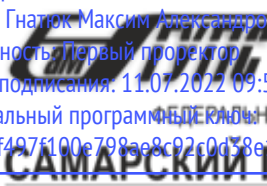


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гнатюк Максим Александрович
Должность: Первый проректор
Дата подписания: 11.07.2022 09:51:21
Уникальный программный ключ:
8873f497f100e798ae8c92c0d38e105c818d5410

 **МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

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)

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

3 семестр

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>А. Последовательности действий В. Интерфейсе С. Наследовании свойстве D. Файловой системе.</p> <p>Взаимодействие программных объектов при объектно – ориентированном программировании (ООП) осуществляется:</p> <p>А. На базе рабочей группы предприятия. В. В иерархических сетях. С. В составе СУБД Microsoft Access. D. Путем передачи сообщений.</p> <p>В ООП используются виды иерархии:</p> <p>А. Часть -- целое. В. Общее -- частное. С. Справедливы оба случая. D. В составе территориально разнесенных узлов и сетей.</p> <p>Основным недостатком ООП является :</p> <p>А. Снижение быстродействия(*) В. Невозможность применения при большом объеме программы С. Невозможность применения при малом объеме программы D. Необходимость разработки развитых средств интерфейса</p> <p>Позднее связывание объектов реализовано в интегрированной среде:</p> <p>А. Delphi всех версий В. С Builder всех версий С. Pascal 7.0 D. Реализовано во всех случаях</p> <p>Этапы логического и физического проектирования отличаются учетом:</p> <p>А. типа операционной системы и используемого оборудования В. механизма сокрытия деталей реализации С. разделения времени D. позднего связывания</p> <p>Разработка программного обеспечения с учетом требований ООП НЕ включает этап:</p> <p>А. анализ и уточнение спецификаций В. реализации С. типизации D. модернизации</p> <p>Объектной декомпозицией называют:</p> <p>А. анализ и уточнение спецификаций В. представление предметной области в виде объектов С. моделирование предметной области D. определение приоритетов сообщений между объектами</p> <p>Сложная предметная область декомпозируется с использованием:</p> <p>А. анализа и уточнения спецификаций В. представления предметной области в виде объектов С. контекстной диаграммы классов D. абстрагирования и установления ассоциаций</p> <p>Над объектом НЕ совершается действие:</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 несет заведующий кафедрой и преподаватель – разработчик оценочных средств.

<p>A. переработки B. создания C. итерации D. селекции</p>	
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: A. Верификация и разбиение полей B. Проверка и добавление записей C. Тестирование и завершение моделирования D. Создание и инициализация полей 12. Деструкцией объекта называется: A. формирование на каждом этапе полного комплекта документации B. уничтожение объекта C. обсуждение вариантов использования объекта 13. Механизм наследования позволяет: A. конструировать новые классы из имеющихся B. задавать поведение объекта C. интерпретировать коды функций 14. Позднее связывание реализуется при помощи методов: A. динамических и виртуальных B. статических C. наследуемых от класса -- родителя 15. При объявлении класса в среде Delphi скрытые элементы класса определяются через ключевое слово: A. published B. protected C. private 16. Абстрактными называют классы, имеющие в своем составе: A. Статические методы B. Методы с отложенной реализацией C. Динамические и виртуальные методы 17. В C# для работы с метаклассами используются: A. Механизм ссылок B. Структуры данных C. Динамические массивы 18. Создание нового сообщения в C# НЕ требует выполнения: A. Описание типа сообщения B. Объявление индекса (номера) сообщения C. Объявление класса объекта – получателя сообщения 19. Основными концепциями объектно – ориентированного программирования являются: A. Инкапсуляция, наследование и полиморфизм B. Алгоритмы, иерархии объектов и IP – адресация C. Исходный, объектный и исполняемый модули 20. Инкапсуляция – это: A. Декомпозиция данных с методами B. Комбинирование данных с процедурами и функциями C. 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	<p>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;</p> <p>Обучающийся владеет: навыками использования и композиции различных программных компонент при проектировании современных программных систем; навыками использования инструментальных средств, применяемых при компонентно-ориентированном программировании;</p>
<p>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.</p>	

<p>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.</p> <p>Алгоритм.</p> <p>ЦИКЛ до конца текст – будет найдено и обработано ровно одно слово</p> <p>НАЙТИ начало слова (в цикле пропустить все пробелы)</p> <p>ЕСЛИ достигли конца текста, ТО ВЫХОД из цикла</p> <p>ЗАПОМНИТЬ позицию начала слова</p> <p>НАЙТИ конец слова (в цикле пропустить все не пробелы)</p> <p>ЗАПОМНИТЬ позицию конца слова</p> <p>ОБРАБОТАТЬ слов – зависит от конкретной задачи</p> <p>КОНЕЦ_ЦИКЛА</p> <p>ПЕЧАТЬ результатов</p> <p>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</p> <p>Задача 3. Удаление лишних пробелов в тексте.</p> <p>Дан текст, состоящий из слов. Напечатать его, оставляя между словами ровно по одному пробелу.</p> <p>Исходные данные: Пример простого текста</p> <p>Результат: Пример простого текста</p>	
<p>ПК-1.4: Осуществляет отладку программ, написанных на языке высокого уровня</p>	<p>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;</p> <p>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.</p>
<p>Задача 4. Проверка правильности математической формулы.</p> <p>Дана символьная строка, содержащая некоторую формулу с круглыми скобками. Написать программу, которая напечатает «ОК», если скобки расставлены правильно, и «НЕТ» в противном случае.</p> <p>Исходные данные:(a+v)*(c-)</p> <p>Результат:ОК</p> <p>Задача 5. Транспонирование матрицы.</p> <p>Дана квадратная матрица целых чисел порядка n. Транспонировать её относительно главной диагонали и напечатать. Другие массивы не использовать. Значения элементов не превосходят 105</p> <p>. Вывести исходную и транспонированную матрицы на консоль и в текстовый файл.</p> <p>Задача 6. Определитель матрицы.</p> <p>Написать программу, которая будет вычислять определитель матрицы порядка N. В консоли задаётся размер матрицы N, и сама матрица.</p>	

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. Deskрипторные 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.