программный код на языках программирования высокого уровня	Обучающийся умеет: применять методы компонентно-ориентированного программирования при проектировании программного обеспечения; выделять компоненты при проектировании программных приложений и систем; реализовывать программные компоненты в соответствии с общими компонентными моделями; Обучающийся владеет: навыками использования и композиции различных программных компонент при проектировании современных программных систем; навыками использования инструментальных средств, применяемых при компонентно-ориентированном программировании;
Задача 1. Подсчет слов в тексте. Написать программу, которая вычислит и напечатает количество слов в заданном тексте	

(взяв текст из окна и выведя его в текстовый файл). Текст состоит из последовательности слов, разделённых пробелами. Слово содержит произвольные символы (буквы), отличные от пробела. Слово не является частью другого слова и не содержит в себе другие слова. За исходный текст взять алгоритм, представленный ниже.

Алгоритм.

ЦИКЛ до конца текст – будет найдено и обработано ровно одно слово

НАЙТИ начало слова (в цикле пропустить все пробелы)

ЕСЛИ достигли конца текста, ТО ВЫХОД из цикла

ЗАПОМНИТЬ позицию начала слова

НАЙТИ конец слова (в цикле пропустить все не пробелы)

ЗАПОМНИТЬ позицию конца слова

ОБРАБОТАТЬ слов – зависит от конкретной задачи

КОНЕЦ_ЦИКЛА

ПЕЧАТЬ результатов

Задача 2. Большие и малые буквы в тексте. Взять текст из текстового файла. Заменить в этом тексте все строчные буквы на прописные и наоборот. Показать исходный текст и полученный текст в окнах на форме. Далее вывести полученный текст в текстовый файл. Исходные данные: Пример Простого Текста Результат: пРИМЕР пРОСТОГО тЕКСТА

Задача 3. Удаление лишних пробелов в тексте.

Дан текст, состоящий из слов. Напечатать его, оставляя между словами ровно по одному пробелу.

Исходные данные: Пример простого текста

Результат: Пример простого текста

ПК-1.4: Осуществляет отладку программ, написанных на языке высокого уровня

Обучающийся умеет: оценивать преимущества и недостатки конкретных компонентных моделей и соответствующих им компонент при решении различных практических задач; использовать существующие компонентные модели и имеющиеся для них наборы программных компонент при решении задач программной инженерии; ориентироваться в потоке научной информации для изучения и использования новых технологий программирования;

Обучающийся владеет: навыками самостоятельного научного исследования в области разработки современных программных систем и инженерных методов их построения; навыками практической реализации результатов научного исследования. навыками выступлений с научными докладами.

Задача 4. Проверка правильности математической формулы.

Дана символьная строка, содержащая некоторую формулу с круглыми скобками. Написать программу, которая напечатает «ОК», если скобки расставлены правильно, и «НЕТ» в противном случае.

Исходные данные: (a+v)*(c-)

Результат: ОК

Задача 5. Транспонирование матрицы.

Дана квадратная матрица целых чисел порядка n. Транспонировать её относительно главной диагонали и напечатать. Другие массивы не использовать. Значения элементов не превосходят 105

. Вывести исходную и

транспонированную матрицы на консоль и в текстовый файл.

Задача 6. Определитель матрицы.

Написать программу, которая будет вычислять определитель матрицы порядка N. В консоли задаётся размер матрицы N, и сама матрица.

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

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 всего задания, использовал при выполнении неправильные алгоритмы, допустил грубые ошибки при программировании, сформулировал неверные выводы по результатам 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 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.