

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информатики и информационных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Современные технологии разработки программного обеспечения

Кафедра информационных технологий
и моделирования экономических процессов
факультета информатики и информационных технологий

Образовательная программа
09.04.03 Прикладная информатика

Направленность (профиль) программы
Прикладная информатика в аналитической экономике

Уровень высшего образования
Магистратура

Форма обучения
Очная

Статус дисциплины: **входит в обязательную часть ОПОП**

Махачкала, 2019

Рабочая программа дисциплины составлена в 2019 году в соответствии с требованиями ФГОС ВО по направлению подготовки 09.04.03 Прикладная информатика (уровень магистратуры) от «19» сентября 2017г. № 916.

Разработчик: кафедра информационных технологий и моделирования экономических процессов, Гаджиев Н.К., к.э.н., доцент

Рабочая программа дисциплины одобрена:

на заседании кафедры ИТиМЭП от «26» июня 2019 г., протокол № 10

Зав. кафедрой А.А.А. Адамадзиев К.Р.

(подпись)

на заседании Методической комиссии факультета информатики и информационных технологий от «02» июля 2019г., протокол № 10.

Председатель М.К.Б. Камилов М.-К.Б.

(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением «03» июль 2019 г. А.А.А.

(подпись)

Аннотация рабочей программы дисциплины

Дисциплина «Современные технологии разработки программного обеспечения» входит в обязательную часть образовательной программы магистратуры по направлению 09.04.03 Прикладная информатика в аналитической экономике.

Дисциплина реализуется на факультете информатики и информационных технологий кафедрой информационных технологий и моделирования экономических процессов.

Содержание дисциплины охватывает круг вопросов, связанных с формированием компетенций в области разработки, проектирования, тестирования, отладки, внедрения и сопровождения программного обеспечения (ПО) вычислительной техники с использованием современных технологий.

Дисциплина нацелена на формирование следующих компетенций выпускника: универсальных УК-2, УК-3, УК-4; общепрофессиональных - ОПК-8.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: текущий контроль в форме опроса, тестов, контрольных работ и промежуточный контроль в форме экзамена.

Объем дисциплины 6 зачетные единицы, в том числе в академических часах по видам учебных занятий

Семестр	Учебные занятия							Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)
	в том числе:							
	всего	Контактная работа обучающихся с преподавателем					СРС, в том числе экзамен	
		всего	Лекции	Лабораторные занятия	Практические занятия	КСР		
1	216	46	10	18	18		170	экзамен

1. Цели освоения дисциплины

Целями освоения дисциплины «Современные технологии разработки программного обеспечения» являются формирование у слушателей теоретических и практических навыков по разработке надежного, качественного программного обеспечения с применением современных технологий программирования, методов и средств коллективной разработки.

2. Место дисциплины в структуре ОПОП магистратуры

Дисциплина «Современные технологии разработки программного обеспечения» входит в обязательную часть ОПОП программы магистратуры по направлению 09.04.03 Прикладная информатика в аналитической экономике.

Данную учебную дисциплину дополняет параллельное или последующее освоение следующих дисциплин: «Информатика и программирование», «Базы данных», «Современные технологии проектирования программного обеспечения», «Программирование на языке высокого уровня».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код и наименование компетенции из ФГОС ВО	Код и наименование индикатора достижения компетенций (в соответствии с ПООП (при наличии))	Планируемые результаты обучения
УК-2. Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1. Знать: методы управления проектами; этапы жизненного цикла проекта. УК-2.2. Уметь: разрабатывать и анализировать альтернативные варианты проектов для достижения намеченных результатов; разрабатывать проекты, определять целевые этапы и основные направления работ. УК-2.3. Владеть навыками разработки проектов в избранной профессиональной сфере; методами оценки эффективности проекта, а также потребности в ресурсах.	Знает: функциональные и технологические стандарты разработки и документирования программных комплексов Умеет: использовать современные технологии разработки программного обеспечения Владеет: способностью формулировать требования к создаваемым программным комплексам.
УК-3. Способен организовывать и руководить работой	УК-3.1. Знать: методики формирования команд; методы эффективного р Руководства коллективами.	Знает: современное состояние нормативно-правовой базы, уровня и направлений развития вычислительной техники,

<p>команды, вырабатывая командную стратегию для достижения поставленной цели</p>	<p>УК-3.2. Уметь: разрабатывать командную стратегию; организовывать работу коллективов; управлять коллективом; разрабатывать мероприятия по личностному, образовательному и профессиональному росту. УК-3.3. Владеть методами организации и управления коллективом, планированием его действий.</p>	<p>программных средств Умеет: использовать сетевые технологии для решения экономических задач, разрабатывать программные модели Владеет: навыками поиска необходимых нормативно-правовых документов при решении прикладных задач</p>
<p>УК-4. Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия</p>	<p>УК-4.1. Знать: современные коммуникативные технологии на государственном и иностранном языках; закономерности деловой устной и письменной коммуникации. УК-4.2. Уметь: применять на практике коммуникативные технологии, методы и способы делового общения. УК-4.3. Владеть методикой межличностного делового общения на государственном и иностранном языках, с применением профессиональных языковых форм и средств.</p>	<p>Знает: новейшие тенденции, методы и направления в области разработки программного обеспечения Умеет: решать задачи производственной и технологической деятельности на профессиональном уровне Владеет: способностью приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии</p>
<p>ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов</p>	<p>ОПК-8.1. Знать архитектуру информационных систем предприятий и организаций; методологии и технологии реинжиниринга, проектирования и аудита прикладных информационных систем различных классов; инструментальные средства поддержки технологии проектирования и аудита информационных систем и сервисов; методы оценки экономической эффективности и качества, управления надежностью и информационной безопасностью; особенности процессного подхода к управлению прикладными ИС; современные ИКТ в процессном управлении; системы управления</p>	<p>Знает: задачи и методы исследования и обеспечения качества и надёжности программных компонентов, технологии, используемые при постановке и решении прикладных задач Умеет: использовать современные программные средства для проектирования программного обеспечения Владеет: задачами и методами исследования и обеспечения качества, безопасности и надёжности программных компонентов, инструментами и методами передовых технологий программирования, навыками анализа рынка ПО и оценки перспектив внедрения</p>

	<p>качеством; концептуальное моделирование процессов управления знаниями; архитектуру систем управления знаниями; онтологии знаний; подсистемы сбора, фильтрации, накопления, доступа, генерации и распространения знаний;</p> <p>ОПК-8.2. Уметь выбирать методологию и технологию проектирования информационных систем; обосновывать архитектуру ИС; управлять проектами ИС на всех стадиях жизненного цикла, оценивать эффективность и качество проекта; применять со-временные методы управления проектами и сервисами ИС; использовать инновационные подходы к проектированию ИС; принимать решения по информатизации предприятий в условиях неопределенности; проводить реинжиниринг прикладных и информационных процессов; обосновывать архитектуру системы правления знаниями;</p>	
--	---	--

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 6 зачетных единиц, 216 академических часа.

4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные занятия	Контроль самостоятельной работы		
Модуль 1. Проектирование программного обеспечения									
1	Задача проектирования программных систем	9	1-3	2				14	Опрос, тестирование
2	Организация процесса проектирования программного обеспечения.	9	4-6		2	2		14	Опрос, тестирование, домашняя контрольная работа

	Технологии создания программных систем.								
	<i>Итого по модулю 1:</i>			2	2	2		28	
Модуль 2. Стандарты в области разработки программного обеспечения									
1	Стандарты и методики, используемые при разработке ПС.	9	7-9	2	2	2		14	Опрос, тестирование, домашняя контрольная работа
2	Технологические средства разработки программного обеспечения.	9	10-12		2	2		14	Опрос, тестирование, домашняя контрольная работа
	<i>Итого по модулю 2:</i>			2	4	4		28	
Модуль 3. Методы анализа и проектирования программного обеспечения									
1	Методы анализа и проектирования программных систем.	9	13-15	2	2	2		14	Опрос, тестирование, домашняя контрольная работа
2	Методы отладки и тестирования программных средств. Методы защиты программ и данных.	9	15-18		2	2		14	Опрос, тестирование, домашняя контрольная работа
	<i>Итого по модулю 3:</i>			2	4	4		28	
Модуль 4. Документирование программного обеспечения									
1	Документирование и оценка качества программных продуктов.			2	2	2		14	Опрос, тестирование, домашняя контрольная работа
2	Организация процесса проектирования ПС. Средства поддержки проекта.				2	2		12	Опрос, тестирование, домашняя контрольная работа
	<i>Итого по модулю 4:</i>			2	4	4		26	
Модуль 5. Методология объектно-ориентированного программирования									
1	Принципы ООП. Объектная декомпозиция предметной области.			2	2	2		12	Опрос, тестирование, домашняя контрольная работа
2	Проектирование интерфейса с пользователем. Структуры диалога. Поддержка пользователя. Многооконные интерфейсы.				2	2		12	Опрос, тестирование, домашняя контрольная работа
	<i>Итого по модулю 5:</i>			2	4	4		24	
	Экзамен (подготовка, сдача)	9	21-23					36	
	ИТОГО:			10	18	18		170	

4.3. Содержание дисциплины, структурированное по темам (разделам).

Модуль 1. Проектирование программного обеспечения

Тема 1. Задача проектирования программных систем.

Аудиторное изучение: Программа. Характеристика программ. Программное обеспечение. Виды программного обеспечения. Программный продукт. Характерные особенности программного продукта. Программный комплекс. Программное средство. Определение требований к программному средству.

Программная система. Сложность и сложные системы. Источники сложности. Признаки работоспособной сложной системы. Классификация программных систем по сложности. Проблемы проектирования сложных программных средств.

Понятие архитектуры программного средства.

Общие принципы разработки программных средств. Специфика разработки программных средств.

Самостоятельное изучение:

Основные классы архитектур программных средств (цельная монолитная программа, комплекс автономно выполняемых программ, слоистая программная система, коллектив параллельно выполняемых программ). Структуры программных архитектур: логическая (концептуальная), модульная, процессная, физическая.

Тема 2. Организация процесса проектирования программного обеспечения. Технологии создания программных систем.

Аудиторное изучение: Общая система понятий технологии программирования. Технология. Процесс. Стадия. Технологический подход. Технология создания (конструирования) ПО. Методы, средства, процедуры технологии конструирования ПО. Требования, предъявляемые к технологиям. Критерии оценки технологий. Выбор технологий.

Понятие жизненного цикла программного средства. Традиционные (классические) этапы разработки программ: анализ, проектирование, реализация, тестирование. Краткая характеристика каждого этапа.

Стратегии конструирования ПО: водопадная (однократный проход), инкрементная, эволюционная.

Модели жизненного цикла ПО (последовательные, итеративные или инкрементальные, эволюционные). Достоинства. Недостатки

Современная реализация инкрементной модели – экстремальное программирование XP.

Модель быстрой разработки приложений (RAD)- инкрементная стратегия конструирования.

Классификация и характеристика технологических процессов.

Классический набор процессов. Стандартный набор процессов, основанный на стандарте ISO 12207:1995.

Классификация технологических подходов. Краткая характеристика каждого.

Самостоятельное изучение: Роль технологии программирования на разных этапах развития программирования. История и эволюция технологий программирования.

Классический жизненный цикл. Макетирование. Быстрая разработка приложений (RAD). XP-процесс (экстремальное программирование).

Классический набор технологических процессов: возникновение и исследование идеи, управление, анализ требований, проектирование, программирование, тестирование и отладка, ввод в действие, эксплуатация и сопровождение, завершение эксплуатации. Краткое описание каждого.

Основные процессы (приобретение, поставка, разработка, эксплуатация, сопровождение). Вспомогательные процессы (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем). Организационные процессы (управление, создание инфраструктуры, усовершенствование, обучение). Взаимосвязь между процессами.

Основные стадии технологических процессов.

Характеристика рационального унифицированного процесса (RUP). Характеристика синтезирующего программирования. Характеристика сборочного программирования. Характеристика конкретизирующего программирования. Характеристика технологии стерильного цеха. Характеристика XP-процесса. Двенадцать методов базиса XP. Понятие идеального XP-процесса. Подходы исследовательского программирования.

Проблемы и перспективы развития технологических подходов.

Основные особенности и проблемы современных программных проектов. Современные тенденции в программной инженерии.

Модуль 2. Стандарты в области разработки программного обеспечения

Тема 3. Стандарты и методика, используемые при разработке программных средств

Аудиторное изучение: Виды стандартов. Методики проектирования. Стандартизация жизненного цикла программного средства в системе государственных стандартов и стандартов ISO. Стандарт пользовательского интерфейса. Стандарт проектирования. Стандарт оформления документации. Стандартизация процесса разработки программ и программной документации.

Самостоятельное изучение: ГОСТ 19.701-90. Международный стандарт ISO/IEC 12207. Стандарты комплекса ГОСТ 34. Руководящий документ РД 50-34.698-90, устанавливающий требования к содержанию документов на автоматизированные системы. Документирование на фазах жизненного цикла по требованиям ЕСПД и стандартов ISO. Оценка качества программного продукта и меры по обеспечению качества в соответствии с положениями ГОСТ Р и стандартами ISO. Основы Программной Инженерии (по SWEBOOK).

Основы построения профессионального интерфейса. Стандартные элементы интерфейса. Выбор шрифтов, цветов. Принципы построения интерфейсов: «Золотое сечение», «Кошелек Миллера», «Принцип группировки», «Бритва Оккама или KISS», «Видимость окружает полезность», «Умное заимствование». Эвристические правила Якоба Нильсена (правило обратной связи, информированность пользователя, средства обеспечения обратной связью, время оповещения, равенство между системой и реальным миром, свобода действий пользователя, последовательность и стандарты, предупреждение ошибок, понимание лучше, чем запоминание, гибкость и эффективность использования, эстетичный и минималистический дизайн, распознавание и исправление ошибок, описание ошибки, описание решения проблемы, справка и документация). Доработка лабораторной работы.

Тема 3. Технологические средства разработки программного обеспечения.

Аудиторное изучение: Инструментарий технологии программирования. Состав и назначение инструментария технологий программирования. Инструменты разработки программных средств. Средства для создания приложений: локальные средства, интегрированные среды. Case-средства. Общая характеристика Case-средств. Классификация Case-средств.

Инструментальная среда разработки. Средства поддержки проекта. Отладчики.

Самостоятельное изучение: История возникновения систем программирования. Принципы функционирования систем программирования: функции текстовых редакторов в системах программирования, компилятор как составная часть системы программирования, назначение и функции компоновщика, загрузчики и отладчики, библиотеки подпрограмм как составная часть систем программирования. Примеры современных систем программирования. Системы программирования компании Borland/Inprise (Turbo Pascal, Borland Delphi, Borland C++ Builder). Системы программирования фирмы Microsoft (Microsoft Visual Basic, Microsoft Visual C++). Концепция .NET. Разработка программного обеспечения для сети Интернет (Язык HTML, языки программирования Java и Java Script).

Автоматизация процесса разработки программного обеспечения. Автоматическое документирование ПО и автоматическая кодогенерация. Анализ требований и проектирование спецификаций.

Понятие Case-средств. Определение потребностей в Case-средствах. Оценка и выбор Case-средств.

Характеристики Case-средств: ERWin, BPWin, PowerDesigner, Rational Rose. Характеристика пакета создания диаграмм Microsoft Visio.

Технологии создания программного обеспечения – Model Driven Architecture, характеристика Bold – инструмента реализации MDA в Delphi.

Характеристика программного комплекса управления проектами – Microsoft Project. Использование OLE-автоматизации в приложениях. Доработка лабораторных работ.

Модуль 3. Методы анализа и проектирования программного обеспечения

Тема 5. Методы анализа и проектирования программных систем. Методы проектирования структуры ПО. Использование декомпозиции и абстракции при проектировании ПО. Декомпозиция системы. Специфики процедур и данных.

Аудиторное изучение: Классификация методов проектирования. По степени автоматизации проектных работ: неавтоматизированное (традиционное) проектирование и автоматизированное проектирование (Case-технология и ее элементы). По принятой методологии процесса разработки: метод функциональной декомпозиции (структурное проектирование программных средств), информационное моделирование предметной области и связанных с ней приложений, объектно-ориентированное проектирование программных средств. В зависимости от объекта структурирования: функционально-ориентированные методы (ориентированные на обработку), методы структурирования данных (ориентированные на данные). Методы, ориентированные на обработку: модульное программирование, функциональная декомпозиция, проектирование с использованием потока данных, технология структурного анализа проекта. Методы проектирования, основанные на использовании структур данных: методология Джексона, методология Уорнера, метод иерархических диаграмм, объектно-ориентированная методология.

Структура программных продуктов. Внутренняя структура. Функциональная структура приложения. Модульная структура программных продуктов.

Метод пошаговой детализации для проектирования структуры ПО. Декомпозиция системы методом пошаговой детализации. Структурная схема программы.

Технология подготовки и решения задач на ЭВМ. Этапы процесса разработки программ для структурного подхода и для объектно-ориентированного подхода. Краткая характеристика этапов создания программного обеспечения: постановка задачи, анализ и исследование задачи, проектирование программы (разработка алгоритма, структуры программы, разработка пользовательского интерфейса), реализация (написание кода программы на выбранном языке программирования, тестирование и отладка кода), составление документации, сопровождение программы, модификация программы.

Краткая характеристика последовательности этапов и используемых методов.

Визуальное моделирование программной системы. Понятие модель. Графические (визуальные) модели.

Структурный подход к разработке программных средств. Методы структурного анализа и проектирования ПО. Принципы инженерии ПО. Два базовых: принцип «разделяй и властвуй» и принцип иерархического упорядочивания. Принципы второстепенные: абстрагирования, формализации, «упрятивания», концептуальной общности, полноты, непротиворечивости, логической независимости, независимости данных, структурирования данных, доступа конечного пользователя.

Средства структурного подхода: диаграммы потоков данных, ER-диаграммы, диаграммы декомпозиции, структурные схемы (архитектура программного продукта в виде иерархии взаимосвязанных программных модулей с идентификацией связей между ними).

Метод функционального моделирования SADT. Общие сведения. Функциональная модель. Состав функциональной модели. Построение иерархии диаграмм (метод моделирования IDEF0-IDEF3). Типы связей между функциями.

Информационное моделирование предметной области. Моделирование потоков данных (процессов). Состав диаграмм потоков данных. Построение иерархии диаграмм потоков данных.

Объектно-ориентированный подход к проектированию программных средств. Методы объектно-ориентированного анализа и проектирования ПО. Сущность объектно-ориентированного подхода. Принципы объектно-ориентированного представления: абстрагирование, инкапсуляция, модульность, иерархия. Классы. Общая характеристика. Виды отношений между классами. Ассоциации классов. Объектная модель.

Унифицированный язык моделирования UML. Структурные модели: диаграммы классов, диаграммы компонентов, диаграммы размещения. Примеры диаграмм.

Модели поведения: диаграммы деятельности, диаграммы схем состояний, диаграммы последовательности, диаграммы сотрудничества (кооперации), диаграммы взаимодействия, диаграммы вариантов использования (Use Case). Примеры диаграмм.

Самостоятельное изучение: Этапы создания программных продуктов при неавтоматизированной разработке программ: составление ТЗ на программирование, технический проект, рабочая документация (описание применения, руководство пользователя, руководство программиста), ввод в действие.

Пример использования структурного подхода. Описание предметной области (организации).

Пример использования объектно-ориентированного подхода.

Создание моделей UML, структурных схем классов и объектов, схем сценариев выполнения, взаимодействий, последовательностей, компонентов, топологии, деятельности и состояний с использованием нотации UML.

Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов.

Модели реализации объектно-ориентированных программных систем. Компонентные диаграммы. Компоненты. Интерфейсы. Компонировка системы. Использование компонентных диаграмм. СОМ- компонентная объектная модель. Доработка лабораторных работ.

Тема 6. Методы отладки и тестирования программного средства. Методы защиты программ и данных

Аудиторное изучение: Введение в тестирование. Основные понятия. Типичные ошибки (синтаксические, логические, времени исполнения). Защита от ошибок. Методы защиты от ошибок, методы тестирования. Тест и тестирование. Этапы процесса тестирования. Стратегии тестирования (черный ящик, белый ящик). Подходы к тестированию: сверху вниз, снизу вверх. Отладка. Принципы и виды отладки. Заповеди отладки. Автономная отладка и тестирование программного модуля. Комплексная отладка и тестирование программного средства. Стандартная техника отладки.

Защита программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, исследования. Средства защиты: технические, юридические, организационные. Методы защиты ПО. Проверка оригинального носителя. Ввод серийного номера. Активация программного обеспечения. Использование электронных ключей. Подходы к защите программ от несанкционированного копирования. Криптографические методы защиты. Метод привязки к идентификатору. Методы, основанные на работа с переходами и стеком. Манипуляции с кодом программы. Методы противодействия динамическим способам снятия защиты программ от копирования. Уязвимости методов защиты.

Самостоятельное изучение: Обеспечение защищенности программных средств. Защита программных средств от исследования. Классификация средств исследования программ. Режимы исследования: статический (изучение исходного текста программы на основе дизассемблирования) и динамический (изучение алгоритма программы на основе выполнения трассировки программы). Методы противодействия дизассемблированию: шифрование; архивация; использование самогенерирующих кодов. Трассировка программ с помощью отладчиков. Методы защиты программ от исследования. Защита на основе шифрования секретной части программы, на блокировании доступа к исполняемому коду программы в оперативной памяти со стороны отладчиков.

Тестовые данные. Стратегия проектирования тестов.

Модуль 4. Документирование программного обеспечения

Тема 7. Документирование и оценка качества программных продуктов

Аудиторное изучение: Понятие качества программного обеспечения. Стандартизация характеристик качества. Критерии качества. Дерево характеристик качества. Выбор показателей качества.

Оценка качества программных продуктов. Методы контроля качества. Методы оценки качества. Методы обеспечения качества.

Надежность и функциональность программных средств. Обеспечение надежности и функциональности программного средства.

Документирование программных средств. Внешнее описание программного средства. Спецификация качества программного средства. Функциональная спецификация программного средства.

Самостоятельное изучение: Характеристики качества ПО. Документация, создаваемая в процессе разработки программных средств. Пользовательская документация программных средств. Документация по сопровождению программных средств. Правовые аспекты распространения ПО. Свободно распространяемое ПО. Доработка лабораторных работ.

Тема 8. Организация процесса проектирования ПС

Аудиторное изучение: Понятие проекта. Представление проекта в виде «черного ящика». Классификация проектов. Класс проекта. Тип проекта. Масштаб проекта. Участники процесса разработки и их роль в разработке проекта.

Окружение проекта: структура организации-исполнителя проекта, организационная культура, заинтересованные в проекте лица. Виды деятельности, входящие в управление проектом: управление содержанием проекта и качеством, метрики ПО, управление ресурсами, специфика управления персоналом, управление рисками, управление коммуникациями и информационным обеспечением.

Технологии коллективной разработки (технические командные роли, психологические командные роли, типы совместной деятельности). Авторская разработка. Общинная модель разработки.

Самостоятельное изучение: Управление разработкой ПС. Управление проектом. Методы управления проектами (метод критического пути, метод анализа и оценки программ ПЕРТ). Современные подходы к управлению проектом. Планирование проекта. Методики оценок времени и затрат. Доработка лабораторных работ.

Модуль 5. Методология объектно-ориентированного программирования

Тема 9. Принципы ООП. Объектная декомпозиция предметной области.

Аудиторное изучение:

Определение ООП. Основная цель ООП. Инкапсуляция. Наследование. Полиморфизм. Основное достоинство ООП. Недостаток ООП. Основные средства разработки классов.

Формальное понятие класса, объекта, свойства, события. Объекты и сообщения. Описание класса. Интерфейс и реализация класса. Поля и методы. Типы методов. Создание метода. Конструктор. Деструктор. Переопределение методов. Обработчик события. Объявление объекта. Инициализация полей. Обращение к полям и методам. Объекты и сообщения. Состояние объекта. Поведение объектов. Типы операций над объектом: создание, селекция, модификация, итерация, уничтожение. Два типа отношений между объектами: отношение использования (активный, пассивный, объект-посредник) и отношение включения.

Особенности работы с объектами в Delphi. Разделение графической и логической составляющих программы.

Средства разработки классов. Библиотеки классов. Ограничение доступа к полям и методам. Области видимости. Сообщения. Типы сообщений.

Самостоятельное изучение: Архитектура программы при ООП. Объектная декомпозиция. Типы методов. Интерфейсы класса. Проектирование классов. Примеры разработки класса для реализации объекта. Доработка лабораторных работ.

Тема 10. Проектирование интерфейса с пользователем. Структуры диалога. Поддержка пользователя. Многооконные интерфейсы. Реализация пользовательских интерфейсов в Delphi.

Аудиторное изучение: Проектирование интерфейса пользователя: диалоговый режим, графический интерфейс пользователя. Классификация систем, поддерживающих диалоговые процессы. Понятие о визуальном программировании. Объектно-ориентированные технологии OLE, ActiveX, COM.

Самостоятельное изучение: Технология разработки программного обеспечения – Model Driven Architecture – архитектура, управляемая моделью. Характеристика технологии. Этап создания модели приложения, которая определяет состав, структуру и поведение программного продукта. Bold – реализация MDA в Delphi. Основные возможности. Преимущества использования Bold. Три уровня создания приложения: уровень данных, бизнес-уровень, графический интерфейс пользователя.

Понятие технологии автоматизации. Доступ к данным из различных приложений Microsoft Office. Создание компонентов COM и ActiveX. Доработка лабораторных работ.

4.3.2. Содержание лабораторно-практических занятий по дисциплине.

Модуль 1. Проектирование программного обеспечения

Лабораторная работа №1. Стандарты проектирования ПО.

Лабораторная работа №2. Знакомство с Case-средствами: ERWin, BPWin, PowerDesigner, Rational Rose. Изучение возможностей пакета создания диаграмм Microsoft Visio.

Модуль 2. Стандарты в области разработки программного обеспечения

Лабораторная работа № 3. Использование методологии IDEF0, DFD и программного пакета Microsoft Visio для анализа предметной области. Создание диаграмм.

Лабораторная работа № 4. Использование методологии IDEF0, DFD и программного комплекса BPWin для проектирования и разработки программного обеспечения. Создание диаграмм.

Модуль 3. Методы анализа и проектирования программного обеспечения

Лабораторная работа № 5. Объектно-ориентированный подход к проектированию ПО. Использование Case-системы PowerDesigner для проектирования и разработки программного обеспечения. Создание моделей: Business Process Model; Free Model; Multi-Model Report; Object-Oriented Model.

Лабораторная работа № 6. Создание моделей с использованием языка UML в Microsoft Visio.

Лабораторная работа №7. Разработка диаграмм на языке UML в Rational Rose (по вариантам): диаграмма вариантов использования; диаграмма классов; диаграмма состояний; диаграмма деятельностей; диаграмма компонентов.

Модуль 4. Документирование программного обеспечения

Лабораторная работа №8. Разработка руководства пользователя с использованием утилиты HelpScribe.

Лабораторная работа №9. Автоматизация управления проектами с помощью ПО MS Project, MS Visio.

Лабораторная работа №10. Особенности работы с объектами в Delphi. Разделение графической и логической составляющих программы.

Модуль 5. Методология объектно-ориентированного программирования

Лабораторная работа №11. Создание MDA приложений в Visual Studio: создание бизнес-уровня; создание модели приложения; создание графического интерфейса; создание уровня данных; работа с приложением.

Лабораторная работа №12. Доступ к данным из различных приложений Microsoft Office. Создание компонентов COM и ActiveX.

5. Образовательные технологии

Образовательные технологии, используемые при реализации различных видов учебной работы по дисциплине «Современные технологии разработки программного обеспечения» предусматривают широкое использование в учебном процессе активных и интерактивных форм проведения занятий. Лекции с проблемным изложением проводятся с применением мультимедийного оборудования в виде презентаций. Данные лекции доступны для обучающихся при подготовке к разного вида контролю и СРС. Лекции-дискуссии, деловые игры (рассмотрение конкретной ситуации), конкретное обсуждение ситуаций.

В учебном процессе широко применяются компьютерные технологии. Поэтому все занятия проводятся в лаборатории, оборудованной ПК и мультимедийным оборудованием.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Самостоятельная работа студентов (СРС) включает контролируемую и внеаудиторную самостоятельную работу, направлена на повышение качества обучения, углубление и закрепление знаний студента, развитие аналитических навыков по проблематике учебной дисциплины, активизацию учебно-познавательной деятельности студентов и снижение аудиторной нагрузки. Часть программного материала выносится для самостоятельного внеаудиторного изучения с последующим текущим или итоговым контролем знаний на занятиях или экзамене. Контроль СРС и оценка ее результатов организуется как самоконтроль (самооценка) студента, а также как контроль и оценка со стороны преподавателя, например в ходе собеседования. Баллы, полученные по СРС студентом, обязательно учитываются при итоговой аттестации по курсу. Формы контроля СРС включают: тестирование; устную беседу по теме с преподавателем; выполнение индивидуального задания и др.

Роль студента в СРС - самостоятельно организовывать свою учебную работу по предложенному преподавателем, методически обеспеченному плану. СРС по курсу учитывает индивидуальные особенности слушателей и включает не только задания, связанные с решением типовых задач, но также творческие задания, требующие самостоятельно «добывать» знания из разных областей, группировать и концентрировать их в контексте конкретной решаемой задачи. Технология обучения предусматривает выработку навыков презентации результатов выполненного индивидуального задания и создание условий для командной работы над комплексной темой с распределением функций и ответственности между членами коллектива. Оценка результатов выполнения индивидуального задания осуществляется по критериям, известным студентам, отражающим наиболее значимые аспекты контроля за выполнением этого вида работ.

Разделы и темы для самостоятельного	Виды и содержание самостоятельной работы
--	---

изучения	
<p>Принципы выбора характеристик качества в проектах программных средств. Пример выбора и формирования требований к характеристикам качества программного средства</p>	<p>-конспектирование первоисточников и другой учебной литературы; -проработка учебного материала (по конспектам лекций учебной и научной литературе) и подготовка докладов на семинарах и практических занятиях, к участию в тематических дискуссиях; -поиск и обзор научных публикаций и электронных источников информации, подготовка заключения по обзору; -работа с тестами и вопросами для самопроверки;</p>
<p>Принципы верификации и тестирования программ. Процессы и средства тестирования программных компонентов. Технологические этапы и стратегии систематического тестирования программ. Процессы тестирования структуры программных компонентов. Примеры оценок сложности тестирования программ. Тестирование обработки потоков данных программными компонентами</p>	<p>-конспектирование первоисточников и другой учебной литературы; -проработка учебного материала (по конспектам лекций учебной и научной литературе) и подготовка докладов на семинарах и практических занятиях, к участию в тематических дискуссиях; -поиск и обзор научных публикаций и электронных источников информации, подготовка заключения по обзору; -работа с тестами и вопросами для самопроверки; -решение задач, упражнений; - решение домашних контрольных задач.</p>
<p>Процессы оценивания характеристик и испытания программных средств. Организация и методы оценивания характеристик сложных комплексов программ. Средства для испытаний и определения характеристик сложных комплексов программ. Оценивание надежности и безопасности функционирования сложных программных средств. Оценивание эффективности использования ресурсов ЭВМ программным продуктом</p>	<p>-конспектирование первоисточников и другой учебной литературы; -проработка учебного материала (по конспектам лекций учебной и научной литературе) и подготовка докладов на семинарах и практических занятиях, к участию в тематических дискуссиях; -поиск и обзор научных публикаций и электронных источников информации, подготовка заключения по обзору; -работа с тестами и вопросами для самопроверки; - решение домашних контрольных задач.</p>
<p>Организация и методы сопровождения программных средств. Этапы и процедуры при сопровождении программных средств. Задачи и процессы переноса программ и данных на иные платформы. Ресурсы, для обеспечения</p>	<p>-конспектирование первоисточников и другой учебной литературы; -проработка учебного материала (по конспектам лекций учебной и научной литературе) и подготовка докладов на семинарах и практических занятиях, к участию в тематических дискуссиях; -поиск и обзор научных публикаций и электронных источников информации, подготовка заключения по обзору; -работа с тестами и вопросами для самопроверки;</p>

сопровождения и мониторинга программных средств	- решение домашних контрольных задач.
Принципы выбора характеристик качества в проектах программных средств. Пример выбора и формирования требований к характеристикам качества программного средства	-конспектирование первоисточников и другой учебной литературы; -проработка учебного материала (по конспектам лекций учебной и научной литературе) и подготовка докладов на семинарах и практических занятиях, к участию в тематических дискуссиях; -поиск и обзор научных публикаций и электронных источников информации, подготовка заключения по обзору; -работа с тестами и вопросами для самопроверки; - решение домашних контрольных задач.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Код и наименование компетенции из ФГОС ВО	Код и наименование индикатора достижения компетенций (в соответствии с ПООП (при наличии))	Планируемые результаты обучения	Процедура освоения
УК-2. Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1. Знать: методы управления проектами; этапы жизненного цикла проекта. УК-2.2. Уметь: разрабатывать и анализировать альтернативные варианты проектов для достижения намеченных результатов; разрабатывать проекты, определять целевые этапы и основные направления работ. УК-2.3. Владеть навыками разработки проектов в избранной профессиональной сфере; методами оценки эффективности проекта, а также потребности в ресурсах.	Знает: функциональные и технологические стандарты разработки и документирования программных комплексов Умеет: использовать современные технологии разработки программного обеспечения Владеет: способностью формулировать требования к создаваемым программным комплексам.	Устный опрос. Тестирование. Расчетно-графическое задание
УК-3. Способен организовывать и руководить	УК-3.1. Знать: методики формирования команд; методы эффективного р	Знает: современное состояние нормативно-	Устный опрос. Тестирование.

<p>работой команды, вырабатывая командную стратегию для достижения поставленной цели</p>	<p>Руководства коллективами. УК-3.2. Уметь: разрабатывать командную стратегию; организовывать работу коллективов; управлять коллективом; разрабатывать мероприятия по личностному, образовательному и профессиональному росту. УК-3.3. Владеть методами организации и управления коллективом, планированием его действий.</p>	<p>правовой базы, уровня и направлений развития вычислительной техники, программных средств Умеет: использовать сетевые технологии для решения экономических задач, разрабатывать программные модели Владеет: навыками поиска необходимых нормативно-правовых документов при решении прикладных задач</p>	<p>Расчетно-графическое задние</p>
<p>УК-4. Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия</p>	<p>УК-4.1. Знать: современные коммуникативные технологии на государственном и иностранном языках; закономерности деловой устной и письменной коммуникации. УК-4.2. Уметь: применять на практике коммуникативные технологии, методы и способы делового общения. УК-4.3. Владеть методикой межличностного делового общения на государственном и иностранном языках, с применением профессиональных языковых форм и средств.</p>	<p>Знает: новейшие тенденции, методы и направления в области разработки программного обеспечения Умеет: решать задачи производственной и технологической деятельности на профессиональном уровне Владеет: способностью приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии</p>	<p>Устный опрос. Тестирование. Расчетно-графическое задние</p>
<p>ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов</p>	<p>ОПК-8.1. Знать архитектуру информационных систем предприятий и организаций; методологии и технологии реинжиниринга, проектирования и аудита прикладных информационных систем</p>	<p>Знает: задачи и методы исследования и обеспечения качества и надёжности программных компонентов, технологии, используемые при постановке и реше-</p>	<p>Устный опрос. Тестирование. Расчетно-графическое задние</p>

	<p>различных классов; инструментальные средства поддержки технологии проектирования и аудита информационных систем и сервисов; методы оценки экономической эффективности и качества, управления надежностью и информационной безопасностью; особенности процессного подхода к управлению прикладными ИС; современные ИКТ в процессном управлении; системы управления качеством; концептуальное моделирование процессов управления знаниями; архитектуру систем управления знаниями; онтологии знаний; подсистемы сбора, фильтрации, накопления, доступа, генерации и распространения знаний; ОПК-8.2. Уметь выбирать методологию и технологию проектирования информационных систем; обосновывать архитектуру ИС; управлять проектами ИС на всех стадиях жизненного цикла, оценивать эффективность и качество проекта; применять со-временные методы управления проектами и сервисами ИС; использовать инновационные подходы к проектированию ИС; принимать решения по информатизации предприятий в условиях неопределенности; проводить реинжиниринг прикладных и информационных процессов; обосновывать архитектуру системы</p>	<p>нии прикладных задач Умеет: использовать современные программные средства для проектирования программного обеспечения Владеет: задачами и методами исследования и обеспечения качества, безопасности и надежности программных компонентов, инструментами и методами передовых технологий программирования, навыками анализа рынка ПО и оценки перспектив внедрения</p>	
--	--	---	--

7.2. Типовые контрольные задания

Примерные тестовые задания для текущего контроля и промежуточной аттестации.

Выберите правильный ответ из списка

1. *Инструментальные средства быстрого прототипирования ИС - ...*
 - языки программирования высокого уровня
 - приложения в развитых СУБД – класс DEVELOPER
 - интегрированные инструменты быстрой разработки приложений – класс BUILDER
 - приложений – класс BUILDER
2. *Сущность прототипной (RAD) технологии - ...*
 - создание на ранней стадии реализации действующей интерактивной модели системы, так называемой системы-прототипа, позволяющей наглядно продемонстрировать пользователю будущую систему, уточнить его требования, оперативно модифицировать интерфейсные элементы
 - формализация процесса проектирования ИС
 - активное вовлечение конечных пользователей в процесс разработки предназначенных для них интерактивных систем
3. *К технологии традиционного прототипирования ИС относятся.....*
 - демонстрация работы прототипа
 - разработка системы прототипа
 - доработка системы прототипа
 - разработка новых спецификаций требований
 - разработка постановки задачи
 - документирование готового приложения
4. *Инструментальными средствами быстрого прототипирования ИС являются ...*
 - приложения в развитых СУБД - класс DEVELOPER
 - интегрированные инструменты быстрой разработки приложений - класс BUILDER
 - языки программирования высокого уровня
5. *Технология итерационного прототипирования ИС - ...*
 - разработка новой постановки задачи
 - демонстрация работы прототипа
 - разработка приложения
 - доработка системы прототипа
 - документирование готового приложения
 - разработка системы прототипа
6. *К основным приемам быстрой разработки прототипа ИС относятся ...*
 - разработка приложения итерациями
 - использование большого количества технических средств
 - использование автоматических генераторов (мастеров)
 - высокая параллельность работ
 - повторное использование частей проекта
 - низкая производительность проектировщиков
7. *Диаграммами, выступающими в качестве инструментальных средств функционально-ориентированного анализа и проектирования являются...*
 - диаграммы деятельности (Activity diagram)
 - SSD (System Structure Diagram) - диаграмма структуры программного приложения
 - диаграммы взаимодействия объектов (Interaction diagram)
 - диаграммы состояний (Statechart diagram)
 - DFD (Data Flow Diagram) - диаграмма потоков данных
8. *Признаки классификации CASE-средств по режиму коллективной разработки проекта - ...*
 - не поддерживающие коллективную разработку
 - по виду загруженности разработчиков
 - работающие под управлением WINDOWS
 - ориентированные на режим объединения подпроектов
 - ориентированные на режим реального времени разработки проекта

9. *Признаки классификации CASE-средств по типу операционной системы (ОС) работающие под управлением -...*
- WINDOWS 3.11 и выше
 - Microsoft Office
 - OS/2 и др.
 - UNIX
10. *Диаграммы, выступающие в качестве инструментальных средств объектно-ориентированного анализа и проектирования - ...*
- диаграммы компонентов (Component diagram)
 - диаграммы деятельности (Activity diagram)
 - диаграммы пакетов (Package diagram)
 - диаграммы размещения (Deployment diagram)
 - DFD (Data Flow Diagram) - диаграмма потоков данных
 - диаграммы взаимодействия объектов (Interaction diagram)
11. *Преимуществом CASE-технологии является ...*
- возможность коллективной разработки ЭИС в режиме реального времени
 - возможность повторного использования компонентов разработки
 - улучшение климата в коллективе разработчиков
 - поддержание адаптивности и сопровождения ИС
 - снижение усталости разработчика

Вопросы для экзамена

1. Жизненный цикл программного обеспечения. Содержание этапов классического ЖЦ. Качество ПО. Критерии качества.
2. Виды архитектур программного средства.
3. Основные области в программировании: методологии и технологии программирования; языки и системы программирования; операционные и архитектурные платформы. Краткая характеристика каждой группы.
4. Методология программирования. Атрибуты методологии. Ядра методологий. Основные методологии: императивного программирования, объектно-ориентированного программирования, функционального программирования, логического программирования, активного программирования (экстремального). Краткая характеристика.
5. Технология: система обозначений, процесс и инструментальные средства.
6. Технологии программирования. Основные понятия и определения.
7. Технологический подход. Классификация подходов.
8. Сложность и сложные системы. Источники сложности. Признаки работоспособной сложной системы. Методы борьбы со сложностью. Моделирование, абстракция, декомпозиция.
9. Функциональная и объектная декомпозиция.
10. Три стратегии конструирования программного обеспечения.
11. Водопадная модель разработки приложений. Недостатки и кризис последовательной модели разработки ПО.
12. Инкрементная модель разработки приложений.
13. Спиральная модель разработки приложений.
14. Тяжеловесные и облегченные процессы разработки приложений.
15. Характеристики XP-процесса.

16. Технологический процесс. Классификация технологических процессов.

17. История и эволюция технологий программирования. Проблемы и перспективы развития.

18. Основные подходы к организации процесса создания и использования программных средств.

19. Методы проектирования программ.

20. Структурное программирование.

21. Модульное программирование. Преимущества модульного программирования. Определение модуля. Свойства модуля. Модульная структура приложения.

22. Сущность объектного подхода к разработке программ.

23. Объектно-ориентированное программирование. Основное понятие ООП. Объекты и классы. Определения, назначение. Принцип инкапсуляции. Принцип наследования. Принцип полиморфизма.

24. Основы визуального программирования. Элементы процесса визуализации. Принципы визуального программирования. Элементы управления. Проектирование формы. Визуальные и объектные аспекты форм. Архитектура приложения. Многооконные интерфейсы.

25. Виды пользовательских интерфейсов и требования к ним.

26. Проектирование пользовательского интерфейса. Рекомендации по проектированию пользовательского интерфейса.

27. Принципы построения пользовательского интерфейса.

28. Три размерности согласованности пользовательского интерфейса.

29. Два вида стилей взаимодействия между пользователем и компьютером и способы для связи.

30. Принципы использования цвета при проектировании эргономичного интерфейса.

31. Тексты и диалоги. Принципы создания текстовых диалогов и отображений.

32. Средства управления графического интерфейса пользователя.

33. Принципы проектирования меню приложения.

34. Режим окон приложения. Три типа окон приложения.

35. Основной элемент интерфейса – экранные формы. Принципы проектирования форм.

36. Проектирование сообщений для пользователя по выполнению необходимых действий на пути решения задачи.

37. Тестирование программ. Методы тестирования.

38. Приемы отладки программных средств. Автономная отладка. Комплексная отладка.

39. Отладка программ. Причины ошибок Типы ошибок в программе: времени выполнения, синтаксические, логические. Защита от ошибок.

40. Диалоговые окна. Стандартные диалоги. Процедуры и функции, реализующие диалоги.

41. Стандарты, необходимые для создания программного обеспечения.

42. Основные программные и эксплуатационные документы (по ГОСТ).
43. Спецификация требований к программной системе. Требования к данным, функциональности, интерфейсу, надежности, безопасности, эффективности. Описание предметной области. Требования к окружению. Архитектура системы.
44. Назначение и содержание «Технического задания».
45. Определение жизненного цикла ПО.
46. Группы процессов, входящие в состав ЖЦ ПО.
47. Наиболее часто используемые процессы в реальных проектах.
48. Определение модели ЖЦ ПО. Используемые модели ЖЦ в настоящее время.
49. Характеристика модели «Водопад».
50. Характеристика спиральной модели.
51. Инструментальные среды разработки и сопровождения ПС. Термин «Case-средства», основные возможности Case-средств.
52. Классификация Case-средств.
53. Структура Case-средства.
54. Сущность прототипной (RAD) технологии. Основные возможности и преимущества быстрой разработки.
55. Классификация инструментальных средств быстрого прототипирования.
56. Основы COM-технологии: понятие интерфейса.
57. Сущность автоматизации OLE.
58. История и эволюция языков моделирования. Блок-схемы, диаграммы.
59. Язык моделирования UML. Возникновение и развитие стандарта. Основные концепции моделирования (абстрагирование, множественность представлений). Обзор стандартных элементов и пакетов. Виды диаграмм.
60. Язык моделирования UML. Диаграммы классов. Классы, объекты, отношения ассоциации.
61. Язык моделирования UML. Отношения зависимости, обобщения, реализации. Стереотипы, интерфейсы.
62. Язык моделирования UML. Диаграмма вариантов использования. Применение совместно со спецификацией функций.
63. Язык моделирования UML. Диаграмма развертывания. Статические и динамические диаграммы, как средства изображения структур и процессов.
64. Язык моделирования UML. Диаграмма состояний. Применение при изображении пользовательского интерфейса и протоколов.
65. Язык моделирования UML. Диаграмма деятельности. Аналогии и преимущества перед блок-схемами.
66. Язык моделирования UML. Диаграмма последовательности. Диаграмма кооперации.
67. Компоненты. Каркасы. Паттерны. Паттерны порождения, структуры и управления.
68. Метрики (количественные оценки) ПО. Оценки сложности ПО, качества ПО, эффективности процесса разработки.

69. Правовые аспекты распространения ПО. Copyright и лицензирование. Свободно распространяемое ПО. Аттестация ПС.

7.3. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 60% и промежуточного контроля - 40%.

Текущий контроль по дисциплине включает:

- посещение занятий - 10 баллов,
- участие на практических занятиях - 20 баллов,
- выполнение лабораторных заданий - 35 баллов,
- выполнение домашних (аудиторных) контрольных работ - 35 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 20 баллов,
- письменная контрольная работа - 40 баллов,
- тестирование - 40 баллов.

8. Учебно-методическое обеспечение дисциплины.

а) адрес сайта курса

<http://cathedra.dgu.ru/?id=111>.

а) основная литература:

1. Киселева Т.В. Программная инженерия. Часть 1 [Электронный ресурс] : учебное пособие / Т.В. Киселева. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 137 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69425.html>
2. Кознов Д.В. Введение в программную инженерию [Электронный ресурс] / Д.В. Кознов. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 306 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/52146.html>
3. Соловьев Н.А. Введение в программную инженерию [Электронный ресурс] : учебное пособие / Н.А. Соловьев, Л.А. Юркевская. — Электрон. текстовые данные. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017. — 112 с. — 978-5-7410-1685-5. — Режим доступа: <http://www.iprbookshop.ru/71267.html>

б) Дополнительная литература:

1. Ехлаков Ю.П. Введение в программную инженерию [Электронный ресурс] : учебное пособие / Ю.П. Ехлаков. — Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011. — 148 с. — 978-5-4332-0018-0. — Режим доступа: <http://www.iprbookshop.ru/13923.html>

2. Липаев В.В. Экономика программной инженерии заказных программных продуктов [Электронный ресурс] : дополнение к учебному пособию «Программная инженерия сложных заказных программных продуктов» (для бакалавров) / В.В. Липаев. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2015. — 139 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/27303.html>

3. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 285 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/39552.html>

4. Ружников В.А. Экономика программной инженерии [Электронный ресурс] : учебное пособие / В.А. Ружников, М.А. Вержаковская, В.Ю. Аронов. — Электрон. текстовые данные. — Самара: Поволжский государственный университет телекоммуникаций и информатики, 2016. — 91 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73844.html>

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. Портал доступа к электронным образовательным ресурсам ДГУ [Электронный ресурс] <http://dgu.ru>;

2. Электронно-библиотечная система IPRbooks [Электронный ресурс] <http://www.iprbookshop.ru>

3. Электронно-библиотечная система «Университетская библиотека он-лайн» [Электронный ресурс] - <http://biblioclub.ru>;

4. Электронно-библиотечная система издательства «Инфра» [Электронный ресурс] <http://znanium.com>

5. IT-портал [Электронный ресурс] <http://citforum.ru>

6. Портал Национального открытого университета «Интуит» [Электронный ресурс] <http://www.intuit.ru>

10. Методические указания для обучающихся по освоению дисциплины.

Для изучения теоретического курса студентам необходимо использовать лекционный материал, учебники и учебные пособия из списка основной и дополнительной литературы, интернет источники.

По дисциплине «Современные технологии разработки программного обеспечения» в конце каждого модуля проводится контрольная работа.

В контрольную работу включаются теоретические вопросы и задачи тех типов, которые были разобраны на предшествующих практических занятиях.

Рабочей программой дисциплины «Современные технологии разработки программного обеспечения» предусмотрена самостоятельная работа студентов в объеме 48 часов. Самостоятельная работа проводится с целью углубления знаний по дисциплине и предусматривает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;
- подготовку к практическим занятиям;
- выполнение индивидуальных заданий;
- подготовку к контрольным работам, зачету и экзаменам.

С самого начала изучения дисциплины студент должен четко уяснить, что без систематической самостоятельной работы успех невозможен. Эта работа должна регулярно начинаться сразу после лекционных и практических занятий, для закрепления только что пройденного материала. После усвоения теоретического материала можно приступить к самостоятельному решению задач из учебников и пособий, входящих в список основной литературы.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Интернет-ресурсы, мультимедиа, электронная почта для коммуникации со студентами, MS Visual Studio, Microsoft Office, Case-средства.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Компьютерный класс, оборудованный для проведения лекционных и практических занятий средствами оргтехники, персональными компьютерами, объединенными в сеть с выходом в Интернет; установленное лицензионное и свободное программное обеспечение.