

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

УТВЕРЖДАЮ

Проректор
по учебно-методической работе

А.А.Панфилов

20/5г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

"Технологии программирования"

Направление подготовки: **09.03.04 «Программная инженерия»**

Профиль подготовки: **Разработка программно-информационных систем**

Уровень высшего образования: **бакалавриат**

Форма обучения: **очная**

Семестр	Трудоемкость зач. ед./час.	Лекций, час.	Практич. занятий, час.	Лаборат. рабо, час.	СРС, час.	Форма промежу- точного кон- троля (экз./зачет)
3	5 ЗЕТ, 180 ч.	36	18	18	72	Экзамен, 36
4	4 ЗЕТ, 144 ч.	36		18	90	Зачет с оценкой
5	5 ЗЕТ, 180 ч.	18	18	18	90	Экзамен, 36, КП
Итого	14 ЗЕТ, 504 ч.	90	36	54	252	Экзамен, Зачет с оцен- кой, Экзамен, КП

Владимир 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является изучение основных подходов к разработке программных систем с использованием технологий Microsoft .NET.

Объектами профессиональной деятельности бакалавров являются: информационные процессы, технологии, системы и сети, их инструментальное (преимущественно программное и техническое) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации программных систем в различных областях. Кроме этого изучаются основные процессы разработки программного обеспечения на уровне технологических процессов и фаз.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Технологии программирования» относится к базовой части блока Б1-дисциплины учебного плана. Дисциплина логически, содержательно и методически тесно связана с рядом теоретических дисциплин и практик, изучающих ОПОП.

Для успешного изучения дисциплины студенты должны быть знакомы с дисциплинами «Основы алгоритмизации и программирования», «Информатика», «Управление данными», «Технологии обработки информации».

Дисциплина является основой для успешного прохождения бакалаврами всех видов практик, выполнения и последующей защиты ВКР.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающиеся должны обладать следующими общекультурными и профессиональными компетенциями:

- готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3);
- способностью готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты (ПК-15);
- владением навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения (ПК-19);
- владением навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации (ПК-21);
- способностью создавать программные интерфейсы (ПК-22).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1) Знать: платформу программирования Microsoft .NET и основные ее возможности для разработки приложений, организацию сетевого взаимодействия программных компонентов и взаимодействие с базами данных; основы объектно-ориентированного подхода к разработке и проектированию сложных программных систем; способы получения и подключения к проекту внешних библиотек и компонентов (ОПК-3, ПК-15, ПК-19, ПК-21, ПК-22);

2) Уметь: устанавливать и настраивать средства разработки приложений с использованием профессиональных инструментов; выбирать эффективные методические приемы, технические и информационные средства для реализации выбранных алгоритмов решения задач; ориентироваться в основных технических характеристиках программных средств и технологий; читать исходный код ранее разработанного компонента и дорабатывать его;

собирать единое приложение на основе различных программных компонентов (в т.ч. внешних, ориентируясь на их спецификацию и предоставляемый программный интерфейс взаимодействия; предоставлять результаты своей работы в виде технической документации и презентационных материалов (ОПК-3, ПК-15, ПК-19, ПК-21, ПК-22);

3) Владеть: методами моделирования и формирования алгоритмов решения задач; навыками работы с программным обеспечением разработки приложений; методами проведения моделирования систем; навыками анализа результатов моделирования и представления алгоритмов решающих задачи; методами и инструментами сборки приложений; программными инструментами моделирования систем и представления результатов моделирования в виде документов, диаграмм, аналитической и прочей проектной документации (ОПК-3, ПК-15, ПК-19, ПК-21, ПК-22).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 14 зачетных единиц, 504 часа.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах /)	Формы текущего контроля успеваемости, форма промежуточной аттестации
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Платформа MS .NET. Архитектура, особенности	3	1-2	4	2	2		8		1/13	
2	Стандартные типы данных, поддерживаемые .NET и языком С#. Числа, строки, массивы	3	3-4	4	2	2		8		1/13	
3	Пользовательские типы данных. Классы, структуры, интерфейсы	3	5-6	4	4	2		8		2/20	Рейтинг-контроль № 1
4	Операторы языка С# для организации вычислительного процесса	3	7-10	8	4	2		8		2/14	
5	Библиотека классов .NET Framework. Пространства имен. Назначение и использование. Обзор	3	11-12	4	2	4		12		2/20	Рейтинг-контроль № 2
6	ООП. Инкапсуляция. Наследование. Полиморфизм	3	13-16	8	2	4		16		2/14	
7	ООП. Взаимодействие классов.	3	17-18	4	2	2		12		2/25	Рейтинг-контроль № 3
	Всего			36	18	18		72		28/39	экзамен
8	Основы Web-программирования	4	1-2	4		2		12		2/33	
9	Web-формы и элементы управления формы. Обработка данных	4	3-4	4		2		12		2/33	
10	Элементы управления для отображения данных на страницах	4	5-6	4		2		14		2/33	Рейтинг-контроль № 1
11	Web-приложения	4	7-10	8		2		14		4/40	
12	Сеансы и сессии. Модели хранения состояний	4	11-12	4		4		14		4/50	Рейтинг-контроль № 2

13	Web-сервисы	4	13-15	6		4		12		3/30	
14	Безопасность ASP.NET. Аутентификация	4	16-18	6		2		12		3/38	Рейтинг-контроль № 3
	Всего			36		18		90		20/37	зачет с оценкой
15	Шаблон проектирования MVC при разработке Web-приложений	5	1-2	2	2	2		10		2/33	
16	Методы разработки ПО через тестирование. Test Driven Development	5	3-4	2	2	4		10		4/50	
17	Подходы к коллективной разработке ПО. Репозиторий проекта. Совместное использование кода на примере Tortoise SVN	5	5-6	2	4	2		12		4/50	Рейтинг-контроль № 1
18	Методология UML как способ проектирования, разработки и сопровождения ПО	5	7-10	4	2	2		12		4/50	
19	Критерии и метрики оценки качества ПО	5	11-12	2	2	2		12		4/67	Рейтинг-контроль № 2
20	Жизненный цикл программного продукта. Классический ЖЦ.	5	13-14	2	2	2		12		2/33	
21	Методы управления командой разработчиков.	5	15-16	2	2	2		12		2/33	
22	Ведение проекта.	5	17-18	2	2	2		10		2/33	Рейтинг-контроль № 3
	Всего			18	18	18		90		24/44	экзамен
Всего				90	36	54		252	КП	72/40	Экзамен, зачет с оценкой, экзамен

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В рамках дисциплины предусматриваются занятия, проводимые с использованием компьютерных образовательных технологий.

В процессе преподавания дисциплины также применяется накопительная бально-рейтинговая система оценки, включающая результаты текущего контроля успеваемости и промежуточной аттестации.

Для реализации подхода на основе компетенций предлагается интегрировать в учебный процесс интерактивные образовательные технологии, включая информационные и коммуникационные технологии (ИКТ), при осуществлении различных видов учебной работы:

- учебную дискуссию;
- электронные средства обучения (слайд-лекции, компьютерные тесты).

Лекционные занятия проводятся в аудиториях, оборудованных компьютерами, электронными проекторами, что позволяет сочетать активные и интерактивные формы проведения занятий. Чтение лекций сопровождается демонстрацией компьютерных слайдов.

Лабораторные работы проводятся в компьютерном классе.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

По дисциплине предусмотрено три текущих контрольных мероприятия (рейтинг-контроля) в 3,4,5 семестрах, промежуточная аттестация - экзамен (в 3 и 5 семестрах) и зачет с оценкой (4 семестр).

3 семестр

Примерный перечень вопросов для текущего контроля:

Рейтинг-контроль 1

1. Что такое .NET Framework?
2. Какие основные компоненты .NET Framework определяют ее архитектуру?
3. В чем преимущества использования .NET Framework и управляемого кода по сравнению с использованием неуправляемого кода или вообще отказ от использования .NET Framework?
4. Что такое CLR?
5. Типы данных в языке C#;
6. Массивы и структуры. Объявление и особенности работы;
7. Арифметические операции и оператор присвоения. Постфиксные и префиксные формы записи арифметических операций. Оператор присвоения;
8. Логические операторы;
9. Операторы ветвления. Полная и сокращенная форма операторов ветвления.
10. Оператор цикла с предусловием;
11. Оператор цикла с постусловием;
12. Итеративный цикл.

Рейтинг-контроль 2

1. Что такое класс?
2. Что такое объект
3. Основные принципы ООП. Перечислить. Дать характеристику и привести пример.
4. Понятия форм и элементов управления Windows.
5. Виды элементов управления.
6. Понятие события и обработчика событий
7. Дать понятие термина «интерфейс»?
8. Чем отличается интерфейс от абстрактного класса?
9. Поддерживают ли реализацию методы интерфейса?
10. Какие объекты языка C# могут быть членами интерфейсов?
11. Каким количеством классов может быть реализован интерфейс?
12. Может ли класс реализовывать множественные интерфейсы?
13. Необходима ли реализация методов интерфейса в классе, включающее этот интерфейс?
14. Возможно ли наследование интерфейсов?

Рейтинг-контроль 3

1. Дать понятие «делегата»
2. В чем основные преимущества и особенности использования делегатов?
3. Когда осуществляется выбор вызываемого метода при использовании делегатов?
4. Возможно ли использование делегата для вызова метода соответствующего подписи делегата?
5. Возможен ли вызов метода в том случае, если его сигнатура не соответствует сигнатуре делегата?
6. Как осуществляется создание цепочки методов для многоадресных делегатов?
7. Какие операторы языка C# используются для создания цепочки методов для многоадресных делегатов?
8. Каким образом осуществляется удаление цепочки методов для многоадресных делегатов?

9. Какие операторы языка C# используются для удаления цепочки методов для многоадресных делегатов?
10. Что понимается под термином «событие»?
11. Являются ли события членами классов?
12. Как выполняется описание событий? Проиллюстрируйте его фрагментом программы на языке C#.
13. Каковы механизмы языка C# для поддержки событий?
14. Что понимается под термином «широковещательное событие» и на основе какого механизма строятся широковещательные события?
15. Дать понятие анонимной функции и лямбда-выражения
16. В чем основные преимущества и особенности использования лямбда-выражений?
17. В чем особенность объявления и использования метода с сигнатурой, объявленной как:
18. Возможно ли использование лямбда-выражений без делегатов?

Примерный перечень вопросов к экзамену:

1. Платформа .NET как основа разработки современных бизнес-приложений Концепции архитектуры и особенности (JIT-компиляторы, IL-язык, компоненты)
2. Классы как основа объектно-ориентированного программирования. Поля, методы и свойства класса. Назначение и использование
3. Языковые особенности. Особенности языка (классы, свойства, пространства имен), назначение и использование
4. Стандартные типы, поддерживаемые .NET и языком C#. Массивы и строки. Обработка массивов. Примеры
5. Пользовательские типы данных. Классы, структуры, интерфейсы, перечисления, делегаты Назначение и примеры работы. Объекты
6. Операторы языка C# для организации вычислительного процесса. Обзор. Циклы. Примеры
7. Операторы языка C# для организации вычислительного процесса. Обзор. Операторы ветвления. Примеры
8. Концепция ООП. Классы и объекты. Создание и уничтожение объектов. Конструкторы и деструкторы. Назначение и использование. Примеры.
9. Концепция ООП. Основные подходы и этапы построения объектной модели.
10. Классы и формы отношений между классами. Инкапсуляция. Уровни доступа к членам класса
11. Классы и формы отношений между классами. Наследование и полиморфизм.
12. Вложенные классы и совместное использование вложенных классов.
13. Интерфейсы. Назначение. Описание и использование. Примеры.
14. Делегаты. Описание и использование. Лямбда выражения. Примеры.
15. События в C#. Назначение и использование. Примеры.
16. Потоки и процессы. Многопоточность. Управление потоками.
17. Потоки и процессы. Состояние потоков. Синхронизация.
18. Потоки и процессы. Тупики. Необходимые условия возникновения тупиков. Блокировки.
19. Библиотека классов .NET Framework. Пространства имен. Назначение и использование. Обзор
20. Библиотека классов .NET Framework. Стандартные пространства имен и классы для работы (System.IO, System.Data, System.NET и т.п). Примеры.

Примерный перечень заданий для самостоятельной работы обучающегося:

1. Операторы управления вычислительным процессом. Работа с базовыми типами данных и массивами

2. Инкапсуляция, наследование, полиморфизм в С#
3. Взаимодействие и сотрудничество классов между собой. Механизмы поддержания взаимодействия в С#
4. Разработка простейшего .NET приложения на языке С# с использованием библиотеки классов Microsoft .NET;
5. Разработка .NET приложения с использованием управляющих конструкций вычислительного процесса;
6. Разработка .NET приложения с использованием механизмов ООП поддерживаемых в С#

4 семестр

Примерный перечень вопросов для текущего контроля:

Рейтинг-контроль 1

1. Что такое WEB-приложение, и каковы его основные особенности?
2. Что такое серверный элемент управления? Чем он отличается от HTML элемента управления?
3. Каким образом осуществляется связывание данных и представления?
4. Какие типы объектов могут быть присвоены свойству DataSource?
5. Как (и на какой стороне – на сервере или на клиенте) осуществляется валидация данных?
6. Каким образом валидация осуществляется на клиенте?
7. Как узнать (на стороне сервера), прошла ли страница валидацию?
8. Как валидатор "привязывается" к элементу управления, который он проверяет?
9. Что такое WEB-сервис? Основные особенности.
10. Каковы основные элементы типового WEB-сервиса?

Рейтинг-контроль 2

1. Основные особенности использования технологии ADO.NET и ее возможности.
2. Основные классы, используемые для организации слоя доступа к данным, их особенности.
3. Как извлекаются и обрабатываются данные на уровне приложения.

Рейтинг-контроль 3

1. Что такое MVC? Какова его архитектура?
2. Как работает шаблон MVC?
3. В чем ценность шаблона MVC?
4. Что такое DAO? Каково его предназначение?
5. Что такое суррогатный ключ? Из чего он строится? Каково его предназначение?
6. Что такое аутентификация?
7. Что такое авторизация?
8. В чём отличия аутентификации от авторизации?
9. Значение аутентификации в приложениях MVC?
10. Значение авторизации в приложениях MVC?

Примерный перечень вопросов к зачету с оценкой.:

1. Зачем используется WebMethods на страницах ?
2. Что такое MasterPage и зачем они ?
3. Жизненный цикл обработки страницы (Page processing life cycle.)

4. События обработки страницы (Page Life Cycle Events)
5. Что такое фильтры ответов Response Filters ?
6. Что такое шаблон проектирования Model/View/Controller? Как и зачем его применяют?
7. Последовательность событий при загрузке HTML страницы на клиенте и что на каждом событии делается?
8. Какова максимальная длина строки запроса?
9. Чем Get запрос отличается от Post запроса?
10. Назовите методы аутентификации в .Net?
11. Объясните 2 технологии ASP.NET используемые для отслеживания сессии?
12. В чем разница между Server.Transfer и Response.Redirect?
13. Какие есть способы управления состоянием сессии в ASP.NET?
14. Какое пространство имен сопоставляется серверным элементам управления ASP.NET?
15. Какая из директив является аналогом директивы Page для файла global.asax?
16. Какие существуют способы проверки подлинности пользователей в ASP.NET приложениях?
17. Что такое сессионный объект?
18. Какие существуют способы организации хранения и доступа к сессионным объектам?

Примерный перечень заданий для самостоятельной работы обучающегося:

1. Какой Web-сервер необходимо установить для использования ASP.NET?
2. Какое расширение имеет файл в котором реализуется веб сервис на базе технологии .NET Framework?
3. Какой зарезервированный IP-адрес для подключения к серверу, запущенному на компьютере, с которого поступил запрос на подключение?
4. Как называется корневой каталог Microsoft Internet Information Server (IIS), создаваемый по умолчанию при его инсталляции?
5. С помощью какого тега указывается какая часть кода из ASPX-файла должна выполняться на серверной стороне?
6. С помощью какой директивы в ASP.NET импортируется пространство имен?
7. Почему такие разные типы данных как, например, Repeater и массивы (Array) одинаково хорошо поддерживают работу с источниками данных?
8. Фоновый код. Особенности. Преимущества использования. Иллюстрирующий пример
9. Дать определение Web-приложения и указать его основные характеристики и компоненты
10. Файл web.config. Назначение и особенности использования. Дать пример
11. Какое событие генерируется при инициализации Web-страницы? Привести пример.
12. Какое событие генерируется при загрузке Web-страницы? Привести пример.
13. Особенность передачи данных формы от клиента к серверу способом GET. Привести иллюстрирующий пример
14. Понятие Web-элемента управления. Назначение и особенности использования. Привести пример
15. Понятие Web-формы ASP.NET приложения. Назначение и особенности использования. Привести пример
16. Особенность передачи данных формы от клиента к серверу способом POST. Привести иллюстрирующий пример
17. Какое свойство страницы и элементов управления используется для автоматического запоминания средой текущего состояния страницы и элементов управления?
18. Каким свойством объекта Page можно воспользоваться для определения того, была ли загружена страница в первый раз или она сгенерирована после обработки запроса?

Примерный перечень вопросов для текущего контроля:

Рейтинг-контроль 1

1. Понятие внешних модулей и необходимость в них
2. Назначение библиотеки log4net
3. Дать понятие TDD, основные этапы цикла TDD
4. Для чего необходимы Assert в тестах?
5. Что такое цикломатическая сложность кода?
6. Что такое NUnit и как его использовать в своей работе?
7. Понятие репозитория и версии репозитория.
8. Назначение и преимущества использования систем контроля версий
9. Что такое commit?
10. Что такое рабочая версия проекта и что такое репозиторий проекта?
11. Когда возникают конфликты при работе с репозитрием? Виды конфликтов и способы их разрешения?
12. Какова стандартная структура репозитория проекта? Каких рекомендаций следует придерживаться при ведении проекта в системе контроля версий?
13. Какие модели ведения версионности файлов поддерживает SVN?
14. Когда надо делать новые ветки проекта? Что это дает?

Рейтинг-контроль 2

1. Что такое прецедент?
2. Каковы основные элементы, выносимые на диаграмму UseCase?
3. Что включает в себя расширенное описание прецедента?
4. Пояснить суть отношений между прецедентами (включение и расширение).
5. Понятие и назначение диаграммы классов?
6. Перечислить основные элементы, выносимые на диаграмму классов. Пояснить назначение каждого из них.
7. Перечислить виды отношений между классами. Раскрыть суть каждого из них.
8. Описать порядок выделения классов предметной области и их вынесение на диаграмму классов.
9. Пояснить суть динамического моделирования программной системы. Указать назначение и цель моделирования.
10. Перечислить и охарактеризовать основные виды диаграмм, предназначенных для динамического моделирования системы.
11. Что такое линия жизни объекта?

Рейтинг-контроль 3

1. Дать понятие потока. Для чего применяются потоки?
2. Что такое параллелизм и когда он возможен?
3. Механизмы использования параллелизма в коде в .NET
4. Понятие профилирования. Для чего оно применяется?
5. Понятие выборки и горячего пути?
6. Описать сеанс производительности.
7. Понятие TDD. Для чего и как оно применяется
8. В чем основные отличия TDD от BDD.

9. Какие ключевые слова существуют в feature-файле фреймворка SpecFlow и для чего они предназначены?

Примерный перечень вопросов к экзамену:

1. Процедурно-ориентированный и объектно-ориентированный подходы к разработке ПО
2. Этапы жизненного цикла разработки и развития ПС. Особенности
3. Системный анализ и системное проектирование ПС. Программа как система
4. Репозитория разработки ПС RUP. Особенности
5. Язык UML. Назначение. Возможности
6. UML. Диаграмма классов. Выделение классов предметной области и выявление отношений между ними. Этапы построения объектной модели и формальные признаки ее усовершенствования
7. Классы и отношения между классами. Реализация отношений между классами в C#
8. UML. Диаграмма состояний объекта и последовательностей. Особенности синтеза
9. Диаграмма прецедентов. Роль прецедентов при разработке ПС. Виды прецедентов и отношения между ними. Правила описания
10. Формирование требований к ПС на основе прецедентов. Функции ПС
11. Развертывание и представление ПС. Диаграмма компонентов и развертывания
12. Анализ требований при проектировании ПС. Диаграмма кооперации и видов деятельности
13. Классический жизненный цикл разработки ПС. Частные реализации и особенности
14. Экстремальное программирование
15. Унифицированный процесс разработки ПС. Этапы и итерации. Особенности
16. Планирование и управление проектом. Командная разработка ПС
17. Критерии и метрики определения качества и сложности разработки ПС. Функционально и размерно-ориентированные метрики. Метрики ООПС (метрики Чидамбера-Кеммерера).
18. Web-программирование. Обработка данных. GET и POST запросы. Анатомия Web-формы. Серверные элементы управления
19. Страничные директивы и страничные события. Механизм. Фоновый код.
20. Данные и источники данных. Элементы управления, обработка и связывание с источниками данных. Механизм
21. Проверяющие элементы управления. Назначение и использование
22. Многостраничные приложения. Связи между формами
23. Web-приложения. Архитектура и управление работой
24. Сеансы и сессии. Назначение и принципы работы. Состояния
25. Web-сервисы. Общие принципы организации. SOAP
26. Web-сервисы. Язык WSDL. Пользовательские типы данных
27. Безопасность в ASP.NET. Принципы идентификации и аутентификации. Авторизация
28. Шаблон проектирования MVC. Взгляд MS .NET. Особенности работы
29. Методы разработки ПО через тестирование (TDD). Взгляд с точки зрения .NET. Особенности.
30. Многопоточность и параллелизм в .NET. Особенности.
31. Техническое задание. Состав и структура технического задания. Назначение.
32. Тестирование программных систем. Виды тестирования.
33. Коллективная разработка и коллективное владение кодом. Методы организации работы через SVN. Разновидности репозитория и общедоступные сервисы.
34. Идеология Windows Workflow Foundation (WWF). Перспективы и возможности

Примерный перечень заданий для самостоятельной работы обучающегося:

1. Что такое процесс создания ПО? Назначение.

2. Что такое фаза (итерация) процесса? Для чего необходима?
3. Почему возможно и целесообразно стандартизировать процесс разработки ПО на уровне компании?
4. Что такое модель процесса разработки ПО? Необходимость?
5. В чем достоинства и недостатки спиральной модели? Каковы ограничения этой модели?
6. В чем достоинства и недостатки классического ЖЦ? В чем ее роль?
7. Произвести количественную оценку классов, представленных на диаграмме классов (следующий слайд) с использованием набора метрик Чидамбера-Кемерера
8. Почему себестоимость исправления выявленной ошибки в ПО выше на поздних этапах разработки системы?
9. Почему водопадная модель до сих пор востребована и применима на практике?
10. Для чего осуществляется совершенствование процесса разработки ПО?

Самостоятельная работа обучающихся заключается в самостоятельном изучении отдельных тем, практической реализации типовых заданий по эти темам. Контроль выполнения самостоятельной работы проводится при текущих контрольных мероприятиях и на промежуточной аттестации по итогам освоения. Учебно-методическое обеспечение самостоятельной работы – основная литература [1,2,3].

Примерный перечень тем к курсовому проектированию:

1. Программная система учета знаний учащихся.
2. Программная система автоматизации и учета подвижного состава автобусного предприятия.
3. Программная система учета пациентов в информационной системе «Поликлиника»
4. Программная система движения материалов на оптовой базе
5. Программная система автоматизации учета торговой деятельности компьютерного магазина
6. Программная система автоматизации документооборота малого предприятия
7. Программная система "Туристическое агентство"
8. Информационно-программная система «Рекламное агентство»
9. Программная система автоматизации учета расходных материалов на
10. Программная система «Страховое агентство»
11. Программная система «Деканат университета»
12. Программная система «Дистанционное обучение студента» (тема допускает два исполнителя)
13. Программная система «Кадровое агентство»
14. Программная система «Кадастровое управление»
15. Регистрационная система «Паспортный стол»

16. Программная система «База знаний отрасли»
17. Подсистема учета расходных материалов промышленного предприятия
18. Программная система учета пассажирских перевозок информационной системы «Автотранспортное предприятие»
19. Программная система муниципальной образовательной информационной сети

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. В.В. Подбельский, Язык С#. Базовый курс: учеб. пособие / - 2-е изд., перераб. и доп. - М.: Финансы и статистика, 2015.- 408 с.: ил.
<http://www.studentlibrary.ru/book/ISBN9785279035342.html>
2. Агапов В.П. Основы программирования на языке С# [Электронный ресурс]: учебное пособие/ Агапов В.П.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2012.- 128 с.
<http://www.iprbookshop.ru/16366.html>
3. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual С# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2011.— 180 с.
<http://www.iprbookshop.ru/19258>

б) дополнительная литература:

1. В.В. Вершинин Программирование для Microsoft.NET : методические указания к лабораторным работам по дисциплине "Программирование" : в 2 ч. / В. В. Вершинин, С. В. Чебыкин ; Владимирский государственный университет (ВлГУ), 2010.
<http://e.lib.vlsu.ru/bitstream/123456789/2111/3/00692.pdf>
2. Васильев В.Н. Основы программирования на языке С+ [Электронный ресурс]: учебное пособие/ Васильев В.Н.— Электрон. текстовые данные.— Волгоград: Волгоградский институт бизнеса, Вузовское образование, 2010.— 72 с. <http://www.iprbookshop.ru/11341.html>
3. Савельев А.О. Проектирование и разработка веб-приложений на основе технологий Microsoft [Электронный ресурс]/ Савельев А.О., Алексеев А.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2010.— 240 с.
<http://www.iprbookshop.ru/16729.html>

в) периодические издания:

1. Вестник компьютерных и информационных технологий ISSN 1810-7206.

г) интернет-ресурсы

- www.edu.ru – портал российского образования
- www.elbib.ru – портал российских электронных библиотек
- www.distance-learning.ru – портал, посвященный дистанционному обучению
- www.eLibrary.ru – научная электронная библиотека
- www.moodle.com – портал разработчиков Moodle
- library.vlsu.ru - научная библиотека ВлГУ
- <http://ispi.cdo.vlsu.ru> – учебный сайт кафедры ИСПИ ВлГУ

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

- Лекционная аудитория (410-2): 30 посадочных мест, мультимедийный проектор с экраном.
- Компьютерный класс (414-2): 25 посадочных мест, 13 персональных компьютеров со специализированным программным обеспечением, мультимедийный проектор с экраном.
- Электронные учебные материалы на сервере Центра дистанционного обучения.
- Доступ в Интернет.

Рабочая программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.04 "Программная инженерия" (программа подготовки "Разработка программно-информационных систем").

Рабочую программу составил:

_____ доцент каф. ИСПИ В.В. Вершинин

Рецензент: _____ директор по производству ЗАО "БСЦ МСК" Р.Н. Выгорчук

Программа рассмотрена и одобрена на заседании кафедры ИСПИ

Протокол № 711 от 06.04.15 года.

Заведующий кафедрой _____ И.Е. Жигалов

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 09.03.04 "Программная инженерия"


Протокол № 7 от 06.04.15 года.

Председатель комиссии _____ И.Е. Жигалов

**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

Рабочая программа одобрена на 2016/17 учебный год.

Протокол заседания кафедры № 1 от 29.08.16 года.

Заведующий кафедрой  Жигалов И. Э.

Рабочая программа одобрена на 2014/18 учебный год.

Протокол заседания кафедры № 1 от 30.08.14 года.

Заведующий кафедрой  Жигалов В. Э.

Рабочая программа одобрена на 2018/19 учебный год.

Протокол заседания кафедры № 1 от 30.08.18 года.

Заведующий кафедрой  Жигалов И. Э.

Рабочая программа одобрена на 2019/20 учебный год.

Протокол заседания кафедры № 1 от 28.08.19 года.

Заведующий кафедрой  Жигалов И. Э.

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ года.

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ года.

Заведующий кафедрой _____