

**Министерство просвещения Российской Федерации**  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
**«Дагестанский государственный педагогический  
университет»**

Кафедра информатики и вычислительной техники



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.В.ДВ.01.01 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Направление подготовки - 44.03.05 Педагогическое образование**

**Направленность (профили)- "Информатика" и "Дополнительное образование (Робототехника)"**

**Квалификация выпускника: Бакалавр**

**Форма обучения – очная, заочная**

Форма обучения	Семестр	Трудоемкость	Виды учебной работы					СРС	Форма аттестации
			Лекции	Практ. занятия	Лабор. занятия	Промежуточный контроль			
очная	10	72	12		20		40	зачет	
заочная	10	72	4		4		64	зачет	

**Махачкала, 2022**

**Автор(ы) рабочей программы дисциплины (модуля):**

*Доцент, к.ф.-м..н. Рагимханова Г.С.*

**Программа утверждена на заседаниях:**

кафедры информатики и вычислительной техники (*протокол № 10 от «20» июня 2022 г.*)

Зав. кафедрой: Эсетов Ф.Э., к.п.н., доцент



(подпись)

Учёного совета института физико-математического и информационно-технологического образования (*протокол № 10 от «27» июня 2022 г.*)

Председатель: Бакмаев А.Ш., к.п.н., доцент



(ФИО, ученое звание)

(подпись)

учебно-методического совета ДГПУ (*протокол № 4 от «28» июня 2022 г.*)

Председатель УМС: Дибиров И.А.



(подпись)

## 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

**Целью** освоения дисциплины «Программная инженерия» являются формирование знаний, умений, навыков и личностных качеств, характеризующих готовность бакалавра к планированию и достижению профессиональной карьеры.

Код компетенции	Содержание компетенции	Индикаторы достижения компетенций
ПК-2	Способен осуществлять целенаправленную воспитательную деятельность	<p>ПК-2.1. Демонстрирует умение постановки воспитательных целей, проектирования воспитательной деятельности и методов ее реализации в соответствии с требованиями ФГОС ОО и спецификой учебного предмета.</p> <p>ПК-2.2. Демонстрирует способы организации и оценки различных видов внеурочной деятельности ребенка (учебной, игровой, трудовой, спортивной, художественной и т.д.), методы и формы организации коллективных творческих дел, экскурсий, походов, экспедиций и других мероприятий (по выбору).</p> <p>ПК-2.3. Выбирает и демонстрирует способы оказания консультативной помощи родителям (законным представителям) обучающихся по вопросам воспитания, в том числе родителям детей с особыми образовательными потребностями.</p>
ОПК-2	Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)	<p>ОПК 2.1. Разрабатывает программы учебных предметов, курсов, дисциплин (модулей), программы дополнительного образования в соответствии с нормативно-правовыми актами в сфере образования.</p> <p>ОПК 2.2. Проектирует индивидуальные образовательные маршруты освоения программ учебных предметов, курсов, дисциплин (модулей), программ дополнительного образования в соответствии с образовательными потребностями обучающихся.</p> <p>ОПК 2.3. Осуществляет отбор педагогических и других технологий, в том числе информационно-коммуникационных, используемых при</p>

		разработке основных и дополнительных образовательных программ и их элементов
--	--	--

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.ДВ.01.01 «Программная инженерия» относится к **вариативной части** учебного плана (основной профессиональной образовательной программы) подготовки бакалавров по направлению 44.03.05 Педагогическое образование.

Дисциплина Б1.В.ДВ.01.01 «Программная инженерия» базируется на компетенциях, знаниях и умениях, сформированных в ходе изучения дисциплин «Вводный курс информатики», «Элементы школьного курса информатики», «Архитектура компьютера», «Программирование», «Избранные вопросы программирования».

Компетенции сформированные в процессе изучения дисциплины необходимы для выполнения заданий (учебной, производственной практик, научно-исследовательской работы и выпускной квалификационной работы).

## 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Дисциплина направлена на формирование следующих компетенций выпускника: ПК-1.

В результате изучения дисциплины обучающиеся должны:

Код компетенции	Знает	Умеет	Владеет
ПК-2	способы организации и оценки различных видов внеурочной деятельности ребенка, методы и формы организации коллективных мероприятий	оказывать консультативную помощь родителям (законным представителям) обучающихся по вопросам воспитания, в том числе родителям детей с особыми образовательными потребностями.	навыками постановки воспитательных целей, проектирования воспитательной деятельности и методов ее реализации в соответствии с требованиями ФГОС ОО и спецификой учебного предмета.
ОПК-2	специфику и структуру основных образовательных программ по информатике, программ	разрабатывать отдельные компоненты образовательной программы; разрабатывать	навыками анализа основных и дополнительных программ в соответствии с требованиями современного

дополнительного образования;	элементы образовательных программ для разных профилей обучения;	образования;
основные элементы педагогических и других технологий, используемых при разработке образовательных программ.	составлять индивидуальные учебные планы, в соответствии с образовательными потребностями обучающихся, в том числе, на углублённом уровне.	навыками использования педагогических, информационно-коммуникационных технологий при разработке отдельных компонентов образовательных программ.

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 2 зачетные единицы (72 часа). Дисциплина изучается в 2 семестре.

#### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Трудоёмкость		
	час.	В т.ч. по семестрам	
		№1	№2
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>72</b>	<b>72</b>	
<b>1. Контактная работа:</b>			
лекции (общее кол-во часов, включая практическую подготовку)	12	12	
практические занятия, семинары и пр. (общее кол-во часов, включая практическую подготовку)			
лабораторные занятия (общее кол-во часов / включая практическую подготовку)	20	20	
курсовое проектирование			
групповые, индивидуальные консультации и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем			
<b>2. Объем самостоятельной работы обучающихся (СРС)</b>	<b>40</b>	<b>40</b>	
Вид промежуточного контроля:		зачет	

#### ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Трудоёмкость		
	час.	В т.ч. по семестрам	
		№1	№2
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>72</b>	<b>72</b>	
<b>1. Контактная работа:</b>			
лекции (общее кол-во часов, включая практическую подготовку)	4	4	
практические занятия, семинары и пр. (общее кол-во часов, включая практическую подготовку)			
лабораторные занятия (общее кол-во часов / включая	4	4	

Вид учебной работы	Трудоёмкость		
	час.	В т.ч. по семестрам	
		№1	№2
практическую подготовку)			
курсовое проектирование			
групповые, индивидуальные консультации и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем			
<b>2. Объем самостоятельной работы обучающихся (СРС)</b>	<b>64</b>	<b>64</b>	
Вид промежуточного контроля:		зачет	

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ) очная форма обучения

№ п/п	Наименование темы (раздела) дисциплины (модуля)	Общая трудоёмкость в акад. часах	Трудоёмкость по видам учебных занятий (в акад. часах)			
			Лек/ пр.подг.	Лаб / пр.подг.	Пр/ пр.подг.	СР
1	<b>Предмет программной инженерии.</b> Программная инженерия, основные понятия. Основные поддисциплины программной инженерии.	4		2		2
2	<b>Жизненный цикл и этапы разработки программного обеспечения.</b> Международный стандарт ISO\IEC 12207, описывающий структуру жизненного цикла программного обеспечения и его процессы. Процессы жизненного цикла разработки программного обеспечения: постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение. Международные стандарты оценки качества процессов создания программного обеспечения.	4	2			2
3	<b>Определение требований к программному обеспечению и исходных данных для его проектирования.</b> Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Обследование системы, общение с заказчиком, планирование разработки. Разработка технического задания.	4		2		2

	<p>Основные разделы технического задания. Принципиальные решения начальных этапов проектирования: выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор подхода к разработке, выбор метода проектирования, выбор или формирование стандартов разработки. Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.</p> <p>Разработка технического задания на создание системы "Учет успеваемости студентов". Сведения об успеваемости студентов должны храниться в течении всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.</p>					
4	<p><b>Анализ требований и определение спецификаций. Проектирование. Разработка программного обеспечения.</b></p> <p>Для учебного примера "Система бронирования билетов для авиакомпании" произвести: Анализ требований и определение спецификаций. Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; анализ требований к системе (определение ее функциональных возможностей, пользовательских требований, требований надежности и безопасности, требований к внешним интерфейсам).</p> <p>Проектирование.</p> <p>Проектирование архитектуры системы (определение состава необходимого оборудования, программного обеспечения и т.д.).</p> <p>Проектирование архитектуры программного обеспечения (определение структуры программного обеспечения, документирование интерфейсов его компонентов, требований к тестам и плана интеграции). Детальное проектирование программного</p>	6	2			4

	<p>обеспечения (подробное описание компонентов программного обеспечения и интерфейсов между ними, обновление пользовательской документации, разработка и документирование требований к тестам, обновление плана интеграции компонентов).          Разработка программного обеспечения. Кодирование и тестирование программного обеспечения (разработка и документирование каждого компонента, а также совокупности процедур и данных для их тестирования). Интеграция программного обеспечения (сборка программных компонентов в соответствии с планом интеграции).          Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; Проектирование архитектуры программного обеспечения; Кодирование и тестирование программного обеспечения.</p>					
5	<p><b>Обзор методологий проектирования программных продуктов.</b>          Модели процесса создания ПО. Каскадная и эволюционная модели. Итерационные модели процесса создания ПО. Модель пошаговой разработки, спиральная модель. Технология быстрой разработки приложений RAD (Rapid Application Development).</p>	4		2		2
6	<p><b>Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).</b>          Унифицированный язык моделирования UML. Модели спецификации разрабатываемого программного обеспечения при использовании UML: использования, логическая, реализации, процессов, развертывания. Основные типы UML-диаграмм, входящие в различные модели: диаграммы прецедентов, диаграммы классов, диаграммы взаимодействия, диаграммы состояний, диаграммы видов деятельности, диаграммы</p>	8	2			6

	<p>компонентов, диаграммы базы данных, диаграммы развертывания. Этапы проектирования ПО с применением UML: моделирование прецедентов, создание модели бизнес-объектов, создание концептуальной модели данных, разработка требований к системе, проектирование моделей базы данных и приложений, проектирование физической реализации системы.</p> <p>Разбиение студентов на команды, выделение действующих лиц и построение диаграммы вариантов использования для учебного примера "Система бронирования билетов для авиакомпании"</p> <p>Переход от диаграмм вариантов использования к диаграммам классов.</p> <p>Создание диаграмм классов на примере предыдущей задачи.</p> <p>Построение концептуальной модели для системы решения комбинаторно-оптимизационных задач. Построение диаграммы последовательностей системы для варианта использования</p> <p>?Выполнение задания решения комбинаторно-оптимизационных задач?. Построение диаграммы деятельности, уточняющую вариант использования ?Выполнение задания решения комбинаторно-оптимизационных задач?.</p> <p>Использование языка UML для проектирования системы учета успеваемости студентов.</p>					
7	<p><b>Методология проектирования приложений MFS. Основные компоненты и модели MFS.</b></p> <p>Основные модели MFS: команды, процесса, приложения, архитектуры корпорации, проектирования решений, управление инфраструктурой, стоимости владения проектом. Базовые компоненты MFS: Solution Development Discipline (SDD) дисциплина разработки приложений, Designing Component Solutions (DCS) проектирование компонентного ПО, Enterprise</p>	6		2		4

	Architecture Planning планирование архитектуры предприятия, Infrastructure Deployment and Management управление технологической инфраструктурой.					
8	<b>Модель команды.</b> Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд. Распределение задач между командами. Распределение ролей в командах.	4		2		2
9	<b>Модель команды.</b> Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд. Распределение задач между командами. Распределение ролей в командах.	4	2			2
10	<b>Модель управления рисками.</b> Основные сведения о рисках. Планирование управления рисками. Процесс управления рисками: выявление, анализ и приоритезация, планирование, мониторинг, корректирование, извлечение уроков. Управление рисками как составная часть жизненного цикла проекта. Разработка плана действий в отношении рисков.	4		2		2
11	<b>Модель процессов. Общая схема процесса разработки. Фазы процесса разработки: выработка концепции, планирование.</b> Схема процесса разработки. Основные структурные единицы схемы: циклы, фазы и вехи. Цикличность всех уровней процесса разработки. Примеры циклов в	4		2		2

	<p>различных задачах на различных уровнях: контроль версий, ежедневная сборка, итерация, версия продукта. Фазы ивехи процесса разработки: выработка концепции, планирование проекта, разработка решения, стабилизация решения, внедрение решения. Распределение задач по фазам. Выбор продолжительности фаз. Фаза выработки концепции. Основные задачи фазы: создание ядра проектной группы, формирование видения проекта, подготовка документа общего описания и рамок проекта, подготовка документа оценки рисков и представление главных рисков проекта, выявление и анализ бизнес-требований. Результаты фазы: Общее описание и рамки проекта, Документ оценки рисков, Описание структуры проекта. Фаза планирования. Основные задачи фазы: анализ и документирование проектных требований, подготовка функциональной спецификации, разработка дизайнов, подготовка рабочих планов, оценка проектных затрат и сроков разработки различных составляющих проекта. Категории проектных требований. Уровни процесса проектирования. Результаты фазы: Функциональная спецификация, План управления рисками, Сводный план и сводный календарный график проекта.</p> <p>Каждая команда: - Формирует видение проекта. - Выделяет и выполняет оценку рисков. - Выявляет и анализирует бизнес-требования. - Определяет структуру проекта.</p> <p>Каждая команда: - Разрабатывает дизайн и архитектуру решения. - Создает функциональную спецификацию. - Разрабатывает планы проекта. - Разрабатывает календарный график проекта. Разрабатывает концепцию решения.</p>					
12	<p><b>Модель процессов. Фазы разработки, стабилизации.</b>          Разработка решения. Фаза разработки. Основная задача фазы -</p>	4	2			2

	<p>создание компонент решения, включая как документацию, так и программный код. Результаты фазы: "Исходный исполняемый код приложений", "Скрипты установки и конфигурирования", "Окончательная функциональная спецификация", "Материалы поддержки решения", "Спецификации и сценарии тестов".</p> <p>Стабилизация решения. Фаза стабилизации. Основные задачи фазы: тестирование разработанного решения, приоритезация и устранение ошибок, подготовка решения к выпуску, пилотное внедрение. Результаты фазы: Окончательный продукт, Документация выпуска, Материалы поддержки решения, Результаты и инструментарий тестирования, Исходный и исполнимый код приложений, Проектная документация, Анализ пройденной фазы.</p> <p>Каждая команда: - Создает прототип приложения. - Разрабатывает необходимые компоненты решения. - Тестирует решение.</p>					
13	<p><b>Внедрение решения. Фаза внедрения.</b></p> <p>Основные задачи фазы: внедрение технологий и компонент решения, стабилизация внедренного решения, передача работ персоналу поддержки и сопровождения, получение со стороны заказчика окончательного одобрения результатами проекта, анализ выполненной работы и удовлетворенности заказчика.</p> <p>Результаты фазы: Информационные системы эксплуатации и поддержки, Процедуры и процессы, Базы знаний, отчеты, журналы протоколов, Версии проектных документов, массивы данных и программный код, разработанные во время проекта, Отчет о завершении проекта, Окончательные версии всех проектных документов, Показатели удовлетворенности заказчика и потребителей,</p>	4		2		2

	Описание последующих шагов. Каждая команда: - Внедряет решение в эксплуатацию в другую команду.					
14	<b>Модель приложения масштаба предприятия.</b> Модель приложения масштаба предприятия. Шесть основных под-моделей: Бизнес-модель приложения; Пользовательская модель; Логическая модель; Технологическая модель; Физическая модель; Модель разработки. Сервисы приложений.	4	2			2
15	<b>Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.</b> Организация работы гостиницы Ведение справочников: Номера, Услуги, Клиенты Функции: Ведение справочников, поселение и выселение клиентов, бронирование мест, учёт оказанных услуг Выходные документы: Счёт за проживание и услуги, Список проживавших на момент времени, Список номеров, Прейскурант услуг. 2. Организация работы больницы. Ведение справочников: Пациенты, Болезни, Палаты, Врачи, История болезни Функции: Ведение справочников, приём пациента, ведение истории болезни, выписка. Выходные документы: Список пациентов, Список врачей, Карточка больного.	4		2		2
16	<b>Разработка прикладных программных систем с использованием FMS.</b> Организация работы аптеки. Ведение справочников: Группы лекарств, Лекарства, Производители, Поставщики Функции: ведение справочников, учёт прихода и продаж лекарств Выходные документы: Отчёт по наличию лекарств на складе по группам, Отчёт по продажам по группам, Счёт-фактура. 2. Организация работы кафе. Ведение справочников: Продукты, Блюда, Заказы Функции: Ведение справочников, хранение рецептов, расчёт себестоимости блюда, приём	4		2		2

заказов Выходные документы: Меню, Счёт заказа, Отчёт по продуктам на складе, Заказы за период. 3. Деятельность службы трудоустройства. 4. Организация работы службы общественного питания. 5. Организация работы службы социальной помощи.					
Итого:	72	12	20		40

### заочная форма обучения

№ п/п	Наименование темы (раздела) дисциплины (модуля)	Общая трудоёмкость в акад. часах	Трудоёмкость по видам учебных занятий (в акад. часах)			
			Лек/ пр.подг.	Лаб / пр.подг.	Пр/ пр.подг.	СР
1	<b>Предмет программной инженерии.</b> Программная инженерия, основные понятия. Основные поддисциплины программной инженерии.	4	2			2
2	<b>Жизненный цикл и этапы разработки программного обеспечения.</b> Международный стандарт ISO\IEC 12207, описывающий структуру жизненного цикла программного обеспечения и его процессы. Процессы жизненного цикла разработки программного обеспечения: постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение. Международные стандарты оценки качества процессов создания программного обеспечения.	4	2			2
3	<b>Определение требований к программному обеспечению и исходных данных для его проектирования.</b> Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Обследование системы, общение с заказчиком, планирование разработки. Разработка технического задания. Основные разделы технического задания. Принципиальные решения начальных этапов проектирования:	4		2		2

	<p>выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор подхода к разработке, выбор метода проектирования, выбор или формирование стандартов разработки. Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.</p> <p>Разработка технического задания на создание системы "Учет успеваемости студентов". Сведения об успеваемости студентов должны храниться в течении всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.</p>					
4	<p><b>Анализ требований и определение спецификаций. Проектирование. Разработка программного обеспечения.</b></p> <p>Для учебного примера "Система бронирования билетов для авиакомпании" произвести: Анализ требований и определение спецификаций. Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; анализ требований к системе (определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам).</p> <p>Проектирование.</p> <p>Проектирование архитектуры системы (определение состава необходимого оборудования, программного обеспечения и т.д.).</p> <p>Проектирование архитектуры программного обеспечения (определение структуры программного обеспечения, документирование интерфейсов его компонентов, требований к тестам и плана интеграции). Детальное проектирование программного обеспечения (подробное описание компонентов программного обеспечения и интерфейсов между</p>	6		2		4

	<p>ними, обновление пользовательской документации, разработка и документирование требований к тестам, обновление плана интеграции компонентов).          Разработка программного обеспечения. Кодирование и тестирование программного обеспечения (разработка и документирование каждого компонента, а также совокупности процедур и данных для их тестирования). Интеграция программного обеспечения (сборка программных компонентов в соответствии с планом интеграции).          Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; Проектирование архитектуры программного обеспечения;          Кодирование и тестирование программного обеспечения.</p>					
5	<p><b>Обзор методологий проектирования программных продуктов.</b>          Модели процесса создания ПО. Каскадная и эволюционная модели. Итерационные модели процесса создания ПО. Модель пошаговой разработки, спиральная модель. Технология быстрой разработки приложений RAD (Rapid Application Development).</p>	4				4
6	<p><b>Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).</b>          Унифицированный язык моделирования UML. Модели спецификации разрабатываемого программного обеспечения при использовании UML:          использования, логическая, реализации, процессов, развертывания. Основные типы UML-диаграмм, входящие в различные модели: диаграммы прецедентов, диаграммы классов, диаграммы взаимодействия, диаграммы состояний, диаграммы видов деятельности, диаграммы компонентов, диаграммы базы данных, диаграммы развертывания. Этапы проектирования ПО с</p>	8				8

	<p>применением UML: моделирование прецедентов, создание модели бизнес-объектов, создание концептуальной модели данных, разработка требований к системе, проектирование моделей базы данных и приложений, проектирование физической реализации системы.</p> <p>Разбиение студентов на команды, выделение действующих лиц и построение диаграммы вариантов использования для учебного примера "Система бронирования билетов для авиакомпании"</p> <p>Переход от диаграмм вариантов использования к диаграммам классов.</p> <p>Создание диаграмм классов на примере предыдущей задачи.</p> <p>Построение концептуальной модели для системы решения комбинаторно-оптимизационных задач. Построение диаграммы последовательностей системы для варианта использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Построение диаграммы деятельности, уточняющую вариант использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Использование языка UML для проектирования системы учета успеваемости студентов.</p>					
7	<p><b>Методология проектирования приложений MFS. Основные компоненты и модели MFS.</b></p> <p>Основные модели MFS: команды, процесса, приложения, архитектуры корпорации, проектирования решений, управление инфраструктурой, стоимости владения проектом. Базовые компоненты MFS: Solution Development Discipline (SDD) дисциплина разработки приложений, Designing Component Solutions (DCS) проектирование компонентного ПО, Enterprise Architecture Planning планирование архитектуры предприятия, Infrastructure Deployment and</p>	6				6

	Management управление технологической инфраструктурой.					
8	<p><b>Модель команды.</b></p> <p>Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд. Распределение задач между командами. Распределение ролей в командах.</p>	4				4
9	<p><b>Модель команды.</b></p> <p>Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд. Распределение задач между командами. Распределение ролей в командах.</p>	4				4
10	<p><b>Модель управления рисками.</b></p> <p>Основные сведения о рисках. Планирование управления рисками. Процесс управления рисками: выявление, анализ и приоритезация, планирование, мониторинг, корректирование, извлечение уроков. Управление рисками как составная часть жизненного цикла проекта. Разработка плана действий в отношении рисков.</p>	4				4
11	<p><b>Модель процессов. Общая схема процесса разработки. Фазы процесса разработки: выработка концепции, планирование.</b></p> <p>Схема процесса разработки. Основные структурные единицы схемы: циклы, фазы и вехи. Цикличность всех уровней процесса разработки. Примеры циклов в различных задачах на различных уровнях: контроль версий, ежедневная сборка, итерация,</p>	4				4

	<p>версия продукта. Фазы и вехи процесса разработки: выработка концепции, планирование проекта, разработка решения, стабилизация решения, внедрение решения. Распределение задач по фазам. Выбор продолжительности фаз. Фаза выработки концепции. Основные задачи фазы: создание ядра проектной группы, формирование видения проекта, подготовка документа общего описания и рамок проекта, подготовка документа оценки рисков и представление главных рисков проекта, выявление и анализ бизнес-требований. Результаты фазы: Общее описание и рамки проекта, Документ оценки рисков, Описание структуры проекта. Фаза планирования. Основные задачи фазы: анализ и документирование проектных требований, подготовка функциональной спецификации, разработка дизайнов, подготовка рабочих планов, оценка проектных затрат и сроков разработки различных составляющих проекта. Категории проектных требований. Уровни процесса проектирования. Результаты фазы: Функциональная спецификация, План управления рисками, Сводный план и сводный календарный график проекта. Каждая команда: - Формирует видение проекта. - Выделяет и выполняет оценку рисков. - Выявляет и анализирует бизнес-требования. - Определяет структуру проекта. Каждая команда: - Разрабатывает дизайн и архитектуру решения. - Создает функциональную спецификацию. - Разрабатывает планы проекта. - Разрабатывает календарный график проекта. Разрабатывает концепцию решения.</p>					
12	<p><b>Модель процессов. Фазы разработки, стабилизации.</b> Разработка решения. Фаза разработки. Основная задача фазы - создание компонент решения, включая как документацию, так и программный код. Результаты фазы:</p>	4				4

	<p>"Исходный исполняемый код приложений", "Скрипты установки и конфигурирования", "Окончательная функциональная спецификация", "Материалы поддержки решения", "Спецификации и сценарии тестов".</p> <p>Стабилизация решения. Фаза стабилизации. Основные задачи фазы: тестирование разработанного решения, приоритезация и устранение ошибок, подготовка решения к выпуску, пилотное внедрение. Результаты фазы: Окончательный продукт, Документация выпуска, Материалы поддержки решения, Результаты и инструментарий тестирования, Исходный и исполнимый код приложений, Проектная документация, Анализ пройденной фазы.</p> <p>Каждая команда: - Создает прототип приложения. - Разрабатывает необходимые компоненты решения. - Тестирует решение.</p>					
13	<p><b>Внедрение решения. Фаза внедрения.</b></p> <p>Основные задачи фазы: внедрение технологий и компонент решения, стабилизация внедренного решения, передача работ персоналу поддержки и сопровождения, получение со стороны заказчика окончательного одобрения результатами проекта, анализ выполненной работы и удовлетворенности заказчика.</p> <p>Результаты фазы: Информационные системы эксплуатации и поддержки, Процедуры и процессы, Базы знаний, отчеты, журналы протоколов, Версии проектных документов, массивы данных и программный код, разработанные во время проекта, Отчет о завершении проекта, Окончательные версии всех проектных документов, Показатели удовлетворенности заказчика и потребителей, Описание последующих шагов.</p> <p>Каждая команда: - Внедряет решение в эксплуатацию в другую</p>	4				4

	команду.					
14	<p><b>Модель приложения масштаба предприятия.</b></p> <p>Модель приложения масштаба предприятия. Шесть основных под-моделей: Бизнес-модель приложения; Пользовательская модель; Логическая модель; Технологическая модель; Физическая модель; Модель разработки. Сервисы приложений.</p>	4				4
15	<p><b>Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.</b></p> <p>Организация работы гостиницы  Ведение справочников: Номера, Услуги, Клиенты  Функции: Ведение справочников, поселение и выселение клиентов, бронирование мест, учёт оказанных услуг  Выходные документы: Счёт за проживание и услуги, Список проживавших на момент времени, Список номеров, Прейскурант услуг.  2. Организация работы больницы. Ведение справочников: Пациенты, Болезни, Палаты, Врачи, История болезни  Функции: Ведение справочников, приём пациента, ведение истории болезни, выписка.  Выходные документы: Список пациентов, Список врачей, Карточка больного.</p>	4				4
16	<p><b>Разработка прикладных программных систем с использованием FMS.</b></p> <p>Организация работы аптеки.  Ведение справочников: Группы лекарств, Лекарства, Производители, Поставщики  Функции: ведение справочников, учёт прихода и продаж лекарств  Выходные документы: Отчёт по наличию лекарств на складе по группам, Отчёт по продажам по группам, Счёт-фактура.  2. Организация работы кафе. Ведение справочников: Продукты, Блюда, Заказы  Функции: Ведение справочников, хранение рецептов, расчёт себестоимости блюда, приём заказов  Выходные документы: Меню, Счёт заказа, Отчёт по продуктам на складе, Заказы за</p>	4				4

период. 3. Деятельность службы трудоустройства. 4. Организация работы службы общественного питания. 5. Организация работы службы социальной помощи.					
Итого:	72	4	4		64

## **5.1. Содержание разделов дисциплины (модуля)**

### **Тема 1. Предмет программной инженерии.**

Программная инженерия, основные понятия. Основные поддисциплины программной инженерии.

### **Тема 2. Жизненный цикл и этапы разработки программного обеспечения.**

Международный стандарт ISO\IEC 12207, описывающий структуру жизненного цикла программного обеспечения и его процессы. Процессы жизненного цикла разработки программного обеспечения: постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение. Международные стандарты оценки качества процессов создания программного обеспечения.

### **Тема 3. Определение требований к программному обеспечению и исходных данных для его проектирования.**

Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Обследование системы, общение с заказчиком, планирование разработки. Разработка технического задания. Основные разделы технического задания. Принципиальные решения начальных этапов проектирования: выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор подхода к разработке, выбор метода проектирования, выбор или формирование стандартов разработки. Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.

Разработка технического задания на создание системы "Учет успеваемости студентов". Сведения об успеваемости студентов должны храниться в течении всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

### **Тема 4. Анализ требований и определение спецификаций. Проектирование. Разработка программного обеспечения.**

Для учебного примера "Система бронирования билетов для авиакомпании" произвести: Анализ требований и определение спецификаций. Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; анализ требований к системе (определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам). Проектирование.

Проектирование архитектуры системы (определение состава необходимого оборудования, программного обеспечения и т.д.). Проектирование архитектуры программного обеспечения (определение структуры программного обеспечения, документирование интерфейсов его компонентов, требований к тестам и плана интеграции). Детальное проектирование программного обеспечения (подробное описание компонентов программного обеспечения и интерфейсов между ними, обновление

пользовательской документации, разработка и документирование требований к тестам, обновление плана интеграции компонентов).

Разработка программного обеспечения. Кодирование и тестирование программного обеспечения (разработка и документирование каждого компонента, а также совокупности процедур и данных для их тестирования). Интеграция программного обеспечения (сборка программных компонентов в соответствии с планом интеграции).

Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; Проектирование архитектуры программного обеспечения; Кодирование и тестирование программного обеспечения.

### **Тема 5. Обзор методологий проектирования программных продуктов.**

Модели процесса создания ПО. Каскадная и эволюционная модели. Итерационные модели процесса создания ПО. Модель пошаговой разработки, спиральная модель. Технология быстрой разработки приложений RAD (Rapid Application Development).

### **Тема 7. Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).**

Унифицированный язык моделирования UML. Модели спецификации разрабатываемого программного обеспечения при использовании UML: использования, логическая, реализации, процессов, развертывания. Основные типы UML-диаграмм, входящие в различные модели: диаграммы прецедентов, диаграммы классов, диаграммы взаимодействия, диаграммы состояний, диаграммы видов деятельности, диаграммы компонентов, диаграммы базы данных, диаграммы развертывания. Этапы проектирования ПО с применением UML: моделирование прецедентов, создание модели бизнес-объектов, создание концептуальной модели данных, разработка требований к системе, проектирование моделей базы данных и приложений, проектирование физической реализации системы.

Разбиение студентов на команды, выделение действующих лиц и построение диаграммы вариантов использования для учебного примера "Система бронирования билетов для авиакомпании" Переход от диаграмм вариантов использования к диаграммам классов.

Создание диаграмм классов на примере предыдущей задачи. Построение концептуальной модели для системы решения комбинаторно-оптимизационных задач. Построение диаграммы последовательностей системы для варианта использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Построение диаграммы деятельностей, уточняющую вариант использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Использование языка UML для проектирования системы учета успеваемости студентов.

### **Тема 8. Методология проектирования приложений MFS. Основные компоненты и модели MFS.**

Основные модели MFS: команды, процесса, приложения, архитектуры корпорации, проектирования решений, управление инфраструктурой, стоимости владения проектом. Базовые компоненты MFS: Solution Development Discipline (SDD) дисциплина разработки приложений, Designing Component Solutions (DCS) проектирование компонентного ПО, Enterprise Architecture Planning планирование архитектуры предприятия, Infrastructure Deployment and Management управление технологической инфраструктурой.

### **Тема 9. Модель команды.**

Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение

пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд.

Распределение задач между командами. Распределение ролей в командах.

### **Тема 10. Модель управления рисками.**

Основные сведения о рисках. Планирование управления рисками. Процесс управления рисками: выявление, анализ и приоритезация, планирование, мониторинг, корректирование, извлечение уроков. Управление рисками как составная часть жизненного цикла проекта.

Разработка плана действий в отношении рисков.

### **Тема 11. Модель процессов. Общая схема процесса разработки. Фазы процесса разработки: выработка концепции, планирование.**

Схема процесса разработки. Основные структурные единицы схемы: циклы, фазы и вехи. Цикличность всех уровней процесса разработки. Примеры циклов в различных задачах на различных уровнях: контроль версий, ежедневная сборка, итерация, версия продукта. Фазы и вехи процесса разработки: выработка концепции, планирование проекта, разработка решения, стабилизация решения, внедрение решения. Распределение задач по фазам. Выбор продолжительности фаз. Фаза выработки концепции. Основные задачи фазы: создание ядра проектной группы, формирование видения проекта, подготовка документа общего описания и рамок проекта, подготовка документа оценки рисков и представление главных рисков проекта, выявление и анализ бизнес-требований. Результаты фазы: Общее описание и рамки проекта, Документ оценки рисков, Описание структуры проекта. Фаза планирования. Основные задачи фазы: анализ и документирование проектных требований, подготовка функциональной спецификации, разработка дизайнов, подготовка рабочих планов, оценка проектных затрат и сроков разработки различных составляющих проекта.

Категории проектных требований. Уровни процесса проектирования. Результаты фазы:

Функциональная спецификация, План управления рисками, Сводный план и сводный календарный график проекта.

Каждая команда: - Формирует видение проекта. - Выделяет и выполняет оценку рисков. - Выявляет и анализирует бизнес-требования. - Определяет структуру проекта. Каждая команда: - Разрабатывает дизайн и архитектуру решения. - Создает функциональную спецификацию. - Разрабатывает планы проекта. - Разрабатывает календарный график проекта. Разрабатывает концепцию решения.

### **Тема 12. Модель процессов. Фазы разработки, стабилизации.**

Разработка решения. Фаза разработки. Основная задача фазы - создание компонент решения, включая как документацию, так и программный код. Результаты фазы: "Исходный и исполняемый код приложений", "Скрипты установки и конфигурирования", "Окончательная функциональная спецификация", "Материалы поддержки решения", "Спецификации и сценарии тестов". Стабилизация решения. Фаза стабилизации. Основные задачи фазы: тестирование разработанного решения, приоритезация и устранение ошибок, подготовка решения к выпуску, пилотное внедрение. Результаты фазы: Окончательный продукт,

Документация выпуска, Материалы поддержки решения, Результаты и инструментарий тестирования, Исходный и исполнимый код приложений, Проектная документация,

Анализ пройденной фазы.

Каждая команда: - Создает прототип приложения. - Разрабатывает необходимые компоненты решения. - Тестирует решение.

### **Тема 13. Внедрение решения. Фаза внедрения.**

Основные задачи фазы: внедрение технологий и компонент решения, стабилизация внедренного решения, передача работ персоналу поддержки и сопровождения, получение со стороны заказчика окончательного одобрения результатами проекта, анализ выполненной работы и удовлетворенности заказчика. Результаты фазы: Информационные системы эксплуатации и поддержки, Процедуры и процессы, Базы знаний, отчеты, журналы протоколов, Версии проектных документов, массивы данных и программный код, разработанные во время проекта, Отчет о завершении проекта, Окончательные версии всех проектных документов, Показатели удовлетворенности заказчика и потребителей,

Описание последующих шагов.

Каждая команда: - Внедряет решение в эксплуатацию в другую команду.

### **Тема 14. Модель приложения масштаба предприятия.**

Модель приложения масштаба предприятия. Шесть основных под-моделей: Бизнес-модель приложения; Пользовательская модель; Логическая модель; Технологическая модель; Физическая модель; Модель разработки. Сервисы приложений.

### **Тема 15. Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.**

Организация работы гостиницы Ведение справочников: Номера, Услуги, Клиенты  
Функции: Ведение справочников, поселение и выселение клиентов, бронирование мест, учёт оказанных услуг  
Выходные документы: Счёт за проживание и услуги, Список проживавших на момент времени, Список номеров, Прейскурант услуг.  
2. Организация работы больницы. Ведение справочников: Пациенты, Болезни, Палаты, Врачи, История болезни  
Функции: Ведение справочников, приём пациента, ведение истории болезни, выписка.  
Выходные документы: Список пациентов, Список врачей, Карточка больного.

### **Тема 16. Разработка прикладных программных систем с использованием FMS.**

Организация работы аптеки. Ведение справочников: Группы лекарств, Лекарства, Производители, Поставщики  
Функции: ведение справочников, учёт прихода и продаж лекарств  
Выходные документы: Отчёт по наличию лекарств на складе по группам, Отчёт по продажам по группам, Счёт-фактура.  
2. Организация работы кафе. Ведение справочников: Продукты, Блюда, Заказы  
Функции: Ведение справочников, хранение рецептов, расчёт себестоимости блюда, приём заказов  
Выходные документы: Меню, Счёт заказа, Отчёт по продуктам на складе, Заказы за период.  
3. Деятельность службы трудоустройства.  
4. Организация работы службы общественного питания.  
5. Организация работы службы социальной помощи.

## **6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

<b>№ п/п</b>	<b>Наименование раздела дисциплины</b>	<b>Вид самостоятельной работы обучающихся</b>
1	<b>Предмет программной инженерии.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
2	<b>Жизненный цикл и этапы разработки программного обеспечения.</b>	подготовка к лабораторным занятиям; подготовка к лекциям;

		выполнение аудиторной контрольной работы.
3	<b>Определение требований к программному обеспечению и исходных данных для его проектирования.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
4	<b>Анализ требований и определение спецификаций. Проектирование.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
5	<b>Разработка программного обеспечения.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
6	<b>Обзор методологий проектирования программных продуктов.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
7	<b>Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
8	<b>Методология проектирования приложений MFS. Основные компоненты и модели MFS.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
9	<b>Анализ и составление программ.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
10	<b>Модель команды.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
11	<b>Модель управления рисками.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
12	<b>Модель процессов. Общая схема процесса разработки. Фазы процесса разработки: выработка концепции, планирование.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
13	<b>Модель процессов. Фазы разработки, стабилизации.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
14	<b>Внедрение решения. Фаза внедрения. Модель приложения масштаба предприятия.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
15	<b>Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.
16	<b>Разработка прикладных программных систем с использованием FMS.</b>	подготовка к лабораторным занятиям; подготовка к лекциям; выполнение аудиторной контрольной работы.

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 7.1. Оценочные материалы для проведения текущего контроля успеваемости

№ п/п	Наименование темы (раздела) дисциплины (модуля)	Средства текущего контроля успеваемости	Перечень компетенций
1	Предмет программной инженерии.	Контрольная работа, тест.	ПК-2, ОПК-2
2	Жизненный цикл и этапы разработки программного обеспечения.	Контрольная работа, тест.	ПК-2, ОПК-2
3	Определение требований к программному обеспечению и исходных данных для его проектирования.	Контрольная работа, тест.	ПК-2, ОПК-2
4	Анализ требований и определение спецификаций. Проектирование.	Контрольная работа, тест.	ПК-2, ОПК-2
5	Разработка программного обеспечения.	Контрольная работа, тест.	ПК-2, ОПК-2
6	Обзор методологий проектирования программных продуктов.	Контрольная работа, тест.	ПК-2, ОПК-2
7	Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).	Контрольная работа, тест.	ПК-2, ОПК-2
8	Методология проектирования приложений MFS. Основные компоненты и модели MFS.	Контрольная работа, тест.	ПК-2, ОПК-2
9	Анализ и составление программ.	Контрольная работа, тест.	ПК-2, ОПК-2
10	Модель команды.	Контрольная работа, тест.	ПК-2, ОПК-2
11	Модель управления рисками.	Контрольная работа, тест.	ПК-2, ОПК-2
12	Модель процессов. Общая схема процесса разработки.	Контрольная работа, тест.	ПК-2, ОПК-2

	<b>Фазы процессаразработки: выработка концепции, планирование.</b>		
13	<b>Модель процессов. Фазы разработки, стабилизации.</b>	Контрольная работа, тест.	ПК-2, ОПК-2
14	<b>Внедрение решения. Фаза внедрения. Модель приложения масштаба предприятия.</b>	Контрольная работа, тест.	ПК-2, ОПК-2
15	<b>Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.</b>	Контрольная работа, тест.	ПК-2, ОПК-2
16	<b>Разработка прикладных программных систем с использованием FMS.</b>	Контрольная работа, тест.	ПК-2, ОПК-2

## 7.2. Оценочные материалы для проведения промежуточной аттестации

### 1. Семестр – 10; форма аттестации – зачет.

### 2. Примерный перечень вопросов к зачету.

1. Этапы разработки ПО по методологии (UNIFIED PROCESS).
2. Понятия: постановка задачи, фазы постановки задачи, процесс постановки задачи.
3. Процесс разработки ПО (проектирование, реализация, валидация, тестирование).
4. Что такое UML (жизненный цикл разработки, области приложений, языки реализации и платформы)?
5. Виды UML сущностей.
6. Разница между диаграммой или моделью. Типы диаграмм. Синтаксис диаграммы.
7. Понятие UP. UP и RUP.
8. Аксиомы и итерации UP.
9. Фазы UP. Распределение работ по фазам.
10. Что такое требование? Примеры функциональных требований. Примеры нефункциональных требований. Атрибуты требований.
11. Что такое проектирование ПО? Влияние нефункциональных требований на архитектуру ПО. Методы системной организации.
12. Понятие модели репозитория. Достоинства, недостатки. Клиент-серверная модель. Достоинства, недостатки.
13. Многослойная модель. Достоинства, недостатки.
14. Объектно-ориентированное проектирование. Процесс объектно-ориентированного решения задач.
15. Solid (понятие, расшифровать). Принцип разделения интерфейсов. Принцип инверсии зависимостей, проблемы, которые он решает.

16. Поэтапная разработка. Постепенная разработка, разработка на основе прототипов. Инкрементальная разработка. Достоинства, недостатки.
17. Спиральная разработка. Риски.
18. Компонентно-ориентированная разработка. Достоинства, недостатки.
19. Опишите основные критерии, определяющие уровень формализма, необходимый для разработки ПО.
20. Классы анализа. Состав Классов анализа. Выявление классов анализа. SRC- анализ.
21. Отношения между объектами и классами. Диаграммы объектов. Диаграммы классов. Ассоциация. Синтаксис ассоциации.
22. Зависимости. Виды зависимостей. Наследование.
23. Понятия: модуль, модульность, информационная закрытость.
24. Связность модуля. Корректные и некорректные типы связности(примеры). Связность по совпадению, логическая связность.
25. Временная связность, процедурная связность. Коммуникативная связность, последовательная связность. Связанность модулей. Типы связанности.
26. Принцип проектирования модульной структуры ПО.
27. Мера и метрика, мотивация для метрик. Размерно-ориентированные метрики.
28. Физические и логические строки кода. SLOC. Виды метрик сложности.
29. Метрика Холстеда. Оценки данной метрики.
30. Цикломатическая сложность. Граф потока управления программы. Метрики Чепина. Вычисления метрики Чепина.
31. Понятие тестирования. Тестовый случай. Виды тестирования. Процесс тестирования. Тестирование по стратегии черного ящика. Тестирование по стратегии белого ящика.
32. Тестирование по стратегии серого ящика.
33. Тестовое покрытие, покрытие требований, покрытие кода. JUnit. Примеры

**3. Перечень компетенций и индикаторов их достижения, описание критериев оценивания компетенций представляются в таблице**

Код компетенции, индикаторы достижения компетенции (ИДК)	Уровни освоения компетенций			
	Продвинутый	Базовый	Пороговый	Не освоены компетенции
	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
	«зачтено»			«не зачтено»
<b>ПК-2. Способен осуществлять целенаправленную воспитательную деятельность</b>				
ИДК 1.1 ПК-2.1. Знает способы организации и оценки различных видов внеурочной деятельности ребенка, методы и формы организации коллективных	<i>Критерий 1</i> Обладает твердым и полным знанием материала, владеет дополнительной информацией. Дает полный, развернутый ответ	<i>Критерий 1</i> Знает материал в запланированном объеме. Ответ достаточно полный, но не отражает некоторые аспекты.	<i>Критерий 1</i> Допускает неточности в формулировках. Знает только основной материал.	<i>Критерий 1</i> Не знает значительной части материала. Отвечает на вопрос частично. Не отвечает на поставленные вопросы.

мероприятий				
	<p><i>Критерий 2</i>          Раскрывает структуру и состав изучаемых разделов информатики, демонстрирует сформированные системные знания. Успешно справляется с решением всех поставленных математических задач</p>	<p><i>Критерий 2</i>          Раскрывает структуру и состав некоторых изучаемых разделов информатики. При решении предметных задач допускает единичные ошибки</p>	<p><i>Критерий 2</i>          Фрагментарно описывает структуру и состав изучаемых разделов информатики. Допускает множественные ошибки при решении предметных задач</p>	<p><i>Критерий 2</i>          Не знает структуру и содержание изучаемых разделов информатики. Не справляется с решением предложенных предметных задач</p>
	<p><i>Критерий 3</i>          Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости. Обладает диапазоном практических умений, требуемых для решения определенных проблем в нестандартной ситуации.</p>	<p><i>Критерий 3</i>          Знает основные понятия и ключевые факты в пределах изучаемой области. Обладает диапазоном практических умений, требуемых для решения определенных проблем в пределах изучаемой области.</p>	<p><i>Критерий 3</i>          Обладает базовыми общими знаниями и основными умениями, требуемыми для выполнения простых задач</p>	<p><i>Критерий 3</i>          Неспособен самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения.</p>
<p>ИДК 1.2. ПК-2.2. Умеет оказывать консультативную помощь родителям (законным представителям) обучающихся по вопросам воспитания, в том числе родителям детей с особыми образовательными потребностями.</p>	<p><i>Критерий 1</i>          Обладает твердым и полным знанием материала, владеет дополнительной информацией. Дает полный, развернутый ответ</p>	<p><i>Критерий 1</i>          Знает материал в запланированном объеме. Ответ достаточно полный, но не отражает некоторые аспекты.</p>	<p><i>Критерий 1</i>          Допускает неточности в формулировках. Знает только основной материал.</p>	<p><i>Критерий 1</i>          Не знает значительной части материала. Отвечает на вопрос частично. Не отвечает на поставленные вопросы.</p>

	<i>Критерий 2</i> Самостоятельно анализирует теоретический материал, умеет применять теоретическую базу при выполнении практических заданий, предлагает собственный метод решения.	<i>Критерий 2</i> Правильно применяет теоретическую базу при выполнении практических заданий.	<i>Критерий 2</i> Способен решать задачи по заданному алгоритму. Испытывает затруднения при анализе теоретического материала и его применении на практике.	<i>Критерий 2</i> Не может установить связь теории с практикой. Не может проанализировать теоретический материал и обосновать его использование на практике.
	Критерий 3 Умеет отбирать материал в зависимости от уровня сложности и логики изложения; умеет применять учебный материал в различных формах обучения в соответствии с требованиями ФГОС ОО	Критерий 3 Способен отбирать материал в зависимости от уровня сложности, но допускает неточности в применении учебного материала в различных формах обучения в соответствии с требованиями ФГОС ОО	Критерий 3 Испытывает затруднения в отборе материала, связанные с логикой изложения и с применением учебного материала в различных формах обучения в соответствии с требованиями ФГОС ОО	Не умеет соотносить содержание изучаемых дисциплин с содержанием школьного курса информатики
<b>ОПК-2. Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)</b>				
ИДК 1.1 ОПК-2.1. Знает специфику и структуру основных образовательных программ по информатике, программ дополнительного образования;  основные элементы педагогических и других технологий, используемых при разработке образовательных	<i>Критерий 1</i> Обладает твердым и полным знанием материала, владеет дополнительной информацией. Дает полный, развернутый ответ	<i>Критерий 1</i> Знает материал в запланированном объеме. Ответ достаточно полный, но не отражает некоторые аспекты.	<i>Критерий 1</i> Допускает неточности в формулировках. Знает только основной материал.	<i>Критерий 1</i> Не знает значительной части материала. Отвечает на вопрос частично. Не отвечает на поставленные вопросы.

программ				
	<p><i>Критерий 2</i>          Раскрывает структуру и состав изучаемых разделов информатики, демонстрирует сформированные системные знания. Успешно справляется с решением всех поставленных математических задач</p>	<p><i>Критерий 2</i>          Раскрывает структуру и состав некоторых изучаемых разделов информатики. При решении предметных задач допускает единичные ошибки</p>	<p><i>Критерий 2</i>          Фрагментарно описывает структуру и состав изучаемых разделов информатики. Допускает множественные ошибки при решении предметных задач</p>	<p><i>Критерий 2</i>          Не знает структуру и содержание изучаемых разделов информатики. Не справляется с решением предложенных предметных задач</p>
	<p><i>Критерий 3</i>          Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости. Обладает диапазоном практических умений, требуемых для решения определенных проблем в нестандартной ситуации.</p>	<p><i>Критерий 3</i>          Знает основные понятия и ключевые факты в пределах изучаемой области. Обладает диапазоном практических умений, требуемых для решения определенных проблем в пределах изучаемой области.</p>	<p><i>Критерий 3</i>          Обладает базовыми общими знаниями и основными умениями, требуемыми для выполнения простых задач</p>	<p><i>Критерий 3</i>          Неспособен самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения.</p>
<p>ИДК 1.2.          ОПК-2.2. Умеет разрабатывать отдельные компоненты образовательной программы;          разрабатывать элементы образовательных программ для разных профилей</p>	<p><i>Критерий 1</i>          Обладает твердым и полным знанием материала, владеет дополнительной информацией. Дает полный, развернутый ответ</p>	<p><i>Критерий 1</i>          Знает материал в запланированном объеме. Ответ достаточно полный, но не отражает некоторые аспекты.</p>	<p><i>Критерий 1</i>          Допускает неточности в формулировках. Знает только основной материал.</p>	<p><i>Критерий 1</i>          Не знает значительной части материала. Отвечает на вопрос частично. Не отвечает на поставленные вопросы.</p>

<p>обучения; составлять индивидуальные учебные планы, в соответствии с образовательными потребностями обучающихся, в том числе, на углублённом уровне.</p>				
	<p><i>Критерий 2</i> Самостоятельно анализирует теоретический материал, умеет применять теоретическую базу при выполнении практических заданий, предлагает собственный метод решения.</p>	<p><i>Критерий 2</i> Правильно применяет теоретическую базу при выполнении практических заданий.</p>	<p><i>Критерий 2</i> Способен решать задачи по заданному алгоритму. Испытывает затруднения при анализе теоретического материала и его применении на практике.</p>	<p><i>Критерий 2</i> Не может установить связь теории с практикой. Не может проанализировать теоретический материал и обосновать его использование на практике.</p>
	<p>Критерий 3 Умеет отбирать материал в зависимости от уровня сложности и логики изложения; умеет применять учебный материал в различных формах обучения в соответствии с требованиями ФГОС ОО</p>	<p>Критерий 3 Способен отбирать материал в зависимости от уровня сложности, но допускает неточности в применении учебного материала в различных формах обучения в соответствии с требованиями ФГОС ОО</p>	<p>Критерий 3 Испытывает затруднения в отборе материала, связанные с логикой изложения и с применением учебного материала в различных формах обучения в соответствии с требованиями ФГОС ОО</p>	<p>Не умеет соотносить содержание изучаемых дисциплин с содержанием школьного курса информатики</p>

## 8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 8.1. Перечень основной учебной литературы

1. Архитектура и проектирование программных систем: Монография / С.В. Назаров. - М.: НИЦИнфра-М, 2013. - 351 с.: 60x88 1/16. - (Научная мысль). (обложка) ISBN 978-5-16-005735-4, 200экз.  
<http://znaniyum.com/bookread.php?book=353187>
2. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина,

Е.В.Кокорева,Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с.: ил.; 60x90 1/16. - (Высшее обр.). (п) ISBN 978-5-8199-0342-1, 500 экз.

<http://znanium.com/bookread.php?book=389963>

3. Практическая программная инженерия на основе учебного примера [Электронный ресурс] / Л. А. Мачашек, Б. Л. Лионг ; пер. с англ. ? 2-е изд. (эл.). ? М. : БИНОМ. Лаборатория знаний, 2012. ? 956 с. : ил. ? (Программисту).

<http://e.lanbook.com/view/book/8766/>

## **8.2. Перечень дополнительной учебной литературы**

1. Проектирование программного обеспечения экономических информационных систем : учебник / А. М. Вендров .? 2-е изд., перераб. и доп. ? М. : Финансы и статистика, 2005 .? 544 с.

2. Проектирование информационных систем: Учебное пособие / Н.Н. Заботина. - М.: ИНФРА-М, 2011. - 331 с.: 60x90 1/16 + CD-ROM. - (Высшее образование). (переплет, cd rom)ISBN 978-5-16-004509-2, 500 экз. <http://znanium.com/bookread.php?book=209816>

## **8.3. Перечень Интернет-ресурсов, необходимых для освоения дисциплины (модуля)**

1.ЭБС «Консультант студента» (<http://www.studentlibrary.ru>). Основным разработчиком проекта является издательская группа «ГЭОТАР-Медиа»

2.ЭБС «Рукопт» (<http://www.rucont.ru>). ОАО «Центральный коллектор библиотек «БИБКОМ» проект Контекстум)

3. ЭБС «Лань» (<http://e.lanbook.com>).

## **8.4. Перечень информационных технологий и программного обеспечения**

Для осуществления образовательного процесса по дисциплине необходимо использование следующего лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства:

1. Microsoft Office 2010
2. Среды программирования PascalABC.NET, Python

При проведении обучения используются следующие информационные системы и программы:

1. Электронная библиотека курса, конспекты лекций, программное обеспечение, задания для лабораторных и практических занятий и самостоятельной работы, варианты тестовых заданий для проверки текущих и остаточных знаний студентов, варианты заданий для текущего и промежуточного контроля знаний обучающихся

2. Компьютерное и мультимедийное оборудование.

3. Система компьютерного тестирования (MyTestX).

4. ИС “Рейтинг студентов” – учет учебной деятельности студентов с использованием балльно-рейтингового метода оценивания.

5. При проведении обучения по дисциплине используются активные и интерактивные формы обучения, включая: лекции-визуализации, лекции-беседы, лекции с разбором конкретных ситуаций.

Лекции-визуализации используются на этапе введения студентов в новую тему. Они основаны на использовании в качестве наглядного материала мультимедийной презентации, содержащей такие формы наглядности, как схемы, рисунки, диаграммы и т.д. После освоения студентам базовых знаний по изучаемой теме проводятся лекции беседы, когда студентам адресуются вопросы для обсуждения в начале лекции и по ее ходу. Для пояснения материала изучаемой темы на практическом примере используются лекции с разбором конкретных ситуаций.

## **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для осуществления образовательного процесса по дисциплине необходима следующая материально-техническая база:

- библиотечный фонд (учебная, учебно-методическая, справочная литература);
- компьютеризированные рабочие места для обучаемых с доступом в сеть Интернет;
- аудитории, оборудованные проекционной техникой.

Для проведения лекционных занятий используется лекционный зал ИМФиИТО, оборудованный проектором и интерактивной доской (ауд. №44).

Для проведения лабораторных занятий используются компьютерные классы кафедры информатики и вычислительной техники (ауд. № 43, 47)), оборудованные современными персональными компьютерами с соответствующим программным обеспечением:

- ауд. № 43 - компьютерный зал:

ПЭВМ в сборе: CPUAMD Athlon (tm)4840 Quad Core Processor-3,10 GHz/DDR 4 Gb/HDD 500 Gb. Монитор: MUY19HJLJCQ959494B – 12 шт;

Все персональные компьютеры подключены к сети университета и имеют выход в глобальную сеть Интернет.

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Приступая к изучению дисциплины, обучающимся целесообразно ознакомиться с ее рабочей программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, а также с предлагаемым перечнем заданий.

### ***Рекомендации по подготовке к аудиторным занятиям***

#### ***Лекционные занятия***

Умение сосредоточенно слушать лекции, активно воспринимать излагаемые сведения – это важнейшее условие освоения данной дисциплины. Каждая из лекций сопровождается компьютерной презентацией. Кроме того, в конце каждой лекции с целью создания условий для осмысления содержания лекционного материала обучающимся предлагается ответить на вопрос для размышления. Краткие записи лекций, их конспектирование помогает усвоить материал. Поэтому в ходе лекционных занятий необходимо вести конспектирование учебного материала, обращая внимание на самое важное и существенное в нем. Имеет смысл оставить в рабочих конспектах поля, на которых делать пометки, замечания, дополнения. Целесообразно разработать собственную "маркографию" (значки, символы), сокращения слов.

#### ***Практические занятия***

В ходе подготовки к практическим занятиям необходимо изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в

периодических изданиях: журналах, газетах и т.д. При этом важно учитывать рекомендации преподавателя и требования учебной программы. Важно также опираться на конспекты лекций. В ходе занятия важно внимательно слушать выступления своих однокурсников. При необходимости задавать им уточняющие вопросы, активно участвовать в обсуждении изучаемых вопросов. В ходе своего выступления целесообразно использовать как технические средства обучения, так и традиционные, то есть доску и мел (при необходимости).

#### ***Организация внеаудиторной деятельности обучающихся***

Внеаудиторная деятельность обучающегося по данной дисциплине предполагает самостоятельный поиск информации, необходимой, во-первых, для выполнения заданий самостоятельной работы (инвариантной и вариативной частей) и, во-вторых, подготовку к текущей и промежуточной аттестации. Успешная организация времени по усвоению данной дисциплины во многом зависит от наличия у обучающегося умения самоорганизовать себя и своё время для выполнения предложенных домашних заданий.

#### ***Подготовка к зачету (экзамену)***

В процессе подготовки к зачету обучающемуся рекомендуется так организовать свою учебу, чтобы все виды работ и заданий, предусмотренные рабочей программой, были выполнены в срок. Основное в подготовке к зачету - это повторение всего материала учебной дисциплины. В дни подготовки к зачету необходимо избегать чрезмерной перегрузки умственной работой, чередуя труд и отдых. При подготовке к сдаче зачета старайтесь весь объем работы распределять равномерно по дням, отведенным для подготовки к зачету, контролировать каждый день выполнения работы. Лучше, если можно перевыполнить план. Тогда всегда будет резерв времени. При подготовке к зачету целесообразно повторять пройденный материал в строгом соответствии с учебной программой, примерным перечнем учебных вопросов, заданий, которые выносятся на зачет и содержащихся в данной программе.

## **11. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Под специальными условиями для получения образования обучающихся с ограниченными возможностями здоровья понимаются условия обучения, воспитания и развития таких студентов, включающие в себя использование при необходимости адаптированных образовательных программ и методов обучения и воспитания, специальных учебников, учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего необходимую помощь, проведение групповых и индивидуальных коррекционных занятий, обеспечение доступа в здания вуза и другие условия, без которых невозможно или затруднено освоение образовательных программ обучающихся с ограниченными возможностями здоровья.

Обучение в рамках учебной дисциплины обучающихся с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Обучение по учебной дисциплине обучающихся с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах.

В целях доступности обучения по дисциплине обеспечивается:

- 1) для лиц с ограниченными возможностями здоровья по зрению:

- наличие альтернативной версии официального сайта института в сети «Интернет» для слабовидящих;

- весь необходимый для изучения материал, согласно учебному плану (в том числе, для обучающихся по индивидуальным учебным планам) предоставляется в электронном виде на диске.

- индивидуальное равномерное освещение не менее 300 люкс;

- присутствие ассистента, оказывающего обучающемуся необходимую помощь;

- обеспечение возможности выпуска альтернативных форматов печатных материалов (крупный шрифт или аудиофайлы);

- обеспечение доступа обучающегося, являющегося слепым и использующего собаку-проводника, к зданию института.

2) для лиц с ограниченными возможностями здоровья по слуху:

- наличие микрофонов и звукоусиливающей аппаратуры коллективного пользования (аудиоколонки);

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, материально-технические условия должны обеспечивать возможность беспрепятственного доступа обучающихся в учебные помещения, столовые, туалетные и другие помещения организации, а также пребывания в указанных помещениях (наличие пандусов, поручней, расширенных дверных проемов и других приспособлений).

Перед началом обучения могут проводиться консультативные занятия, позволяющие студентам с ограниченными возможностями адаптироваться к учебному процессу.

В процессе ведения учебной дисциплины профессорско-преподавательскому составу рекомендуется использование социально-активных и рефлексивных методов обучения, технологий социокультурной реабилитации с целью оказания помощи обучающимся с ограниченными возможностями здоровья в установлении полноценных межличностных отношений с другими обучающимися, создании комфортного психологического климата в учебной группе.

Особенности проведения текущей и промежуточной аттестации по дисциплине для обучающихся с ограниченными возможностями здоровья устанавливаются с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и другое). При необходимости предоставляется дополнительное время для подготовки ответа на зачете или экзамене.



## АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ): «Программная инженерия»

**Цель освоения дисциплины (модуля):** изучение разделов: предмет программной инженерии, жизненный цикл и этапы разработки программного обеспечения, определение требований к программному обеспечению и исходных данных для его проектирования, анализ требований и определение спецификаций, проектирование, разработка программного обеспечения обзор методологий проектирования программных продуктов, визуальное моделирование при анализе и проектировании, основы Unified Modeling Language (UML), методология проектирования приложений MFS, основные компоненты и модели MFS, модель команды, модель управления рисками, модель процессов, общая схема процесса разработки, фазы процесса разработки: выработка концепции, планирование, модель процессов, фазы разработки, стабилизации, внедрение решения, фаза внедрения, модель приложения масштаба предприятия, визуальное проектирование прикладных программных систем с использованием унифицированного языка UML, разработка прикладных программных систем с использованием FMS.

### 1. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» относится к вариативной части образовательной программы бакалавриата по направлению 44.03.05 Педагогическое образование.

### 2. Требования к результатам освоения дисциплины(модуля):

Код компетенции	Содержание компетенции	Индикаторы достижения компетенций
ПК-2	Способен осуществлять целенаправленную воспитательную деятельность	<p>ПК-2.1. Демонстрирует умение постановки воспитательных целей, проектирования воспитательной деятельности и методов ее реализации в соответствии с требованиями ФГОС ОО и спецификой учебного предмета.</p> <p>ПК-2.2. Демонстрирует способы организации и оценки различных видов внеурочной деятельности ребенка (учебной, игровой, трудовой, спортивной, художественной и т.д.), методы и формы организации коллективных творческих дел, экскурсий, походов, экспедиций и других мероприятий (по выбору).</p> <p>ПК-2.3. Выбирает и демонстрирует способы оказания консультативной помощи родителям (законным представителям) обучающихся по вопросам воспитания, в том числе родителям детей с особыми образовательными потребностями.</p>

ОПК-2	Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)	<p>ОПК 2.1. Разрабатывает программы учебных предметов, курсов, дисциплин (модулей), программы дополнительного образования в соответствии с нормативно-правовыми актами в сфере образования.</p> <p>ОПК 2.2. Проектирует индивидуальные образовательные маршруты освоения программ учебных предметов, курсов, дисциплин (модулей), программ дополнительного образования в соответствии с образовательными потребностями обучающихся.</p> <p>ОПК 2.3. Осуществляет отбор педагогических и других технологий, в том числе информационно-коммуникационных, используемых при разработке основных и дополнительных образовательных программ и их элементов</p>
-------	--	--

3. **Общая трудоемкость дисциплины (модуля) составляет 2 зачетные единицы (72 часа).**

4. **Семестр: 10**

5. **Основные разделы дисциплины (модуля):**

**Тема 1. Предмет программной инженерии.**

Программная инженерия, основные понятия. Основные поддисциплины программной инженерии.

**Тема 2. Жизненный цикл и этапы разработки программного обеспечения.**

Международный стандарт ISO\IEC 12207, описывающий структуру жизненного цикла программного обеспечения и его процессы. Процессы жизненного цикла разработки программного обеспечения: постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение. Международные стандарты оценки качества процессов создания программного обеспечения.

**Тема 3. Определение требований к программному обеспечению и исходных данных для его проектирования.**

Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Обследование системы, общение с заказчиком, планирование разработки. Разработка технического задания. Основные разделы технического задания. Принципиальные решения начальных этапов проектирования: выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор подхода к разработке, выбор метода проектирования, выбор или формирование стандартов разработки. Распределение финансовых и временных затрат на реализацию каждого из этапов разработки программного обеспечения.

Разработка технического задания на создание системы "Учет успеваемости студентов". Сведения об успеваемости студентов должны храниться в течении всего

срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

#### **Тема 4. Анализ требований и определение спецификаций. Проектирование. Разработка программного обеспечения.**

Для учебного примера "Система бронирования билетов для авиакомпании" произвести: Анализ требований и определение спецификаций. Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; анализ требований к системе (определение ее функциональных возможностей, пользовательских требований, требований надежности и безопасности, требований к внешним интерфейсам). Проектирование.

Проектирование архитектуры системы (определение состава необходимого оборудования, программного обеспечения и т.д.). Проектирование архитектуры программного обеспечения (определение структуры программного обеспечения, документирование интерфейсов его компонентов, требований к тестам и плана интеграции). Детальное проектирование программного обеспечения (подробное описание компонентов программного обеспечения и интерфейсов между ними, обновление пользовательской документации, разработка и документирование требований к тестам, обновление плана интеграции компонентов).

Разработка программного обеспечения. Кодирование и тестирование программного обеспечения (разработка и документирование каждого компонента, а также совокупности процедур и данных для их тестирования). Интеграция программного обеспечения (сборка программных компонентов в соответствии с планом интеграции).

Выбор модели жизненного цикла, стандартов, методов и средств разработки, составление плана работ; Проектирование архитектуры программного обеспечения; Кодирование и тестирование программного обеспечения.

#### **Тема 5. Обзор методологий проектирования программных продуктов.**

Модели процесса создания ПО. Каскадная и эволюционная модели. Итерационные модели процесса создания ПО. Модель пошаговой разработки, спиральная модель. Технология быстрой разработки приложений RAD (Rapid Application Development).

#### **Тема 7. Визуальное моделирование при анализе и проектировании. Основы Unified Modeling Language (UML).**

Унифицированный язык моделирования UML. Модели спецификации разрабатываемого программного обеспечения при использовании UML: использования, логическая, реализации, процессов, развертывания. Основные типы UML-диаграмм, входящие в различные модели: диаграммы прецедентов, диаграммы классов, диаграммы взаимодействия, диаграммы состояний, диаграммы видов деятельности, диаграммы компонентов, диаграммы базы данных, диаграммы развертывания. Этапы проектирования ПО с применением UML: моделирование прецедентов, создание модели бизнес-объектов, создание концептуальной модели данных, разработка требований к системе, проектирование моделей базы данных и приложений, проектирование физической реализации системы.

Разбиение студентов на команды, выделение действующих лиц и построение диаграммы вариантов использования для учебного примера "Система бронирования билетов для авиакомпании" Переход от диаграмм вариантов использования к диаграммам классов.

Создание диаграмм классов на примере предыдущей задачи. Построение концептуальной модели для системы решения комбинаторно-оптимизационных задач. Построение диаграммы последовательностей системы для варианта использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Построение

диаграммы деятельности, уточняющую вариант использования ?Выполнение задания решения комбинаторно-оптимизационных задач?. Использование языка UML для проектирования системы учета успеваемости студентов.

### **Тема 8. Методология проектирования приложений MFS. Основные компоненты и модели MFS.**

Основные модели MFS: команды, процесса, приложения, архитектуры корпорации, проектирования решений, управление инфраструктурой, стоимости владения проектом. Базовые компоненты MFS: Solution Development Discipline (SDD) дисциплина разработки приложений, Designing Component Solutions (DCS) проектирование компонентного ПО, Enterprise Architecture Planning планирование архитектуры предприятия, Infrastructure Deployment and Management управление технологической инфраструктурой.

### **Тема 9. Модель команды.**

Основные направления по которым организуются команды специалистов: управление продуктом, управление программой, разработка, тестирование, обучение пользователей, сопровождение (логистика). Основные задачи, обязанности каждой из команд, требуемые профессиональные качества членов команд. Распределение задач между командами. Распределение ролей в командах.

### **Тема 10. Модель управления рисками.**

Основные сведения о рисках. Планирование управления рисками. Процесс управления рисками: выявление, анализ и приоритезация, планирование, мониторинг, корректирование, извлечение уроков. Управление рисками как составная часть жизненного цикла проекта. Разработка плана действий в отношении рисков.

### **Тема 11. Модель процессов. Общая схема процесса разработки. Фазы процесса разработки: выработка концепции, планирование.**

Схема процесса разработки. Основные структурные единицы схемы: циклы, фазы и вехи. Цикличность всех уровней процесса разработки. Примеры циклов в различных задачах на различных уровнях: контроль версий, ежедневная сборка, итерация, версия продукта. Фазы и вехи процесса разработки: выработка концепции, планирование проекта, разработка решения, стабилизация решения, внедрение решения. Распределение задач по фазам. Выбор продолжительности фаз. Фаза выработки концепции. Основные задачи фазы: создание ядра проектной группы, формирование видения проекта, подготовка документа общего описания и рамок проекта, подготовка документа оценки рисков и представление главных рисков проекта, выявление и анализ бизнес-требований. Результаты фазы: Общее описание и рамки проекта, Документ оценки рисков, Описание структуры проекта. Фаза планирования. Основные задачи фазы: анализ и документирование проектных требований, подготовка функциональной спецификации, разработка дизайнов, подготовка рабочих планов, оценка проектных затрат и сроков разработки различных составляющих проекта.

Категории проектных требований. Уровни процесса проектирования. Результаты фазы:

Функциональная спецификация, План управления рисками, Сводный план и сводный календарный график проекта.

Каждая команда: - Формирует видение проекта. - Выделяет и выполняет оценку рисков. - Выявляет и анализирует бизнес-требования. - Определяет структуру проекта.

Каждая команда: - Разрабатывает дизайн и архитектуру решения. - Создает

функциональную спецификацию. - Разрабатывает планы проекта. - Разрабатывает календарный график проекта. Разрабатывает концепцию решения.

### **Тема 12. Модель процессов. Фазы разработки, стабилизации.**

Разработка решения. Фаза разработки. Основная задача фазы - создание компонент решения, включая как документацию, так и программный код. Результаты фазы: "Исходный и исполняемый код приложений", "Скрипты установки и конфигурирования", "Окончательная функциональная спецификация", "Материалы поддержки решения", "Спецификации и сценарии тестов". Стабилизация решения. Фаза стабилизации. Основные задачи фазы: тестирование разработанного решения, приоритезация и устранение ошибок, подготовка решения к выпуску, пилотное внедрение. Результаты фазы: Окончательный продукт, Документация выпуска, Материалы поддержки решения, Результаты и инструментарий тестирования, Исходный и исполнимый код приложений, Проектная документация, Анализ пройденной фазы.

Каждая команда: - Создает прототип приложения. - Разрабатывает необходимые компоненты решения. - Тестирует решение.

### **Тема 13. Внедрение решения. Фаза внедрения.**

Основные задачи фазы: внедрение технологий и компонент решения, стабилизация внедренного решения, передача работ персоналу поддержки и сопровождения, получение со стороны заказчика окончательного одобрения результатами проекта, анализ выполненной работы и удовлетворенности заказчика. Результаты фазы: Информационные системы эксплуатации и поддержки, Процедуры и процессы, Базы знаний, отчеты, журналы протоколов, Версии проектных документов, массивы данных и программный код, разработанные во время проекта, Отчет о завершении проекта, Окончательные версии всех проектных документов, Показатели удовлетворенности заказчика и потребителей, Описание последующих шагов.

Каждая команда: - Внедряет решение в эксплуатацию в другую команду.

### **Тема 14. Модель приложения масштаба предприятия.**

Модель приложения масштаба предприятия. Шесть основных под-моделей: Бизнес-модель приложения; Пользовательская модель; Логическая модель; Технологическая модель; Физическая модель; Модель разработки. Сервисы приложений.

### **Тема 15. Визуальное проектирование прикладных программных систем с использованием унифицированного языка UML.**

Организация работы гостиницы Ведение справочников: Номера, Услуги, Клиенты  
Функции: Ведение справочников, поселение и выселение клиентов, бронирование мест, учёт оказанных услуг  
Выходные документы: Счёт за проживание и услуги, Список проживавших на момент времени, Список номеров, Прейскурант услуг. 2. Организация работы больницы. Ведение справочников: Пациенты, Болезни, Палаты, Врачи, История болезни  
Функции: Ведение справочников, приём пациента, ведение истории болезни, выписка. Выходные документы: Список пациентов, Список врачей, Карточка больного.

### **Тема 16. Разработка прикладных программных систем с использованием FMS.**

Организация работы аптеки. Ведение справочников: Группы лекарств, Лекарства, Производители, Поставщики  
Функции: ведение справочников, учёт прихода и продаж лекарств  
Выходные документы: Отчёт по наличию лекарств на складе по группам,

Отчёт по продажам по группам, Счёт-фактура. 2. Организация работы кафе. Ведение справочников: Продукты, Блюда, Заказы Функции: Ведение справочников, хранение рецептов, расчёт себестоимости блюда, приём заказов Выходные документы: Меню, Счёт заказа, Отчёт по продуктам на складе, Заказы за период. 3. Деятельность службы трудоустройства. 4. Организация работы службы общественного питания. 5. Организация работы службы социальной помощи.

**6. Формы текущего контроля успеваемости и промежуточной аттестации: зачет.**

**Автор:** *Рагимханова Г. С., доцент кафедры информатики и ВТ*