Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Курганский государственный университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Первый проректор
_____ Т.Р. Змызгова
« 31 »августа 2021 г.

Рабочая программа учебной дисциплины

ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

образовательной программы высшего образования –
программы бакалавриата

09.03.03 Прикладная информатика

Направленность
Интеллектуальные информационные системы и технологии
Форма обучения: очная, заочная

09.03.04 Программная инженерия

Направленность
Программное обеспечение автоматизированных систем
Форма обучения: очная, заочная

Курган 2021

Рабочая программа дисциплины «Основы программной инженерии» составлена в соответствии с учебным планом программы бакалавриата: «Прикладная информатика», (Интеллектуальные информационные системы и технологии), «Программная инженерия» (Программное обеспечение автоматизированных систем), утвержденным для очной и заочной форм обучения 30 августа 2021 г.

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» 30.08.2021 года, протокол № 1.

Рабочую программу составил:
к.т.н., доцент кафедры ПОАС _____/В. К. Волк/

Согласовано:

Заведующий кафедрой
«Программное обеспечение
автоматизированных систем» _____ /В. К. Волк/

Специалист
по учебно-методической работе
Учебно-методического отдела _____/Г. В Казанкова/

Начальник
Управления образовательной
деятельности _____/С.Н. Синицын/

СОДЕРЖАНИЕ

1. ОБЪЕМ ДИСЦИПЛИНЫ

09.03.03 – Прикладная информатика
Общая трудоемкость – 3 зач. ед. (108 акад. часов)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)			
	Очная форма обучения		Заочная форма обучения	
	Всего	4-й семестр	Всего	5-й семестр
Аудиторные занятия:	**48**	**48**	**8**	**8**
Лекции	16	16	2	2
Лабораторные работы	32	32	6	6
Самостоятельная работа:	**60**	**60**	**100**	**100**
Выполнение контрольной работы	18	18	18	18
Подготовка к зачету	18	18	18	18
Прочие виды	24	24	64	64
Вид промежуточной аттестации		ЗАЧЕТ		ЗАЧЕТ

09.03.04 – Программная инженерия
Общая трудоемкость – 3 зач. ед. (108 акад. часов)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)			
	Очная форма обучения		Заочная форма обучения	
	Всего	4-й семестр	Всего	5-й семестр
Аудиторные занятия:	**48**	**48**	**8**	**8**
Лекции	16	16	2	2
Лабораторные работы	32	32	6	6
Самостоятельная работа:	**60**	**60**	**100**	**100**
Выполнение контрольной работы	18	18	18	18
Подготовка к зачету	18	18	18	18
Прочие виды	24	24	64	64
Вид промежуточной аттестации		ЗАЧЕТ		ЗАЧЕТ

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

Дисциплина «Основы программной инженерии» включена в вариативную часть блока 1 учебных планов образовательных программ. Дисциплина базируется на курсах «Основы программирования» и «Объектно-ориентированное программирование» и создает методологическую основу для изучения дисциплин технологического блока: «Базы данных», «Разработка и анализ требований», "Технологии проектирования программных/информационных систем», «Управление качеством и тестирование ПО», «Управление программными проектами».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплины – введение в промышленные технологии разработки программного обеспечения.

Задачи дисциплины:

изучение:

- основных понятий, методологических основ и стандартов программной инженерии;
- структуры процессов жизненного цикла программного продукта;
- основных моделей жизненного цикла программного продукта.

практическое освоение:

- основ языка визуального моделирования (UML), используемого при анализе и проектировании ПО;
- CASE-средств поддержки программных проектов.

Компетенции, формируемые в результате освоения дисциплины:

09.03.03 – Прикладная информатика:

- способность моделировать прикладные (бизнес) процессы и предметную область, использовать методы и инструментальные средства разработки программных проектов на стадиях технического задания, технологии концептуального, функционального и логического проектирования (ПК-4).

09.03.04 – Программная инженерия:

- владение стандартами и моделями жизненного цикла программного продукта (ПК-4);
- владение методами и инструментальными средствами разработки программных проектов на стадиях технического задания, концептуального, функционального и логического проектирования (ПК-5).

В результате освоения дисциплины студент должен демонстрировать следующие ***результаты обучения***:

Должен знать:

- историю развития технологий разработки программного обеспечения (ПО);
- профессиональную терминологию программной инженерии;
- состав и содержание типовых стадий жизненного цикла ПО;

- состав и структуру стандартных процессов жизненного цикла ПО;
- основные модели жизненного цикла ПО;
- основы языка визуального моделирования программных систем (UML);
- типовую ролевую модель команды программного проекта;
- профессионально-этические принципы программной инженерии (кодекс этики IEEE-CS/ACM).

Должен уметь:

- использовать UML-ориентированные CASE-средства для документирования программных проектов.

Должен владеть

- стандартными приемами графического моделирования проектируемой программной системы на разных стадиях программного проекта.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

4.1.1 Очная форма обучения

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабораторные работы
1	Предмет и базовые понятия программной инженерии	**2**	**-**
2	Процессы и модели жизненного цикла ПО	**4**	**-**
	Рубежный контроль №1	**2**	**-**
3	Визуальное моделирование при анализе и проектировании программных систем	**8**	**30**
	Рубежный контроль №2		**2**
Всего по дисциплине:		**16**	**32**

4.1.2 Заочная форма обучения

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабораторные работы
1	Предмет и базовые понятия программной инженерии	**1**	**-**
2	Процессы и модели жизненного цикла ПО	**1**	**-**
3	Визуальное моделирование при анализе и проектировании программных систем	**-**	**6**
Всего по дисциплине:		**2**	**6**

4.2 Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Раздел №1. **ПРЕДМЕТ И БАЗОВЫЕ ПОНЯТИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ**		
Лекция 1. ***Введение. Программная инженерия: история и базовые понятия*** Цели и задачи изучения дисциплины; взаимосвязи с другими дисциплинами учебных планов, обзор рабочей программы и учебно-методических материалов. Краткий исторический очерк: модульное программирование, структурный и объектно-ориентированный подходы к программированию и проектированию программных систем. Предмет и методология программной инженерии: понятие инженерной деятельности; отличия программной инженерии от традиционных инженерных областей; отличия программной инженерии от информатики; проектирование и моделирование; понятие и классификация CASE-средств. Терминология программной инженерии.	**2**	**1**
Раздел №2. **ПРОЦЕССЫ И МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА ПО**		
Лекция 2. ***Жизненный цикл ПО*** Понятие и основные стадии жизненного цикла (ЖЦ) промышленного изделия; особенности ЖЦ ПО. Стандартизация в программной инженерии: классификация стандартов (международные, государственные, отраслевые, корпоративные); обзор стандартов качества ПО. Состав и структура процессов ЖЦ ПО (стандарт ISO/ГОСТ 12207); Ролевая модель команды программного проекта; обзор профессиональных и образовательных стандартов IT-отрасли.	**2**	**1**
Лекция 3. ***Модели жизненного цикла ПО*** Понятие модели ЖЦ. Особенности и области применения моделей (каскадная модель, V-модель, эволюционные модели, спиральная модель, модели быстрой разработки).	**2**	**0**
Рубежный контроль №1.	**2**	**-**
Раздел №3. **ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРИ АНАЛИЗЕ И ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ СИСТЕМ**		
Лекция 4. ***Проектирование ПО*** Проектирование как процесс преобразования информационных моделей объекта. Задачи и базовые принципы проектирования сложных объектов (абстрагирование, декомпозиция, многоэтапность, итерационность). Визуализация при проектировании. Обзор систем графического моделирования. Понятие и классификация CASE-средств. Язык графического моделирования UML: история стандартизации UML; структура и базовые понятия языка; модели и UML-диаграммы.	**2**	**0**

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Лекция 5. ***UML-диаграммы вариантов использования (UseCase-diagram)*** Use Case-модель как результат функциональной декомпозиции проектируемой системы на начальной стадии программного проекта. Назначение и область применения Use-Case-моделей. Компоненты модели и графическая нотация. Сценарии вариантов использования. Примеры UseCase-диаграмм.	**2**	**0**
Лекция 6. ***UML-диаграммы классов и пакетов (Class-*** и ***Package-diagram)*** Статические модели концептуального и логического уровней. «Пакеты» и «Классы» – два уровня структурной декомпозиции проектируемой системы. Компоненты моделей (пакеты, классы, интерфейсы, отношения), графическая нотация. Примеры диаграмм.	**2**	**0**
Лекция 7. ***UML-диаграммы состояний (State Chart-*** или ***State Machine-diagram)*** Назначение и область применения динамических UML-моделей. Модели логического уровня. Компоненты модели: простые и составные состояния; события и переходы; триггеры. Обозначения и графическая нотация. Примеры диаграмм.	**2**	**0**
Всего часов лекционных занятий	**16**	**2**

4.3 Лабораторный практикум

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Раздел №3. **ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРИ АНАЛИЗЕ И ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ СИСТЕМ**		
Лабораторная работа № 1. ***Подготовка к выполнению учебного программного проекта*** Рассмотрение примера выполнения программного проекта. Формирование команд разработчиков учебных программных проектов. Постановка задачи разработки.	**2**	**0**
Лабораторная работа № 2. ***Разработка UseCase-модели*** Стадия технического задания учебного программного проекта: формирование терминологического словаря предметной области; разработка обобщенной диаграммы вариантов использования.	**6**	**1**

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
Лабораторная работа № 3. ***Разработка диаграммы пакетов, UseCase-моделей и сценариев вариантов использования*** Стадия эскизного проекта: структурная декомпозиция проектируемой системы (разработка диаграммы пакетов); функциональная декомпозиция подсистем (разработка локальных Use Case-диаграмм и сценариев вариантов использования)	**8**	**1**
Лабораторная работа № 4. ***Разработка диаграммы классов*** Стадия технического проекта – разработка статической модели проектируемой системы (компонента системы): структурная декомпозиция пакетов, разработка диаграмм классов.	**8**	**2**
Лабораторная работа № 5. ***Разработка диаграммы состояний*** Стадия технического проекта – разработка динамической модели поведения проектируемой системы (компонента системы): разработка диаграммы состояний.	**6**	**2**
Рубежный контроль №2.	**2**	**-**
Всего часов лабораторных занятий	**32**	**6**

4.4 Контрольная работа

Контрольная работа (в форме индивидуального домашнего задания) выполняется студентами очной и заочной форм обучения и предполагает выполнение учебного мини-проекта. Темы проектов и методические указания по их выполнению и документальному оформлению приведены в соответствующем разделе учебного пособия [2].

Поэтапное выполнение и защита контрольной работы студентами очной формы обучения синхронизированы с выполнением лабораторных работ (п. 4.3 рабочей программы).

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1 Курс лекций

Конспект лекций (краткий обзор рассматриваемых на лекциях вопросов) представлен в формате мультимедийных презентаций и включен в состав учебно-методического комплекса дисциплины, доступного студентам.

Более детальное содержание лекционного материала представлено в учебных пособиях [1, 2], структура которых соответствует тематическому плану изучения дисциплины. Учебные пособия содержат перечни контрольных вопросов, ответы на которые должны быть получены студентами в процессе самостоятельной проработки материала соответствующей лекции.

5.2 Лабораторный (проектный) практикум

Основная цель проектного практикума – ознакомление с технологией объектно-ориентированного проектирования программных систем и получение практического опыта использования поддерживающих язык UML CASE-средств в процессе выполнения учебных программных проектов.

Программой изучения дисциплины предусмотрено выполнение пяти лабораторных работ, первая из которых соответствует подготовительному (предпроектному) этапу, а остальные четыре работы объединены темой учебного программного проекта, в рамках которого последовательно разрабатываются соответствующие UML-диаграммы. Каждая лабораторная работа реализует одну из стадий программного проекта и требует освоения и применения соответствующих CASE-средств.

Учебные проекты выполняются студентами по индивидуальным заданиям (в соответствии с темой контрольной работы) во время, отведенное для самостоятельной подготовки. На аудиторных лабораторных занятиях проводится текущий контроль и защита отчетов о промежуточных результатах выполнения проектов.

Защита проектов проводится в форме собеседования по материалу представленного отчета и сделанного публичного доклада. В процессе защиты оценивается полнота и качество выполнения практических заданий каждым из участников команды проекта, грамотность использования инструментальных средств, качество оформления отчета.

Методические указания по выполнению лабораторных работ и требования к содержанию и оформлению отчетов приведены в соответствующем разделе учебного пособия [2].

5.3 Самостоятельная работа

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Самостоятельная работа студентов по освоению дисциплины включает подготовку к промежуточной аттестации (зачету), выполнение контрольной работы, подготовку к проведению рубежного контроля (для студентов очной формы обучения), проработку лекционного материала, выполнение и оформление результатов лабораторных работ. Рекомендуемое распределение трудоемкости самостоятельной работы приведено таблице 5.1.

Таблица 5.1 – Рекомендуемая трудоемкость самостоятельной работы

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов	
	Очная форма	Заочная форма
Изучение материала лекционного курса:	**12**	**40**
Предмет и базовые понятия программной инженерии	2	4
Процессы и модели жизненного цикла ПО	4	16

Язык UML	6	20
Подготовка и выполнение лабораторных работ	**8**	**24**
Подготовка к рубежному контролю (по 2 часа на контроль)	**4**	**0**
Выполнение контрольной работы	**18**	**18**
Подготовка к зачету	**18**	**18**
Всего:	**60**	**100**

6 КОНТРОЛЬ И АТТЕСТАЦИЯ

6.1 Состав и формы проведения контрольно-аттестационных мероприятий

Программой изучения дисциплины предусмотрены мероприятия текущего и рубежного контроля и промежуточная аттестация (в форме зачета).

График и формы проведения контрольных и аттестационных мероприятий приведены в таблице 6.1.

Таблица 6.1 - График проведения контрольных и аттестационных мероприятий

Виды	Содержание	Форма проведения	Неделя
Текущий контроль	Контроль посещения аудиторных занятий	-	1 – 16
	Контроль выполнения лабораторных работ	Собеседование	4 – 16
Рубежный контроль	№1. Базовые понятия программной инженерии. Процессы и модели ЖЦ ПО.	Тестирование	6
	№2. Защита контрольной работы	Публичная защита результатов выполнения учебного проекта	15
Промежуточная аттестация	Зачет по дисциплине	Тестирование	17

6.2 Процедура оценивания результатов освоения дисциплины

6.2.1 Рубежный контроль

Рубежный контроль №1 проводится в форме фронтального тестирования по теоретической части дисциплины (тематические разделы №1 и №2). Тест содержит 20 вопросов, расчетное время проведения тестирования – 45 минут. Оценивается количество правильных ответов на задания теста и соответственно начисляется балл (см. таблицу 6.2). Студент, ответивший правильно менее, чем на 10 заданий теста, считается не прошедшим тестирование и обязан повторно пройти этот тест во время консультации по дисциплине.

Рубежный контроль №2 проводится в форме публичной защиты результатов выполнения индивидуальных контрольных заданий.

6.2.2 Промежуточная аттестация

Зачет по дисциплине проводится в форме тестирования по всем разделам дисциплины. Тест содержит 30 вопросов, расчетное время проведения тестирования – 45 минут. Оценивается количество правильных ответов и соответственно вычисляется балльная оценка (по 1 баллу за каждый правильный ответ).

6.3 Система балльно-рейтинговой оценки работы студентов

Оценивание результатов выполнения студентами очной формы обучения плановых контрольных и аттестационных мероприятий по дисциплине производится в соответствии с Положением о балльно-рейтинговой системе контроля о оценки академической активности студентов ФГБОУ ВО «Курганский государственный университет».

6.3.1 Критерии оценивания

Оценивание производится по 100-балльной шкале с последующим приведением итоговой 100-балльной рейтинговой оценки к традиционной четырехбалльной.

Рейтинговая оценка студента по дисциплине получается путем суммирования баллов, полученных студентом в течение семестра (максимум 70 баллов) и баллов, полученных им на промежуточной аттестации (максимум 30 баллов).

Максимальные балльные оценки по результатам проведения контрольных и аттестационных мероприятий (для студентов очной формы обучения) приведены в таблице 6.2. Минимальное количество баллов, которыми может быть оценен удовлетворительный ответ студента на зачете, равно 11. Неудовлетворительный ответ оценивается в 0 баллов.

Таблица 6.2 – Рейтинговые балльные оценки по дисциплине

Виды контроля/аттестации по дисциплине	Содержание	Максимальная оценка	
		За одну аттестацию	Всего
Текущий контроль	Контроль посещения аудиторных занятий	1	24
	Контроль выполнения лабораторных работ	4	20
Рубежный контроль	№1. Базовые понятия программной инженерии. Процессы и модели ЖЦ ПО.	12	12
	№2. Защита контрольной работы (индивидуального учебного программного проекта)	14	14
Промежуточная аттестация (зачет)		30	30
Максимальная итоговая оценка, баллов			100

Пересчет 100-балльной рейтинговой оценки студента по дисциплине в традиционную (4-балльную) оценку и в оценку ECTS (Общеевропейская система учета учебной работы) производится в соответствии с таблицей 6.3.

Таблица 6.3 – Соответствие шкал оценивания

Рейтинговая оценка, *баллов*	Виды оценок промежуточной аттестации		
	Традиционная оценка		Оценка ECTS
91-100	5	Зачтено	A
84-90	4		B
74-83			C
68-73	3		D
61-67			E
51-60	2	Не зачтено	Fx
0-50			F

6.3.2 Критерии допуска к промежуточной аттестации

Для допуска к зачету студент должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и при этом он должен выполнить и защитить все лабораторные работы и контрольную работу.

В случае если по результатам текущего и рубежного контроля студентом набрано менее 50 баллов, он может набрать недостающее количество баллов, выполнив дополнительные индивидуальные задания до конца зачетной недели семестра.

Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, также проводится путем выполнения дополнительных индивидуальных заданий.

Состав дополнительных заданий, их количество, формы выполнения и максимальные балльные оценки определяются преподавателем и доводятся до студента в момент выдачи заданий.

Для получения оценки «зачтено» автоматически (без сдачи зачета) студенту достаточно набрать 61 балл по результатам текущего и рубежного контроля в течение семестра.

Студенту преподаватель вправе добавить до 30 дополнительных (бонусных) баллов за активность на учебных занятиях, оригинальность принимаемых решений при выполнении лабораторных работ и индивидуальных контрольных заданий.

6.4 Фонд оценочных средств

6.4.1 Перечень оценочных средств

Фонд оценочных средств по дисциплине включает следующие компоненты, включенные в состав учебно-методического комплекса дисциплины:

1. Задания для пробного и контрольного тестирования по разделам дисциплины «Базовые понятия программной инженерии» и «Процессы и модели ЖЦ ПО» (рубежный контроль №1).
2. Задания для пробного и контрольного тестирования при проведении зачета по дисциплине.
3. Вопросы для подготовки к зачету по дисциплине.
4. Задания для выполнения контрольной работы.
5. Образцы отчетов по лабораторным работам.
6. Образцы отчетов по контрольным заданиям.

Ниже приведены примерные варианты контрольных заданий, дающие представление об их направленности и уровне сложности.

6.4.2 Примерные варианты компонентов фонда оценочных средств

6.4.2.1 Примеры вопросов для проведения рубежного контроля №1

1. Определите понятия "*программный продукт*" и "*программное обеспечение*". Чем отличаются "*коробочные*" программные продукты от "*заказных*"?
2. Что понимается под "*жизненным циклом ПО*"?

3. Что общего и в чем отличия между программной инженерией и другими инженерными отраслями?
4. Определите основные черты технологии *модульного программирования*.
5. Определите основные черты технологии *структурного проектирования* и *структурного программирования*.
6. Определите основные черты технологии *объектно-ориентированного* анализа, проектирования и программирования.
7. Используя соответствующие профессиональные стандарты, определите требования к профессиональной компетентности и должностные обязанности специалистов (по квалификационным уровням), работающих по профессии:
   - *специалист по информационным ресурсам;*
   - *специалист по информационным системам;*
   - *программист.*
8. Дайте краткую характеристику *ролевой модели* команды программного проекта. Перечислите основные функции *разработчика*, *тестировщика* и *инженера по качеству*.
9. Определите русскоязычные эквиваленты английских терминов: software product, firmware, system, non-deliverable item, off-the-shelf product, software service.
10. Определите понятие "*жизненный цикл промышленного изделия*", перечислите его основные фазы (этапы).
11. Сравните понятия "*жизненный цикл ПО*" и "*модель жизненного цикла ПО* ".

6.4.2.2 Примеры вопросов для проведения рубежного контроля №2

1. Что специфицирует *отношение обобщения* между «акторами» на UML-диаграмме вариантов использования?
2. Что специфицирует *отношение включения* между вариантами использования на UseCase-диаграмме вариантов использования?
3. Что специфицирует *отношение обобщения* между классами на UML-диаграмме классов?
4. Определите понятия «событие» и «переход», используемые при разработке State Machine-диаграмм.
5. Определите понятия «состояние», «составное состояние», «параллельное состояние» и «последовательное состояние», используемые при разработке State Machine-диаграмм.
6. Определите понятия «историческое состояние», «глубокое историческое состояние», используемые при разработке State Machine-диаграмм.

6.4.2.3 Примеры вопросов для проведения зачета по дисциплине

1. Перечислите стадии разработки ПО и приведите основное содержание каждой из них.
2. Прокомментируйте понятия "*стадия разработки*", "*жизненный цикл ПО*" и "*модель жизненного цикла ПО*". Перечислите известные Вам модели ЖЦ ПО.
3. Опишите основные черты и области эффективного применения:
   - каскадной модели ЖЦ ПО;
   - эволюционной модели;
   - модели пошаговой разработки;
   - спиральной модели.
4. Для чего используют модели сложных систем при их проектировании? Перечислите основные задачи, решаемые проектировщиками систем с помощью моделирования.
5. Какие цели преследует визуальное моделирование систем?
6. Расшифруйте сокращенные названия диаграмм *SADT, ERD, DFD* и переведите на русский язык. Для чего используются диаграммы перечисленных типов?
7. Какие задачи позволяет решать методология *SADT*? Для чего используются элементы модели *activity* и *arrow*? Что такое ICOM? Опишите *SADT*-диаграммами процесс выдачи книг читателю абонемента публичной библиотеки (на двух уровнях декомпозиции).
8. Перечислите компоненты ER-модели, дайте определения всем компонентам. Разработайте ER-модель данных для учета книжного фонда публичной библиотеки (студенческого абонемента университетской библиотеки, читального зала научных работников).
9. Дайте общую характеристику языка UML. Какова область применения этого языка? Чем принципиально отличается язык UML от языка программирования высокого уровня?
10. Определите понятие "*пакет*" языка UML. Для чего могут быть использованы *пакеты* и как они изображаются на UML-диаграммах?
11. Перечислите основные типы UML-диаграмм.
12. Определите понятие "*интерфейс*" как компонент *UseCase*-диаграммы.
13. Определите понятие "*связь* (*отношение*)" как компонент *UseCase*-диаграммы.
14. Понятие "Класс" в языке UML, графическое представление этого элемента на UML-диаграммах классов.
15. Понятие и формат спецификаций *атрибута и метода класса*.

16. Отношения *ассоциации* и *обобщения* между классами: понятие, характеристики, обозначение на UML-диаграммах
17. UML-диаграмма состояний: назначение и использование, основные компоненты и их графические обозначения.

7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Основная литература

1. Брауде Э. Технология разработки программного обеспечения. пер. с англ. – Санкт-Петербург: Питер, 2004.
2. Волк В.К. Введение в программную инженерию : учебное пособие. – Курган : Изд-во Курганского гос. ун-та, 2018 – 141 с.
3. Волк В.К. Практическое введение в программную инженерию: учебное пособие. – СПб.: Изд-во «Лань», 2019. – 100 с.: ил. – (Учебники для вузов. Специальная литература).
4. Леоненков А. Самоучитель UML − 2-е издание.: − Санкт-Петербург: БХВ-ПетербургСоммервилл И. Инженерия программного обеспечения − 6-е издание : пер. с англ. – Москва : Издательский дом «Вильямс», 2002. – 624 с. : ил.

7.2 Дополнительная литература

4. Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 3-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2010 – 720 с.
5. Липаев В.В. Программная инженерия. Методологические основы : учебник − Москва : ТЕИС, 2006. − 608 с.

7.4 Информационно-справочные материалы

6. UML 2.5 Diagrams Overview. URL: https://www.uml-diagrams.org/.
7. Рекомендации по преподаванию программной инженерии и информатики в университетах (Software engineering 2004: curriculum guidelines for undergraduate degree programs in software engineering). /пер. с англ.URL: www.intuit.ru. ИНТУИТ.РУ − Москва: «Интернет-университет информационных технологий», 2007.

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Техническое обеспечение

№	Наименование	Использование
1	Комплект: ноутбук, медиа-проектор, экран	Для демонстрации иллюстративного материала при чтении лекций.
2	Персональный компьютер стандартной комплектации	Используется в качестве инструмента и объекта исследования при выполнении лабораторных и контрольных работ.

8.2 Программное обеспечение

№	Наименование	Использование
1	StarUML™. The Open Source UML/MDA Platform.	Используются в качестве Case-средства поддержки программных проектов при выполнении лабораторных и контрольных работ.

9. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1. ЭБС «Лань»
2. ЭБС «Консультант студента»
3. ЭБС «Znanium.com»
4. «Гарант» - справочно-правовая система

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально- техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе.

11. Для студентов, обучающихся с использованием дистанционных образовательных технологий

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2 либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до сведения обучающихся.

Аннотация
рабочей программы учебной дисциплины
Основы программной инженерии
Образовательных программ высшего образования:
Программы бакалавриата (очная и заочная формы обучения):
09.03.03 Прикладная информатика (*Интеллектуальные информационные системы и технологии*)
09.03.04 Программная инженерия (*Программное обеспечение автоматизированных систем*)

Трудоемкость освоения дисциплины – 3 зач. ед. (108 акад. часов)

Семестры: 4-й (для очной формы обучения)

5-й (для заочной формы обучения)

Содержание дисциплины

Дисциплина «Основы программной инженерии» включена в вариативную часть учебных планов образовательных программ. Дисциплина базируется на курсах «Основы программирования» и «Объектно-ориентированное программирование» и создает методологическую основу для изучения дисциплин технологического блока: «Базы данных», «Разработка и анализ требований», "Технологии проектирования программных/информационных систем», «Управление качеством и тестирование ПО», «Управление программными проектами».

Основная цель изучения дисциплины – введение в промышленные технологии разработки программного обеспечения.

Задачи дисциплины:
изучение:

- основных понятий, методологических основ и стандартов программной инженерии;
- структуры процессов жизненного цикла программного продукта;
- основных моделей жизненного цикла программного продукта.

практическое освоение:

- основ языка визуального моделирования (UML), используемого при анализе и проектировании ПО;
- CASE-средств поддержки программных проектов.