

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Усынин Максим Валерьевич
Должность: Ректор
Дата подписания: 25.12.2024 11:41:07
Уникальный программный ключ:
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

Частное образовательное учреждение высшего образования
«Международный Институт Дизайна и Сервиса»
(ЧОУВО МИДиС)

Кафедра математики и информатики

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО
КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ
СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Направление подготовки: 09.03.03 Прикладная информатика
Направленность (профиль): Разработка компьютерных игр и приложений с
виртуальной и дополненной реальностью
Квалификация выпускника: Бакалавр
Год набора: 2022

Автор-составитель: Чеботарев С.С.

СОДЕРЖАНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания	4
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	6
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	11

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Современные технологии программирования» направлен на формирование следующих компетенций:

Код и наименование компетенции выпускника	Код и наименование индикатора достижения компетенций
ПК-1 Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	<p>ПК-1.1. Разрабатывает код информационных систем и баз данных информационных систем.</p> <p>ПК-1.2. Осуществляет верификацию кода, баз данных и структуры баз данных информационных систем</p> <p>ПК-1.3. Устраняет обнаруженные несоответствия с применением методик тестирования разрабатываемых информационных систем</p>
ПК-2 Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе	<p>ПК-2.1. Применять методы обследования организации и анализа входной информации для формирования требований к информационной системе</p> <p>ПК-2.2. Осуществлять деятельность по проведению переговоров и презентаций для информирования заказчиков о возможностях информационной системы.</p> <p>ПК-2.3. Выявлять информационные потребности пользователей, определяет возможности достижения соответствия информационных систем первоначальным требованиям заказчика, разрабатывает стратегии управления заинтересованными сторонами в проекте.</p>

№ п/п	Код компетенции	Наименование компетенции	Этапы формирования компетенций
1.	ПК-1	Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	<p><i>1 Этап – Знать:</i> ПК-1.1. Способы разработки кода информационных систем</p> <p><i>2 Этап – Уметь:</i> ПК-1.2. Осуществлять верификацию кода, баз данных и структуры баз данных информационных систем</p> <p><i>3 Этап – Владеть:</i> ПК-1.3. Навыками устранения обнаруженных несоответствий с применением методик тестирования разрабатываемых информационных систем</p>
2.	ПК-2	Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе	<p><i>1 Этап – Знать:</i> ПК-2.1. Методы обследования организации и анализа входной информации для формирования требований к информационной системе</p> <p><i>2 Этап – Уметь:</i> ПК-2.2. Осуществлять деятельность по проведению переговоров и презентаций</p>

		системе	для информирования заказчиков о возможностях информационной системы
			<p><i>3 Этап – Владеть:</i> ПК-2.3. Навыками выявления информационных потребностей пользователей, навыками определения возможности достижения соответствия информационных систем первоначальным требованиям заказчика; навыками разработки стратегии управления заинтересованными сторонами в проекте</p>

2. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

№ п/п	Код компетенции	Наименование компетенции	Критерии оценивания компетенций на различных этапах формирования	Шкала оценивания
1.	ПК-1	Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках со-временных бизнес-приложений)	<p><i>1 Этап – Знать:</i> ПК-1.1. Способы разработки кода информационных систем</p> <p><i>2 Этап – Уметь:</i> ПК-1.2. Осуществлять верификацию кода, баз данных и структуры баз данных информационных систем</p> <p><i>3 Этап – Владеть:</i> ПК-1.3. Навыками устранения обнаруженных несоответствий с применением методик тестирования разрабатываемых информационных систем</p>	<p>Зачет «ЗАЧТЕНО»</p> <ol style="list-style-type: none"> 1. Хорошее знание программного материала. 2. Знание основных пакетов прикладных программ. 3. Точность и обоснованность выводов. 4. Логичное изложение вопроса, соответствие изложения научному стилю. 5. Негрубая ошибка при выполнении практического задания. 6. Наличие незначительных неточностей в употреблении терминов, классификаций 7. Правильные ответы на дополнительные вопросы. <p>«НЕ ЗАЧТЕНО»</p> <ol style="list-style-type: none"> 1. Незнание значительной части программного материала. 2. Неспособность привести примеры пакетов прикладных программ 3. Неумение выделить главное, сделать выводы и обобщения. 4. Грубые ошибки при выполнении практического задания. 5. Неправильные ответы на дополнительные вопросы <p>Экзамен</p>
2	ПК-2	Способен проводить обследование организаций, выявлять информационные потребности	<p><i>1 Этап – Знать:</i> ПК-2.1. Методы обследования организации и анализа входной информации для формирования</p>	

		пользователей, формировать требования к информационной системе	требований информационной системе к	<p style="text-align: center;">«ОТЛИЧНО»</p> <ol style="list-style-type: none"> 1. Глубокое и прочное усвоение программного материала. 2. Знание пакетов прикладных программ. 3. Знание основных принципов построения пакетов прикладных программ.
		2 Этап – Уметь: ПК-2.2. Осуществлять деятельность по проведению переговоров и презентаций для информирования заказчиков о возможностях информационной системы		<ol style="list-style-type: none"> 4. Знание основных задач прикладных программ. 5. Свободное владение пакетами прикладных программ. 6. Точность и обоснованность выводов. 7. Безошибочное выполнение практического задания. 8. Точные, полные и логичные ответы на дополнительные вопросы
		3 Этап – Владеть: ПК-2.3. Навыками выявления информационных потребностей пользователей, навыками определения возможности достижения соответствия информационных систем первоначальным требованиям заказчика; навыками разработки стратегии управления заинтересованными сторонами в проекте		<p style="text-align: center;">«ХОРОШО»</p> <ol style="list-style-type: none"> 1. Хорошее знание программного материала. 2. Недостаточно полное изложение теоретического вопроса экзаменационного билета. 3. Наличие незначительных неточностей в употреблении терминов, классификаций. 4. Знание основных пакетов прикладных программ. 5. Неполнота представленного иллюстративного материала. 6. Точность и обоснованность выводов. 7. Логичное изложение вопроса, соответствие изложения научному стилю. 8. Негрубая ошибка при выполнении практического задания. 9. Правильные ответы на дополнительные вопросы. <p style="text-align: center;">«УДОВЛЕТВОРИТЕЛЬНО»</p> <ol style="list-style-type: none"> 1. Поверхностное усвоение программного материала. 2. Недостаточно полное изложение теоретического вопроса экзаменационного билета. 3. Затруднение в приведении примеров, подтверждающих теоретические положения. 4. Наличие неточностей в употреблении терминов, классификаций. 5. Неумение четко сформулировать

				<p>выводы.</p> <p>6. Отсутствие навыков научного стиля изложения.</p> <p>7. Грубая ошибка в практическом задании.</p> <p>8. Неточные ответы на дополнительные вопросы.</p> <p>«НЕУДОВЛЕТВОРИТЕЛЬНО»</p> <p>1. Незнание значительной части программного материала.</p> <p>2. Неспособность привести примеры пакетов прикладных программ</p> <p>3. Неумение выделить главное, сделать выводы и обобщения.</p> <p>4. Грубые ошибки при выполнении практического задания.</p> <p>5. Неправильные ответы на дополнительные вопросы.</p>
--	--	--	--	--

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1 ЭТАП – ЗНАТЬ

Устный опрос на практическом занятии по темам:

Раздел 1. Базовые паттерны программирования

- Тема 1.1. Наблюдатель
- Тема 1.2. Итератор
- Тема 1.3. Декоратор объектов
- Тема 1.4. Адаптер
- Тема 1.5. Фабрика
- Тема 1.6. Стратегия

Раздел 2. Паттерны параллельного программирования

- Тема 2.1. Работа с задачами (Tasks).
- Тема 2.2. Концепции библиотеки PFX
- Тема 2.3. Технология PLINQ
- Тема 2.4. Async методы

2 ЭТАП – УМЕТЬ

Практические задания

Кейс-задачи

Задание №1: Об “обедающих философах”

Для реализации потребуется пять процессоров.

Суть задачи следующая: пять философов сидят за круглым столом. Они проводят жизнь, чередуя приемы пищи и размышления.

В центре стола находится большое блюдо спагетти. Философам, чтобы съесть порцию спагетти, требуется две вилки.

Вилкок всего пять: между каждой парой философов лежит по одной вилке.

Каждому философу дозвоительно пользоваться только вилками, которые лежат рядом с ним (слева и справа).

Задача - написать программу, моделирующую поведение философов. Программа должна избегать ситуации, в которой все философы голодны, то есть ни один из них не может взять себе две вилки (например, когда каждый философ держит по одной вилке и не хочет отдавать ее). Раз вилкок всего пять, то одновременно могут есть не более, чем двое философов. Два сидящих рядом философа не могут есть одновременно. Предположим, что периоды раздумий и приемов пищи различны - для их имитации в программе можно использовать генератор случайных чисел. Имитация поведения каждого философа может быть разбита на следующие блоки: поразмыслить, взять вилки, поесть, отдать вилки. Вилки являются разделяемым ресурсом.

Задание №2: “о восьми ферзях”

Нужно расставить на шахматной доске восемь ферзей так, чтобы они не атаковали друг друга, разработайте программу, которая строит все 92 решения этой задачи.

Обнаружив часть решений, процессы должны передать их на 0-й процесс.

Программа должна найти все решения и завершить работу. Управляющий процесс должен все решения записать в файл результатов.

Задание №3: "о коммивояжере"

Эта классическая задача имеет практическое применение, например, при планировании обслуживания населения городским общественным транспортом. Дано N городов и симметричная матрица $A(N,N)$. Значением элемента $A(i,j)$ является расстояние между городами i и j . Коммивояжер начинает путь в городе 1 и должен посетить по одному разу каждый город, закончив свой путь снова в городе 1. Требуется найти путь, минимизирующий расстояние, которое придется проехать коммивояжеру, а результат сохранить в векторе $V(N)$. Значением вектора должна быть перестановка целых чисел от 1 до N , соответствующая порядку посещения городов коммивояжером.

Разработайте параллельную программу для решения поставленной задачи, используя модель "управляющий-рабочие".

Задание №4: “решето Эратосфена”

Запрограммируйте параллельную программу, реализующую алгоритм “решето Эратосфена” для нахождения всех простых чисел меньше p . Решетом Эратосфена называют следующий способ.

Выпишем подряд все целые числа от 2 до p .

Первое простое число 2. Запомним его, а все большие числа, кратные 2, вычеркнем.

Первое из оставшихся чисел 3. запомним его, а все большие числа, кратные 3, вычеркнем.

Первое из оставшихся чисел 5 (4 уже вычеркнуто как кратное 2), запомним его, а все большие числа кратные 5, вычеркнем и т.д.

Задание №5: метод Монте-Карло

Запрограммируйте параллельную программу, реализующую алгоритм метода Монте-Карло для вычисления площади круга $x+y < 1$

Методы Монте-Карло - это общее название группы методов для решения различных задач с помощью случайных последовательностей. Эти методы (как и вся теория вероятностей) выросли из попыток людей улучшить свои шансы в азартных играх. Этим объясняется и тот факт, что название этой группе методов дал город Монте-Карло - столица европейского игорного бизнеса.

При реализации параллельного алгоритма каждый процессор генерирует свое пространство случайных чисел и их обрабатывает. Запишите в файл результат выполнения параллельной программы, содержащий общее количество сгенерированных случайным образом точек n , количество попавших в заданную область точек t , приближенное вычисленное значение площади круга, погрешность вычисления.

Выполнение проектов

Проект предполагает создание проекта на C#, отвечающего требованиям стиля:

- создание DLL;
- визуальный интерфейс, выделенный в отдельный поток, позволяющий наблюдать и управлять процессом бизнес-логики;
- визуализация результатов работы;
- другие правила хорошего стиля.

Содержательно проект предполагает создание как последовательной, так и параллельной версии решаемой задачи. Теоретическое и практическое сравнение эффективности работы разных версий в зависимости от числа используемых процессоров (ядер). В результате проведенного анализа должны быть сделаны обоснованные выводы о целесообразности распараллеливания данной задачи и какой эффект может дать распараллеливание.

Варианты проектов:

1. Игра "Жизнь";
2. Игра «Балда» - подбор слов;
3. Множество Мандельброта и его аналоги;
4. Обедаящие философы;
5. Алгоритмы Data Mining
 - кластеризации K-Means (k- средних);
 - кластеризации, основанный на мере близости;
 - Apriori;
 - C4.5;
 - Cart.

3 ЭТАП – ВЛАДЕТЬ

Вопросы к зачету

1. Классификация вычислительных систем.
2. Уровни параллелизма в многоядерных архитектурах.
3. Проблемы разработки параллельных приложений.
4. Декомпозиция задач. Масштабирование подзадач.
5. Синхронизация. Моделирование параллельных приложений.
6. Основы параллельного программирования .NET. Состояния потоков.
7. Переключение контекста. Свойства потока Thread.
8. Локальное хранилище потока.
9. Пул (pool) потоков.
10. Средства синхронизации.
11. Основные конкурентные коллекции, особенности и примеры использования.
12. Task Parallel Library. Статусы задач.
13. Работа с данными в задачах.
14. Вложенные задачи.
15. Механизмы отмены задач.
16. Исключения в задачах.
17. Задачи-продолжения.

Вопросы к экзамену

1. Класс Parallel.
2. Параллельные циклы.
3. Параметры цикла.
4. Разделение данных.
5. Вычисление агрегированных значений.
6. Пакетная обработка итераций.
7. Метод AsParallel().
8. Основные этапы выполнения PLINQ-запроса.
9. Функциональная чистота.
10. Вызов блокирующих функций или функций с интенсивным вводом/выводом.
11. Эффективность. Буферизация. Разделение данных.
12. Агрегирование вычислений.
13. Асинхронные паттерны. Интерфейса IAsyncResult.
14. Task и await.
15. Паттерн Task-based Asynchronous Pattern.

Тестирование

Задание:

В каких случаях не возникает проблема гонки данных

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1 один поток осуществляет запись в разделяемую переменную типа double, другой поток читает значения.

Вариант 2 потоки инкрементируют значения, которые хранятся в разделяемом массиве; каждый поток работает с элементом по уникальному индексу.

Вариант 3 один поток добавляет элементы в коллекцию типа FIFO (очередь), другой поток извлекает элементы из коллекции.

Вариант 4 потоки осуществляют запись в разделяемую переменную типа int

Задание:

Выберете правильные утверждения, соответствующие системам с общей памятью

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1 существует возможность параллелизма как на уровне потоков одного процесса, так и на уровне процессов

Вариант 2 обеспечивается параллелизм только на уровне приложений как разделение процессорного времени

Вариант 3 для взаимодействия параллельных потоков не требуется специальных интерфейсов передачи сообщений

Вариант 4 для взаимодействия параллельных частей требуется применять средства синхронизации

Задание:

Выберете правильные утверждения, соответствующие системам с общей памятью

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1 при выполнении параллельной программы может возникать ситуация гонки данных

Вариант 2 синхронизация необходима только при выполнении программы в режиме разделения времени

Вариант 3 для реализации параллельного выполнения программы обязательно применение средств синхронизации

Вариант 4 применение средств синхронизации позволяет решить проблему гонки данных
Задание:

Оптимизация относительного расположения данных в памяти позволяет снять проблему:
(Отметьте один правильный вариант ответа.)

Вариант 1 гонка данных

Вариант 2 ложное разделение кэша

Вариант 3 сбалансированность нагрузки

Вариант 4 взаимоблокировка

Задание:

Параллельный алгоритм состоит из двух подзадач и обеспечивает ускорение в два раза на двудерной системе. Чему равна эффективность алгоритма системе с четырьмя процессорами?

(Отметьте один правильный вариант ответа.)

Вариант 1 0.5

Вариант 2 2

Вариант 3 1

Вариант 4 0.25

Задание:

Какое предельное ускорение может быть получено согласно закону Амдала для параллельной программы, содержащей 20% последовательного кода

(Отметьте один правильный вариант ответа.)

Вариант 1 нет правильных ответов

Вариант 2 2

Вариант 3 5

Вариант 4 10

Задание:

Что такое мультипрограммный режим работы?

(Отметьте один правильный вариант ответа.)

Вариант 1 параллельная разработка нескольких программ

Вариант 2 параллельное выполнение разных программ

Вариант 3 параллельное выполнение одной программы

Вариант 4 выполнение приложений мультимедиа

Задание:

Если полагать, что единственным общим ресурсом является оперативная память, то какие проблемы характерны для параллельных вычислений, выполняемых на системах с распределенной памятью (кластеров):

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1 гонка данных (DataRace)

Вариант 2 блокировка

Вариант 3 клинч

Вариант 4 синхронизация

Вариант 5 распределить – собрать (Map – Reduce)

Задание:

Что такое блокировка в параллельных вычислениях?

(Отметьте один правильный вариант ответа.)

Вариант 1 блокирование критической секции программы с целью исключения возможности ее одновременного выполнения в параллельных процессах

Вариант 2 блокирование области памяти с целью исключения возможности ее использования в параллельных процессах

Вариант 3 блокирование программы за исключением критических секций для получения возможности одновременного выполнения критических секций в нескольких параллельных процессах

Вариант 4 блокирование программы с целью исключения возможности ее использования в параллельных процессах

Задание:

Сколько классов компьютеров в классификации Флинна:

(Отметьте один правильный вариант ответа.)

Вариант 1 2

Вариант 2 4

Вариант 3 8

Вариант 4 16

Задание:

Из закона Амдаля следует:

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1 для любой задачи можно добиться сколь угодно высокого ускорения за счет увеличения числа используемых процессоров

Вариант 2 для некоторой задачи за счет увеличения числа используемых процессоров нельзя добиться 10-кратного ускорения

Вариант 3 для любой задачи ускорение ограничено величиной, зависящей от доли последовательных вычислений, не подлежащих распараллеливанию

Вариант 4 для любой задачи за счет увеличения числа используемых процессоров можно добиться более чем 10-кратного ускорения

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1 ЭТАП – ЗНАТЬ

Критерии оценки устных ответов студентов

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения,

	применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно.
«хорошо»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно, но допускает 1-2 ошибки, которые исправляет сам.
«удовлетворительно»	знает и понимает основные положения по содержанию задания; излагает материал неполно, но допускает неточности в определении понятий или формулировке правил; не достаточно глубоко и доказательно обосновать свои суждения иллюстрирует ответ примерами только по предложенной литературе; излагает материал последовательно и допускает 3-4 ошибки.
«неудовлетворительно»	допускает существенные ошибки в формулировке определений и правил, искажающие их смысл; излагает материал непоследовательно, неуверенно и не по существу задания; допускает существенные ошибки, не позволяющие раскрыть смысл задания, являющиеся серьезным препятствием к успешному овладению следующим материалом.

2 ЭТАП – УМЕТЬ

Критерии оценивания результатов выполнения практических заданий

Оценка	Критерии
«отлично»	Работа выполнена и оформлена верно; использован оптимальный метод решения;
«хорошо»	Работа выполнена и оформлена в целом верно с небольшими ошибками;
«удовлетворительно»	Работа выполнена и оформлена с ошибками; использован неполный метод решения;
«неудовлетворительно»	Работа не выполнена в срок

Критерии оценивания работы с кейс-задачами

Оценка	Критерии
«отлично»	разработана архитектура классов; разработано приложение; на защите кейс-задачи были получены верные ответы на все дополнительные вопросы
«хорошо»	разработана архитектура классов; разработано приложение; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«удовлетворительно»	разработана архитектура классов; разработано приложение, но в приложении имеются ошибки и недоработки; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«неудовлетворительно»	разработана архитектура классов; не было разработано приложение

3 ЭТАП – ВЛАДЕТЬ

Критерии оценивания знаний на зачете

Оценка	Критерии
«зачтено»	<ol style="list-style-type: none"> 1. Хорошее знание программного материала. 2. Знание основных пакетов прикладных программ. 3. Точность и обоснованность выводов. 4. Логичное изложение вопроса, соответствие изложения научному стилю. 5. Негрубая ошибка при выполнении практического задания. 6. Наличие незначительных неточностей в употреблении терминов, классификаций. 7. Правильные ответы на дополнительные вопросы.
«не зачтено»	<ol style="list-style-type: none"> 1. Незнание значительной части программного материала. 2. Неспособность привести примеры пакетов прикладных программ. 3. Неумение выделить главное, сделать выводы и обобщения. 4. Грубые ошибки при выполнении практического задания. 5. Неправильные ответы на дополнительные вопросы.

Критерии оценивания знаний на экзамене

Оценка	Критерии
«отлично»	<ol style="list-style-type: none"> 1. Глубокое и прочное усвоение программного материала. 2. Знание пакетов прикладных программ. 3. Знание основных принципов построения пакетов прикладных программ. 4. Знание основных задач прикладных программ. 5. Свободное владение пакетами прикладных программ. 6. Точность и обоснованность выводов. 7. Безошибочное выполнение практического задания. 8. Точные, полные и логичные ответы на дополнительные вопросы.
«хорошо»	<ol style="list-style-type: none"> 1. Хорошее знание программного материала. 2. Недостаточно полное изложение теоретического вопроса экзаменационного билета. 3. Наличие незначительных неточностей в употреблении терминов, классификаций. 4. Знание основных пакетов прикладных программ. 5. Неполнота представленного иллюстративного материала. 6. Точность и обоснованность выводов. 7. Логичное изложение вопроса, соответствие изложения научному стилю. 8. Негрубая ошибка при выполнении практического задания. 9. Правильные ответы на дополнительные вопросы.
«удовлетворительно»	<ol style="list-style-type: none"> 1. Поверхностное усвоение программного материала. 2. Недостаточно полное изложение теоретического вопроса экзаменационного билета. 3. Затруднение в приведении примеров, подтверждающих теоретические положения. 4. Наличие неточностей в употреблении терминов, классификаций. 5. Неумение четко сформулировать выводы. 6. Отсутствие навыков научного стиля изложения. 7. Грубая ошибка в практическом задании. 8. Неточные ответы на дополнительные вопросы.
«неудовлетворительно»	<ol style="list-style-type: none"> 1. Незнание значительной части программного материала. 2. Неспособность привести примеры пакетов прикладных программ.

	<ol style="list-style-type: none">3. Неумение выделить главное, сделать выводы и обобщения.4. Грубые ошибки при выполнении практического задания.5. Неправильные ответы на дополнительные вопросы.
--	--

Критерии оценивания результатов теста

Полная версия тестовых вопросов содержится в электронно-информационной системе вуза. Студенты проходят тестирование в компьютерном классе. Оценка успешности прохождения теста определяется следующей сеткой: от 0% до 29% – «неудовлетворительно», от 30% до 59% – «удовлетворительно»; 60% – 79 % – «хорошо»; 80% -100% – «отлично».