

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Горно-Алтайский государственный университет»
(ФГБОУ ВО ГАГУ, ГАГУ, Горно-Алтайский государственный университет)

Программная инженерия рабочая программа дисциплины (модуля)

Закреплена за кафедрой **кафедра экономики, туризма и прикладной информатики**

Учебный план 09.03.03_2026_826.plx
09.03.03 Прикладная информатика
Инжиниринг информационных систем

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **4 ЗЕТ**

Часов по учебному плану	144	Виды контроля в семестрах: экзамен 7
в том числе:		
аудиторные занятия	60	
самостоятельная работа	47	
часов на контроль	34,75	

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	7 (4.1)		Итого	
	Неделя		12 5/6	
Вид занятий	уп	рп	уп	рп
Лекции	20	20	20	20
Лабораторные	30	30	30	30
Практические	10	10	10	10
Консультации (для студента)	1	1	1	1
Контроль самостоятельной работы при проведении аттестации	0,25	0,25	0,25	0,25
Консультации перед экзаменом	1	1	1	1
В том числе инт.	18	18	18	18
Итого ауд.	60	60	60	60
Контактная работа	62,25	62,25	62,25	62,25
Сам. работа	47	47	47	47
Часы на контроль	34,75	34,75	34,75	34,75
Итого	144	144	144	144

Программу составил(и):

ст. преподаватель, Жилин Дмитрий Иванович

Рабочая программа дисциплины

Программная инженерия

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

утвержденного учёным советом вуза от 29.01.2026 протокол № 2.

Рабочая программа утверждена на заседании кафедры

кафедра экономики, туризма и прикладной информатики

Протокол от 09.04.2026 протокол № 9

Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2021-2022 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2021 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2022-2023 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2022 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2023-2024 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2023 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2024-2025 учебном году на заседании кафедры **кафедра экономики, туризма и прикладной информатики**

Протокол от _____ 2024 г. № ____
Зав. кафедрой Газукина Юлия Геннадьевна

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	<i>Цели:</i> - формирование у студентов теоретических и практических навыков по изучению и использованию современных технологий разработки программного обеспечения.
1.2	<i>Задачи:</i> Основными задачами изучения дисциплины являются формирование у студентов умений и навыков по проблемам оценки требований, проектирования, разработки, качества, повышения надежности и документирования программного обеспечения, а также по вопросам управления коллективной разработкой программного обеспечения.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.В
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Высокоуровневые методы информатики и программирования
2.1.2	Интернет-технологии. Интернет-программирование
2.1.3	Разработка и стандартизация программных средств и информационных технологий
2.1.4	Информатика и программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Выполнение и защита выпускной квалификационной работы
2.2.2	Преддипломная практика

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-2: Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение	
ИД-1.ПК-2: Определяет эффективные методы, способы и технологии разработки, внедрения и адаптации прикладного программного обеспечения	
Знать: технологии разработки, внедрения и адаптации прикладного программного обеспечения на базовом уровне. Уметь: разрабатывать, внедрять и адаптировать прикладное программное обеспечение на базовом уровне. Владеть: практическими навыками разработки, внедрения и адаптации прикладного программного обеспечения на базовом уровне.	
ИД-2.ПК-2: Разрабатывает эффективные алгоритмы и программы с использованием современных технологий программирования.	
Знать: технологии разработки, внедрения и адаптации прикладного программного обеспечения на достаточном уровне. Уметь: разрабатывать, внедрять и адаптировать прикладное программное обеспечение на достаточном уровне. Владеть: практическими навыками разработки, внедрения и адаптации прикладного программного обеспечения на достаточном уровне.	
ИД-3.ПК-2: Осуществляет внедрение и адаптацию программного обеспечения для решения конкретных прикладных задач	
Знать: технологии разработки, внедрения и адаптации прикладного программного обеспечения на продвинутом уровне. Уметь: разрабатывать, внедрять и адаптировать прикладное программное обеспечение на продвинутом уровне. Владеть: практическими навыками разработки, внедрения и адаптации прикладного программного обеспечения на продвинутом уровне.	
ПК-13: Способен проводить тестирование компонентов ИС и программного обеспечения ИС	

ИД-1.ПК-13: Определяет и осуществляет выбор современных технологии и методов тестирования, специализированного программного обеспечения автоматизации тестирования ИС.
Знать: теорию и методологию тестирования компонентов программного обеспечения ИС на базовом уровне. Уметь: проводить различные виды тестирования компонентов программного обеспечения ИС на базовом уровне. Владеть: практическими навыками модульного тестирования компонентов программного обеспечения ИС на базовом уровне.
ИД-2.ПК-13: Применяет основные инструментальные средства тестирования компонентов ИС и программного обеспечения ИС
Знать: теорию и методологию тестирования компонентов программного обеспечения ИС на достаточном уровне. Уметь: проводить различные виды тестирования компонентов программного обеспечения ИС на достаточном уровне. Владеть: практическими навыками модульного и регрессионного тестирования компонентов программного обеспечения ИС на достаточном уровне.
ИД-3.ПК-13: Разрабатывает программу и методику тестирования компонентов ИС и программного обеспечения ИС
Знать: теорию и методологию тестирования компонентов программного обеспечения ИС на продвинутом уровне. Уметь: проводить различные виды тестирования компонентов программного обеспечения ИС на продвинутом уровне. Владеть: практическими навыками модульного, регрессионного и нагрузочного тестирования компонентов программного обеспечения ИС на продвинутом уровне
ИД-4.ПК-13: Проводит тестирование компонентов ИС и программного обеспечения ИС в соответствии с программой и методикой тестирования
Знать: виды и методы тестирования программных средств Уметь: выполнять тестирование программных средств по различным сценариям Владеть: навыками формирования отчетов по результатам тестирования программных средств.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
	Раздел 1. Лекции						
1.1	Лекция 1. О предмете изучения: Программная инженерия Программное обеспечение Литература /Лек/	7	0,5	ИД-1.ПК-2 ИД-2.ПК-2 ИД-3.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13 ИД-3.ПК-13 ИД-4.ПК-13	Л1.1 Л1.2 Л1.3Л2.1 Л2.2	0	
1.2	Лекция 2. Процесс разработки программного обеспечения: Процесс Совершенствование процесса Классические модели процесса Литература /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.3	Лекция 3. Рабочий продукт, дисциплина обязательств, проект: Рабочий продукт Дисциплина обязательств Проект Литература /Лек/	7	1,5	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	

1.4	Лекция 4. Архитектура ПО: Обсуждение Определение Множественность точек зрения Язык UML Литература /Лек/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	2	
1.5	Лекция 5. Управление требованиями: Проблема Виды и свойства требований Варианты формализации требований Цикл работы с требованиями /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.6	Лекция 6. Конфигурационное управление: Проблема Единицы конфигурационного управления Управление версиями Управление сборками Понятие baseline /Лек/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	2	
1.7	Лекция 7. Тестирование Управление качеством Тестирование Работа с ошибками /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.8	Лекция 8. Диаграммные техники в работе со знаниями Метод случаи использования Итеративный цикл автор/рецензент Карты памяти /Лек/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.9	Лекция 9. MSF История и текущий статус Основные принципы Модель команды Прочие особенности /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.10	Лекция 10. СМММ Что такое СМММ? Уровни зрелости процессов по СМММ Области усовершенствования /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.11	Лекция 11. «Гибкие» (agile) методы разработки Общее Extreme Programming Scrum /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.12	Лекция 12. Обзор технологии Microsoft Visual Studio Team System (VSTS) Обзор Состав продукта Правила инсталляции Пакет Team Explorer /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.13	Лекция 13. VSTS: управление элементами работ (Work Items) Определение, свойства, жизненный цикл Средства использования /Лек/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	2	
1.14	Лекция 14. VSTS: конфигурационное управление Система контроля версий Автоматические сборки /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
1.15	Лекция 15. VSTS: тестирование Система отслеживания ошибок Модульные тесты Пакеты тестов Автоматическое тестирование Web-приложений /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	

1.16	Лекция 16. VSTS: поддержка различных моделей процесса Поддержка шаблонов процесса Обзор существующих шаблонов MSF for Agile Software Development Scrum /Лек/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
	Раздел 2. Лабораторные работы						
2.1	Процессы программного обеспечения (ПО) /Лаб/	7	4	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.2	Требования к ПО и спецификация требований /Лаб/	7	4	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.3	Разработка и аттестация ПО /Лаб/	7	4	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.4	Развитие ПО /Лаб/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.5	Управление проектом ПО /Лаб/	7	4	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.6	Среды и средства поддержки ПО /Лаб/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	2	
2.7	Определение требований к ИС и разработка технического задания. Планирование работ. /Лаб/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	2	
2.8	Использование средств автоматизации тестирования программного обеспечения. /Лаб/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.9	Использование систем контроля версий исходного кода программ. /Лаб/	7	4	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
2.10	Автоматизация управления проектом по разработке и внедрению автоматизированной информационной системы. /Лаб/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	1	
	Раздел 3. Практические занятия						
3.1	Обзор программных продуктов для одновременной работы нескольких участников /Пр/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	

3.2	Знакомство и создание проекта Microsoft Visual Studio Team System /Пр/	7	2	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.3	Работа с системой отслеживания ошибок /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.4	Работа с системой контроля версий /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.5	Разработка модульных тестов /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.6	Создание и конфигурация автоматической сборки /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.7	Настройка шаблона процесса /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
3.8	Разработка технического задания по ГОСТу /Пр/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
	Раздел 4. Самостоятельная работа						
4.1	Процессы программного обеспечения (ПО) /Ср/	7	8	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
4.2	Требования к ПО и спецификация требований /Ср/	7	7	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
4.3	Разработка и аттестация ПО /Ср/	7	8	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
4.4	Развитие ПО /Ср/	7	8	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
4.5	Управление проектом ПО /Ср/	7	8	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	

4.6	Среды и средства поддержки /Ср/	7	8	ИД-1.ПК-2 ИД-2.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13	Л1.2 Л1.3Л2.1 Л2.2	0	
	Раздел 5. Консультации						
5.1	Консультация по дисциплине /Конс/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-3.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13 ИД-3.ПК-13 ИД-4.ПК-13		0	
	Раздел 6. Промежуточная аттестация (экзамен)						
6.1	Подготовка к экзамену /Экзамен/	7	34,75	ИД-1.ПК-2 ИД-2.ПК-2 ИД-3.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13 ИД-3.ПК-13 ИД-4.ПК-13		0	
6.2	Контроль СР /КСРАтт/	7	0,25	ИД-1.ПК-2 ИД-2.ПК-2 ИД-3.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13 ИД-3.ПК-13 ИД-4.ПК-13		0	
6.3	Контактная работа /КонсЭк/	7	1	ИД-1.ПК-2 ИД-2.ПК-2 ИД-3.ПК-2 ИД-1.ПК-13 ИД-2.ПК-13 ИД-3.ПК-13 ИД-4.ПК-13		0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Пояснительная записка

1. Назначение фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Программная инженерия

2. Фонд оценочных средств включает контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме тестовых заданий для проведения входного контроля, первой и второй промежуточной аттестации, а также вопросы экзамена и примерная тематика рефератов

5.2. Оценочные средства для текущего контроля

Контрольные тесты и задания

Название вопроса*: 1(ПК-2)

Формулировка вопроса: Рефакторинг кода — это процесс изменения внутренней структуры программы без изменения её внешнего поведения.

Ключ: Верно

Название вопроса*: 2(ПК-2)

Формулировка вопроса: Какой инструмент чаще всего используется для контроля версий в командной разработке?

Варианты ответов: 1)Microsoft Word, 2)Adobe Photoshop, 3)Git, 4)Windows Explorer

Ключ: 3

Название вопроса*: 3(ПК-2)

Формулировка вопроса: Полиморфизм - это принцип объектно-ориентированного программирования (ООП), согласно которому один класс (подкласс) наследует свойства и методы другого класса (суперкласса).

Ключ: Неверно

Название вопроса*: 4(ПК-2)

Формулировка вопроса: Какой метод чаще используется для оценки качества внедренного ПО?

Варианты ответов: 1) Количество строк кода, 2) Отзывы пользователей и метрики надежности, 3) Количество коммитов в репозитории, 4) Размер исполняемого файла

Ключ: 2

Название вопроса: 1(ПК-13)

Формулировка вопроса: Выберите верные соответствия

Значение:

Модульное тестирование (Unit Testing)

Интеграционное тестирование (Integration Testing)

Системное тестирование (System Testing)

Регрессионное тестирование (Regression Testing)

Верный ответ:

Тестирование отдельных функций или методов в изоляции.

Проверка взаимодействия между модулями системы.

Проверка всей системы на соответствие требованиям.

Повторное тестирование после изменений.

Название вопроса: 2(ПК-13)

Формулировка вопроса: Автоматизированное тестирование полностью заменяет ручное тестирование и делает его ненужным.

Ключ: Неверно

Название вопроса: 3(ПК-13)

Формулировка вопроса: Какой инструмент НЕ используется для модульного тестирования?

Варианты ответов: 1) JUnit, 2) Selenium, 3) pytest, 4) NUnit

Ключ: 2

Название вопроса: 4(ПК-13)

Формулировка вопроса: Какой метод тестирования используется для проверки устойчивости системы при высокой нагрузке?

Варианты ответов: 1) Юнит-тестирование, 2) Нагрузочное тестирование, 3) Регрессионное тестирование, 4) Дымовое тестирование

Ключ: 2

Оценочные средства для текущего контроля

Примерные вопросы для проверки теоретических знаний по дисциплине

Входной контроль

1. К какому типу проектов относятся проекты по разработке ПО:

a. и к творческим, и к промышленным проектам

b. к творческим проектам

c. к промышленным проектам

2. Какие возвраты невозможны при разработке по водопадной модели:

a. возврат от кодированию к тестированию

b. возврат от тестирования к кодированию

c. возврат от тестирования к анализу

3. В чем заключается согласованность ПО:

a. в том, что ПО основывается на объективных посылках

b. в том, что ПО должно быть согласовано с большим количеством интерфейсов

c. в согласованности заказчика и исполнителя

4. Для чего используется рабочий продукт:

a. для контроля разработки

b. для устранения накладных расходов

- c. для контроля разработки
- 5. Какая стратегия нацелена на решение конкретных проблем компании:
 - a. technology push
 - b. organization pull
 - c. обе стратегии
- 6. Какой вопрос решается в сфере программной инженерии:
 - a. бизнес-реинжиниринг
 - b. вопросы создания компьютерных программ и/или программного обеспечения
 - c. вопрос организации и улучшения процесса разработки ПО
- 7. Какая область объединяет различные инженерные дисциплины по разработке всевозможных искусственных систем:
 - a. системотехника
 - b. бизнес-реинжиниринг
 - c. информатика
- 8. Какое свойство определяет процедуры внесения изменений в требования:
 - a. тестируемость и проверяемость
 - b. модифицируемость
 - c. прослеживаемость
- 9. Для чего предназначены диаграммы конечных автоматов:
 - a. для задания поведения реактивных систем
 - b. для моделирования структуры объектно-ориентированных приложений классов, их атрибутов и заголовков методов, наследования
 - c. для моделирования компонентной структуры распределенных приложений
- 10. Какие тесты представляют собой последовательность действий тестировщика или разработчика, приводящую к воспроизведению ошибки:
 - a. никакие
 - b. любые
 - c. ручные

Примерный перечень вопросов для прохождения первой и второй промежуточной аттестации

- 1. Какую роль выполняет Scrum-мастер?
 - a. обеспечивает максимальную работоспособность и продуктивную работу команды
 - b. постановка для итерации реально достижимых и приоритетных для проекта в целом задач
 - c. представляет в проекте интересы заказчика
- 2. К какому типу проектов относятся проекты по разработке ПО:
 - a. к творческим проектам
 - b. к промышленным проектам
 - c. и к творческим, и к промышленным проектам
- 3. Какие участники имеют право принимать участие в совещаниях Daily Scrum Meeting в Scrum?
 - a. любые заинтересованные лица
 - b. Scrum-команда (Scrum Team)
 - c. Владелец продукта (Product Owner)
 - d. Scrum-мастера (Scrum Master)
- 4. В какой модели каждый виток представляет собой фазу разработки?
 - a. ни в одной из указанных моделей
 - b. в спиральной модели
 - c. в водопадной модели
 - d. в любой модели
- 5. Что реализуют модели, представленные диаграммами UML?
 - a. фазу разработки ПО
 - b. вид деятельности
 - c. точку зрения на программную систему
- 6. В чем заключается согласованность ПО:
 - a. в согласованности заказчика и исполнителя
 - b. в том, что ПО основывается на объективных послылках
 - c. в том, что ПО должно быть согласовано с большим количеством интерфейсов
- 7. В чем отличие информатики от программной инженерии?
 - a. в том, что информатика нацелена на разработку формальных, математизированных подходов к программированию
 - b. в том, что информатика нацелена на решение проблем производства
 - c. ни в чем
- 8. Каковы достоинства водопадной модели?
 - a. ограничение возможности возвратов на произвольный шаг назад
 - b. неустойчивость к сбоям в финансировании проекта
 - c. требование полного окончания фазы-деятельности
 - d. интеграция всех результатов в конце разработки
- 9. При использовании какого метода тестирования реализация системы недоступна тестировщикам:
 - a. в том, что информатика нацелена на решение проблем производства
 - b. в том, что информатика нацелена на решение проблем производства
 - c. ни в чем

- a. при использовании метода белого ящика
 b. при использовании метода черного ящика
 c. при использовании любого метода тестирования
10. В какой модели каждый виток представляет собой фазу разработки?
 a. в любой модели
 b. ни в одной из указанных моделей
 c. в водопадной модели
 d. в спиральной модели
11. Что такое нагрузочное тестирование:
 a. тестирование системы на корректную работу с большими объемами данных
 b. тестирование системы на устойчивость к непредвиденным ситуациям
 c. тестирование всей системы в целом, как правило, через ее пользовательский интерфейс
12. Что определяют варианты использования:
 a. как функции, так и требования
 b. только требования к системе
 c. только функции системы
13. Что такое конфигурационное управление?
 a. специальная деятельность по поддержанию файловых активов проекта в порядке
 b. управление версиями файлов
 c. автоматизированный процесс трансформации исходных текстов ПО в пакет исполняемых модулей
14. Какое свойство зависит от размера программных объектов?
 a. согласованность
 b. сложность
 c. нематериальность
 d. изменяемость
15. Какой вопрос решается в сфере программной инженерии:
 a. вопрос управления командой разработчиков
 b. бизнес-реинжиниринг
 c. вопросы создания компьютерных программ и/или программного обеспечения
16. Какие тесты представляют собой последовательность действий тестировщика или разработчика, приводящую к воспроизведению ошибки:
 a. никакие
 b. ручные
 c. любые
17. Какие требования определяют то, что система должна уметь делать?
 a. функциональные
 b. все требования
 c. нефункциональные
18. При выполнении какого вида тестирования две и более компонент тестируются на совместимость?
 a. при выполнении интеграционного тестирования
 b. при выполнении системного тестирования
 c. при выполнении модульного тестирования
19. При выполнении какого вида тестирования система тестируется на устойчивость к непредвиденным ситуациям?
 a. при выполнении нагрузочного тестирования
 b. при выполнении стрессового тестирования
 c. при выполнении интеграционного тестирования
20. Какой вид деятельности процесса разработки ПО акцентирует внимание на принципах реализации ПО?
 a. проектирование
 b. составление функциональных требований к ПО
 c. тестирование

Краткие методические указания.

Промежуточный тест проводится в электронной форме во время последнего в учебном периоде лабораторного занятия. Тест состоит из 20 тестовых заданий. На выполнение теста отводится 20 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем.

Критерии оценки.

	Описание
5	Процент правильных ответов от 90% до 100%
4	Процент правильных ответов от 75 до 89%
3	Процент правильных ответов от 60 до 74%

5.3. Темы письменных работ (эссе, рефераты, курсовые работы и др.)

Примерные темы докладов по дисциплине:

1. Эволюция сложных программных систем.
2. Методы документирования архитектуры программных систем.

3. Методы документирования требований к программным системам.
4. Управление знаниями в процессе разработки программных систем.
5. CASE технологии разработки программных систем.
6. Модели программных систем.
7. Модели структурного анализа программных проектов.
8. Модели объектно-ориентированного анализа программных проектов.
9. Построение процесса разработки программных систем.
10. Бизнес аспекты разработки программных систем.
11. Методы выбора организационной формы реализации программного проекта.
12. Человеческий фактор при разработке ПО.
13. Модели и методы оценки личностных характеристик исполнителей и команды в целом. Обзор, сравнительный анализ
14. Модели ROI для оценки эффективности компаний-разработчиков программного обеспечения. Описание, примеры использования, инструменты поддержки
15. Оценка затрат программных проектов методом функциональных точек. Описание, примеры использования, инструменты поддержки
16. Регрессионная модель оценки затрат программных проектов COCOMO II. Описание, примеры использования, инструменты поддержки
17. Оценка программных проектов в модели SLIM. Описание, примеры использования, инструменты поддержки
18. Количественные методики оценки рисков программных проектов. Описание, примеры использования, инструменты поддержки
19. Метрические показатели в оценке программных проектов.
20. Метод определения точек тестирования, основанный на анализе цикломатической сложности Мак-Кейба. Описание, примеры использования, инструменты поддержки
21. Инструменты моделирования и трассировки программных требований. Обзор, сравнительный анализ
22. Инструменты верификации программных проектов. Обзор, сравнительный анализ
23. Инструменты оптимизации программных проектов. Обзор, сравнительный анализ
24. Инструменты тестирования программного обеспечения (генераторы тестов, схемы выполнения тестов, оценка тестов, управление тестами). Обзор, сравнительный анализ
25. Инструменты сопровождения (поддержки) программного обеспечения. Обзор, сравнительный анализ
26. Инструменты обеспечения качества программного обеспечения. Обзор, сравнительный анализ
27. Инструменты управления конфигурацией программного обеспечения. Обзор, сравнительный анализ
28. Инструменты планирования и отслеживания программных проектов. Обзор, сравнительный анализ
29. Инструменты, реализующие поддержку инфраструктуры разработки. Обзор, сравнительный анализ
30. Применение open source программных средств для создания UML моделей программного обеспечения.
31. DevOps – новая методология разработки ПО, да ?
32. ITIL, что нужно знать программному инженеру ?

Критерии оценки

Актуальность проблемы и четкость ее постановки.

Соответствие содержания работы заявленной теме.

Четкость и конкретность формулировки проблемы, цели и задач работы. Четкость описания методов реализации проекта.

Самостоятельный анализ фактов по заявленной теме.

Последовательность и логичность этапов реализации проекта.

Соответствие результатов проекта поставленным задачам. Практическая значимость проекта.

Степень реализации проекта на практике.

Наличие собственной оценки эффективности реализации проекта и оценка перспектив развития проекта.

5.4. Оценочные средства для промежуточной аттестации

1. Программная инженерия – понятие, предпосылки и история. Отличие от информатики.
2. Программное обеспечение и программные продукты. Стоимость ПО.
3. Программный процесс – понятие и модели.
4. Методы программной инженерии
5. CASE средства – определение, назначение, примеры.
6. Кодекс этики IEEE-CS/ACM – характер требований и принципы.
7. Стандарты программной инженерии и их разработчики
8. Стратегии разработки ПО – однократные, инкрементные и эволюционные. Адаптивность процесса разработки. Характеристики методологий. Особенности гибких (agile) методологий разработки.
9. Экстремальное программирование (XP) – описание процесса, методологии.
10. Методология SCRUM – роли, артефакты и организация процесса.
11. Программные требования – определение, уровни и свойства. Функциональные не функциональные требования.
12. Процесс разработки требований: роли, способы выявления требований.
13. Анализ и уточнение требований. Приоритизация требований.
14. Спецификация требований.
15. Изменение требований. Политика и анализ влияния изменения.
16. Управление состоянием требований. Трассировка требований, матрица прослеживания требований.

17. Проектирование программного обеспечения по SWEBOOK – структура области знаний.
 18. Руководство программным проектом – четыре «П» разработки. Планирование программного проекта. Структура плана управления.
 19. Ресурсы программного проекта. Сотрудники и роли проекта.
 20. Управление риском: понятие риска, влияние риска, действия при управлении риском.
 21. Анализ риска. Стандарты управления рисками.
 22. Групповая работа над проектом: задачи, типы систем контроля версий (СКВ), операции в СКВ.
 23. Модели качества процессов разработки ПО. Модель зрелости процесса разработки, уровни зрелости модели СММ.

Критерии оценки

«отлично», 91-100%, повышенный уровень

Студент демонстрирует сформированность дисциплинарных компетенций на итоговом уровне, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.

«хорошо», 75-90%, пороговый уровень

Студент демонстрирует сформированность дисциплинарных компетенций на среднем уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

«удовлетворительно», 60-74%, пороговый уровень

Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по дисциплинарной компетенции, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

«неудовлетворительно», менее 60%, уровень не сформирован

Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже базового, проявляется недостаточность знаний, умений, навыков.

«неудовлетворительно», менее 60%, уровень не сформирован

Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Киселева Т.В.	Программная инженерия. Часть 1: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2017	http://www.iprbookshop.ru/69425.html
Л1.2	Романов Е.Л.	Программная инженерия: учебное пособие	Новосибирск: Новосибирский государственный технический университет (НГТУ), 2017	http://www.iprbookshop.ru/91681.html
Л1.3	Кознов Д. В.	Введение в программную инженерию: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024	https://www.iprbookshop.ru/133932.html

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Ружников В.А., Вержаковская М.А., Аронов В.Ю.	Экономика программной инженерии: учебное пособие	Самара: Поволжский государственный университет телекоммуникаций и информатики, 2016	http://www.iprbookshop.ru/73844.html

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.2	Киселева Т.В.	Программная инженерия: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2017	http://www.iprbookshop.ru/83193.html

6.3.1 Перечень программного обеспечения

6.3.1.1	MS Office
6.3.1.2	Яндекс.Браузер
6.3.1.3	LibreOffice
6.3.1.4	Kaspersky Endpoint Security для бизнеса СТАНДАРТНЫЙ
6.3.1.5	MS Windows
6.3.1.6	Oracle VM VirtualBox
6.3.1.7	NVDA
6.3.1.8	Visual Studio
6.3.1.9	РЕД ОС

6.3.2 Перечень информационных справочных систем

6.3.2.1	Гарант
6.3.2.2	Электронно-библиотечная система IPRbooks
6.3.2.3	База данных «Электронная библиотека Горно-Алтайского государственного университета»

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

	метод проектов	
	дискуссия	
	проблемная лекция	
	лекция-визуализация	
	выполнение лабораторных и практических работ	
	работа с программными продуктами	

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Номер аудитории	Назначение	Основное оснащение
317 А2	Компьютерный класс, класс деловых игр, центр (класс) деловых игр, класс имитации деятельности предприятия, лаборатория имитации деятельности предприятия, учебно-тренинговый центр (лаборатория), лаборатория информационно-коммуникативных технологий. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Интерактивная доска с проектором, экран, подключение к интернету, ученическая доска, презентационная трибуна

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Важнейшим этапом практического занятия является самостоятельная работа обучающихся, которая складывается из нескольких разделов:

1. Теоретическая самоподготовка обучающихся по некоторым учебным темам, входящим в примерный тематический учебный план
 2. Знакомство с дополнительной учебной литературой и другими учебными методическими материалами, закрепляющими некоторые практические навыки обучающихся (учебными материалами в интернете, онлайн-курсами).
- Методические рекомендации по составлению опорного конспекта

Конспект, план-конспект – это работа с источником. Цель – зафиксировать, переработать тот или иной учебный текст. Конспект представляет собой дословные выписки из текста источника. При этом конспект – это не полное переписывание чужого текста. Обычно при написании конспекта сначала прочитывается текст-источник, в нём выделяются основные положения, подбираются примеры, идёт перекомпоновка материала, а уже затем оформляется текст конспекта. Конспект

может быть полным, когда работа идёт со всем текстом источника или неполным, когда интерес представляет какой-либо один или несколько вопросов, затронутых в источнике.

План-конспект представляет собой более детальную проработку источника: составляется подробный, сложный план, в котором освещаются не только основные вопросы источника, но и частные. К каждому пункту или подпункту плана подбираются и выписываются цитаты.

Конспектом называется краткое последовательное изложение содержания статьи, книги, лекции. Его основу составляют план, тезисы, выписки, цитаты. Конспект воспроизводит не только мысли оригинала, но и связь между ними, в конспекте отражается не только то, о чем говорится в работе, но и что утверждается, и как доказывается.

Существуют разнообразные виды и способы конспектирования. Одним из наиболее распространенных является так называемый текстуальный конспект, который представляет собой последовательную запись текста книги или лекции.

Такой конспект точно передает логику материала и максимум информации.

Общую последовательность действий при составлении текстуального конспекта можно определить таким образом:

1. Уяснить цели и задачи конспектирования.
2. Ознакомиться с материалом в целом и выделить информационно значимые разделы текста.
3. Внимательно прочитать текст параграфа, главы и отметить информационно значимые места.
4. Составить конспект.

Опорный конспект по логике должен содержать все то, что студент собирается предъявить преподавателю в письменном виде. Это могут быть формулы, формулировки законов, определения, структурные схемы.

Основные требования к содержанию опорного конспекта

1. Полнота – это значит, что в нем должно быть отражено все содержание вопроса.
2. Логически обоснованная последовательность изложения.

Основные требования к форме записи опорного конспекта

1. Опорный конспект должен быть понятен не только вам, но и преподавателю.
2. По объему он должен составлять примерно один - два листа, в зависимости от объема содержания вопроса.
3. Должен содержать, если это необходимо, несколько отдельных пунктов, обозначенных номерами или пробелами.
4. Не должен содержать сплошного текста.
5. Должен быть аккуратно оформлен (иметь привлекательный вид).

Методика составления опорного конспекта

1. Разбить текст на отдельные смысловые пункты.
2. Выделить пункт, который будет главным содержанием ответа.
3. Придать плану законченный вид (в случае необходимости вставить дополнительные пункты, изменить последовательность расположения пунктов).
4. Записать получившийся план в тетради в виде опорного конспекта, вставив в него все то, что должно быть, написано – определения, формулы, выводы, формулировки, выводы формул, формулировки законов и т.д.

Порядок организации самостоятельной работы студентов

Самостоятельная работа бакалавров по данной дисциплине предполагает: - самостоятельный поиск ответов и необходимой информации по предложенным вопросам; - выполнение заданий для самостоятельной работы; - изучение теоретического и лекционного материала, а также основной и дополнительной литературы при подготовке к семинарским занятиям, научным дискуссиям, написании докладов; - самостоятельное изучение отдельных вопросов, не рассматриваемых на практических занятиях, по перечню, предусмотренному в методической разработке данного курса; - подготовка к контрольным работам по темам, предусмотренным программой данного курса; - самостоятельное изучение материалов официальных сайтов для выступления на семинарских занятиях и для подготовки заданий, предусмотренных методической разработкой по данному курсу; - выполнение индивидуальных заданий для КСР по отдельным темам дисциплины, представленным в методической разработке. Объем заданий рассчитан максимально на 2-4 часа в неделю.

Алгоритм самостоятельной работы студентов: 1 этап – поиск в литературе и изучение теоретического материала на предложенные преподавателем темы и вопросы; 2 этап – осмысление полученной информации из основной и дополнительной литературы, освоение терминов и понятий, механизма решения задач; 3 этап – составление плана ответа