

Приложение 2 к РПД
Клиент-серверные технологии разработки
программного обеспечения
09.03.01 Информатика и вычислительная техника
Направленность (профиль)
Виртуальные технологии и дизайн
Форма обучения – очная
Год набора – 2022

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

1. Общие сведения

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	09.03.01 Информатика и вычислительная техника
3.	Направленность (профиль)	Виртуальные технологии и дизайн
4.	Дисциплина (модуль)	Б1.В.ДВ.04.01 Клиент-серверные технологии разработки программного обеспечения
5.	Форма обучения	Очная
6.	Год набора	2022

2. Перечень компетенций

– ПК-1: Способен к разработке и применению алгоритмических и программных решений в области прикладного программного обеспечения
--

3. Критерии и показатели оценивания компетенций на различных этапах их формирования

Этапы формирования компетенций (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
Введение в дисциплину	ПК-1	-основы новых информационных технологий для разработки клиент-серверного ПО	-анализировать архитектуру предприятия с помощью приложений архитектур клиент-сервисов	-основными клиент-серверными технологиями	Участие в дискуссии
Понятие клиент-серверного ПО	ПК-1	-особенности разработки, организации, дистрибуции и монетизации клиент-серверного ПО	-теоретически обосновать необходимость применения ИТ и выбор для решения конкретных профессиональных задач	- навыками использования ИС и ИКТ-решений для разработки ПО	Выполнение Лабораторных работ, Контрольное тестирование
Инструментарий разработки клиент-серверного ПО	ПК-1	-IDE написания программного кода -системы управления проектом -системы коммуникации команды -системы контроля версий	-пользоваться специализированными программными средствами для разработки клиент-серверного ПО	-навыками работы с системами контроля версий	Выполнение Лабораторных работ, Контрольное тестирование
Применение ООП технологий в рамках клиент-серверной разработки ПО	ПК-1	-основы технологии объектно-ориентированной декомпозиции программных систем -базовые шаблоны проектирования -отношения между классами	-использовать принципы ООП при создании клиент-серверного ПО	-принципами построения объектно-ориентированных программных систем на C++	Выполнение Лабораторных работ, Контрольное тестирование
Решение задач программной визуализации при разработке клиент-серверного ПО	ПК-1	-основные методы компьютерной геометрии -основы векторной и растровой графики -технологии реализации алгоритмов компьютерной графики	-программно реализовывать основные алгоритмы растровой и векторной графики -использовать графические стандарты и библиотеки	-методологией решения задач программной визуализации	Выполнение Лабораторных работ, Контрольное тестирование
Технологии	ПК-1	-основные NoSQL	-оптимизировать работу	-методами оптимизации	Выполнение

Этапы формирования компетенций	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
разработки серверной компоненты клиент-серверного ПО		технологии -архитектуру высоконагруженных систем	клиентских приложений с распределёнными базами данных	sql запросов -технологиями NoSQL	Лабораторных работ, Контрольное тестирование
Методики обеспечения и контроля качества клиент-серверного ПО	ПК-1	- основные понятия, категории и подходы к управлению качеством ПО	-использовать методы анализа и управления качеством	-основными инструментами управления качеством, информационными технологиями в обеспечении качества	Выполнение Лабораторных работ, Контрольное тестирование
Продуцирование текстового контента клиент-серверных программных продуктов.	ПК-1	-критерии уникальности текстового контента	-разрабатывать текстовый контент в ПО	-методологией локализации и интернационализации ПО	Выполнение Лабораторных работ, Контрольное тестирование
Механизмы серверного конфигурирования клиент-серверного ПО.	ПК-1	-методы конфигурации клиент-серверного ПО	-конфигурировать клиент-серверное ПО	-навыками создания конфигураций клиент-серверного ПО	Выполнение Лабораторных работ, Контрольное тестирование

Шкала оценивания в рамках балльно-рейтинговой системы

«неудовлетворительно» – 60 баллов и менее;

«удовлетворительно» – 61-80 баллов

«хорошо» – 81-90 баллов

«отлично» – 91-100 баллов

4. Критерии и шкалы оценивания

1. Тест по разделу дисциплины

Процент правильных ответов	До 60	61-80	81-100
Количество баллов за решенный тест	0	6	12

2. Критерии оценки выполнения лабораторной работы

- 6 баллов выставляется, если студент вовремя и полностью выполнил задание на лабораторную работу, правильно и полностью описал и изложил необходимые результаты в отчете, аргументировав их на защите лабораторной работы.
- 5 балла выставляется, если студент выполнил полностью задание на лабораторную работу, правильно описал и изложил необходимые результаты в отчете, аргументировав их на защите лабораторной работы, но задержал сдачу работы на одну неделю.
- 4 балла выставляется, если студент выполнил полностью задание на лабораторную работу, правильно описал и изложил необходимые результаты в отчете, аргументировав их на защите лабораторной работы, но задержал сдачу работы на две недели.
- 3 балла выставляется, если студент выполнил полностью задание на лабораторную работу, правильно описал и изложил необходимые результаты в отчете, аргументировав их на защите лабораторной работы, но задержал сдачу работы более чем три недели.
- 0 баллов - если студент не выполнил задания и/или предоставил отчет.

3. Критерии оценки выступления с презентацией (доклад, реферат)

Характеристика выступления с презентацией	количество баллов
Содержание	
• Сформулирована цель работы	0,5
• Понятны задачи и ход работы	0,5
• Информация изложена полно и четко	0,5
• Иллюстрации усиливают эффект восприятия текстовой части информации	0,5
• Сделаны выводы	0,5
Оформление презентации	
• Единый стиль оформления	0,5
• Текст легко читается, фон сочетается с текстом и графикой	0,5
• Все параметры шрифта хорошо подобраны, размер шрифта оптимальный и одинаковый на всех слайдах	0,5
• Ключевые слова в тексте выделены	0,5
Эффект презентации	
• Общее впечатление от просмотра презентации	0,5
Мак количество баллов	5

4. Критерии оценки разработки и защиты проекта

Характеристики работы студента	количество баллов
- студент глубоко и всесторонне усвоил проблему; - уверенно, логично, последовательно и грамотно его излагает; - опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью; - умело обосновывает и аргументирует выдвигаемые им идеи; - делает выводы и обобщения; - свободно владеет понятиями	20
- студент твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы;	15

Характеристики работы студента	количество баллов
<ul style="list-style-type: none"> - не допускает существенных неточностей; - увязывает усвоенные знания с практической деятельностью; - аргументирует научные положения; - делает выводы и обобщения; - владеет системой основных понятий 	
<ul style="list-style-type: none"> - тема раскрыта недостаточно четко и полно, то есть студент освоил проблему, по существу излагает ее, опираясь на знания только основной литературы; - допускает несущественные ошибки и неточности; - испытывает затруднения в практическом применении знаний; - слабо аргументирует научные положения; - затрудняется в формулировании выводов и обобщений; - частично владеет системой понятий 	10
<ul style="list-style-type: none"> - студент не усвоил значительной части проблемы; - допускает существенные ошибки и неточности при рассмотрении ее; - испытывает трудности в практическом применении знаний; - не может аргументировать научные положения; - не формулирует выводов и обобщений; - не владеет понятийным аппаратом 	0

5. Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

5.1 Тестовое задание по теме «Применение ООП технологий в рамках клиент-серверной разработки ПО»

1. Базовым понятием объектно-ориентированного программирования является:

- а) метод;
- б) поле;
- в) ориентация;
- г) **объект.**

2. Наследование, это:

- а) реализация только одной идеи устранения дублирования данных;
- б) особое отношение между объектами одного класса;
- в) **такое отношение между классами, когда один класс повторяет структуру и поведение другого класса.**

3. Механизм, который объединяет данные и методы, манипулирующие этими данными, и защищает и то и другое от внешнего вмешательства или неправильного использования, это:

- а) наследование;
- б) полиморфизм;
- в) **инкапсуляция;**
- г) абстракция данных.

4. Класс, который получается в результате наследования, называется:

- а) родительский;
- б) основной;
- в) **порожденный.**

5. Класс может использоваться для создания:

- а) одного объекта;

- б) двух объектов;
- в) пяти объектов;
- г) более пяти объектов.

6. Основными разделами класса являются:

- а) public, protected, virtual;
- б) **private, public, protected;**
- в) private, public, friend;
- г) private, protected, overload.

7. Конструкторы бывают следующих типов:

- а) конструкторы по умолчанию, встраиваемый конструктор;
- б) **конструкторы по умолчанию, конструктор копирования;**
- в) конструктор копирования, встраиваемый конструктор;
- г) конструкторы по умолчанию, конструктор копирования, встраиваемый конструктор.

8. Указатель, который указывает на текущий объект и является скрытым параметром класса, это:

- а) inline;
- б) **this;**
- в) static;
- г) protected.

9. Функция, вызов которой и выполняемые при этом действия зависят от типа объекта, для которого она вызвана, называется:

- а) дружественными;
- б) **встраиваемыми;**
- в) виртуальными;
- г) конструкторами.

10. Ключевым словом, в определении некоторого класса, которое помогает объявить функцию дружественной этому классу, является:

- а) private;
- б) this;
- в) protected;
- г) **friend;**
- д) virtual.

5.2 Тестовое задание по теме «Решение задач программной визуализации при разработке клиент-серверного ПО»

1. Какая компания официально ввела понятие GPU:

- а) **NVidia;**
- б) ATI;
- в) SGI;
- г) 3Dlabs.

2. Какой метод отвечает за удаление невидимых поверхностей:

- а) алгоритм Брезенхема;
- б) **z-буфер;**
- в) bump mapping.

3. Какая спецификация низкоуровневой компьютерной графики является открытой:

- а) GLSL;
- б) OpenCV;
- в) **OpenGL;**

г) IrisGL.

4. На чем основывается работа OpenGL:

- а) граф зависимостей;
- б) машина состояний;**
- в) недетерминированный конечный автомат.

5. Какой тип архитектуры реализуется для построения изображения:

- а) конвейерная;**
- б) последовательная;
- в) X86

6. Что такое программа-шейдер:

- а) программа, выполняющаяся на CPU;
- б) программа, выполняющаяся на GPU;**
- в) программа, выполняющаяся на DSP.

7. Каких типов шейдеров не существует:

- а) геометрических;
- б) пиксельных;
- в) объектных;**
- г) фрагментных.

8. К какому типу преобразований относится преобразование переноса:

- а) аффинное;**
- б) линейное;
- в) нелинейное.

9. Из чего состоит полигональная сетка:

- а) вершин;
- б) полигонов-треугольников;**
- в) кортежей.

10. Как реализуются сложные преобразования:

- а) умножением матриц;**
- б) сложением матриц;
- в) сложением векторов.

Ключи: правильные ответы выделены «жирным»

5.3 Вопросы к зачету:

1. Шаблоны проектирования. Примеры. Использование
2. Моделирование требований к ПО. Варианты использования. Выявление вариантов использования. Пример описания варианта использования.
3. Взаимосвязи вариантов использования (включение, расширение). Рекомендации по структурированию вариантов использования.
4. Описание не функциональных требований к ПО.
5. Ассоциации между классами. Составление и агрегирование иерархий. Иерархия обобщения/специализации. Ограничения. Статическое моделирование контекста системы.
6. Категоризация классов с помощью стереотипов UML. Моделирование внешних классов.
7. Критерии структурирования объектов и классов. Классы и объекты моделирования приложений.
8. Внешние классы и классы границ ПО. Граничные классы и объекты.
9. Моделирование объектов взаимодействия. Нумерация последовательности сообщений на диаграмме взаимодействия.

10. Моделирование динамического взаимодействия. Моделирование динамического взаимодействия без учета состояния. Примеры моделирование динамического взаимодействия без учета состояния.
11. Конечные автоматы и переходы между состояниями. Примеры диаграмм состояний. События и условия сохранения.
12. Иерархические диаграммы состояний. Разработки диаграмм состояния на основе вариантов использования. Примеры.
13. Этапы моделирования динамического взаимодействия с учетом состояния.
14. Моделирование сценариев взаимодействия с помощью диаграмм взаимодействия и диаграмм состояния.
15. Шаблоны архитектур ПО.
16. Проектирование клиент-серверных архитектур ПО. Проектирование архитектуры подсистем ПО.
17. Проблемы проектирования архитектуры ПО.
18. Критерии структурирования подсистем. Выбор сообщений, передаваемых между подсистемами.
19. Проектирование интерфейсов и операций классов.
20. Классы для описания абстрактных данных.
21. Классы для описания конечных автоматов.
22. Классы для описания графического интерфейса пользователей.
23. Классы описания бизнес логики.
24. Использование наследования в проектировании. Спецификация интерфейсов классов.
25. Архитектурные шаблоны структур клиент-серверного ПО. Архитектурные шаблоны коммуникации для клиент-серверных архитектур.
26. Промежуточное ПО в клиент-серверных системах. Проектирование сервисных подсистем.
27. Проектирование на основе перехода от статических моделей к реляционным БД.
28. Проектирование ориентированных на сервисы архитектур ПО Программные архитектуры шаблонов брокеров.
29. Технологическая поддержка ориентированных на сервисы. Архитектурные программные шаблоны транзакций.
30. Проектирование интерфейса сервисов в сервис-ориентированной архитектуре. Координация сервисов в сервис-ориентированной архитектуре.
31. Проектирование сервис-ориентированных архитектур.
32. Проектирование распределенных программных архитектур, основанных на компонентах. Составные подсистемы и компоненты. Моделирование компонент с помощью UML.