

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ
Декан факультета ИВТ



Д.Ю. Чалый

«_18_» __мая_____ 2020 г.

Рабочая программа дисциплины
«Разработка программных приложений»

Направление подготовки
09.03.03 Прикладная информатика
Профиль
«Прикладная информатика в экономике»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 23 апреля 2020 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 7 от
17 мая 2020 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Разработка программных приложений» являются ознакомление студентов с разработкой программ и отдельных компонентов .NET Framework, идеологией создания Windows-приложений и Web-приложений, языком C# как одним из основных языков программирования в среде .NET Framework, библиотекой классов Common Language Runtime, изучение средств создания, отладки и развертывания приложений и служб, предоставляемых средой разработки Visual Studio .NET.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Разработка программных приложений» относится к базовой части ОП бакалавриата.

Для изучения данной дисциплины студентам рекомендуется предварительно изучить один из языков высокого уровня, иметь навыки объектно-ориентированного программирования и начальные знания об операционной системе Windows. Знания и навыки, полученные при изучении данной дисциплины, используются учащимися при изучении других дисциплин компьютерного цикла, а также при подготовке курсовых и дипломных работ.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Код компетенции	Формулировка компетенции	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1 демонстрирует знания методов алгоритмизации, языков и технологий программирования, пригодных для практического применения. ОПК-7.2 демонстрирует умение применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач ОПК-7.3 демонстрирует навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач	Знать: – постановки задач линейной алгебры; – основные понятия и формулировки теорем линейной алгебры. Уметь: – решать задачи по линейной алгебре; – доказывать основные и вспомогательные утверждения и теоремы из курса линейной алгебры. Владеть: – математическим аппаратом линейной алгебры; – навыками использования аппарата линейной алгебры при решении конкретных задач.
Профессиональные компетенции		
ПК-2 Способность разрабатывать и адаптировать прикладное	ПК-2.1. Знает современные методы разработки и реализации алгоритмов математических моделей	Знать: – принципы объектно-ориентированного программирования, устройство и основные механизмы .NET Framework; – основы программирования на языке C#;

программное обеспечение	на базе языков и пакетов прикладных программ моделирования. ПК-2.2. Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования.	<p>– принципы разработки и тестирования программ и отдельных программных модулей для работы в среде .NET Framework;</p> <p>– технологию LINQ и основы работы с базами данных;</p> <p>– основные приемы отладки и тестирования программ и отдельных программных модулей.</p> <p>Уметь:</p> <p>– проектировать пользовательский интерфейс и функциональность Windows- и Web-приложений;</p> <p>– пользоваться инструментальными средствами Visual Studio для отладки и тестирования.</p> <p>Владеть навыками:</p> <p>– программирования, отладки и тестирования Windows- и Web-приложений для .NET Framework;</p> <p>– использования инструментальных средств Visual Studio для создания и запуска автоматических тестов.</p>
-------------------------	--	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 7 зач. ед., 252 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1.	Введение в C# и .NET Framework	5	1		2				
2.	Программные конструкции C#	5	2		4			6	Выполнение учебных упражнений
3.	Объявление и вызов методов	5	1		2	1		3	Выполнение учебных упражнений
4.	Обработка исключений	5	1		2			3	Выполнение учебных упражнений
5.	Работа с файлами	5	2		4	1		4	Выполнение учебных упражнений
6.	Создание новых типов	5	1		2			6	Выполнение учебных

									упражнений
7.	Инкапсуляция данных и методов	5	1		2			4	Выполнение учебных упражнений
8.	Наследование от классов и реализация интерфейсов	5	2		4	1		6	Выполнение учебных упражнений
9.	Управление временем жизни объектов и работа с ресурсами	5	1		2			1,7	Выполнение учебных упражнений
10.	Инкапсуляция данных и определение перегруженных операций	5	1		2			3	Выполнение учебных упражнений
11.	Использование делегатов и обработка событий	5	1		2	1		3	Выполнение учебных упражнений
12.	Использование коллекций и создание параметризованных типов	5	2		4	1		4	Выполнение учебных упражнений
13.	Создание и использование пользовательских коллекций	5	2		4	1		4	Выполнение учебных упражнений
	Всего за 5 семестр		18		36	6	0,3	47,7	Зачет
14.	Использование технологии LINQ для запроса данных	6	1		4			3	Выполнение учебных упражнений
15.	Многопоточность и асинхронное программирование	6	1		2	1		2	Выполнение учебных упражнений
16.	Введение в ASP.NET.	6	1					2	Выполнение учебных упражнений
17.	Создание Web-приложений.	6			4			2	Выполнение учебных упражнений
18.	Проектирование Web-форм.	6	1		2			2	Выполнение учебных упражнений
19.	Жизненный цикл Web-страницы.	6	1						Выполнение учебных упражнений
20.	Развитие интерфейса и функциональности Web-приложения.	6	1		2	1		4	Выполнение учебных упражнений
21.	Создание и использование пользовательских элементов управления.	6	1		2			4	Выполнение учебных упражнений
22.	Проверка корректности ввода.	6	1		2			4	Выполнение учебных упражнений
23.	Отладка Web-приложений.	6			4			4	Выполнение учебных упражнений
24.	Основы работы с базами данных в Web-приложениях.	6	1		2			2	Выполнение учебных упражнений
25.	Программируемый доступ к базе данных.	6	1		2			2	Выполнение учебных упражнений
26.	Использование LINQ в Web-приложениях.	6	1		2	1		2	Выполнение учебных упражнений
27.	Работа с динамическими данными в ASP.NET.	6	1		2	1		3	Выполнение учебных упражнений
28.	Использование AJAX в ASP.NET.	6	1		2			2	Выполнение учебных упражнений
29.	AJAX Control Toolkit.	6	1		2			3	Выполнение учебных упражнений

30.	Использование служб Microsoft Windows Communication Foundation.	6	1		2			3	Выполнение учебных упражнений
31.	Управление состоянием Web-приложений.	6	1					2	Выполнение учебных упражнений
32.	Конфигурирование Web-приложения.	6	1					2	Выполнение учебных упражнений
33.	Безопасность Web-приложений.	6	1						Выполнение учебных упражнений
	Всего за 6 семестр		18		36	6	0,5	48	Экзамен
	Всего		36		72	12	1	95,7	

Для самостоятельной работы используются учебные упражнения, представляющие собой создание учебного приложения или библиотеки по подробной инструкции. Упражнения доступны в электронной форме в сети факультета. Распределение их по темам соответствует темам лекций.

Экзамен (зачет) в каждом семестре состоит из двух частей. В теоретической части студент должен ответить на вопросы по темам, приведенным в приложении. Практическая часть в первом семестре представляет собой разработку многокомпонентного проекта, написанного на языке C#. Тип интерфейса WPF или Windows Forms по усмотрению студента. Во втором семестре студент должен разработать Web-приложение, основанное либо на Web-формах, либо на технологии MVC.

Содержание разделов дисциплины:

- 1. Введение в C# и .NET Framework.**
 - 1.1. Введение в .NET Framework 4.
 - 1.2. Создание проектов в среде разработки Visual Studio.
 - 1.3. Создание приложения на C#.
 - 1.4. Создание приложений с графическим интерфейсом.
 - 1.5. Документирование приложения.
 - 1.6. Отладка приложений в среде Visual Studio.
- 2. Программные конструкции C#.**
 - 2.1. Объявление и инициализация переменных.
 - 2.2. Выражения и операции.
 - 2.3. Создание и использование массивов.
 - 2.4. Операторы выбора.
 - 2.5. Использование операторов цикла.
- 3. Объявление и вызов методов.**
 - 3.1. Объявление и вызов методов.
 - 3.2. Параметры по умолчанию и выходные параметры.
- 4. Обработка исключений.**
 - 4.1. Перехват исключений.
 - 4.2. Выбрасывание исключений.
- 5. Работа с файлами.**
 - 5.1. Доступ к файловой системе.
 - 5.2. Чтение и запись файлов с использованием потоков.
- 6. Создание новых типов.**
 - 6.1. Создание и использование перечислений.
 - 6.2. Создание и использование классов.
 - 6.3. Создание и использование структур.
 - 6.4. Сравнение ссылочных типов и типов значений.
- 7. Инкапсуляция данных и методов.**

- 7.1. Контроль видимости членов типа.
- 7.2. Разделяемые методы и данные.
- 8. Наследование от классов и реализация интерфейсов.**
 - 8.1. Использование наследования для создания новых ссылочных типов.
 - 8.2. Определение и реализация интерфейсов.
 - 8.3. Определение абстрактных классов.
- 9. Управление временем жизни объектов и работа с ресурсами.**
 - 9.1. Процесс сборки мусора.
 - 9.2. Управление ресурсами.
- 10. Инкапсуляция данных и определение перегруженных операций.**
 - 10.1. Создание и использование свойств.
 - 10.2. Создание и использование индексаторов
 - 10.3. Перегрузка операций
- 11. Использование делегатов и обработка событий.**
 - 11.1. Объявление и использование делегатов.
 - 11.2. Использование лямбда-выражений.
 - 11.3. Обработка событий.
- 12. Использование коллекций и создание параметризованных типов.**
 - 12.1. Использование коллекций.
 - 12.2. Создание и использование обобщенных типов.
 - 12.3. Определение обобщенных интерфейсов и вариантность.
 - 12.4. Использование обобщенных методов и делегатов.
- 13. Создание и использование пользовательских коллекций.**
 - 13.1. Реализация собственного класса коллекции.
 - 13.2. Реализация перечислителя в собственном классе коллекции.
- 14. Использование технологии LINQ для запроса данных.**
 - 14.1. Использование методов расширения LINQ и операций запроса.
 - 14.2. LINQ to SQL.
 - 14.3. ADO.NET и LINQ поверх DataSet.
 - 14.4. LINQ to XML.
 - 14.5. Построение динамических запросов LINQ и выражений.
- 15. Многопоточность и асинхронное программирование.**
 - 15.1. Работа с потоками в .NET Framework.
 - 15.2. Введение в параллельное программирование на платформе .NET.
- 16. Введение в ASP.NET.**
 - 16.1. Обзор технологии ASP.NET.
 - 16.2. Структура и файлы ASP.NET Web-приложения.
- 17. Создание Web-приложений.**
 - 17.1. Инструменты и шаблоны Microsoft Visual Studio для Web-программирования
 - 17.2. Создание Web-приложений ASP.NET в среде разработки Visual Studio
- 18. Проектирование Web-форм.**
 - 18.1. Способы создания Web-форм.
 - 18.2. Категории элементов управления.
 - 18.3. Атрибуты страниц и элементов управления.
 - 18.4. Методы передачи их состояния на сервер.
 - 18.5. Использование страниц с фоновым кодом.
 - 18.6. Добавление процедур обработки событий для Web-серверных элементов управления.
 - 18.7. Создание и использование классов и компонентов.
- 19. Жизненный цикл Web-страницы.**
 - 19.1. Жизненный цикл Web-страницы.
 - 19.2. Событие Page_Load.

- 19.3. Postback-процесс.
- 20. Развитие интерфейса и функциональности Web-приложения.**
 - 20.1. Создание и использование ведущих страниц.
 - 20.2. Контент-страницы.
 - 20.3. Вложенные ведущие страницы.
- 21. Создание и использование пользовательских элементов управления.**
 - 21.1. Проектирование интерфейса и функциональности пользовательских элементов управления.
- 22. Проверка корректности ввода.**
 - 22.1. Проверка корректности пользовательского ввода.
 - 22.2. Стандартные и пользовательские элементы управления для проверки корректности ввода.
- 23. Отладка Web-приложений.**
 - 23.1. Использование объектов Trace и Debug для отладки Web-приложений.
 - 23.2. Удаленная отладка.
- 24. Основы работы с базами данных в Web-приложениях.**
 - 24.1. Работа с реляционной моделью данных в Visual Studio.NET.
 - 24.2. Обзор технологии ADO.NET.
 - 24.3. Установление связи с базой данных.
 - 24.4. Связывание серверных элементов с источниками данных.
- 25. Программируемый доступ к базе данных.**
 - 25.1. Использование объекта DataSet.
 - 25.2. Загрузка нескольких таблиц в DataSet и организация связи между ними.
 - 25.3. Использование объекта DataReader для программируемого доступа к базе данных.
- 26. Использование LINQ в Web-приложениях.**
 - 26.1. Использование LINQ для работы с данными.
 - 26.2. Работа с данными в формате XML с использованием провайдера LINQ для XML.
 - 26.3. Работа с данными через провайдера LINQ для SQL и LINQ для сущностей.
- 27. Работа с динамическими данными в ASP.NET.**
 - 27.1. Введение в динамические данные.
 - 27.2. Применение динамических данных в ASP.NET.
 - 27.3. Настройка динамических данных в ASP.NET.
- 28. Использование AJAX в ASP.NET.**
 - 28.1. Технология AJAX и ее реализация в ASP.NET и Visual Studio.
 - 28.2. Расширение стандартных элементов управления ASP.NET возможностями AJAX.
- 29. AJAX Control Toolkit.**
 - 29.1. Установка AJAX Control Toolkit.
 - 29.2. Специализированные элементы управления AJAX из пакета AJAX Control Toolkit.
- 30. Использование служб Microsoft Windows Communication Foundation.**
 - 30.1. Обзор сервисов Windows Communication Foundation.
 - 30.2. Обращение к службам Windows Communication Foundation.
 - 30.3. Работа со службами WCF Data.
- 31. Управление состоянием Web-приложений.**
 - 31.1. Управление состоянием.
 - 31.2. Профили ASP.NET.
 - 31.3. Кэширование ASP.NET.
- 32. Конфигурирование Web-приложения.**
 - 32.1. Конфигурация Web-приложения ASP.NET.
 - 32.2. Настройка и развертывание Web-приложения Microsoft ASP.NET.
- 33. Безопасность Web-приложений.**
 - 33.1. Обзор вопросов безопасности Web-приложений.
 - 33.2. Настройка процессов аутентификации и авторизации.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекции проводятся в компьютерном классе с использованием проектора для демонстрации презентаций и демонстрационных примеров. Такой подход позволяет разобрать достаточно большой объем материала и продемонстрировать весьма объемные примеры кода, что было бы невозможным при использовании традиционных средств. Лекции по времени совмещены с лабораторными занятиями, что дает возможность сразу закрепить прослушанный материал на практике и при необходимости получить консультации лектора.

Практическое занятие – занятие в компьютерном классе, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

Основной формой практической работы студентов по усвоению данного курса является выполнение ими самостоятельных проектов в рамках лабораторных (по каждой теме) и самостоятельных занятий по данному курсу. Все задания, заготовки проектов, демонстрационные и вспомогательные учебные материалы предоставляются учащимся в электронном виде и постоянно доступны в сети факультета. Это позволяет в случае пропуска занятия самостоятельно отработать пропущенную тему.

Промежуточная аттестация производится в форме отчетов студентов по самостоятельно выполненным проектам, окончательная аттестация в форме экзамена.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются:

- Windows 7 в качестве основной операционной системы;
- **OfficeStd 2013 RUS OLP NL Acdmc 021-10232** для создания слайдов презентаций, используемых для чтения лекций;
- Интегрированная среда разработки Microsoft Visual Studio для подготовки демонстрационных примеров, а также заготовок лабораторных работ и учебных упражнений;
- **OfficeStd 2013 RUS OLP NL Acdmc 021-10232** для формирования инструкций по выполнению учебных упражнений, а также текстов материалов для промежуточной и текущей аттестации;
- Microsoft Access, Microsoft SQL Server и учебная БД Northwind в для обучения работе с базами данных.

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Васильчиков, В. В. Программирование на языке С# для .NET Framework : курс лекций. Часть 1 : учебное пособие / В. В. Васильчиков ; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2013. — 196 с.
2. Васильчиков, В. В. Программирование на языке С# для .NET Framework : курс лекций. Часть 2 : учебное пособие / В. В. Васильчиков ; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2014. — 200 с.
3. Васильчиков, В. В. Дополнительные вопросы программирования для .NET Framework: учебно-методическое пособие / В. В. Васильчиков ; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2017. — 60 с.

б) дополнительная:

4. С# 4.0 и платформа .NET 4 для профессионалов : пер. с англ. / К. Нейгел, Б. Ивьен, Д. Глинн, К. Уотсон. — М. : И. Д. Вильямс, 2011. — 1440 с.
5. Троелсен, Э. Язык программирования С# 2010 и платформа .NET : пер. с англ. / Э. Троелсен. — М. : И. Д. Вильямс, 2011. — 1392 с.
6. Трей, Н. С# 2010 — Ускоренный курс для профессионалов : пер. с англ. / Н. Трей. — М. : И. Д. Вильямс, 2010. — 592 с.
7. Макки, А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов : пер. с англ. / А. Макки. — М. : И. Д. Вильямс, 2010. — 416 с.
8. Шилдт, Г. С# 4.0 Полное руководство : пер. с англ. / Г. Шилдт. — М. : И. Д. Вильямс, 2011. — 1056 с. Эспозито Д. Программирование с использованием Microsoft ASP.NET 3.5 : пер. с англ. / Д. Эспозито. — СПб. : И. Д. Питер, 2013. — 544 с.
9. Эспозито Д. Разработка веб-приложений с использованием ASP.NET и AJAX : пер. с англ. / Д. Эспозито. — СПб. : И. Д. Питер, 2012. — 400 с.
10. Фримен, А. ASP.NET 4.5 с примерами на С# 5.0 для профессионалов : пер. с англ. / А. Фримен. — М. : И. Д. Вильямс, 2014. — 1120 с.
11. Мак-Дональд М., Фримен, А., Шпушта М. Microsoft ASP.NET 4 с примерами на С# 2010 для профессионалов : пер. с англ. / М. Мак-Дональд. — М. : И. Д. Вильямс, 2011. — 1424 с.

в) ресурсы сети «Интернет»

- Visual Studio 2010.
- Информация по платформе Microsoft .NET Framework, языкам программирования, технологиям, справочная система: <https://msdn.microsoft.com/ru-ru/vstudio/default.aspx>

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Зав. кафедрой

вычислительных и программных систем, к.т.н. _____ В.В. Васильчиков

Приложение №1
к рабочей программе дисциплины
"Разработка программных приложений"

Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине

1. Типовые контрольные задания или иные материалы, необходимые для оценки
знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы
формирования компетенций

1.1. Контрольные задания и иные материалы, используемые в процессе текущей
аттестации

Задания для самостоятельной работы

В третьем семестре студенты должны изучить основы .NET Framework, язык С# и научиться создавать Windows-приложения для работы в среде исполнения .NET Framework.

Для самостоятельной работы используются учебные упражнения, по каждой теме данного учебного курса предлагается несколько. Большинство из них являются обязательными для выполнения. Заготовки проектов, сценарии работы и пошаговые инструкции для выполнения каждого из упражнений выложены в локальной сети факультета.

Большинство из них являются обязательными для выполнения. Заготовки проектов, сценарии работы и пошаговые инструкции для выполнения каждого из упражнений выложены в локальной сети факультета. Упражнения заключаются в создании проекта приложения с графическим интерфейсом и последовательном развитии его функциональности. По завершении всех этапов развития приложения готовый проект сдается преподавателю. Для прохождения аттестации в течение семестра студенты должны выполнить все обязательные упражнения и сдать завершённый проект.

Для оценки качества сданного проекта применяются следующие критерии оценивания

Оценка	Критерии
Отлично	ПК-8: Умеет выбрать подходящий тип проекта, разработать его архитектуру и грамотно реализовать ее в виде отдельных компонентов на языке С#. Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует подходящие события и реализует их обработчики. Создает полноценное .NET-приложение с графическим интерфейсом. Хорошо владеет средствами LINQ для работы с данными. Программа выполняется для всех тестов. ПК-15: Умеет использовать механизм исключений, грамотно пользуется отладчиком. Поясняет код и изменяет его при необходимости
Хорошо	ПК-8: Умеет выбрать подходящий тип проекта, разработать его архитектуру и реализовать ее в виде отдельных компонентов на языке С#. Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует события и реализует их обработчики. Создает .NET-

	<p>приложение с графическим интерфейсом. Владеет средствами LINQ для работы с данными. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается.</p> <p>ПК-15: Умеет использовать механизм исключений, пользуется отладчиком. Поясняет код и изменяет его при необходимости с небольшими неточностями</p>
Удовлетворительно	<p>ПК-8: Умеет выбрать подходящий тип проекта, использовать его для отдельных компонентов на языке C#. В основном может спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Понимает связь между событием и обработчиком, с некоторыми подсказками может выбрать подходящее событие. Создает .NET-приложение с графическим интерфейсом. Не в полной мере владеет средствами LINQ. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев.</p> <p>ПК-15: Неуверенно пользуется отладчиком. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи</p>
Неудовлетворительно	<p>ПК-8: Не может спроектировать графический интерфейс и реализовать его. Не понимает устройства многокомпонентного приложения. Не может создать приложение с графическим интерфейсом в среде разработки или создает неработающий проект. Не владеет средствами LINQ. Программа не выполняется для большинства тестов.</p> <p>ПК-15: Не владеет средствами и приемами отладки. Не может пояснить код и изменить его.</p>

Примерный список дополнительных вопросов при защите собственного проекта в третьем семестре

1. В чем состоит назначение Common Language Runtime?
2. Сколько классов может содержать приложение на C#?
3. Перечислите три способа передачи данных методу через его параметры и соответствующие ключевые слова языка C# (если таковые есть)
4. Опишите разницу между интерфейсом, абстрактным классом и конкретным классом
5. Пусть у вас есть два .cs файла. Файл alpha.cs содержит класс Alpha, в котором есть internal метод Method. Файл beta.cs file содержит класс Beta, в котором также есть internal метод Method. Может ли Alpha.Method быть вызван из Beta.Method и наоборот?
6. В чем разница в использовании оператора преобразования типа и оператора as?
7. Что такое делегат?
8. В каком порядке вызываются методы, подписанные на событие? Может ли случиться так, что не все методы будут вызваны?
9. Какая часть сборки содержит информацию об импортируемых и экспортируемых типах и о версии сборки?
10. Какая команда используется для того, чтобы поместить сборку в глобальный кэш сборок?
11. Когда следует использовать приватный конструктор?
12. В чем состоит назначение слабых ссылок?
13. Перечислите хотя бы три типа потоков (в смысле streams) .NET Framework и их назначение

14. Чем определяется, будут ли параметры и возвращаемые значения для методов удаленного объекта передаваться по значению или по ссылке?
15. В чем состоит назначение Common Type System?
16. Сколько методов Main может содержать приложение на C#?
17. Какие элементы описания метода образуют его сигнатуру?
18. Что такое полиморфизм? Какое он имеет отношение к раннему и позднему связыванию?
19. Перечислите способы, с помощью которых можно определить тип объекта
20. Содержит ли данный код синтаксические ошибки, если да, то какие?

```

namespace Outer.Inner
{
class Wibble { }
}
namespace Test
{
using Outer.Inner;
class SpecialWibble: Inner.Wibble { }
}

```

21. В каких случаях преобразование типа должно быть явным?
22. Что такое подписка на событие и как она происходит?
23. Какие сборки могут быть помещены в глобальный кэш сборок?
24. Какая команда используется для того, чтобы сгенерировать открытый и закрытый ключ?
25. Когда происходят операции упаковки и распаковки?
26. Как используются поколения в механизме сбора мусора?
27. Напишите строчку кода для создания объекта WebRequest для URI:
http://www.microsoft.com/default.htm?foo=bar
28. В чем состоит назначение проху применительно к ремоутингу?

1.2. Список вопросов по темам для проведения зачета по курсу "Разработка программных приложений" в третьем семестре

1. Обзор платформы Microsoft .NET и языка C#

Архитектура и возможности платформы Microsoft .NET. Обзор языка C#. Структура программы. Простейший ввод/вывод. Средства отладки. Создание простой программы на C#

2. Основы языка C#

Использование простых типов данных. Создание и использование перечислений и структур. Операторы и операции языка C#. Исключения в языке C#.

Методы и параметры. Механизмы передачи параметров и результатов методов. Создание и использование методов. Создание и использование массивов в языке C#.

3. Реализация принципов объектно-ориентированного программирования в языке C#

Основные принципы объектно-ориентированного программирования и их реализация в языке C#. Абстрактные и конкретные классы, интерфейсы.

4. Ссылочные типы данных

Создание и использование классов. Ссылочные типы данных, преобразование типов, упаковка и распаковка. Классы Exception, String, StringBuilder. Определение типов объектов. Иерархия объектов в C#. Наиболее употребительные пространства имен.

Определение и использование переменных ссылочных типов. Преобразование типов данных в C#. Правила явного и неявного преобразования типов. Упаковка и распаковка. Работа с интерфейсами.

5. Создание и уничтожение объектов

Конструкторы для классов и структур. Конструкторы по умолчанию.

Создание объектов. Инициализация readonly-данных. Цикл жизни объектов. Сборка мусора. Использование деструкторов. Интерфейс IDisposable. Управление ресурсами. Шаблон реализации интерфейса IDisposable.

6. Механизм наследования в C#

Механизм и правила наследования в C#. Использование виртуальных методов и интерфейсов. Использование абстрактных классов. Использование наследования для реализации интерфейсов.

7. Агрегация и пространства имен

Использование классов, методов и данных с модификатором доступа internal. Агрегация. Использование шаблона программирования Factory.

Способы разрешения конфликтов имен. Использование пространств имен. Директива using-namespace. Псевдонимы. Использование модулей и сборок.

8. Операции, делегаты и события

Определение и использование перегруженных операций. Создание и использование делегатов. Определение и использование событий. Определение и использование свойств и индексов.

9. Атрибуты

Атрибуты. Основные типы предопределенных атрибутов. Определение и использование собственных атрибутов.

10. Работа с компонентами

Механизм исполнения управляемого кода в .NET Framework. Создание компонента и использование его из Windows- и Web-приложений.

11. Развертывание приложений

Варианты развертывания приложений. Совместное использование сборок. Сборки с сильным именем. Глобальный кэш сборки. Сборка и развертывание приложений с частными и разделяемыми компонентами. Контроль версий

12. Common Type System

Стандартная система типов. Работа с типами. Работа с интерфейсами. Переопределение методов класса System.Object. Вопросы эффективности, связанные с процессами упаковки и распаковки.

13. Строки, массивы, коллекции

Работа со строками. Основные методы классов String и StringBuilder. Массивы и коллекции. Их свойства и методы. Основные интерфейсы. Работа с интерфейсами IEnumerable и Comparable.

14. Использование делегатов и событий

Делегаты и события: сценарии использования, синтаксис в C#, механизм работы. Использование делегатов и событий на примере создания простого чат-сервера.

15. Управление памятью и прочими ресурсами

Управление ресурсами. Использование кода финализации. Работа сборщика мусора для простых объектов и для объектов с деструкторами. Явное управление ресурсами. Интерфейс IDisposable, поколения. Работа с потоками данных и файлами.

16. Разработка сетевых приложений

Сценарии работы сетевых приложений. Работа с сетевыми протоколами. Разработка простого клиент-серверного приложения на основе протокола TCP.

17. Сериализация

Использование сериализации и форматов.

18. Ремоутинг

Сценарии активации серверной и клиентской стороны. Организация удаленного взаимодействия объектов.

19. XML Web-службы

XML Web-службы. Использование утилиты Wsdl для построения клиентской части. Использование XML Web-службы: создание прокси и тестового приложения-клиента.

1.3. Правила проведения зачета в третьем семестре по курсу "Разработка программных приложений"

Для получения оценки "зачтено" студент должен выполнить следующие требования:

- в течение семестра выполнить и сдать все обязательные упражнения для самостоятельной работы;
- разработать и защитить собственный проект .NET-приложения, созданного в среде Microsoft Visual Studio, требования к содержанию проекта перечислены ниже;
- в процессе защиты своего проекта правильно ответить на вопросы по данному проекту, а также на вопросы для проведения зачета, перечисленные выше.

Основные требования к содержанию самостоятельного проекта для сдачи зачета

- Тема для приложения выбирается студентом самостоятельно.
- Приложение должно быть создано в среде Microsoft Visual Studio на языке C#, причем разработка ведется "с нуля", не допускается переделка ранее созданных кем-либо приложений. Приложение может иметь графический интерфейс WPF (рекомендуется) или Windows Forms.
- Приложение должно иметь многокомпонентную архитектуру, т.е. иметь как минимум одну сборку в виде файла dll. Все компоненты создаются в рамках общего решения (файл .sln).
- В программе должно быть разработано несколько классов. На оценке работы сказывается развитость их функциональности. Рекомендуемые элементы перечисляются ниже:
 - Наличие собственного хранилища данных (сущностей) в виде массива или коллекции (лучше типизированной);
 - Наличие методов, свойств, индексов для работы с этим хранилищем;
 - Использование интерфейсов и/или абстрактных классов при проектировании иерархии классов;
 - Использование механизма исключений для работы с нестандартными ситуациями;
 - Предпочтительно использовать свойства (возможно, автоматические), а не поля для хранения данных;
 - Переопределение операций.

- Использование конструкций LINQ для работы с данными.
- Работа с файлами, использование стандартных диалогов для их открытия.
- Работа любым типом базы данных: SQL-сервер, файл SQL, XML и т.п.
- Оценка за проект зависит от степени его развитости, качества интерфейса и полезности для пользователя.

В случае неудовлетворительной защиты проекта, либо его отсутствия, студент должен сдать зачет в специально назначенное для этого время. Ему потребуется в учебном классе в отведенный временной промежуток создать и отладить в среде Microsoft Visual Studio многокомпонентное .NET-приложение со строго сформулированными характеристиками.

В случае сдачи зачета в такой форме оценка "зачтено" выставляется только в случае удовлетворения приложения всем требованиям, перечисленным в задании.

В четвертом семестре студенты должны изучить основы ASP.NET и научиться создавать Web-приложения для работы в среде исполнения .NET Framework.

Для самостоятельной работы используются учебные упражнения, по каждой теме данного учебного курса предлагается несколько. Большинство из них являются обязательными для выполнения. Заготовки проектов, сценарии работы и пошаговые инструкции для выполнения каждого из упражнений выложены в локальной сети факультета.

Упражнения заключаются в создании проекта Web-приложения и последовательном развитии его функциональности. По завершении всех этапов развития приложения готовый проект сдается преподавателю. Для прохождения аттестации в течение семестра студенты должны выполнить все обязательные упражнения и сдать завершённый проект.

Для оценки качества сданного проекта применяются следующие критерии оценивания

Оценка	Критерии
Отлично	<p>ПК-8: Умеет выбрать подходящий тип проекта, разработать его архитектуру и грамотно реализовать ее в виде отдельных компонентов на языке C#. Умеет спроектировать Web-интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует подходящие события и реализует их обработчики. Создает полноценное Web-приложение. Хорошо владеет средствами LINQ для работы с данными. Программа выполняется для всех тестов.</p> <p>ПК-15: Умеет использовать механизм исключений, грамотно пользуется отладчиком. Поясняет код и изменяет его при необходимости</p>
Хорошо	<p>ПК-8: Умеет выбрать подходящий тип проекта, разработать его архитектуру и реализовать ее в виде отдельных компонентов на языке C#. Умеет спроектировать Web-интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует события и реализует их обработчики. Создает Web-приложение. Владеет средствами LINQ для работы с данными. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается.</p> <p>ПК-15: Умеет использовать механизм исключений, пользуется отладчиком. Поясняет код и изменяет его при необходимости с небольшими неточностями</p>

Удовлетворительно	<p>ПК-8: Умеет выбрать подходящий тип проекта, использовать его для отдельных компонентов на языке C#. В основном может спроектировать Web-интерфейс и реализовать его с использованием подходящих элементов управления. Понимает связь между событием и обработчиком, с некоторыми подсказками может выбрать подходящее событие. Создает Web-приложение. Не в полной мере владеет средствами LINQ. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев.</p> <p>ПК-15: Неуверенно пользуется отладчиком и трассировкой. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи</p>
Неудовлетворительно	<p>ПК-8: Не может спроектировать Web-интерфейс и реализовать его. Не может создать приложение с Web-интерфейсом в среде разработки или создает неработающий проект. Не владеет средствами LINQ. Программа не выполняется для большинства тестов.</p> <p>ПК-15: Не умеет пользоваться отладчиком и трассировкой. Не может пояснить код и изменить его.</p>

Примерный список дополнительных вопросов при защите собственного проекта в четвертом семестре

1. В чем состоит назначение Common Language Runtime?
2. В чем состоит назначение Common Type System?
3. Какие элементы описания метода образуют его сигнатуру?
4. Что такое полиморфизм? Какое он имеет отношение к раннему и позднему связыванию?
5. Перечислите способы, с помощью которых можно определить тип объекта
6. Что такое подписка на событие и как она происходит?
7. Напишите строчку кода для создания объекта WebRequest для URI:
http://www.microsoft.com/default.htm?foo=bar
8. В чем состоит назначение проху применительно к использованию WCF служб?
9. Как в обработчике события Page_Load определить, в первый раз запрашивается страница или это результат обратной посылки?
10. В чем разница в использовании объектов Trace и Debug?
11. Сколько объектов DataTables может содержать DataSet?
12. Как после создания объектов SqlConnection, SqlDataAdapter, и DataSet objectотобразить данные в объекте DataGridView?
13. В чем разница между объектами DataSet и DataView?
14. В чем разница между объектами DataSet и DataReader?
15. Как получить доступ к XML Web службе или WCF службе из Web формы?
16. Назовите три категории событий, которые обрабатываются в коде файла global.asax
17. В чем разница между объектами Cache и кэшем вывода?
18. Для чего нужны мастер-страницы?
19. Что такое контент-страница?
20. Для чего нужны заказные элементы управления?
21. Какие файлы используются для конфигурирования ASP.NET Web приложения?
22. Почему можно удалить файлы с фоновым кодом после развертывания ASP.NET Web приложения?
23. В чем разница между аутентификацией и авторизацией?

24. Назовите три метода проведения аутентификации, поддерживаемых ASP.NET

1.4. Список вопросов по темам для проведения экзамена по курсу "Разработка программных приложений" в четвертом семестре

1. Обзор технологии ASP.NET

Структура и файлы ASP.NET Web-приложения. Обзор среды разработки Visual Studio .NET. Создание Web-приложений ASP.NET. Создание компонента и его использование на Web-форме.

2. Создание ASP.NET Web-форм

Способы создания ASP.NET Web-форм. Категории элементов управления. Атрибуты страниц и элементов управления. Методы передачи их состояния на сервер. Использование страниц с фоновым кодом. Добавление процедур обработки событий.

3. Обработка событий Web-страницы

Жизненный цикл Web-страницы. Событие Page_Load, Postback-процесс. Связывание элементов управления на странице. Обработка событий Web-страницы. Использование объектов Trace и Debug для отладки ASP.NET Web-приложений. Удаленная отладка.

4. Проверка корректности пользовательского ввода

Создание и использование элементов управления для проверки корректности ввода. Создание и использование пользовательских элементов управления.

5. Технология ADO.NET

Работа с реляционной моделью данных в Visual Studio .NET. Обзор технологии ADO.NET.

6. Отображение данных на Web-форме

Отображение данных на Web-форме с использованием List-bound элементов управления. Доступ к данным с использованием технологии ADO.NET. Программируемый доступ к базе данных, Использование объектов DataSet и DataView. Загрузка нескольких таблиц в DataSet и организация связи между ними. Использование объекта DataReader для программируемого доступа к базе данных. Хранимые процедуры, типы их параметров, их использование с объектом DataAdapter и объектом Command.

7. Работа с XML данными

Чтение и запись XML данных. XML и объект DataSet. Использование XML WEB-серверных элементов управления. Обзор XML WEB-служб. Доступ к XML WEB-службе посредством протокола Http. Программируемый доступ к XML WEB-службе. Создание и использование XML WEB-служб.

8. Управление состоянием Web-приложения

Файл Global.asax. Использование и хранение переменных приложения и сеанса. Использование cookies.

9. Оптимизация, конфигурирование и развертывание ASP.NET Web-приложений

Оптимизация и конфигурирование ASP.NET Web-приложений. Развертывание ASP.NET Web-приложений.

10. Безопасность ASP.NET Web-приложений

Обеспечение безопасности ASP.NET Web-приложений. Основные методы и сценарии проведения аутентификации.

1.5. Правила проведения экзамена в четвертом семестре по курсу "Разработка программных приложений"

Для получения положительной оценки студент должен выполнить следующие требования:

- в течение семестра выполнить и сдать все обязательные упражнения для самостоятельной работы;
- разработать и защитить собственный проект ASP.NET-приложения, созданного в среде Microsoft Visual Studio, требования к содержанию проекта перечислены ниже;
- в процессе защиты своего проекта правильно ответить на вопросы по данному проекту, а также на вопросы для проведения экзамена, перечисленные выше.

Основные требования к содержанию самостоятельного проекта для сдачи экзамена

- Тема для приложения выбирается студентом самостоятельно.
- Приложение должно быть создано в среде Microsoft Visual Studio на языке C#, причем разработка ведется "с нуля", не допускается переделка ранее созданных кем-либо приложений. Приложение может иметь Web Forms(рекомендуется) или MVC.
- Приложение должно иметь не менее трех страниц.
- Должны удовлетворяться следующие обязательные требования:
 - Работа любым типом базы данных: SQL-сервер, файл SQL, XML и т.п. Автоматически создаваемая база данных для аутентификации пользователей не считается.
 - Наличие аутентификации пользователей и разных ролей (аутентифицированный пользователь, анонимный пользователь, администратор). Разные возможности для разных ролей.
 - Использование LINQ для работы с данными.
- Дополнительные требования, не являющиеся обязательными, но влияющие на оценку качества проекта:
 - Использование AJAX (хотя бы частично-страничной отправки).
 - Использование мастер-страниц.
 - Использование заказных контролов.
 - Использование контролов валидации.
 - Использование WCF-служб.
 - Использование средств управления состоянием приложения и/или сеанса.
- Оценка за проект зависит от степени его развитости, качества интерфейса и полезности для пользователя.

В случае неудовлетворительной защиты проекта, либо его отсутствия, студент должен сдать экзамен в специально назначенное для этого время. Ему потребуется в учебном классе в отведенный временной промежуток создать и отладить в среде Microsoft Visual Studio Web-приложение в соответствии с полученным заданием.

В случае сдачи экзамена в такой форме максимально возможная оценка "удовлетворительно" выставляется только в случае удовлетворения приложения всем требованиям, перечисленным в задании.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ОПК-7	Проверка выполнения учебных упражнений по темам № 1-8, 10-14, 16-21 Зачет	1-8, 10-14, 16-21	Знать: – основные алгоритмы и концепции современной информатики; Уметь: – использовать классические алгоритмы для решения прикладных задач.	1. Знать основные структуры данных современной информатики. 2. Уметь моделировать работу классических алгоритмов.	<ul style="list-style-type: none"> • Уметь создавать оригинальные структуры данных для решения типовых задач. • Знать преимущества и недостатки использования отдельных алгоритмов для решения типовых задач. 	1. Уметь разрабатывать свои алгоритмы на основе классических для решения прикладных задач и знать способы анализа трудоемкости этих алгоритмов.
Профессиональные компетенции						
ПК-2	Проверка выполнения учебных упражнений по темам № 1-8, 10-14, 16-21 Зачет	1-8, 10-14, 16-21	Знать: – принципы объектно-ориентированного программирования, устройство и основные механизмы .NET Framework;	1. Знание основных понятий ООП. 2. Знание основных конструкций языка C#.	1. Знание основных понятий ООП. Понимание инкапсуляции, наследования, полиморфизма. 2. Знание конструкций	1. Знание основных понятий ООП. Понимание инкапсуляции, наследования, полиморфизма. Четкое понимание достоинств и недостатков различных способов хранения и обработки данных класса. 2. Знание конструкций языка

		<p>– основы программирования на C#;</p> <p>– принципы разработки и тестирования программ и отдельных программных модулей для работы в среде .NET Framework;</p> <p>– технологию LINQ и основы работы с базами данных.</p> <p>Уметь:</p> <p>– проектировать пользовательский интерфейс и функциональность Windows- и Web-приложений</p> <p>Владеть:</p> <p>– навыками программирования, отладки и тестирования Windows- и Web-приложений .NET Framework.</p>	<p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов.</p> <p>4. Программирование требуемой функциональности классов.</p> <p>5. Умение обрабатывать исключительные ситуации.</p> <p>6. Умение проектировать пользовательский интерфейс.</p>	<p>языка C#. Умение выбрать наиболее подходящие для конкретного случая средства языка.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа.</p> <p>4. Программирование требуемой функциональности классов. Исключение дублирования кода.</p> <p>5. Умение обрабатывать исключительные ситуации. Создание и выбрасывание собственных типов исключений.</p> <p>6. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления.</p>	<p>C#. Умение выбрать наиболее подходящие для конкретного случая средства языка, грамотно провести рефакторинг кода.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа, атрибуты.</p> <p>4. Программирование требуемой функциональности классов. Исключение дублирования кода, применение для этого средств Visual Studio. Умение создавать многопоточные приложения</p> <p>5. Умение обрабатывать исключительные ситуации. Создание и выбрасывание собственных типов исключений.</p> <p>6. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления, создавать собственные контролы.</p>
--	--	---	--	---	---

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «не зачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Высокий уровень формирования компетенций соответствует оценке «отлично» за самостоятельные, контрольные работы и экзаменационную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за самостоятельные, контрольные работы и экзаменационную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за самостоятельные, контрольные работы и экзаменационную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2
к рабочей программе дисциплины
"Разработка программных приложений"

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине "Разработка программных приложений" являются лекции, проводимые в виде электронных презентаций, что позволяет сделать материал лекций более наглядными, улучшает информативность и понимание изучаемого курса.

По большинству тем предусмотрены учебные упражнения, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и получение навыков разработки и отладки Windows- и Web-приложений для .NET Framework. Выполнение этих упражнений, а также выполнение индивидуальных заданий позволяет не только понять и закрепить теоретический материал, но и приобрести навык анализа предметной области и построения объектно-ориентированных приложений для .NET Framework на языке C#.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются и обсуждаются на лекциях и лабораторных занятиях. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы концепции объектно-ориентированного анализа и проектирования. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной при выполнении лабораторных работ или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков программирования для .NET Framework, в течение всего периода обучения проводятся консультации и разбор самостоятельного выполнения индивидуальных заданий.

В конце первого семестра студенты сдают зачет, в конце второго – экзамен. Зачет по итогам семестра выставляется по результатам выполнения учебных упражнений и индивидуальных заданий, а также краткого собеседования по вопросам теоретического материала. Экзамен предполагает самостоятельную разработку приложения и ответы на теоретические вопросы.

Освоить вопросы, излагаемые в процессе изучения дисциплины "Разработка программных приложений" самостоятельно студенту достаточно сложно. Это связано и с недостатком опыта построения сложных объектно-ориентированных приложений и с особенностями их исполнения в .NET Framework от традиционных Windows-программ. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет и экзамен по итогам изучения дисциплины студенту будет сложно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](#) (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.