

год начала подготовки 2020

Документ подписан квалифицированной электронной подписью

Сертификат: 023E519200DAAC0FAC74E9329E4F1A669EE

Владелец: "АНО ВО «РОССИЙСКИЙ НОВЫЙ УНИВЕРСИТЕТ»"; АН

Действителен до: 2020-02-28 12:00:00

АНО ВО «Российский новый университет»

**Елецкий филиал Автономной некоммерческой организации высшего образования «Российский новый университет»
(Елецкий филиал АНО ВО «Российский новый университет»)**

кафедра Прикладной экономики

Рабочая программа учебной дисциплины (модуля)

Программная инженерия

(наименование учебной дисциплины (модуля))

09.03.03 Прикладная информатика

(код и направление подготовки/специальности)

Прикладная информатика в экономике

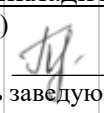
Направленность (профиль)

Рабочая программа учебной дисциплины (модуля) рассмотрена и утверждена на заседании кафедры «04» февраля 2020, протокол № 4.

Заведующий кафедрой Прикладной экономики

(название кафедры)

к.э.н. Преснякова Д.В.

(ученая степень, ученое звание, фамилия и инициалы, подпись  заведующего кафедрой)

Елец
2020 год

1. НАИМЕНОВАНИЕ И ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Учебная дисциплина «Программная инженерия» изучается обучающимися, осваивающими образовательную программу «Прикладная информатика» по профилю Прикладная информатика в экономике в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению 09.03.03 Прикладная информатика (уровень бакалавриата), утвержденным приказом Министерства образования и науки РФ от 19.09.2017 N 922 (ФГОС ВО 3+++).

Основная цель изучения учебной дисциплины состоит в том, чтобы дать обучающимся теоретические знания в области использования современных методик, применяемых при индустриальной разработке программного обеспечения (ПО). В результате прохождения курса студенты приобретут представление о жизненном цикле ПО, технологиях и инструментальных средствах, применяемых на каждом этапе разработки ПО, освоят базовые понятия и принципы проектного менеджмента. Студенты приобретут практические навыки разработки проектной документации, изучат особенности документирования процесса разработки ПО, характерные для отечественной и международной практики.

Изучение учебной дисциплины направлено на подготовку обучающихся к осуществлению деятельности по концептуальному, функциональному и логическому проектированию систем среднего и крупного масштаба и сложности, планированию разработки или восстановления требований к системе, анализу проблемной ситуации заинтересованных лиц, разработке бизнес-требований заинтересованных лиц, постановки целей создания системы, разработки концепции системы и технического задания на систему, организации оценки соответствия требованиям существующих систем и их аналогов, представлению концепции, технического задания на систему и изменений в них заинтересованным лицам, организации согласования требований к системе, разработке шаблонов документов требований, постановке задачи на разработку требований к подсистемам и контроль их качества, сопровождению приемочных испытаний и ввода в эксплуатацию системы, обработке запросов на изменение требований к системе, определенных профессиональным стандартом «Системный аналитик», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 28.10.2014 N 809н (Регистрационный номер №34882).

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Учебная дисциплина Программная инженерия относится к обязательной части учебного плана и изучается на 3, 4 курсе.

2.1. Требования к предварительной подготовке обучающегося:

Изучению данной учебной дисциплины предшествует освоение следующих учебных дисциплин: Информационные системы и технологии, Вычислительные системы, сети и телекоммуникации, Теория систем и системный анализ, Операционные системы.

Параллельно с учебной дисциплиной «Программная инженерия» изучаются дисциплины: Проектирование информационных систем, Базы данных, Исследование операций и методы оптимизации, Математическое и имитационное моделирование, Теория алгоритмов.

Параллельно с данной учебной дисциплиной проходит Производственная практика: технологическая (проектно-технологическая) практика.

2.2. Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Результаты освоения дисциплины «Программная инженерия» являются базой для прохождения обучающимися практики: Производственная практика: преддипломная практика, а также для изучения учебных дисциплин: Разработка программных

приложений, Корпоративные информационные системы, Теория экономических информационных систем, Электронный документооборот, Системы информационной безопасности, Внедрение информационных систем, Реинжиниринг процессов, а также прохождения практики: Производственная практика: преддипломная практика. Развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств обеспечивается чтением лекций, проведением практических занятий, содержание которых разработано на основе результатов научных исследований, проводимых Университетом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения дисциплины обучающийся по программе бакалавриата должен овладеть:

- Способен разрабатывать алгоритмы и программы, пригодные для практического применения; (ОПК-7)

Формируемая компетенция	Планируемые результаты обучения	Соотнесение показателей обучения дисциплины с индикаторами достижения компетенций	
		Код показателя результатов обучения	Код показателя результатов обучения
способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-7)	Знать:		
	– методы извлечения экспертных знаний из специалистов в произвольных предметных областях.	ОПК-7-31	И-ОПК-7.1
	– распространенные виды ошибок при формулировании требований к проектируемой системе.	ОПК-7-32	И-ОПК-7.1
	– способы оценки продолжительности работ при разработке ПО.	ОПК-7-33	И-ОПК-7.1
	– распространенные типы ошибок, возникающих при анализе предметной области, и способы их избежать.	ОПК-7-34	И-ОПК-7.1
	– модели представления и обработки знаний, системы принятия решений.	ОПК-7-35	И-ОПК-7.1
	– методы оптимизации и принятия проектных решений.	ОПК-7-36	И-ОПК-7.1
	Уметь:		
	– проводить формализацию и оформление собранных знаний о предметной области на естественном языке и с помощью моделей в различных нотациях.	ОПК-7-У1	И-ОПК-7.2
	– осуществлять поиск информации в сети Интернет в целях формирования справочного раздела документации.	ОПК-7-У2	И-ОПК-7.2
	– моделировать анализируемые системы на языке UML	ОПК-7-У3	И-ОПК-7.2
	– проводить анализ предметной области, выделять структурные элементы и их взаимосвязи.	ОПК-7-У4	И-ОПК-7.2
	– проводить трассировку требований.	ОПК-7-У5	И-ОПК-7.2
	– разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7-У6	И-ОПК-7.2
	Владеть:		
	– методами сбора и формализации данных о предметной области.	ОПК-7-В1	И-ОПК-7.3
– навыками работы с инструментальными средствами моделирования требований.	ОПК-7-В2	И-ОПК-7.3	

	– навыками работы с инструментальными средствами моделирования архитектуры.	ОПК-7-В3	И-ОПК-7.3
	– навыками работы с инструментальными средствами управления проектами.	ОПК-7-В4	И-ОПК-7.3
	– навыками работы с инструментальными средствами моделирования требований.	ОПК-7-В5	И-ОПК-7.3
	– методами сбора и формализации данных о предметной области.	ОПК-7-В6	И-ОПК-7.3

4. ОБЪЕМ ДИСЦИПЛИНЫ В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ, ВЫДЕЛЕННЫХ НА КОНТАКТНУЮ РАБОТУ ОБУЧАЮЩИХСЯ С ПРЕПОДАВАТЕЛЕМ (ПО ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ) И НА САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ

4.1.Общий объем учебной дисциплины(модуля)

№	Форма обучения	Семестр	Общая трудоемкость		В том числе контактная работа с преподавателем							СР	Контроль	
			В з.е.	В часах	Всего	Л	Пр	КоР	КР П	Зачет	Конс			Экзамен
1	Заочная	2 сессия, 3 курс	1	36	4	4							32	
		1 сессия, 4 курс	2	72	10	4	4	1,7		0,3			58,3	3,7
		2 сессия 4 курс	3	108	15	4	4	1,6	3		2	0,4	86,4	6,6
Итого:			6	216	29	12	8	3,3	3	0,3	2	0,4	176,7	10,3

Дисциплина предполагает изучение 6 тем. Общая трудоемкость дисциплины составляет 6 зачетных единиц (216 часов).

4.2. Распределение учебного времени по темам и видам учебных занятий

б) заочная форма обучения

№	Наименование разделов, тем учебных занятий	Всего часов	Контактная работа с преподавателем							СР	Контроль	Формируемые результаты обучения	
			Всего	Л	ПР	Кор	КПР	Зачет	Конс				Эк
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Основные понятия программной инженерии. Жизненный цикл ПО	25	2	2							23		ОПК-7-31 ОПК-7-32 ОПК-7-33
2	Составление требований к разрабатываемому ПО	41	6	2	4						35		ОПК-7-35 ОПК-7-У4 ОПК-7-У5
3	Анализ и проектирование ПО	36,3	4	4							32,3		ОПК-7-36 ОПК-7-37
4	Промежуточная аттестация (зачет)	5,7	2			1,7		0,3				3,7	
5	Итого 1 сессия	108	14	8	4	1,7	0	0,3	0	0	90,3	3,7	
6	Кодирование (разработка ПО)	17,4									17,4		ОПК-7-31, ОПК-7-32, ОПК-7-У4, ОПК-7-У5,
7	Отладка и тестирование ПО	22	4	2	2						18		ОПК-7-33, ОПК-7-34,

													ОПК-7-У5, ОПК-7-В5, ОПК-7-В6
8	Развертывание и сопровождение ПО	22	4	2	2						18		ОПК-7-35, ОПК-7-36, ОПК-7-У4, ОПК-7-У5, ОПК-7-У6, ОПК-7-В1, ОПК-7-В6
9	Курсовая работа	36	3				3				33		
10	Экзамен	10,6	4			1,6		2	0,4			6,6	
11	Итого 2 сессия	108	15	4	4	1,6	3	2	0,4		86,4	6,6	
12	ИТОГО	216	29	12	8	3,3	3	0,3	2	0,4	176,7	10,3	

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ)

Тема 1. Основные понятия программной инженерии. Жизненный цикл ПО.

Основные определения программной инженерии. Связь программной инженерии с другими науками. История возникновения термина. Базовые стандарты своды знаний программной инженерии. Набор моделей (методологий) совершенствования процессов Capability Maturity Model Integration (CMMI). CMMI for Development (CMMI-DEV), CMMI for Services (CMMI-SVC) и CMMI for Acquisition (CMMI-ACQ).

Жизненный цикл ПО, типы моделей. Стандарты, определяющие понятие и содержание информационных продуктов жизненного цикла. Case-средства моделирования и автоматизации процессов жизненного цикла программного обеспечения. Методологии управления программными проектами: связь с моделями жизненного цикла, критерии выбора, области применения. Тяжеловесные и гибкие методологии управления программными проектами, особенности и отличия. Рассмотрение особенностей организации разработки на примере Scrum методологии.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Тема 2. Составление требований к разрабатываемому ПО.

Роль и место требований в общем процессе разработки ПО. Применимые стандарты.

Техническое задание и Software Requirement Specification: схожесть и различия.

Требования: классификация, представление, требования к требованиям. Виды требований: С- и D-требования.

Практические приемы при составлении требований. Извлечение информации из специалистов. Распространенные ошибки при составлении требований: пропуск, искажение, обобщение; копирование существующей бизнес-логики, навязывание новшеств.

Описание требований на языке UML: иерархия требований, варианты использования (Use Case) и ссылки на требования, диаграмма последовательности действий, диаграмма состояний.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Тема 3. Анализ и проектирование ПО.

Проектирование ПО: общие понятия, методы и технологии проектирования. Подходы к проектированию ПО. Применимые международные и отечественные стандарты. Эскизный проект и Architectural Design Document: схожесть и различия.

Понятие архитектуры ПО, примеры типовых архитектур. Примеры описание архитектур с помощью стандарта IEEE 1471, модели «4+1», фреймворка Захмана, ERM.

Форматы описания архитектуры ПО: блок-схемы, НПО-диаграмма (Hierarchy and Input-Process-Output), ERM-диаграмма (Entity Relation Model), диаграмма потоков данных (Data Flow Diagram), псевдоязыки программирования.

Объектный подход к проектированию ПО (ООП). Основные концепции объектной модели: абстрагирование, модульность, инкапсуляция. Дополнительные концепции объектной модели: типизация, параллелизм, сохраняемость.

Роль и место архитектуры в проекте ПО. Декомпозиция. Обобщенный алгоритм построения архитектуры. Образцы проектирования. Выбор типовой архитектуры.

Архитектурное проектирование и компоновка системы с помощью UML: диаграммы пакетов и диаграммы компонентов. Детальное проектирование и подготовка к кодированию системы с помощью диаграммы классов: особенности, принципы упаковки классов в архитектурные подсистемы. Основные принципы SOLID (SRP, OCP, LSP, ISP, DIP).

Особенности проектирования пользовательского интерфейса. Метрики и оценки качества проектирования: сущностная эффективность, согласованность задач, наблюдаемость задач, единообразие компоновки, визуальная связность.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Тема 4. Кодирование (разработка ПО).

Кодирование (разработка) ПО. Планирование итераций, распределение задач.

Контроль качества: стандарты программирования, метрики для исходного кода.

Организация и инструментальные средства коллективной разработки.

Управление рисками и сроками. Основные виды и источники рисков в программных проектах, методы анализа и управления рисками. Методы календарного планирования и управления отклонениями.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Тема 5. Отладка и тестирование ПО.

Отладка и тестирование ПО: основные понятия. Принципы тестирования: функциональное/модульное тестирование, регрессивное тестирование, нагрузочное тестирование. Приемы и технологии тестирования.

Планирование тестирования, взаимодействие тестировщиков и разработчиков.

Тестирование документации к ПП.

Интеграция, верификация и валидация системы. Инструментальные средства поддержки тестирования.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Тема 6. Развертывание и сопровождение ПО.

Развертывание ПО. Развертывание программной системы на аппаратных средствах: артефакты, узлы, моделирование с помощью диаграммы развертывания UML. Особенности описания и организации процессов с помощью стандартов IEEE, SWEBOK, методологии TOGAF, набора практик DevOps. Документирование процессов.

Сопровождение ПО. Длительность, виды обеспечения, способы достижения технологической и эксплуатационной гибкости.

Вывод из эксплуатации: «подводные камни», необходимость планирования и выделения ресурсов.

Литература:

а) основная: 1-2.

б) дополнительная: 3-5.

Планы семинарских, практических, лабораторных занятий

Тема 2. Практическое занятие: Разработка ТЗ

Продолжительность занятия - 2 часа

Основные вопросы:

1. Получение первичных знаний по предметной области. Анализ предметной области и выделение мест для внедрения ИС
2. Источники и способы получения информации по предметной области
3. Способы формализации и структуризации данных по предметной области

Активность студентов:

Выбрав уникальную (не повторяющуюся в группе) тему из предложенных или согласовав с преподавателем тему по выбору учащегося, студент разрабатывает требования к перспективному программному средству (ПС).

Разработанные требования описываются в формализованном документе (Техническом задании или Software Requirement Document), соответствующем требованиям российских и/или международных стандартов. Результат выполнения Практического задания используется обучаемым при выполнении последующих заданий.

Тема 3. Практическое занятие. Разработка эскизного проекта

Продолжительность занятия - 2 часа

Основные вопросы:

1. Существующие архитектуры ПС и их применимость в конкретном случае
2. Способы описания архитектуры ПС
3. Влияние требований на архитектуру ПС

Активность студентов:
На основании сформулированных требований студент разрабатывает несколько (два или более) альтернативных архитектур ПС и обосновывает выбор наиболее подходящего. Далее часть проектируемой системы подвергается более детальному проектированию (до уровня интерфейса одного или нескольких классов на произвольном языке программирования). Составляется таблица соответствия требований элементам выбранной архитектуры ПС (трассировка требований).

Разработанные требования описываются в формализованном документе (Эскизном проекте или Architecture Design Document), соответствующем требованиям российских и/или международных стандартов. Результат выполнения Практического задания 2 используется обучаемым при выполнении последующих заданий.

Примечание: непосредственная разработка программных компонентов в рамках курса не предполагается.

Тема 4. Практическое занятие. Разработка диаграммы Ганта

Продолжительность занятия - 2 часа

Основные вопросы:

1. Применимые в конкретном случае подходы к выполнению разбиения работ
2. Методы оценки длительности работ
3. Определение потребных для выполнения работ ресурсов

Активность студентов:
На основании созданного эскизного проекта или ADD студент формирует план реализации проекта:

- разбивает общий объем работ на отдельные работы;
- оценивает длительность и взаимосвязь работ;
- определяет и назначает конкретным работам необходимые ресурсы. Результатом выполнения практического задания является Диаграмма Ганта.

Тема 5. Практическое занятие. Разработка Программы и методик приемо-сдаточных испытаний

Продолжительность занятия - 2 часа

Основные вопросы:

1. Способы проверки требований. Особенности демонстрации выполнения требований заказчику

2. Необходимость создания дополнительных программных и/или аппаратных средств для проведения тестирования

3. Особенности тестирования распределенных систем Активность студентов:

На основании сформулированных требований и проекта ПС студент разрабатывает процедуру тестирования одного или нескольких компонентов ПС. Составляется таблица соответствия требований пунктам методики тестирования ПС (трассировка требований).

Разработанные требования описываются в формализованном документе (Программа и методики испытаний), соответствующем требованиям российских стандартов.

Примечание: реальное проведение тестирования ПО в рамках курса не предполагается.

Тема 6. Практическое занятие. Создание эксплуатационной документации

Продолжительность занятия - 2 часа

Основные вопросы:

1. Методы оценки рисков

2. Планирование работ, связанных с поддержанием эксплуатации систем

3. Проблемы обучения персонала, эксплуатирующего систему Активность студентов:

На основании проекта ПС студент разрабатывает один из следующих документов:

– инструкцию системного программиста в части установки одного из компонентов разработанного ПС на новое рабочее место (хост, техническое средство);

– инструкцию оператора для одного из рабочих мест разработанного ПС;

– инструкцию оператора службы технической поддержки в части списка возможных неисправностей, способов идентификации их причин и действий по устранению.

Разработанные документы описываются в формализованном документе, соответствующем требованиям российских стандартов.

Примечание: конкретный тип документа и его содержание выбираются в зависимости от темы работы студента и согласовываются с преподавателем.

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПОДИСЦИПЛИНЕ

6.1. Задания для приобретения, закрепления и углубления знаний, углубления и закрепления ранее приобретенных знаний

№	Задание	Код результата обучения
1.	В чем заключается метод извлечения знаний из экспертов, не требующий специальных знаний и навыков от когнитолога?	ОПК-7-31
2.	В чем заключается метод интервьюирования при сборе требований?	ОПК-7-31
3.	В чем заключаются ошибки в следующей формулировке требования ТЗ: «Система должна работать в реальном масштабе времени и иметь интуитивно»	ОПК-7-32

	понятный пользовательский интерфейс»	
4.	Правильно ли сформулировано требование: «Система должна работать в реальном масштабе времени.»	ОПК-7-32
5.	Как оценить продолжительность работ в системе ?	ОПК-7-33
6.	Причины, по которым длительность некоторых из перечисленных работ может измениться	ОПК-7-33
7.	Сипы ошибок, которые могут возникнуть при анализе вашей предметной области	ОПК-7-34
8.	Способы избежания перечисленных выше ошибок	ОПК-7-34
9.	Способы обнаружения «узких мест» систем.	ОПК-7-35
10.	В чем заключается алгоритм описания системы, который позволит этот способ реализовать	ОПК-7-35
11.	Приведите пример системы, проектировать которую необходимо методом «сверху вниз». Покажите, что метод «снизу вверх» приведет к нерациональному расходу сил и средств. Вариант: Пример системы, которую эффективнее проектировать методом «снизу вверх».	ОПК-7-36
12.	Приведите пример системы, при проектировании которой удобнее использовать подход разбиения на уровневые подсистемы	ОПК-7-36

Основными видами внеаудиторной самостоятельной работы при изучении данного предмета являются: чтение основной и дополнительной литературы (в соответствии с перечнем основной и дополнительной литературы, необходимой для освоения дисциплины) по указанию преподавателя, а также с использованием Интернета; изучение конспектов лекций; учебно-исследовательская работа под руководством преподавателя с использованием компьютерной техники; повторная работа над учебным материалом, подготовка докладов и презентаций для выступления на семинарах, выполнение домашних заданий.

Основные категории учебной дисциплины для самостоятельного изучения:

Методы извлечения знаний по предметной области, в том числе из специалистов в предметной области:

– **пассивные коммуникативные методы** включают наблюдение, анализ протоколов

«мыслей вслух», процедуры извлечения знаний из лекций.

– **активные индивидуальные методы** включают методы анкетирования, интервьюирования, свободного диалога и игры с экспертом.

– **активные групповые методы** включают «мозговой штурм», дискуссии за круглым столом и ролевые игры.

Метод наблюдения – фиксирование всех действий эксперта, его реплик и объяснений.

Метод анализа протоколов «мыслей вслух» предполагает, что эксперт не только комментирует свои действия, но и объясняет цепочку своих рассуждений, приводящих к решению.

Метод «мозгового штурма» – один из наиболее известных и широко применяемых методов генерирования новых идей путем творческого сотрудничества группы специалистов. В процессе такого штурма участники выдвигают и развивают собственные идеи, стимулируя появление новых и комбинируя их. После выдвижения идей выполняются тщательное их обсуждение, оценка и отбор лучших.

В играх с экспертом инженер по знаниям берет на себя чью-нибудь роль в моделируемой ситуации. **Ролевые игры в группе** предусматривают участие в игре нескольких специалистов. Участники игры наделяются определенными ролями, а собственно игра проводится по составленному когнитологом сценарию.

IEEE (Institute of Electrical and Electronics Engineers) — организация, которая поддерживает набор стандартов для управления и выполнения проектов по конструированию ПО и систем.

Абстракция - способность отделить существенные черты предмета (объекта) от второстепенных, видеть идею, которая будет реализована.

Абстрактная архитектура - декомпозированная структура задач домена на подсистемы или иерархию подсистем, на каждом уровне которой фиксируются параметры и ограничения, определяющие выделенные подсистемы.

Агрегация - объединение ряда понятий в новое понятие (отношение типа "часть-целое"), общие признаки которого могут быть суммой признаков других компонентов или быть новым признаком.

Анализ требований - отображения функций системы и ее ограничений в модели предметной области.

Артефакт - любой продукт деятельности специалистов по разработке ПО.

Архитектура программной системы - структура системы в терминах подсистем (компонентов) и интерфейсов между ними, отображающая правила декомпозиции проблемы.

Ассоциация - наиболее общее и существенное отношение, которое устанавливает наличие связей между понятиями без уточнения их содержания и размеров.

Водопадная (каскадная) модель - схема работ, в которой каждая из работ выполняется один раз и в том порядке, который указан в модели жизненного цикла.

Диаграмма - графическое представление сценариев работы системы с помощью классов, состояний, событий и т.п.

Жизненный цикл системы - схема выполнения работ по проектированию системы, начиная с момента принятия решения о необходимости ее создания и заканчивая моментом ее полного изъятия из эксплуатации.

Инженерия требований - сбор, анализ, оформление условий и ограничений на разработку системы в виде спецификации, согласованной как заказчиком, так и исполнителем.

Нефункциональные требования - требования, которые характеризуют организационные, исполнительские, операционные аспекты работы программной системы в среде реализации.

Онтология - совокупность элементарных понятий, терминологии и парадигмы их интерпретации в среде проблемы, которую требуется разработать.

Пакет - программная структура с общим механизмом организации элементов (объектов, классов) в группы, начиная от системы (стереотип "система") и к ее подсистемам различного уровня детализации.

Переносимость системы - возможность изменять сервис системы (ОС, связи, сетевые коммуникации, данные СУБД и т.п.) путем настройки модулей на новые условия среды или платформы.

Программная инженерия - система методов, средств и дисциплины планирования, разработки, эксплуатации и сопровождения программного обеспечения, способного к массовому воспроизводству.

Процесс приобретения - действия, которые инициируют определенный цикл анализа для определения покупателем программной системы или сервиса.

Процесс разработки - действия разработчика по инженерии требований, проектированию, кодированию и тестированию программного продукта.

Процесс сдачи - действия по передаче разработанного продукта покупателю.

Процесс эксплуатации - действия по обслуживанию системы пользователем.

Процесс сопровождения - действия по решению задач системы, поддержкой системы в актуальном состоянии для выполнения функций системы, управлению модификациями или изъятию системы из употребления.

Проектирование концептуальное - уточнение понимания и согласование деталей требований к системе.

Проектирование архитектурное - определение структурных особенностей

строящейся системы.

Проектирование техническое - отображение требований среды функционирования и разработки системы путем определения всех конструктивных элементов и их композиций.

Проектирование детальное - определение подробностей реализации функций для заданной среды и связей между соответствующими компонентами системы.

Реализация программной системы - преобразования проектных решений в работающую систему (синонимы: кодирование, конструирование).

Связь (Relationship) - поименованная ассоциация между двумя сущностями, имеющая значение для рассматриваемой предметной области.

Спецификация - описание алгоритма, правил, ограничений действий объектов с учетом стандартов, критериев качества и др.

Спиральная модель ЖЦ - модель процессов разработки системы, с возможностью возвращаться к любому предыдущему процессу с целью переработки элементов сделанного продукта.

Событие - явление, которое провоцирует смену определенного состояния и переход к другому состоянию в системе.

Структура системы - множество элементов и отношений между ними.

Субъект (актор) – кто-то или что-то вне системы, что взаимодействует с системой.

Сущность (Entity) - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

Сценарий - конкретная последовательность действий, которая иллюстрирует поведение и выполнение экземпляра прецедента.

Требование - соглашение или договор между заказчиком и исполнителем системы относительно свойств ее функций, условий работы в заданной среде.

Унаследованная система - существующая действующая система, созданная с помощью любых методов и технологий для поддержки некоторых процессов бизнеса.

Функциональные требования - это условия и ограничения на цели, функции системы и принципы их выполнения на компьютере.

Эксплуатация - действия по выполнению готовой программной системы.

6.2. Задания, направленные на формирование профессиональных умений

13.	По выбранной теме опишите структуру РИС в виде UML диаграммы компонентов.	ОПК-7-У1
14.	Выделите важные семантические понятия выбранной предметной области, опишите их в виде UML диаграммы классов	ОПК-7-У1
15.	Создайте глоссарий проекта с использованием материалов из сети Интернет	ОПК-7-У2
16.	Опишите роль и место проектируемой системы по материалам из сети Интернет	ОПК-7-У2
17.	Разработайте таблицу типов циркулирующих в РИС сообщений согласно описанной структуре и семантике работы РИС	ОПК-7-У3
18.	Разработайте UML диаграмму последовательности, показывающую обмен сообщениями между приложениями	ОПК-7-У3
19.	Разработайте UML диаграмму активности, показывающую обмен сообщениями между приложениями	ОПК-7-У4
20.	Опишите связи понятий выбранной предметной области в виде онтологии в формате OWL.	ОПК-7-У4
21.	Создайте начальный уровень разбиения на работы, основываясь на верхнем уровне архитектуры системы	ОПК-7-У5
22.	Разбейте одну из работ на более мелкие части. Покажите, как зависимости между работами верхнего уровня «отражаются» на зависимости нижнего уровня	ОПК-7-У5
23.	Для компонента нижнего уровня выберите подходящее программное решение из существующих. Покажите, какие доработки/настройки понадобятся, чтобы использовать готовый компонент в конкретной системе Вариант: Обоснуйте неэффективность создания «костылей»	ОПК-7-У6
24.	Покажите, какой простой алгоритм/программный компонент имеет смысл реализовать «по месту», не адаптируя существующие	ОПК-7-У6

Вариант: Обоснуйте неэффективность создания «велосипедов»	
---	--

6.3. Задания, направленные на формирование профессиональных навыков, владений

25.	Проведите анализ ресурсных возможностей проекта с помощью метрика Function Points.	ОПК-7-B1
26.	Опишите жизненный цикл программного проекта согласно методологии Scrum.	ОПК-7-B1
27.	С помощью диаграммы компонентов опишите и включите в модель создаваемой системы унаследованные приложения и данные.	ОПК-7-B2
28.	Опишите основные понятия предметной области в виде онтологии	ОПК-7-B2
29.	Выделите критический путь в созданной диаграмме	ОПК-7-B3
30.	Создайте график занятости ресурсов	ОПК-7-B3
31.	Загрузите произвольный приложение, доступное в исходных кодах в среду проектирования	ОПК-7-B4
32.	На основании загруженного кода постройте диаграммы классов	ОПК-7-B4
33.	Создайте описание проблемы в Mantis	ОПК-7-B5
34.	Обновите версию ПО в SVN	ОПК-7-B5
35.	Сгенерируйте «пустой» код на выбранном языке программирования на основании UML диаграммы классов	ОПК-7-B6
36.	Обновите диаграмму классов и проверьте, как изменения отображаются на код на ЯП	ОПК-7-B6

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПОДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Средства оценивания текущего контроля:

- письменные краткие опросы в ходе аудиторных занятий на знание категорий учебной дисциплины, указанных в п.6.1.1.;
- задания и упражнения, рекомендованные для самостоятельной работы;
- практические работы.
- задания и упражнения в ходе практических занятий.

7.2. ФОС для текущего контроля

№	Формируемая компетенция	Показатели результата обучения	ФОС текущего контроля
1.	способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-7)	ОПК-7-31	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 1,2 п.6.1.2 Вопросы к зачету 1-4
2.		ОПК-7-32	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 3,4 п.6.1.2 Вопросы к зачету 5-9
3.		ОПК-7-33	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 5,6 п.6.1.2 Вопросы к зачету 10-14
4.		ОПК-7-34	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 7,8 п.6.1.2 Вопросы к зачету 15-19
5.		ОПК-7-35	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 9,10 п.6.1.2 Вопросы к зачету 20-29

6.	ОПК-7-36	Устный и письменный опрос на знание категорий учебной дисциплины. Задания для самостоятельной работы 11,12 п.6.1.2 Вопросы к зачету 30-40
7.	ОПК-7-У1	Задания для самостоятельной работы 13,14 п.6.2 Вопросы к экзамену 1-9
8.	ОПК-7-У2	Задания для самостоятельной работы 15,16 п.6.2 Вопросы к зачету 10-19
9.	ОПК-7-У3	Задания для самостоятельной работы 17,18 п.6.2 Вопросы к экзамену 20-24
10.	ОПК-7-У4	Задания для самостоятельной работы 19,20 п.6.2 Вопросы к экзамену 25-29
11.	ОПК-7-У5	Задания для самостоятельной работы 21,22 п.6.2 Вопросы к экзамену 30-34
12.	ОПК-7-У6	Задания для самостоятельной работы 23,24 п.6.2 Вопросы к экзамену 35-39
13.	ОПК-7-В1	Задания для самостоятельной работы 25,26 п.6.3 Вопросы к экзамену 40-44
14.	ОПК-7-В2	Задания для самостоятельной работы 27,28 п.6.3 Вопросы к экзамену 45-49
15.	ОПК-7-В3	Задания для самостоятельной работы 29,30 п.6.3 Вопросы к экзамену 50-54
16.	ОПК-7-В4	Задания для самостоятельной работы 31,32 п.6.3 Вопросы к экзамену 55-59
17.	ОПК-7-В5	Задания для самостоятельной работы 34,34 п.6.3 Вопросы к экзамену 60-64
18.	ОПК-7-В6	Задания для самостоятельной работы 35,36 п.6.3 Вопросы к экзамену 65-73

7.3 ФОС для промежуточной аттестации

7.3.1. Задания для оценки знаний

№	Формируемая компетенция	Показатели результата обучения	ФОС для оценки знаний
1.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-7)	ОПК-7-31	Вопросы к зачету 1-4
2.		ОПК-7-32	Вопросы к зачету 5-9
3.		ОПК-7-33	Вопросы к зачету 10-14
4.		ОПК-7-34	Вопросы к зачету 15-19
5.		ОПК-7-35	Вопросы к зачету 20-29
6.		ОПК-7-36	Вопросы к зачету 30-40

Вопросы для подготовки к зачету

1. Основные определения: программный продукт, программный проект, программная инженерия.
2. Понятия процесса в программной инженерии. Официальная классификация процессов программной инженерии.
3. Российские и зарубежные стандарты (ГОСТ, ISO, IEEE). Термины стандартов. Особенности описания и классификации процессов, групп процессов.
4. Своды знаний программной инженерии и проектного управления (SWEBOOK, PMBOOK). Основные термины и определения, особенности описания и классификации процессов.
5. Жизненный цикл. Определение, структура. Концепция уровней ЖЦ Скотта Амблера.
6. Отличие ЖЦ программного обеспечения, информационной системы, проекта.
7. Жизненный цикл. Стандарты, определяющие понятие жизненного цикла.
8. Жизненный цикл. Типы моделей жизненного цикла, классификация, варианты модификации и зависимость от стратегии разработки (каскадная, инкрементная и итерационная, спиральная).

9. Жизненный цикл. Преимущества, недостатки, области применения моделей жизненного цикла.
10. Жизненный цикл. Модель Бозма и цикл качества Шухарта-Деминга PDCA.
11. Жизненный цикл. Средства автоматизации и моделирования процессов жизненного цикла программного обеспечения.
12. Capability Maturity Model Integration (CMMI) набор моделей (методологий) совершенствования процессов. CMMI for Development (CMMI-DEV), CMMI for Services (CMMI-SVC) и CMMI for Acquisition (CMMI-ACQ).
13. Анализ предметной области. Извлечение информации из специалистов предметной области.
14. Управление требованиями. Понятие требования к программному обеспечению (Software Requirements).
15. Управление требованиями. Виды требований по уровням. Приведите примеры.
16. Управление требованиями. Виды требований по характеру. Приведите примеры.
17. Характеристики детального требования: прослеживаемость, тестируемость, приоритетность, полнота, согласованность.
18. Управление требованиями. Виды требований Выражение требований в виде текстовых утверждений и графических моделей.
19. Особенности документирования проекта разработки (бизнес-правила, бизнес-требования, видение).
20. Управление требованиями. Анализ требований разработчика (D-требования, developer requirements), требований заказчика (C-требования, consumer requirements).
21. Управление требованиями. Разработка спецификации требований (SRS) и стандарт IEEE Std 830-1998.
22. Управление требованиями (фиксация, внесение изменений, стратегия трассировки).
23. Техническое задание (ТЗ). Назначение, стандарты, регламентирующие структуру и содержание.
24. Особенности моделирования ПО с помощью нотации UML на уровне формирования требований, анализа требований, архитектурном уровне, уровне детального проектирования, уровне физического размещения.
25. Моделирование требований. Разработка модели требований с помощью UML: особенности, элементы и примеры применения диаграмм прецедентов.
26. Моделирование требований. Разработка модели требований с помощью UML: особенности, элементы и примеры применения диаграмм коммуникации.
27. Моделирование требований. Разработка модели требований с помощью UML: особенности, элементы и примеры применения диаграмм конечных автоматов.
28. Особенности спецификации элементов Use Case и расширения функциональных возможностей в процессе изменения требований.
29. Особенности спецификации элементов Use Case: точки расширения (extension points) и их назначение при моделировании.
30. Абстрактные элементы диаграммы Use Case и их назначение при моделировании.
31. Оценки программных проектов. Ресурсная оценка программного проекта помощью метода UCP Густава Карнера.
32. Анализ и моделирование с помощью диаграмм деятельности.
33. Анализ и моделирование с помощью диаграмм коммуникации.
34. Анализ и моделирование с помощью диаграмм последовательности.
35. Анализ и моделирование с помощью диаграмм конечных автоматов. Моделирование событийно-управляемых объектов.
36. Анализ и моделирование с помощью диаграмм конечных автоматов. Моделирование композитных состояний объекта.
37. Особенности процессов анализа и синтеза программных систем. Особенности архитектурного этапа проектирования.

38. Понятие архитектуры системы, типовые архитектуры.
39. Стандарты, определяющие описание архитектуры (IEEE 1471, ISO 42010)
40. Примеры описание архитектур с помощью модели «4+1», фреймворка Захмана, ERM.

Вопросы для подготовки к экзамену

1. Структурирование системы. Декомпозиция подсистем на модули.
2. Структурирование системы. Модульность.
3. Структурирование системы. Информационная закрытость.
4. Структурирование системы. Связность модулей (функциональная, информационная, коммуникативная, процедурная, временная, логическая, связность по совпадению).
5. Структурирование системы. Сцепление модулей.
6. Оценки сложности и иерархической структуры программной системы.
7. Документирование архитектуры программного средства. Эскизный проект и Architectural Design Document (схожесть и различия).
8. Архитектурные паттерны и антипаттерны: классификация, области применения.
9. CRC-карты: назначение, применение.
10. Основные принципы ООП: абстрагирование.
11. Основные принципы ООП: инкапсуляция.
12. Основные принципы ООП: модульность.
13. Основные принципы ООП: типизация.
14. Основные принципы ООП: параллелизм.
15. Основные принципы ООП: сохраняемость.
16. Объекты: общее определение, виды отношений, связи, видимость, агрегация.
17. Классы: общее определение, виды отношений, ассоциации, наследование, агрегация, зависимость.
18. Архитектурное проектирование и компоновка системы с помощью нотации UML: диаграммы пакетов.
19. Архитектурное проектирование и компоновка системы с помощью нотации UML: диаграммы компонентов.
20. Детальное проектирование и подготовка к кодированию системы с помощью диаграммы классов: особенности, принципы упаковки классов в архитектурные подсистемы.
21. Архитектурные принципы проектирования SOLID (SRP, OCP, LSP, ISP, DIP).
22. Требования стандарта IEEE Std 1016-2009 к документированию процесса проектирования. Особенности документирования процесса проектирования на примере SDD.
23. Принципы построения и применения архитектурных паттернов. Паттерн Модель-Представление-Контроллер (MVC).
24. Принципы построения и применения архитектурных паттернов. Паттерн Цепочка обязанностей (Chain of responsibility).
25. Принципы построения и применения архитектурных паттернов. Паттерн Абстрактная фабрика (Abstract factory).
26. Особенности проектирования пользовательского интерфейса.
27. Метрики и оценки качества проектирования интерфейса: сущностная эффективность.
28. Метрики и оценки качества проектирования интерфейса: согласованность задач.
29. Метрики и оценки качества проектирования интерфейса: наблюдаемость задач.
30. Метрики и оценки качества проектирования интерфейса: единообразие компоновки.
31. Метрики и оценки качества проектирования интерфейса: визуальная связность.
32. Компонентная объектная модель: организация интерфейса COM, серверы COM-объектов, работа с COM-объектами.
33. Язык спецификаций для описания интерфейсов IDL. Информация о библиотеке типа в IDL-описании.
34. Процесс тестирования программных средств. Основные понятия и принципы

- тестирования ПО.
35. Способы тестирования базового пути (поточковый граф, цикломатическая сложность).
 36. Способы тестирования проверки логических условий (тестирование ветвей и операций отношений).
 37. Тестирование потоков данных. Особенности и назначение.
 38. Тестирование циклов. Особенности и назначение.
 39. Тестирование элементов. Особенности и назначение.
 40. Тестирование интеграции (нисходящее, восходящее). Особенности и назначение.
 41. Системное тестирование. Особенности и назначение.
 42. Разработка через тестирование TDD.
 43. Инструментальные среды тестирования (JUnit).
 44. Динамический и статический анализ кода. Последовательность и описание шагов процесса тестирования.
 45. Развертывание программной системы на аппаратных средствах: артефакты, узлы, моделирование с помощью диаграммы развертывания UML.
 46. Развертывание программной системы. Особенности описания и организации процессов с помощью стандартов IEEE, SWEBOK.
 47. Развертывание программной системы. Особенности описания и организации процессов с помощью методологии TOGAF, набора практик DevOps. Документирование процессов.
 48. Методологии управления программными проектами: связь с моделями жизненного цикла, критерии выбора, области применения.
 49. Тяжеловесные и гибкие методологии управления программными проектами, особенности и отличия.
 50. Рассмотрение особенностей организации разработки на примере Scrum методологии.
 51. Рассмотрение особенностей организации разработки на примере методологии RUP и OpenUP.
 52. Рассмотрение особенностей организации разработки на примере методологии MSF и MSF for Agile
 53. Рассмотрение особенностей организации разработки на примере методологии 1С:ТБВ и 1С:ТКВ.
 54. Модели оценки характеристик качества и надежности ПО. Размерно-ориентированные метрики.
 55. Модели оценки характеристик качества и надежности ПО. Функционально-ориентированные метрики.
 56. Оценки программных проектов. Оценка на основе LOC-метрик.
 57. Оценки программных проектов. Оценка на основе FP-метрик.
 58. Конструктивная модель стоимости проекта (Constructive Cost Model, COCOMO).
 59. Модель композиции приложения.
 60. Модель раннего этапа проектирования.
 61. Модель пост-архитектуры.
 62. Способы предварительной оценки программного проекта.
 63. Оценка программных систем. Метрики связности по данным.
 64. Оценка программных систем. Метрики связности по методам.
 65. Оценка программных систем. Зависимость изменения между классами CDBC.
 66. Оценка программных систем. Метрики Чидамбера и Кемерера.
 67. Оценка программных систем. Метрики Лоренца-Кидда.
 68. Оценка программных систем. Метрики Фернандо Абреу.
 69. Управление качеством. Определение качества программного обеспечения.
 70. Управление качеством. Модели качества процессов разработки (ISO/IEC 15504, ISO 9001:2000, CMM). Факторы и критерии качества.
 71. Управление качеством. Верификация и валидация программного обеспечения.

год начала подготовки 2020

72. Управление программным проектом. Основные понятия управления программным проектом, проектные ограничения.
73. Управление программным проектом. Структура плана управления программным проектом.
74. Управление программным проектом. Иерархическая структура работ (WBS), сетевые графики и диаграммы.
75. Управление программным проектом. Контроль хода программного проекта с помощью метода освоенного объема (EVM).
76. Управление программным проектом. Управление рисками.
77. Управление программным проектом. Формирование команды проекта.
78. Управление программным проектом. Управление документацией.

Перечень тем к курсовой работе:

1. CASE-средства поддержки процесса разработки ПО.
2. OMG SEMAT – единая теория программной инженерии.
3. Автоматизация программного проекта обслуживания клиентов банка через Интернет.
4. Автоматизация программного проекта правового сопровождения кредитования юридических лиц.
5. Автоматизация программного проекта учёта по подключению сети Интернет.
6. Исследование и реализация методов управления программным проектом.
7. Исследование модели управления человеческими ресурсами программного проекта.
8. Исследование экономической модели программного проекта.
9. Метод «Cleanroom» бездефектной разработки ПО.
10. Методология RAD создания средств разработки программных продуктов.
11. Модели лицензирования программного обеспечения.
12. Построение программного проекта автоматизации складского учёта на предприятии.
13. Построение программного проекта информационного сайта.
14. Разработка веб-системы: социальная сеть (однорукпников, персонажей комиксов, людей по интересам).
15. Разработка модуля (системы) обслуживания посетителей в ресторане.
16. Разработка модуля (системы) обслуживания посетителей в салоне красоты.
17. Разработка модуля (системы) онлайн записи в парикмахерскую.
18. Разработка модуля (системы) организации авиаперевозок грузов.
19. Разработка модуля (системы) организации авиаперевозок пассажиров.
20. Разработка модуля (системы) организации автоперевозок грузов.
21. Разработка модуля (системы) организации денежных переводов.
22. Разработка модуля (системы) организации оптовой торговли.
23. Разработка модуля (системы) организации продаж новых автомобилей в автосалоне.
24. Разработка модуля (системы) организации продаж подержанных автомобилей в автосалоне.
25. Разработка модуля (системы) организации продажи театральных билетов.
26. Разработка модуля (системы) организации проката автомобилей.
27. Разработка модуля (системы) организации проката видео-, аудио- и т.п. продукции.
28. Разработка модуля (системы) организации работы с клиентами в фирме страхования.
29. Разработка модуля (системы) организации розничной торговли.
30. Разработка модуля (системы) организации сессии в ВУЗе.
31. Разработка модуля (системы) организации складского хозяйства.
32. Разработка модуля (системы) поиска авиарейсов.

год начала подготовки 2020

33. Разработка модуля (системы) поиска и бронирования авиабилетов.
34. Разработка модуля (системы) поиска и бронирования туров.
35. Разработка модуля (системы) поиска и возврата пропавших животных.
36. Разработка модуля (системы) учета выдачи книг в библиотеке.
37. Разработка модуля (системы) учета записей на прием к врачам в поликлинике.
38. Разработка модуля (системы) учета поселения гостей в гостинице.
39. Разработка модуля (системы) учета посетителей в поликлинике.
40. Разработка модуля (системы) учета поступлений и списаний книг в библиотеке.
41. Разработка модуля (системы) учета свободных номеров в гостинице.
42. Разработка программного проекта автоматизации информационных потоков организации.
43. Разработка программного проекта документооборота в корпоративной информационной системе организации.
44. Разработка программного проекта Интернет-магазина продаж телекоммуникационного оборудования.
45. Разработка программного проекта информационно-аналитической системы контакт-центра.
46. Разработка программного проекта информационно-справочной системы.
47. Разработка программного проекта клиентской базы данных планирования и оказания услуг в организации.
48. Разработка программного проекта учёта товаров.
49. Реинжиниринг бизнес-процессов обслуживания покупателей.
50. Системный анализ методов управления качеством информационных технологий программного проекта.
51. Системный анализ планирования программного проекта.
52. Системный анализ этапов жизненного цикла программного проекта.
53. Управление персоналом: модель People-CMM (SEI).

Методические указания к курсовой работе:

Курсовая работа направлена на самостоятельную разработку проекта создания программного средства по предложенной теме.

Курсовая работа выполняется в течение семестра. Семестровые задания выдаются индивидуально или на группу (при использовании проектного метода).

Курсовая работа должна включать:

- описание видения и границ проекта (vision,scope);
- разработку требований к программному средству;
- разработку функциональных требований технического задания;
- описание информационных объектов;
- детальное проектирование;
- трассировку требований;
- разработку эскизно-технического проекта.

Описание показателей и критериев оценивания курсовой работы.

Оценка курсовой работы определяется исходя из двух критериев, по которым оценивается качество курсовой работы обучающегося.

Критерии качества курсовой работы и их оценка

Оценка	Критерии оценки показателя компетенции на базовом уровне
Отлично	- соответствие содержания курсовой работы теме работы; - имеется обоснованность и доказательность выводов и предложений; - имеется научное и практическое значение выполненной работы;

	<ul style="list-style-type: none"> - оформлена правильно, имеются несущественные стилистические и грамматические ошибки; - все проектные документы разработаны. - диаграммы построены правильно и обоснованно описаны. - логичность и убедительность изложения, соответствие частей проекта заданию. - пояснительная записка написана грамотно и не содержит фактических ошибок.
Хорошо	<ul style="list-style-type: none"> - в основном соответствует содержания курсовой работы теме работы; - в основном имеется обоснованность и доказательность выводов и предложений; - в основном имеется научное и практическое значение выполненной работы; - оформлена правильно, имеются некоторые существенные стилистические и грамматические ошибки; - все проектные документы разработаны. - диаграммы построены правильно и обоснованно описаны. - логичность и убедительность изложения почти достигнуты. - пояснительная записка написана грамотно и не содержит фактических ошибок.
Удовлетворительно	<ul style="list-style-type: none"> - частично соответствует содержания курсовой работы теме работы; - частично имеется обоснованность и доказательность выводов и предложений; - частично имеется научное и практическое значение выполненной работы; - оформлена в основном правильно, имеются существенные стилистические и грамматические ошибки, допущены исправления по тексту; - частично разработаны проектные документы. - не все диаграммы построены правильно. - логичность и убедительность изложения почти достигнуты. - пояснительная записка написана и содержит ошибки.
Неудовлетворительно	<ul style="list-style-type: none"> - не соответствует содержания курсовой работы теме работы; - не имеется обоснованность и доказательность выводов и предложений; - не имеется научное и практическое значение выполненной работы; - оформлена небрежно, имеются грубые стилистические и грамматические ошибки; - отсутствуют проектные документы. - не все диаграммы построены правильно. - логичность и убедительность изложения не достигнуты. - пояснительная записка написана и содержит ошибки.

Критерии защиты курсовой работы и ее оценка

Оценка	Критерии оценки показателя компетенции на базовом уровне
Отлично	- ответ правильный, уверенный, четкий и полный.
Хорошо	- ответ в основном полный, уверенный и правильный, однако допущены незначительные погрешности, исправленные после дополнительных вопросов.
Удовлетворительно	- ответ неполный, неуверенный, нечеткий, отдельные положения неправильные, однако путем наводящих вопросов, в основном, достигается необходимая полнота ответов.
Неудовлетворительно	- ответ сумбурный, неправильный, содержит существенные, принципиальные ошибки, студент не понимает сущности излагаемого вопроса или не дает ответа на него.

Критерии итоговой оценки

Оценка	Критерии оценки показателя компетенции на базовом уровне
Отлично	- по двум критериям ответ оценен на «отлично».
Хорошо	- по двум критериям ответ оценен на «хорошо».
Удовлетворительно	по двум критериям ответ оценен на «удовлетворительно».
Неудовлетворительно	по двум критериям ответ оценен на «неудовлетворительно».

7.3.2 Задания для оценки умений.

В качестве фондов оценочных средств для оценки умений обучающегося используются задания 13-24, рекомендованные для выполнения в часы самостоятельной работы (раздел 6.2.)

7.3.3 Задания для оценки навыков, владений, опыта деятельности

В качестве фондов оценочных средств для оценки навыков, владений, опыта деятельности обучающегося используются задания 25-36, рекомендованные для

8. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

8.1 Основная литература

1. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452137>

2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452749>

8.1 Дополнительная литература

3. Батоврин В.К. Системная и программная инженерия. Словарь-справочник [Электронный ресурс] : учебное пособие для вузов / В.К. Батоврин. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 280 с. — 978-5-4488-0129-7. — Режим доступа: <http://www.iprbookshop.ru/63956.html>

4. Липаев, В. В. Программная инженерия сложных заказных программных продуктов : учебное пособие / В. В. Липаев. — М. : МАКС Пресс, 2014. — 309 с. — ISBN 978-5-317-04750-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/27297.html>

5. Полетайкин А.Н. Учебно-методическое пособие по выполнению лабораторных работ по дисциплине «Программная инженерия». Часть I. Реализация жизненного цикла программного обеспечения [Электронный ресурс] : учебно-методическое пособие / А.Н. Полетайкин. — Электрон. текстовые данные. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2016. — 97 с. — 2227-8397.

9. ПЕРЕЧЕНЬ КОМПЛЕКТОВ ЛИЦЕНЗИОННОГО И СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИСПОЛЬЗУЕМОГО ПРИ ИЗУЧЕНИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

При изучении учебной дисциплины (в том числе в интерактивной форме) предполагается применение современных информационных технологий. Комплект программного обеспечения для их использования включает в себя: операционная система Microsoft Windows 7 Pro, офисный пакет программ Microsoft Office Professional Plus 2010, офисный пакет программ Microsoft Office Professional Plus 2007, антивирусная программа Dr. Web Desktop Security Suite, архиватор 7-zip, аудиопроигрыватель AIMP, просмотр изображений FastStone Image Viewer, ПО для чтения файлов формата PDF Adobe Acrobat Reader, ПО для сканирования документов NAPS2, ПО для записи видео и проведения видеотрансляций OBS Studio, ПО для удалённого администрирования Aspiа, правовой справочник Гарант Аэро, онлайн-версия КонсультантПлюс: Студент, электронно-библиотечная система IPRBooks, электронно-библиотечная система Юрайт, математические вычисления Mathcad 14 University, версия 1С для обучения программированию: 1С: Предприятие 8.2 Версия для обучения программированию

10. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).

10.1. Интернет-ресурсы.

1. <https://cyberleninka.ru> – научная электронная библиотека «КИБЕРЛЕНИНКА»
2. <https://elibrary.ru> – научная электронная библиотека
3. <http://www.gpntb.ru> - Государственная публичная научно-техническая библиотека России
4. <http://swebok.sorlik.ru/> Основы программной инженерии по SWEBOK. Русский перевод SWEBOK 2004 с замечаниями и комментариями подготовлены Сергеем Орликом.
5. <https://www.iso.org/standard/67604.html> ISO/IEC TR 19759:2015 (the IEEE Computer Society) Preview Software Engineering -- Guide to the software engineering body of knowledge (SWEBOK).
6. <https://www.iso.org/standard/50518.html> ISO/IEC/IEEE 24765:2010 Systems and software engineering –Vocabulary.
7. <https://www.intuit.ru/studies/courses/4806/1054/info> Технологии командной разработки программного обеспечения. Учебный курс Академии Microsoft.
8. ЭБС IPRbooks (АйПиАрбукс) <http://www.iprbookshop.ru>
9. Образовательная платформа ЮРАЙТ <https://urait.ru>

11. ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ.

Изучение учебной дисциплины «программная инженерия» обучающимися инвалидами и лицами с ограниченными возможностями здоровья осуществляется в соответствии с Приказом Министерства образования и науки РФ от 9 ноября 2015 г. № 1309

«Об утверждении Порядка обеспечения условий доступности для инвалидов объектов и предоставляемых услуг в сфере образования, а также оказания им при этом необходимой помощи» (с изменениями и дополнениями), Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утвержденными Министерством образования и науки РФ 08.04.2014 г. № АК-44/05 вн, Положением об организации обучения студентов – инвалидов и лиц с ограниченными возможностями здоровья, утвержденным приказом ректора Университета от 6 ноября 2015 года № 60/о, Положением о Центре инклюзивного образования и психологической помощи АНО ВО «Российский новый университет», утвержденного приказом ректора от 20 мая 2016 года № 187/о.

Лица с ограниченными возможностями здоровья и инвалиды обеспечиваются электронными образовательными ресурсами, адаптированными к состоянию их здоровья.

Предоставление специальных технических средств обучения коллективного и индивидуального пользования, подбор и разработка учебных материалов для обучающихся с ограниченными возможностями здоровья производится преподавателями с учетом индивидуальных психофизиологических особенностей обучающихся и специфики приема-передачи учебной информации на основании просьбы, выраженной в письменной форме.

С обучающимися по индивидуальному плану или индивидуальному графику проводятся индивидуальные занятия и консультации.

12. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля успеваемости и промежуточной аттестации

Ауд. 403 (компьютерный класс № 4)

Специализированная мебель:

- столы студенческие;
- стулья студенческие;
- стол для преподавателя;
- стул для преподавателя;
- столы компьютерные;
- кресла компьютерные;
- шкаф для хранения раздаточного материала;
- доска (меловая);
- маркерная доска (переносная).

Технические средства обучения:

- проектор;
- ПК для преподавателя с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду вуза;
- ПК для с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду вуза;
- веб-камера;
- экран;
- колонки;
- микрофон.

Специализированное оборудование:

- наглядные пособия (плакаты)

Автор (составитель): доцент Н.А. Гнездилова




(подпись)

год начала подготовки 2020

**Лист внесения изменений в рабочую программу учебной дисциплины
«Программная инженерия»**

Рабочая программа рассмотрена и одобрена на 2020/2021 учебный год.
Протокол № 1 заседания кафедры ПЭ от «03» сентября 2020 г.

Зав. кафедрой


_____/Преснякова Д.В./

Аннотация рабочей программы учебной дисциплины

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Код и направление подготовки 09.03.03 Прикладная информатика

Прикладная информатика в экономике

Учебная дисциплина «Программная инженерия» изучается обучающимися, осваивающими образовательную программу «Прикладная информатика» по профилю Прикладная информатика в экономике в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению 09.03.03 Прикладная информатика (уровень бакалавриата), утвержденным приказом Министерства образования и науки РФ от 19.09.2017 N 922 (ФГОС ВО 3++).

Основная цель изучения учебной дисциплины состоит в том, чтобы дать обучающимся теоретические знания в области использования современных методик, применяемых при индустриальной разработке программного обеспечения (ПО). В результате прохождения курса студенты приобретут представление о жизненном цикле ПО, технологиях и инструментальных средствах, применяемых на каждом этапе разработки ПО, освоят базовые понятия и принципы проектного менеджмента. Студенты приобретут практические навыки разработки проектной документации, изучат особенности документирования процесса разработки ПО, характерные для отечественной и международной практики.

Учебная дисциплина «Программная инженерия» изучается по заочной форме обучения в ходе 2 сессии 3 курса и 1,2 сессии 4 курса.

Изучение учебной дисциплины направлено на подготовку обучающихся к осуществлению деятельности по концептуальному, функциональному и логическому проектированию систем среднего и крупного масштаба и сложности, планированию разработки или восстановления требований к системе, анализу проблемной ситуации заинтересованных лиц, разработке бизнес-требований заинтересованных лиц, постановки целей создания системы, разработки концепции системы и технического задания на систему, организации оценки соответствия требованиям существующих систем и их аналогов, представлению концепции, технического задания на систему и изменений в них заинтересованным лицам, организации согласования требований к системе, разработке шаблонов документов требований, постановке задачи на разработку требований к подсистемам и контроль их качества, сопровождению приемочных испытаний и ввода в эксплуатацию системы, обработке запросов на изменение требований к системе, определенных профессиональным стандартом «Системный аналитик», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 28.10.2014 N 809н (Регистрационный номер №34882).

В результате освоения дисциплины обучающийся по программе бакалавриата должен овладеть общепрофессиональной компетенцией – способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-7).