

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 18.06.2026 14:18:48

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d19d0c800

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
кандидат технических наук,
доцент

А.А. Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ Асинхронное программирование

Направленность (профиль)	09.03.02 Информационные системы и технологии
Профиль	Прикладное программирование в интеллектуальных информационных системах
Форма обучения	очная
Год начала обучения	2026
Реализуется в 8 семестре	

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимся установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Асинхронное программирование»

3. Разработчик Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Дроздова В. И., профессор департамента цифровых, робототехнических систем и электроники

Члены комиссии:

Крахоткина Е.В., доцент департамента цифровых, робототехнических систем и электроники

Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

Представитель организации-работодателя: А. В. Новиков, начальник отдела АСУТП ООО «ЕНДС-Ставрополь»

Экспертное заключение: ФОС по дисциплине «Асинхронное программирование» позволяет оценить уровень сформированности компетенций. Рекомендовать к использованию в учебном процессе.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
<i>Компетенция: ПК-1</i>				
Результаты обучения по дисциплине (модулю): Демонстрирует знания в области применения алгоритмов и их адаптации к практическим задачам <i>Индикатор:</i> ПК-1 _{ид-1.1}	Понимает основные понятия и принципы алгоритмов. Знает наиболее распространенные алгоритмы и их применение в различных областях. Способен применять готовые алгоритмы для решения простых практических задач. Не имеет опыта в адаптации алгоритмов к конкретным практическим задачам.	Обладает глубоким пониманием основных алгоритмов и их принципов работы. Умеет адаптировать и модифицировать готовые алгоритмы для решения задач конкретной области. Имеет опыт применения алгоритмов в практических проектах. Способен обосновывать применение алгоритмов и доказывать их эффективность.	Владеет широким спектром алгоритмов и глубоко разбирается в принципах их работы. Умеет эффективно адаптировать алгоритмы к самым сложным практическим задачам. Имеет опыт разработки собственных алгоритмов и методов их оптимизации. Может проводить обучение и консультирование других специалистов по алгоритмам.	Создает уникальные и высокоэффективные алгоритмические решения для сложных профессиональных задач. Имеет публикации и научные работы по алгоритмам и их применению. Способен выступать на конференциях и вести обучающие курсы по алгоритмике.
Результаты обучения по дисциплине (модулю): Разрабатывает приложения на языке высокого уровня в том числе с использованием интеллектуальных технологий <i>Индикатор:</i> ПК-1 _{ид-1.2}	Понимание основных понятий и принципов языка высокого уровня (например, переменные, операторы, функции). Умение разрабатывать простые приложения с использованием языка высокого уровня. Знание основных интеллектуальных технологий и их возможностей.	Демонстрирует глубокое понимание синтаксиса и особенностей языка высокого уровня; умение разрабатывать сложные приложения с использованием языка высокого уровня. Знает основные алгоритмы и структуры данных. Способен использовать интеллектуальные технологии (например, машинное обучение или нейронные сети) в разработке приложений.	Демонстрирует глубокое знание языка высокого уровня, включая продвинутые возможности и функции. Способен проектировать и реализовывать сложные приложения, используя язык высокого уровня и интеллектуальные технологии. Способен оптимизировать и улучшать производительность приложений. Способен анализировать и решать сложные задачи с использованием интеллектуальных технологий.	Демонстрирует глубокое и полное владение языком высокого уровня и его возможностями. Способен разрабатывать сложные и инновационные приложения, используя язык высокого уровня и интеллектуальные технологии. Способен создавать собственные алгоритмы и модели для решения сложных задач. Способен применять передовые исследования и технологии в области интеллектуальных технологий.

				в разработке приложений.
Результаты обучения по дисциплине (модулю): Применяет современные автоматизированные технологии тестирования и отладки программного обеспечения <i>Индикатор:</i> ПК-1_{ид-1.3}	Знание основных принципов и методов тестирования программного обеспечения. Способен применять базовые инструменты и технологии для автоматизации тестирования. Способен проводить ручное тестирование программного обеспечения и отлаживать проблемы	Демонстрирует понимание и применяет различные методы тестирования, такие как модульное тестирование, интеграционное тестирование, функциональное тестирование и др. Способен создавать и поддерживать автоматизированные тестовые скрипты и сценарии. Применяет инструменты для отладки и профилирования программного обеспечения	Способен создавать и управлять комплексными автоматизированными тестовыми системами. Демонстрирует навыки использования инструментальных средств для контроля качества и управления дефектами, таких как JIRA, QC/ALM и другие. Демонстрирует глубокое понимание принципов разработки и отладки программного обеспечения, чтобы эффективно выявлять и исправлять ошибки	Демонстрирует экспертное знание и опыт применения современных автоматизированных технологий тестирования, таких как контейнеризация, облачное тестирование, DevOps и другие. Способен создавать и реализовывать стратегии тестирования программного обеспечения. Владеет современными инструментами и методами для поиска и устранения уязвимостей безопасности в программном обеспечении
Результаты обучения по дисциплине (модулю): Применяет современные технологии непрерывного развертывания и доставки программного обеспечения <i>Индикатор:</i> ПК-1_{ид-1.4}	Понимает основные принципы непрерывного развертывания и доставки программного обеспечения. Имеет представление о современных инструментах и технологиях, используемых для автоматизации процессов развертывания и доставки ПО. Способен использовать базовые функции инструментов непрерывного развертывания, таких как системы контроля версий и среды разработки	Обладает углубленными знаниями о принципах и процессах непрерывного развертывания и доставки ПО. Владеет различными инструментами и технологиями, используемыми для автоматизации процессов развертывания и доставки ПО. Может использовать продвинутые функции инструментов непрерывного развертывания, таких как CI/CD платформы и системы управления конфигурациями. Способен проектировать и настраивать слож-	Обладает глубокими знаниями и опытом в области непрерывного развертывания и доставки ПО. Владеет широким набором инструментов и технологий для автоматизации процессов развертывания и доставки ПО. Способен разрабатывать и реализовывать инновационные подходы к непрерывному развертыванию и доставке ПО. Может управлять сложными проектами по развертыванию и доставке	Обладает экспертным уровнем по непрерывному развертыванию и доставке ПО и вносит значительный вклад в развитие данной области. Владеет всеми современными инструментами и технологиями, используемыми в непрерывном развертывании и доставке ПО. Способен разрабатывать и реализовывать стратегии и

		ные цепочки развертывания и доставки ПО. Понимает преимущества и ограничения контейнеризации и виртуализации и может использовать их эффективно в своей работе	ПО, включая управление ресурсами, распределением задач и контролем качества. Обладает глубоким пониманием принципов контейнеризации и виртуализации и может использовать их для оптимизации процессов развертывания и доставки ПО	методологии развертывания и доставки ПО для организаций любого масштаба. Может эффективно управлять сложными проектами и командами, занимающимися развертыванием и доставкой ПО. Вносит вклад в научные исследования в области непрерывного развертывания и доставки ПО и активно развивает эту область
Результаты обучения по дисциплине (модулю): Разрабатывает приложения для различных целевых платформ (мобильные, серверные, встраиваемые, оконные) <i>Индикатор:</i> ПК-1 _{ид-1.5}	Основные знания в области разработки приложений для различных платформ. Умение использовать основные инструменты и технологии для разработки приложений (например, различные языки программирования, фреймворки, библиотеки). Способность разрабатывать простые приложения для одной или нескольких платформ.	Демонстрирует знания в области разработки приложений для различных платформ. Способен использовать различные инструменты и технологии для разработки полноценных приложений. Способен разрабатывать сложные приложения для одной или нескольких платформ	Демонстрирует глубокие знания в области разработки приложений для различных платформ. Способен применять передовые инструменты и технологии для разработки масштабных приложений (например, работа с облачными сервисами, многопоточность, оптимизация производительности). Способен разрабатывать сложные и инновационные приложения для одной или нескольких платформ	Демонстрирует обширные знания и использует практический опыт в области разработки приложений для различных платформ. Демонстрирует глубокое понимание принципов работы различных платформ и их особенностей. Способен разрабатывать сложные и высоконагруженные приложения для различных платформ. Способен анализировать и решать сложные технические проблемы, связанные с разработкой приложений для различных платформ
<i>Компетенция: ПК-7</i>				

Результаты обучения по дисциплине (модулю): Демонстрирует знания методик автоматизации составления технической документации и разработки отчетности по утвержденным формам <i>Индикатор:</i> ПК-7_{ид-7.1}	Знаком с основными методиками автоматизации составления технической документации и разработки отчетности. Знает основные инструменты и программное обеспечение, используемые для автоматизации этих процессов. Знания являются поверхностными и отсутствует достаточный опыт работы с ними.	Имеет глубокие знания методик автоматизации составления технической документации и разработки отчетности. Применяет эти методики на практике и имеет определенный опыт работы с инструментами и программным обеспечением. Способен выполнять более сложные задачи в автоматизации данных процессов.	Обладает высокими знаниями и умениями в методиках автоматизации составления технической документации и разработки отчетности. Способен применять различные стратегии и методы при автоматизации этих процессов, а также выполнять сложные и нетиповые задачи с использованием инструментов и программного обеспечения. Имеет опыт руководства в данной области.	Имеет обширный опыт в автоматизации составления технической документации и разработке отчетности. Обладает глубокими знаниями и опытом работы с различными инструментами и программным обеспечением, может разрабатывать и внедрять новые методики автоматизации. Способен решать сложные и нетипичные задачи, а также обучать и консультировать других специалистов.
Результаты обучения по дисциплине (модулю): Составляет техническую документацию на всех стадиях реализации проекта с использованием современных инструментальных средств <i>Индикатор:</i> ПК-7_{ид-7.2}	Знаком с основными принципами и требованиями к составлению технической документации на всех стадиях проекта. Способен создавать базовые документы, такие как технические задания, спецификации и отчеты, используя простые инструменты, такие как текстовые редакторы и таблицы	Применяет свои знания и навыки в составлении технической документации. Способен использовать сложные инструментальные средства, такие как программы для создания блок-схем и диаграмм, а также средства для создания проектных планов и графиков. Способен составлять документацию на всех стадиях проекта с использованием различных форматов и стандартов	Обладает глубокими знаниями и опытом в составлении технической документации. Способен использовать широкий спектр инструментальных средств, включая специализированные программы для разработки технической документации, такие как CASE-системы и средства автоматизации процесса составления документов. Способен может использовать новейшие технологии и методы, такие как документация с применением средств виртуальной реальности или техники моделирования.	Демонстрирует экспертный уровень в области составления технической документации на всех стадиях проекта. Обладает глубокими знаниями в области стандартов и требований к технической документации, а также опытом работы с различными инструментальными средствами. Способен разрабатывать сложные и масштабные документы, включая документацию для крупных проектов и систем. Он может также проводить обучение и консультирование других специалистов в

				области составления технической документации.
Результаты обучения по дисциплине (модулю): Применяет методы визуализации данных для обоснования эффективности проектов <i>Индикатор:</i> ПК-7_{ид-7.3}				

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Задания закрытого типа	

1.	1, 2	Какие типы относятся к типам-делегатам? 1) Action<T> 2) Func<T> 3) List<T> 4) IEnumerable<T>	ПК-1, ПК-7
2.	1	Какой метод позволяет реализовать остановку выполнения программы и ожидание продолжения выполнения в течении заданного количества миллисекунд? 1) Thread.Sleep() 2) System.Console() 3) Math.Wait() 4) Waiting.Stop()	ПК-1, ПК-7
3.	1	Какой класс используется для представления потока в многопоточном приложении? 1) Thread 2) Task 3) MultiTask 4) Waiting 5) Action<T>	ПК-1, ПК-7
4.	2	Какое пространство имен является корневым в библиотеке .NET 1) Data 2) System 3) Object 4) Input 5) IO	ПК-1, ПК-7

5.	1,2,4	В каких фрагментах кода объявляются переменные? 1) <code>int a;</code> 2) <code>int b = 7;</code> 3) <code>a = 123;</code> 4) <code>string str = "Hello, World!!!";</code>	ПК-1, ПК-7
6.	3	С помощью какого зарезервированного слова происходит объявление пространства имен? 1) <code>using</code> 2) <code>break</code> 3) <code>namespace</code> 4) <code>foreach</code> 5) <code>name</code>	ПК-1, ПК-7
7.	1	Даны определения классов на языке C#. <pre> public class Parent { protected int a; } class Child : Parent { int b; void E() { b = a; } } class Child1 : Parent { int c; public Child1() { } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) переменная <code>a</code> является недоступной в производном классе, следовательно, присвоение <code>b = a</code> некорректно 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным	ПК-1, ПК-7
8.	2	Что такое конструктор по умолчанию в языке C#? 1) Конструктор по умолчанию представляет собой метод, который вызывается перед созданием объекта и устанавливает значения всех полей класса в предопределенные значения	ПК-1, ПК-7

		2) Программист, определяя класс, может явно не реализовывать конструктор. В этом случае .NET Framework использует пустой метод в качестве конструктора по умолчанию 3) Программист определяет сколько угодно своих конструкторов, но .NET Framework будет игнорировать их, если не отключен конструктор по умолчанию	
9.	4	Даны определения классов на языке C#. <pre> public class Parent { protected int a; Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) конструктор базового класса не виден из производного класса	ПК-1, ПК-7
10.	4	Даны определения классов на языке C#. <pre> public class Parent { protected int a; protected Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данных фрагментах? 1) конструктор базового класса не виден из производного класса	ПК-1, ПК-7

		2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) ошибок нет	
11.	1	Даны определения классов на языке C#. <pre> public class Parent: Child1 { protected int a; protected Parent() { a = 0; } } class Child1 : Child { int c; } class Child : Parent { int b; void E() { b = a; } } </pre> Существуют ли ошибки в данных фрагментах? 1) классы наследуются по кругу 2) неверное использование модификатора доступа protected в классе Parent 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным 5) ошибок нет	ПК-1, ПК-7
12.	1	Сколько наследников может быть у представленного класса?	ПК-1, ПК-7

		<pre>public class Child : Parent { int b; void E() { b = a; } }</pre> 1) столько, сколько реализует программист (принципиальных ограничений нет) 2) количество наследников ограничивается в базовом классе Parent 3) не более 128	
13.	2	В языке C# программист определил класс следующим образом) <pre>public class Parent { protected int a; protected Parent() { a = 0; } }</pre> Какой класс является базовым для данного класса? 1) System.Data 2) System.Object 3) System.IO 4) нет базового класса	ПК-1, ПК-7
14.	3	В языке C# программист определил класс следующим образом)	ПК-1, ПК-7

		<pre> public class Parent { protected int a; protected Parent() { a = 0; } } </pre> Как характеризуется переменная a по отношению ко всем производным классам? 1) в производных классах переменная a является недоступной как внутри класса, так и из объектной переменной 2) в производных классах переменная a является недоступной внутри класса, но доступной из объектной переменной 3) в производных классах переменная a является видимой внутри класса, но недоступной из объектной переменной 4) в производных классах переменная a является видимой как внутри класса, так и из объектной переменной	
15.	3	В языке C# программист определил классы и создал их объекты следующим образом) <pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } } public class Child : Parent { int b; public Child() :base(0) { b = a; } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } </pre> Какая ошибка была допущена? 1) некорректное использование зарезервированного слова base 2) в конструкторе класса Child нет доступа к переменной a базового класса 3) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора	ПК-1, ПК-7

		4) нельзя создать объект базового класса, так как объявлен класс-наследник 5) ошибок нет	
16.	1	В языке C# программист определил классы и создал их объекты следующим образом) <pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } public Parent() : this(0) { } } public class Child : Parent { int b; public Child() : base() { b = a; } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } </pre> Какая ошибка была допущена? 1) ошибок нет 2) некорректное использование зарезервированного слова base 3) в конструкторе класса Child нет доступа к переменной a базового класса 4) некорректное использование зарезервированного слова this 5) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора	ПК-1, ПК-7
17.	1	В языке C# программист определил классы и интерфейсы следующим образом)	ПК-1, ПК-7

		<pre> public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } public interface IDraw { void Draw(); } public interface IString { string GetStr(); } </pre> Какая ошибка была допущена? 1) нет ошибок 2) класс MainObject не может иметь двух базовых классов 3) некорректное использование зарезервированного слова interface 4) в классе IString метод GetStr не определен	
18.	2	На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class Parent { protected int a; } </pre> <pre> public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } </pre> <pre> static void Main(string[] args) { MainObject ob1 = new MainObject(); ob1.Draw(); Parent ob2 = new Parent(); ob2.Draw(); } </pre> Какая ошибка была допущена?	ПК-1, ПК-7

		1) в классе Parent нет конструктора, поэтому объект ob2 не может быть создан 2) в классе Parent не определен метод Draw, который вызывается через объектную переменную ob2 3) класс MainObject не может иметь двух базовых классов 4) ошибок нет	
19.	2	На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } </pre> Какая ошибка была допущена? 1) класс MainObject не может иметь двух базовых классов 2) класс MainObject не реализует метод Draw, наследуемого интерфейса IDraw 3) класс MainObject содержит метод GetStr, который дублирует метод базового класса IString 4) ошибок нет	ПК-1, ПК-7
20.	1	На языке C# программист определил класс, интерфейсы)	ПК-1, ПК-7

		<pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre>	<pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre>	
		Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; } </pre> Какая ошибка была допущена? 1) нет ошибок 2) в массиве присутствуют элементы различного типа 3) не указан размер массива при его создании 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают		
21.	2	На языке С# программист определил класс, интерфейсы)		
				ПК-1, ПК-7

		<pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre> <pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre> Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; IString ob = new IString(); } </pre> Какая ошибка была допущена? 1) нет ошибок 2) Невозможно создать объект интерфейса ob 3) в массиве присутствуют элементы различного типа 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают	
22.	1	Для чего служит оператор break? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	ПК-1, ПК-7
23.	2	Для чего служит оператор continue? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации цикла	ПК-1, ПК-7

		3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	
24.	3	В переменной Sum по завершении выполнения фрагмента кода будет содержаться... <pre> long Sum = 0; int i; for (i = 1; i <= 1000000; i++) if (i % 3 == 0) Sum += i; </pre> 1) сумма кубов всех чисел от 1 до 1000000 2) 0 3) сумма всех чисел кратных 3 и находящихся в диапазоне от 1 до 1000000	ПК-1, ПК-7
25.		Какие СУБД являются реляционными? +: MS SQL Server -: MongoDB +: MySQL +: PostgreSQL	ПК-1, ПК-7
26.		Какие СУБД являются документоориентированными? +: MongoDB -: MySQL -: PostgreSQL -: MS SQL Server	ПК-1, ПК-7
27.		В каком формате представляются документы в MongoDB? +: JSON -: XML -: XLS -: HTTP	ПК-1, ПК-7
28.		Как происходит создание базы данных при использовании консоли MongoDB? +: база данных создается при вызове функции записи документа (документов), вместе с коллекцией, содержащей документы -: база данных создается специальной командой; MongoDB не предполагает создания базы данных «на лету» -: создать базу данных можно только из клиентского приложения на языке Python	ПК-1, ПК-7

29.	При описании подхода NoSQL используют правило 5V. Какие свойства присутствуют в правиле 5V? +: скорость (velocity) +: объем (volume) -: согласованность (consistency) -: атомарность (atomicity)	ПК-1, ПК-7
30.	При описании подхода NoSQL используют правило 5V. Какие свойства присутствуют в правиле 5V? +: достоверность (veracity) +: объем (volume) -: согласованность (consistency) +: значимость (value)	ПК-1, ПК-7
31.	Какое утверждение о формате JSON корректно? +: это текстовый формат -: это бинарный формат -: это табличный формат -: это мультимедия-контент	ПК-1, ПК-7
32.	Какое утверждение об объекте JSON корректно? +: это неупорядоченное множество пар имя/значение, заключённое в фигурные скобки { } -: это неупорядоченное множество пар имя/значение, заключённое в квадратные скобки [] -: это класс, определяемый на языке Javascript	ПК-1, ПК-7
33.	Какой из объектов JSON определен корректно? +: {“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”} -: /*“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”*/ -: [“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”] -: class P {name” : “Nik Corner”, “salary” : 2500, “address” : “No data”} -: это класс, определяемый на языке Javascript	ПК-1, ПК-7
34.	Какой из объектов JSON определен корректно? +: {“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”} -: <--“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”--!> -: [“name” : “Nik Corner”, “salary” : 2500, “address” : “No data”] -: class P {name” : “Nik Corner”, “salary” : 2500, “address” : “No data”}	ПК-1, ПК-7
35.	В консоли MongoDB необходимо выбрать базу users в качестве текущей. Какую команду консоли следует использовать? +: use users	ПК-1, ПК-7

		-: use database users -: using users -: select users	
36.		какая команда консоли позволяет разработчику получить справку по функционалу консоли MongoDB? +: help -: use -: select -: what	ПК-1, ПК-7
37.		В консоли MongoDB требуется узнать имя текущей используемой базы данных. Какую команду надо использовать для этого? +: db.getName() -: db.help() -: db.use() -: db.select()	ПК-1, ПК-7
38.		Какая команда консоли MongoDB позволяет вывести список баз данных? +: show dbs -: dbs list -: list -: print databases	ПК-1, ПК-7
39.		Какая команда консоли MongoDB позволяет вывести список коллекций текущей базы данных? +: show collections -: print collections -: collections -: db.list()	ПК-1, ПК-7
40.		Какая команда используется для запуска консоли MongoDB? +: mongo -: startmongo -: mongod -: start mongod	ПК-1, ПК-7
41.		Какая команда используется для импорта данных из внешнего источника в базу данных MongoDB? +: mongoimport -: importdata -: importmongo	ПК-1, ПК-7

		-: dataimport -: import	
42.		Какие стратегии для работы с MongoDB доступны разработчику? +: работа с MongoDB посредством консоли +: работа с MongoDB с использованием специального графического приложения Compass -: работа с MongoDB из текстового редактора	ПК-1, ПК-7
43.		Необходимо добавить документ в коллекцию persons текущей базы данных. Какая команда является корректной? +: db.persons.insert ({"name" : "No name"}) -: db.persons ({"name" : "No name"}) -: db.persons.create({"name" : "No name"}) -: db.insert({"name" : "No name"})	ПК-1, ПК-7
44.		Необходимо добавить один документ в коллекцию persons текущей базы данных. Какие команды являются корректными? +: db.persons.insert({"name" : "No name"}) -: db.persons({"name" : "No name"}) -: db.persons.insertOne({"name" : "No name"}) -: db.insert({"name" : "No name"})	ПК-1, ПК-7
45.		Необходимо добавить один документ в коллекцию persons текущей базы данных. Какие команды являются корректными? +: db.persons.insert({"name" : "No name"}) -: db.persons({"name" : "No name"}) -: db.persons.set({"name" : "No name"}) -: db.insert({"name" : "No name"})	ПК-1, ПК-7
46.		Необходимо добавить несколько документов в коллекцию persons текущей базы данных. Какие команды являются корректными? +: db.persons.insert ([{"name" : "No name"}, {"name" : "Max"}]) -: db.persons.select ([{"name" : "No name"}, {"name" : "Max"}]) +: db.persons.insertMany ([{"name" : "No name"}, {"name" : "Max"}]) -: db.persons.insert ({"name" : "No name"}, {"name" : "Max"})	ПК-1, ПК-7
47.		Необходимо выбрать все документы из коллекции zoo текущей базы данных. Какую команду консоли следует использовать? +: db.zoo.find()	ПК-1, ПК-7

		-: db.zoo.select() -: db.zoo.findAndSelect() -: db.find()	
48.		Необходимо выбрать все документы из коллекции cats текущей базы данных. Какую команду консоли следует использовать? +: db.cats.find() -: db.cats.select() -: db.cats.findAndSelect() -: db.find("cats")	ПК-1, ПК-7
49.		Необходимо выбрать все документы из коллекции person текущей базы данных. Какую команду консоли следует использовать? +: db.person.find() -: db.person.select() -: db.person.findAndSelect() -: db.find()	ПК-1, ПК-7
50.		Необходимо выбрать один документ из коллекции zoo текущей базы данных, удовлетворяющий критерию поиска. Какую команду консоли следует использовать? +: db.person.findOne({name: "Tiger"}) -: db.person.find({name: "Tiger"}) -: db.person.One({name: "Tiger"}) -: db.person.selectOne({name: "Tiger"})	ПК-1, ПК-7
51.		Необходимо выбрать один документ из коллекции zoo текущей базы данных, удовлетворяющий критерию поиска. Какую команду консоли следует использовать? +: db.person.findOne({age: 30}) -: db.person.find({age: 30}) -: db.person.One({age: 30}) -: db.person.selectOne({age: 30})	ПК-1, ПК-7
52.		В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} Были выполнены следующие команды: db.goods.insertOne({type: "A", product: "Сыр"})	ПК-1, ПК-7

		db.goods.insert ([{name: "No name"}, {type: "B", name: "Духи"}]) db.goods.getcount() Что будет выведено в консоль? -: 6 +: Error -: 2 -: 3 -: 1 -: 5 -: 4	
		Задания открытого типа	ПК-1, ПК-7
1.	2	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find({type: "Town"})?	ПК-1, ПК-7
2.	0	В коллекции places содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.places.find({name: "Town"})?	ПК-1, ПК-7
3.	1	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find({type: "Village"})?	ПК-1, ПК-7
4.	3	В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find()?	ПК-1, ПК-7
5.	5	В коллекции goods содержатся документы:	ПК-1,

		<pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос db.goods.find()?	ПК-7
6.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос db.goods.find({ type : "B" })?	ПК-1, ПК-7
7.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос db.goods.findOne({ type : "B" })?	ПК-1, ПК-7
8.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос db.goods.find({ type : "B" }).limit(1)?	ПК-1, ПК-7
9.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre>	ПК-1, ПК-7

		Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(3)?</code>	
10.	2	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> <code>{type: "B", name: "Салфетка"}</code> <code>{type: "C", name: "NoName"}</code> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(0)?</code>	ПК-1, ПК-7
11.	3	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> Сколько документов вернет запрос <code>db.goods.find().limit(0)?</code>	ПК-1, ПК-7
12.	1	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(2)?</code>	ПК-1, ПК-7
13.	2	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> Сколько документов вернет запрос <code>db.goods.find({ type : "A" }).limit(2)?</code>	ПК-1, ПК-7
14.	2	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> Сколько документов вернет запрос <code>db.goods.find({ type : "A" }).limit(0)?</code>	ПК-1, ПК-7
15.	2	В коллекции <code>goods</code> содержатся документы: <code>{type: "A", name: "Хлеб"}</code> <code>{type: "A", name: "Молоко"}</code> <code>{type: "B", name: "Спрей"}</code> Что вернет запрос <code>db.goods.find({ type : "A" }).count()?</code>	ПК-1, ПК-7

16.	1	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Что вернет запрос <code>db.goods.find({ type : "B" }).count()</code>?	ПК-1, ПК-7
17.	3	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко", date : "07.2019"} {type: "B", name: "Спрей"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "B", name: "Спрей"}])</pre> <code>db.goods.find({ type: "A" }).count()</code> Что будет выведено в консоль?	ПК-1, ПК-7
18.	2	В коллекции goods содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Молоко"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "A", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "C", name: "Духи"}])</pre> <code>db.goods.find({ name: "Молоко" }).count()</code> Что будет выведено в консоль?	ПК-1, ПК-7

2. Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ не содержит ошибок и решены все задачи. Компетенция ПК-1, ПК-7 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенция ПК-1, ПК-7 освоена на среднем уровне

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенция ПК-1, ПК-7 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенция ПК-1, ПК-7 не сформирована.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка **«отлично»** выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-1, ПК-7 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-1, ПК-7 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-1, ПК-7 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-1, ПК-7 не сформирована.