

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Андреевич

Должность: И.о. директора института перспективной инженерии

Дата подписания: 18.06.2026 12:13:40

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Директор института
перспективной инженерии,
кандидат технических наук,
доцент

Порохня А. А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Web-программирование

Направление подготовки

Профиль

Форма обучения

Год начала обучения

Реализуется в 2 семестре

09.04.02 Информационные системы и технологии

Управление данными

очная

2026

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 обучающимися установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Web-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:

Шагрова Г. В., профессор департамента цифровых, робототехнических систем и электроники

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

Представитель организации-работодателя: Ю. В. Рокотов, директор ООО «Кибер-Софт»

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Web-программирование» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.04.02 «Информационные системы и технологии» и позволяют оценить уровень сформированности компетенций УК-3, ПК-7.

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: УК-3</i>				
ИД-1 _{УК-3} Вырабатывает стратегию сотрудничества; организует отбор членов команды для достижения поставленной цели	Не отвечает (или отвечает некорректно) на вопросы собеседования. Не способен ответить по вопросам коллоквиума. Не предоставил ответы на контрольные вопросы лабораторных работ	При собеседовании некорректно отвечает на вопрос. Демонстрирует незнание отдельных вопросов теоретического материала в рамках коллоквиума. Не ответил на большинство контрольных вопросов лабораторных работ.	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. При ответе на вопросы коллоквиума демонстрирует пробелы в знании отдельных элементов теоретического материала. Не предоставил ответы на отдельные контрольные вопросы лабораторных работ	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. В рамках коллоквиума даны ответы на все вопросы преподавателя. В рамках защиты лабораторных работ формулирует корректные ответы на вопросы для контроля.
ИД-2 _{УК-3} Планирует командную работу, распределяет поручения и делегирует полномочия членам команды	Не выполняет учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Не способен проектировать архитектуру приложений для решения задач.	Выполняет корректно учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает и документирует приложения в рамках индивидуального задания к лабораторным работам. Самостоятельно проектирует архитектуру приложений в рамках выполнения лабораторных работ

ИД-3 _{УК-3} Организирует дискуссии по заданной теме и обсуждение результатов работы команды с привлечением оппонентов	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Не владеет навыками тестирования, отладки и документирования информационных систем.	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.	Владеет навыками по документированию приложений. Демонстрирует навыки использования интегрированных сред разработки.	Владеет навыками по документированию приложений. Демонстрирует навыки уверенного использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.
ИД-4 _{УК-3} Осуществляет мониторинг работы группы разработки объектов профессиональной деятельности, принимает управленческие решения и несет за них ответственность	Не выполняет учебное задание лабораторных работ. Не способен решать разрабатывать приложения в рамках индивидуального задания к лабораторным работам. Не способен проектировать архитектуру приложений для решения задач.	Выполняет корректно учебное задание лабораторных работ. Не способен решать разрабатывать приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает и документирует приложения в рамках индивидуального задания к лабораторным работам. Самостоятельно проектирует архитектуру приложений в рамках выполнения лабораторных работ
<i>Компетенция: ПК-7</i>				
ИД-1 _{ПК-7} Демонстрирует знание методов разработки, контроля качества, внедрения и сопровождения информационных систем и технологий в науке и производстве	Не отвечает (или отвечает некорректно) на вопросы собеседования. Не способен ответить по вопросам коллоквиума. Не предоставил ответы на контрольные вопросы лабораторных работ	При собеседовании некорректно отвечает на вопрос. Демонстрирует незнание отдельных вопросов теоретического материала в рамках коллоквиума. Не ответил на большинство контрольных вопросов лабораторных работ.	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. При ответе на вопросы коллоквиума демонстрирует пробелы в знании отдельных элементов теоретического материала. Не предоставил ответы на отдельные контрольные вопросы лабораторных работ	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. В рамках коллоквиума даны ответы на все вопросы преподавателя. В рамках защиты лабораторных работ формулирует корректные ответы на вопросы для контроля.

ИД-2 _{ПК-7} Осуществляет разработку информационных систем и технологий в науке и производстве	Не выполняет учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Не способен проектировать архитектуры приложений для решения задач.	Выполняет корректно учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает и документирует приложения в рамках индивидуального задания к лабораторным работам. Самостоятельно проектирует архитектуры приложений в рамках выполнения лабораторных работ
ИД-3 _{ПК-7} Контролирует качество информационных систем и технологий в науке и производстве	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Не владеет навыками тестирования, отладки и документирования информационных систем.	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.	Владеет навыками по документированию приложений. Демонстрирует навыки использования интегрированных сред разработки.	Владеет навыками по документированию приложений. Демонстрирует навыки уверенного использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.
ИД-4 _{ПК-7} Осуществляет внедрение и сопровождение информационных систем и технологий в науке и производстве	Не выполняет учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Не способен проектировать архитектуры приложений для решения задач.	Выполняет корректно учебное задание лабораторных работ. Не способен решать разработки приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает и документирует приложения в рамках индивидуального задания к лабораторным работам. Самостоятельно проектирует архитектуры приложений в рамках выполнения лабораторных работ

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Вопросы для собеседования	
1.		1. Основные концепции технологии ASP NET MVC. 2. Основные элементы проекта ASP.NET MVC. 3. Этапы создания проекта ASP.NET MVC. 4. Создание контроллера и представлений в Visual Studio 5. Стилизация приложения ASP.NET MVC. 6. Основные понятия контроллеров. 7. Методы действий и их параметры. 8. Результаты действий. 9. ViewResult и генерация представлений. 10. Переадресация и отправка кодов статуса и ошибок. 11. Отправка файлов в ASP.NET MVC. 12. Контекст запроса HttpContext. 13. Куки. Сессии. 14. Асинхронные методы в ASP.NET MVC. 15. Основные термины и понятия представлений. 16. Строго типизированные представления. 17. Мастер-страницы. 18. Частичные представления. 19. HTML-хелперы. 20. Методы работы с формами. 21. Строго типизированные хелперы 22. Правила определение маршрутов. 23. Работа с маршрутами. 24. Создание ограничений для маршрутов. 25. Генерация исходящих адресов URL. 26. Области в ASP.NET MVC. 27. Создание собственного обработчика маршрутов.	УК-3, ПК-7

		28. Атрибуты маршрутизации. 29. Маршрутизация и вложенные ресурсы. 30. Модели и БД. 31. Подключение к базе данных. 32. Шаблонные хелперы. 33. Редактирование модели. 34. Добавление и удаление модели. 35. Шаблоны формирования. 36. Модели со сложной структурой. 37. Работа со сложными моделями. 38. Модели со связью многие-ко-многим. 39. 1 Работа с моделями со связью многие-ко-многим. 40. Передача массивов и сложных данных в контроллер. 41. Миграция баз данных. 42. Создание пагинации. 43. Переопределение шаблонов формирования. 44. Переопределение шаблонов отображения и редактирования. 45. Фильтрация данных. 46. Domain Model и View Model. 47. Аннотации данных для отображения свойств. 48. Основы валидации. 49. Атрибуты валидации. 50. Валидация модели в контроллере. 51. Отображение ошибок валидации. 52. Создание собственной логики валидации. 53. Фильтры в ASP.NET MVC. Основные понятия. 54. Фильтры аутентификации. 55. Фильтры авторизации. 56. Фильтры исключений. 57. Фильтры действий и результатов. 58. Примеры фильтров действий. 59. Глобальные фильтры и переопределение фильтров. 60. Дополнительные встроенные фильтры.	
--	--	---	--

		61. Логгирование исключений. 62. Основные понятия привязки моделей. 63. Использование DefaultModelBinder. 64. Явная привязка модели. 65. Поставщики значений. 66. Создание привязчика модели. 67. Основы JavaScript. 68. Введение в AJAX. 69. Ajax-Формы. 70. Параметры объекта AjaxOptions. 71. AJAX-ссылки. 72. Формат JSON. 73. AJAX-запросы с помощью jQuery. 74. Пример реализации проекта с использованием AJAX и jQuery: Чат. 75. Пример реализации проекта с использованием AJAX и jQuery: Игра. 76. Основные концепции и понятия технологии OWIN. 77. Основные концепции и понятия технологии Katana. 78. Назначение, принципы и особенности технологии ASP.NET Identity. 79. Базовые классы AspNet Identity. 80. Авторизация через внешние сервисы. 81. Переопределение системы AspNet Identity. 82. Работа с ролями в AspNet Identity. 83. Аутентификация форм. 84. Использование объектов Claim. 85. Настройка валидации пароля в ASP.NET Identity. 86. Валидация пользователя в ASP.NET Identity. 87. Подтверждение Email в ASP.NET Identity. 88. Подтверждение телефона по SMS в ASP.NET Identity. 89. Сброс пароля в ASP.NET Identity. 90. Универсальные провайдеры в MVC. 91. Переопределение универсальных провайдеров. Провайдер ролей. 92. Создание приложения с ASP.NET Identity с нуля. 93. Регистрация и создание пользователей в ASP.NET Identity.	
--	--	---	--

		94. Авторизация пользователей в ASP.NET Identity. 95. Редактирование и удаление пользователей. 96. Добавление ролей в ASP.NET Identity. 97. Аутентификация OWIN и ClaimsIdentity. 98. Авторизация на основе Claims. 99. Назначение и принципы использования бандлов. 100. Назначение и методы минификации. 101. Создание бандлов. 102. Библиотеки из CDN. 103. Назначение и основы применения Bootstrap. 104. Позиционирование элементов. 105. Компоненты Bootstrap. 106. Работа с формами в Bootstrap. 107. Работа с таблицами. 108. Модальные окна. 109. Система сеток Bootstrap. 110. CSS-стили Bootstrap. 111. Декорирование изображений Bootstrap. 112. Декорирование элементов управления Bootstrap. 113. Назначение и основы использования SignalR. 114. Push-уведомления 115. Persistent Connection API 116. Пример приложения с использованием SignalR: графический редактор. 117. Пример приложения с использованием SignalR: чат. 118. jQuery UI и Autocomplete. 119. Диалоговые окна и CRUD-интерфейс. 120. Валидация в диалоговых окнах. 121. Элемент jqGrid в ASP.NET MVC. 122. Пагинация в jqGrid. 123. Поиск по таблице в jqGrid. 124. CRUD-операции в jqGrid. 125. Постраничный вывод в ASP.NET MVC. 126. Методы тестирования приложений.	
--	--	---	--

		127. Понятие и основные принципы Test-Driven Development. 128. Этапы создания проекта для юнит-тестов. 129. Создание юнит-тестов. 130. Слабосвязанные объекты и тестирование работы с БД. 131. Фреймворк Moq. 132. Тестирование создания модели и переадресации. 133. Жизненный цикл приложения и запроса. 134. Модули. 135. Создание собственного модуля. 136. Взаимодействие между модулями. События модулей. 137. HTTP-обработчики. Создание обработчиков. 138. Создание асинхронных HTTP-обработчиков. 139. Переопределение HTTP-обработчика. 140. Выполнение действий по таймеру. 141. Понятие и назначение кэширования. 142. Атрибут OutputCache. 143. Параметры кэширования. 144. Профили кэширования. 145. Динамическое управление кэшем. 146. Кэширование статического контента. 147. MemoryCache. 148. Инверсия управления. 149. IoC-контейнер Ninject. 150. Внедрение зависимостей с параметрами. 151. IoC-контейнер Autofac. 152. IoC-контейнер Castle Windsor. 153. Принципы SOLID. 154. Основные концепции в URL Rewriting. 155. Переменные сервера и обратные ссылки. 156. Примеры правил URL Rewriting. 157. Onion-архитектура. 158. Паттерн Unit of Work. 159. AutoMapper.	
--	--	---	--

		160. Многоуровневая архитектура. 161. Монолитная архитектура. 162. Data Access Layer. 163. Business Logic Layer. 164. Presentation Layer 165. ASP.NET Identity в многоуровневой архитектуре.	
		Задания закрытого типа	
2.		Какие типы относятся к типам-делегатам? 1) Action<T> 2) Func<T> 3) List<T> 4) IEnumerable<T>	УК-3, ПК-7
3.		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()	УК-3, ПК-7
4.		Какой класс используется для представления потока в многопоточном приложении? 1) Thread 2) Task 3) MultiTask 4) Waiting 5) Action<T>	УК-3, ПК-7
5.		Какое пространство имен является корневым в библиотеке .NET 1) Data 2) System 3) Object 4) Input	УК-3, ПК-7

		5) IO	
6.		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>	УК-3, ПК-7
7.		С помощью какого зарезервированного слова происходит объявление пространства имен? 1) using 2) break 3) namespace 4) foreach 5) name	УК-3, ПК-7
8.		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 a 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	УК-3, ПК-7

9.		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 predetermined значения 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	УК-3, ПК-7
10.		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	УК-3, ПК-7
11.		Даны определения классов на языке C#.	УК-3, ПК-7

		<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 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) ошибок нет	
12.		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	УК-3, ПК-7

		3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным 5) ошибок нет	
13.		Сколько наследников может быть у представленного класса? <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	УК-3, ПК-7
14.		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	УК-3, ПК-7

15.	В языке C# программист определил класс следующим образом) <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	УК-3, ПК-7
16.	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	УК-3, ПК-7

		2) в конструкторе класса Child нет доступа к переменной а базового класса 3) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора 4) нельзя создать объект базового класса, так как объявлен класс-наследник 5) ошибок нет	
17.		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 а базового класса 4) некорректное использование зарезервированного слова this 5) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора	УК-3, ПК-7
18.		В языке C# программист определил классы и интерфейсы следующим образом)	УК-3, 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	
19.		На языке C# программист определил класс, интерфейсы и создал объекты следующим образом)	УК-3, ПК-7

		<pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class Parent { protected int a; } public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } static void Main(string[] args) { MainObject ob1 = new MainObject(); ob1.Draw(); Parent ob2 = new Parent(); ob2.Draw(); } </pre> Какая ошибка была допущена? 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	
20.		На языке C# программист определил класс, интерфейсы и создал объекты следующим образом)	УК-3, ПК-7

		<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	
21.		На языке C# программист определил класс, интерфейсы) <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)	УК-3, ПК-7

		<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	
22.		На языке C# программист определил класс, интерфейсы) <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	УК-3, ПК-7

		2) Невозможно создать объект интерфейса <code>ob</code> 3) в массиве присутствуют элементы различного типа 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают	
23.		Для чего служит оператор <code>break</code>? 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	УК-3, ПК-7
24.		Для чего служит оператор <code>continue</code>? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации цикла 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	УК-3, ПК-7
25.		В переменной <code>Sum</code> 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	УК-3, ПК-7
		Задания открытого типа	
26.		В коллекции <code>city</code> содержатся документы: <pre>{type: "Village", name: "Woodburn"}</pre>	УК-3, ПК-7

		<pre>{type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find({type: "Town"})?</pre>	
27.		<pre>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"})?</pre>	УК-3, ПК-7
28.		<pre>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"})?</pre>	УК-3, ПК-7
29.		<pre>В коллекции city содержатся документы: {type: "Village", name: "Woodburn"} {type: "Town", name: "Bookland"} {type: "Town", name: "Prokland"} Сколько документов вернет запрос db.city.find()?</pre>	УК-3, ПК-7
30.		<pre>В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"} Сколько документов вернет запрос db.goods.find()?</pre>	УК-3, ПК-7
31.		<pre>В коллекции goods содержатся документы: {type: "A", name: "Хлеб"} {type: "A", name: "Молоко"}</pre>	УК-3, ПК-7

		<pre>{type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" })</code>?	
32.	1	В коллекции <code>goods</code> 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 <code>db.goods.findOne({ type : "B" })</code>?	УК-3, ПК-7
33.		В коллекции <code>goods</code> 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 <code>db.goods.find({ type : "B" }).limit(1)</code>?	УК-3, ПК-7
34.		В коллекции <code>goods</code> 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 <code>db.goods.find({ type : "B" }).limit(3)</code>?	УК-3, ПК-7
35.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"}</pre>	УК-3, ПК-7

		<pre>{type: "B", name: "Спрей"} {type: "B", name: "Салфетка"} {type: "C", name: "NoName"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(0)?</code>	
36.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find().limit(0)?</code>	УК-3, ПК-7
37.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "B" }).limit(2)?</code>	УК-3, ПК-7
38.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "A" }).limit(2)?</code>	УК-3, ПК-7
39.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre> Сколько документов вернет запрос <code>db.goods.find({ type : "A" }).limit(0)?</code>	УК-3, ПК-7
40.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {type: "B", name: "Спрей"}</pre>	УК-3, ПК-7

		Что вернет запрос <code>db.goods.find({ type : "A" }).count()</code> ?	
41.		В коллекции <code>goods</code> 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>?	УК-3, ПК-7
42.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {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"}]) db.goods.find().count()</pre> Что будет выведено в консоль?	УК-3, ПК-7
43.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко"} {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: "Духи"}]) db.goods.find({ type: "B" }).count()</pre> Что будет выведено в консоль?	УК-3, ПК-7
44.		В коллекции <code>goods</code> содержатся документы: <pre>{type: "A", name: "Хлеб"} {type: "A", name: "Молоко", date : "07.2019"} {type: "B", name: "Спрей"}</pre>	УК-3, ПК-7

		Были выполнены следующие команды: <pre>db.goods.insertOne({type: "А", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "В", name: "Спрей"}]) db.goods.find({ type: "А"}).count()</pre> Что будет выведено в консоль?	
45.		В коллекции goods содержатся документы: <pre>{ name : "Хлеб" , date : "2020"} {type : "А", name: "Молоко" } {type : "В", name: "Спрей", date : "2019"} {type : "В", name: "Салфетка" , date : "2018"} {type : "С", name: "NoName" , date : "2020"}</pre> Сколько документов вернет запрос <code>db.goods.find({ date : "2020" }).limit(0)?</code>	УК-3, ПК-7
46.		В коллекции goods содержатся документы: <pre>{ name : "Хлеб" , date : "2020"} {type : "А", name: "Молоко" } {type : "В", name: "Спрей", date : "2019"} {type : "В", name: "Салфетка" , date : "2020"} {type : "С", name: "NoName" , date : "2020"}</pre> Сколько документов вернет запрос <code>db.goods.find({ date : "2020" }).limit(0)?</code>	УК-3, ПК-7
47.		В коллекции goods содержатся документы: <pre>{type: "А", name: "Хлеб"} {type: "А", name: "Молоко"} {type: "В", name: "Спрей"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "А", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "С", name: "Духи"}]) db.goods.find({ type: "С"}).count()</pre> Что будет выведено в консоль?	УК-3, ПК-7

48.	В коллекции goods содержатся документы: <pre>{type: "А", name: "Хлеб"} {type: "А", name: "Молоко"} {type: "В", name: "Спрей"} {type: "В", name: "Молоко"}</pre> Были выполнены следующие команды: <pre>db.goods.insertOne({type: "А", product: "Сыр"}) db.goods.insert ([{name: "No name"}, {type: "С", name: "Духи"}]) db.goods.find({ name: "Молоко"}).count()</pre> Что будет выведено в консоль?	УК-3, ПК-7
-----	--	------------

2. Описание шкалы оценивания

Рейтинговая система оценки не предусмотрена для студентов, обучающихся на образовательных программах уровня высшего образования магистратуры.

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. Компетенции УК-3, 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. Компетенции УК-3, 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 УК-3, 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. Компетенции УК-3, 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. Компетенции УК-3, ПК-7 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенции УК-3, ПК-7 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенции УК-3, ПК-7 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенции УК-3, ПК-7 не сформирована.