

МИНОБРАЗОВАНИЯ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерных сетей

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая 2023 г.

Рабочая программа дисциплины

«Программная инженерия»

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль)

«Информатика и компьютерные науки»

Квалификация выпускника

Бакалавр

Форма обучения

очная

Программа рассмотрена на
заседании кафедры
от 17 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Программная инженерия» является изучение основных качеств программного обеспечения и принципов его построения, обеспечивающих реализацию этих качеств. Изучение подкрепляется примерами, имеющими практическую направленность и учитывающими современные тенденции развития методологий и технологий программной инженерии. Дисциплина «Программная инженерия» обеспечивает приобретение знаний и умений в области проектирования и разработки информационных систем с использованием современных информационных технологий, а также способствует фундаментализации образования, включая в себя изучение универсальных принципов и методологий разработки программного обеспечения.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Программная инженерия» относится к вариативной части ОП бакалавриата.

Она базируется на знаниях и навыках, полученных студентами при изучении общепрофессиональных дисциплин компьютерного цикла, в частности дисциплин «Основы программирования», «Языки и методы программирования», «Базы данных».

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК-2 Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения, в том числе методы машинного обучения.	ПК – 2.1 Демонстрирует знания в области современных языков и технологий программирования, комплексов прикладных компьютерных программ.	Знать: методологию разработки программного обеспечения; Уметь: выбирать подходящие методы разработки программного обеспечения; находить и работать с новыми технологиями. Владеть навыками: • работы с документацией программного обеспечения.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зач. ед., 144 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)	
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
			Контактная работа						
1.	Введение в программную инженерию	6	2	4,5				1	
2.	Качества программного обеспечения	6	2	4,5		1		1	Задания для самостоятельной работы
3.	Принципы программной инженерии	6	2	4,5		1		2	Задания для самостоятельной работы
4.	Спецификации программного обеспечения	6	2	4,5				2	Задания для самостоятельной работы
5.	Проектирование и архитектура программного обеспечения	6	2	4,5				2	Задания для самостоятельной работы
6.	Верификация программного обеспечения	6	2	4,5		1		2	Задания для самостоятельной работы
7.	Процесс разработки и модели жизненного цикла программного обеспечения	6	2	4,5				2	Задания для самостоятельной работы
8.	Управление программной инженерией	6	4	4,5		1		2	
	Всего за 6 семестр		18	36		4		14	экзамен
	Всего		18	36		4		14	

Содержание разделов дисциплины:

1. Введение в программную инженерию
2. Качества программного обеспечения. Классификация качеств. Основные качества продукта и процесса. Качества и типы программных систем.

3. Принципы программной инженерии. Пример: разработка компилятора. Пример: проектирование системы управления лифтами.
4. Спецификации программного обеспечения. Классификация и качества спецификаций. Способы представления операционных и описательных спецификаций.
5. Проектирование и архитектура программного обеспечения. Типовые компоненты архитектуры. Требования к разбиению программного продукта на модули.
6. Верификация программного обеспечения. Определение и подходы к верификации. Верификация корректности. Отдельные приемы и методики.
7. Процесс разработки и модели жизненного цикла программного обеспечения. Понятие модели процесса разработки. Основные этапы разработки. Основные модели разработки.
8. Управление программной инженерией. Понятие и функции управления. Производительность процесса разработки и оценка затрат. Управление проектом.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Семинар (семинарское занятие) – форма занятия, на котором происходит обсуждение студентами под руководством преподавателя заранее подготовленных докладов, рефератов, проектов. Семинар выполняет следующие функции: систематизация и обобщение знаний по изученному вопросу, теме, разделу (в том числе в нескольких учебных курсах); совершенствование умений работать с дополнительными источниками, сопоставлять изложение одних и тех же вопросов в различных источниках информации; умений высказывать свою точку зрения, обосновывать ее; писать рефераты, тезисы и планы докладов и сообщений, конспектировать прочитанное. План семинара озвучивается заранее и в нем обычно указываются основные вопросы, подлежащие рассмотрению и литература, рекомендуемая всем и отдельным докладчикам.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

Лекция-дискуссия – это взаимодействие преподавателя и студентов, свободный обмен мнениями, идеями и взглядами по исследуемому вопросу. В отличие от лекции-беседы здесь преподаватель при изложении лекционного материала не только использует ответы студентов на свои вопросы, но и организует свободный обмен мнениями в интервалах между логическими разделами.

Лекция с разбором конкретных ситуаций – это по форме та же лекция-дискуссия, но на обсуждение преподаватель ставит не вопрос, а конкретную ситуацию. Как правило, такая ситуация представляется устно или в очень короткой видеозаписи, поэтому изложение ее должно быть очень кратким, но содержать достаточную информацию для оценки характерного явления и обсуждения. Это, так называемая, микроситуация. Слушатели анализируют и обсуждают ее сообща, всей аудиторией. Преподаватель старается активизировать участие в обсуждении отдельными вопросами, обращенными к отдельным слушателям, выясняет их оценку суждениям коллег, предлагает сопоставить с собственной практикой, «сталкивает» между собой различные мнения и тем развивает дискуссию, стремясь направить ее в нужное русло. Затем, опираясь на правильные высказывания и анализируя неправильные, ненавязчиво, но убедительно подводит аудиторию к коллективному выводу или обобщению.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Парамонов, И. В., Инженерия программных систем и комплексов на основе гибкой методологии разработки : учеб.-метод. пособие для студентов, обучающихся по направлению Прикладная математика и информатика / И. В. Парамонов; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2015, 47с

2. Парамонов, И. В., Инженерия программных систем и комплексов на основе гибкой методологии разработки [Электронный ресурс] : учеб.-метод. пособие для студентов, обучающихся по направлению Прикладная математика и информатика / И. В.

Парамонов; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2015, 47с

<http://www.lib.uni Yar.ac.ru/edocs/iuni/20150402.pdf>

б) дополнительная:

1. Мацяшек, Л. А., Практическая программная инженерия на основе учебного примера / Л. А. Мацяшек, Б. Л. Лионг ; пер. с англ. А. М. Епанешникова, В. А. Епанешникова, М., БИНОМ. Лаборатория занятий, 2013,

95бс в) ресурсы сети «Интернет»

Электронно-библиотечная система «Юрайт»(<https://urait.ru/>).

Электронно-библиотечная система «Лань»(

<https://e.lanbook.com/>).

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);
- учебные аудитории для проведения лабораторных занятий;
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Доцент кафедры КС _____ И. В. Парамонов

**Приложение №1 к рабочей программе дисциплины
«Программная инженерия»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для
оценки знаний, умений, навыков и (или) опыта деятельности,
характеризующих этапы
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе
текущей аттестации**

Задания для самостоятельной работы

Пример задания по теме «Качества программного обеспечения»:

Рассмотрите ситуацию, связанную с выходом новой версии крупной библиотеки, не совместимой с предыдущей версией. Какие причины приводят разработчиков к подобным решениям? Каковы последствия таких решений для существующего программного обеспечения? Рассмотрите вопрос с точки зрения качества ПО.

Критерии оценивания:

Оценка	Критерии
Отлично	ОПК-3: Знает качества программного обеспечения, выделяет среди них относящиеся к поставленной задаче — сопровождаемость, способность к эволюции . ПК-1: Выявляет причину описанного в условии задачи решения — исчерпание способности к эволюции . Определяет последствия — прекращение жизненного цикла зависимого программного обеспечения или его переписывание для поддержки новой версии библиотеки . При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.
Хорошо	ОПК-3: Знает качества программного обеспечения, выделяет среди них относящиеся к поставленной задаче. ПК-1: Выявляет причину описанного в условии задачи решения — исчерпание способности к эволюции. Не вполне точно определяет последствия. При ответе на вопрос рассуждает уверенно, но не всегда аргументированно, примеры приводит не всегда.
Удовлетворительно	ОПК-3: Знает качества программного обеспечения, но затрудняется с выделением среди них относящихся к поставленной задаче. ПК-1: Выявляет причину описанного в условии задачи решения — исчерпание способности к эволюции. Неточно или неверно определяет последствия. При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.

Неудовлетворительно	ОПК-3: Не знает или нетвёрдо знает качества программного обеспечения.
---------------------	---

	ПК-1: Не понимает или плохо понимает суть поставленного вопроса. При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.
--	---

Пример задания по теме «Принципы программной инженерии»:

Перечислите основные абстракции объектно-ориентированного программирования. Выявите цели этих абстракций, определите условия, при которых эти абстракции являются адекватными.

Критерии оценивания:

Оценка	Критерии
Отлично	ОПК-3: Знает основные абстракции объектно-ориентированного программирования — классы, интерфейсы, объекты, методы, сообщения . Верно выявляет цели этих абстракций — классификацию объектов реального мира, необходимых при разработке ПО, выделение их существенных общих свойств и поведения . ПК-1: Определяет условия адекватности перечисленных абстракций — наличие ярко выраженных объектов в предметной области приложения, отсутствие их явной привязки к оборудованию компьютерной системы . При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.
Хорошо	ОПК-3: Знает основные абстракции объектно-ориентированного программирования. Верно выявляет цели этих абстракций. ПК-1: Затрудняется в определении условий их адекватности. При ответе на вопрос рассуждает в целом уверенно, но не всегда аргументированно, примеры приводит не всегда.
Удовлетворительно	ОПК-3: Знает основные абстракции объектно-ориентированного программирования. Неточно или неверно выявляет цели этих абстракций. ПК-1: Затрудняется в определении условий их адекватности. При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.
Неудовлетворительно	ОПК-3: Не знает или нетвёрдо знает основные абстракции объектно-ориентированного программирования. ПК-1: Не понимает или плохо понимает суть поставленного вопроса. При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.

Пример задания по теме «Спецификации программного обеспечения»:

Опишите процедуру преобразования диаграммы состояний в код в соответствии с шаблоном проектирования State

Критерии оценивания:

Оценка	Критерии
Отлично	ОПК-3: Знает принципы устройства диаграммы состояний и шаблона проектирования State. Верно устанавливает соответствие между элементами диаграммы состояний и элементами шаблона State. Уверенно строит диаграмму классов в соответствии с шаблоном проектирования State по конкретной диаграмме состояний. Верно пишет программный код в соответствии с шаблоном проектирования State по конкретной диаграмме состояний и может его объяснить.
Хорошо	ОПК-3: Знает принципы устройства диаграммы состояний и шаблона проектирования State. Верно устанавливает соответствие между элементами диаграммы состояний и элементами шаблона State. Не вполне уверенно и, возможно, с небольшими ошибками, строит диаграмму классов в соответствии с шаблоном проектирования State по конкретной диаграмме состояний. Не вполне верно пишет программный код в соответствии с шаблоном проектирования State по конкретной диаграмме состояний и затрудняется с объяснением отдельных его элементов.
Удовлетворительно	ОПК-3: Знает принципы устройства диаграммы состояний и шаблона проектирования State. Верно или с небольшими погрешностями устанавливает соответствие между элементами диаграммы состояний и элементами шаблона State. Не может построить диаграмму классов в соответствии с шаблоном проектирования State по конкретной диаграмме состояний или строит её с ошибками. Неверно пишет программный код в соответствии с шаблоном проектирования State по конкретной диаграмме состояний и не может его объяснить.
Неудовлетворительно	ОПК-3: Не знает или нетвёрдо знает принципы устройства диаграммы состояний и шаблона проектирования State. Неверно устанавливает соответствие между элементами диаграммы состояний и элементами шаблона State или не может его установить. Не может построить диаграмму классов в соответствии с шаблоном проектирования State по конкретной диаграмме состояний. Не может написать программный код в соответствии с шаблоном проектирования State по конкретной диаграмме состояний.

Пример задания по теме «Проектирование и архитектура программного обеспечения»:

Определите отношение между модулями IS_COMPONENT_OF. Чем оно отличается от отношения USES, какими свойствами обладает и чем может быть полезно для проектирования?

Критерии оценивания:

Оценка	Критерии
Отлично	ОПК-3: Знает определение отношения USES. На основе знаний по курсу верно определяет отношение IS_COMPONENT_OF, указывая основной его смысл: полное включение компонента в

	<p>состав другого компонента. Верно выявляет главное отличие между отношениями — отношение USES описывает зависимость, а отношение IS_COMPONENT_OF описывает связь типа «целое-часть». Указывает на главное свойство отношения IS_COMPONENT_OF — то, что это отношение является деревом, в отличие от отношения USES, являющегося связным ориентированным графом без циклов.</p> <p>ПК-1: Определяет главную полезность этого отношения для проектирования — возможность структурной декомпозиции модулей с возможной инкапсуляцией некоторых из них. При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.</p>
Хорошо	<p>ОПК-3: Знает определение отношения USES. На основе знаний по курсу верно определяет отношение IS_COMPONENT_OF, указывая основной его смысл. Указывает на главное свойство отношения IS_COMPONENT_OF.</p> <p>ПК-1: Неверно или неточно выявляет главное отличие между отношениями. При ответе на вопрос рассуждает уверенно, но не всегда аргументированно, примеры приводит не всегда.</p>
Удовлетворительно	<p>ОПК-3: Знает определение отношения USES. На основе знаний по курсу верно определяет отношение IS_COMPONENT_OF, указывая основной его смысл. Не может указать на главное свойство отношения IS_COMPONENT_OF.</p> <p>ПК-1: Неверно или неточно выявляет главное отличие между отношениями. При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.</p>
Неудовлетворительно	<p>ОПК-3: Не знает определение отношения USES и не может сформулировать определение отношения IS_COMPONENT_OF.</p> <p>ПК-1: Не понимает или плохо понимает суть поставленного вопроса. При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.</p>

Пример задания по теме «Верификация программного обеспечения»:

Проанализируйте полезность методики автоматической генерации случайных тестовых данных.

Критерии оценивания:

Оценка	Критерии
Отлично	<p>ОПК-3: Понимает отличия между методиками ручного построения тестовых данных и автоматической их генерации. Выделяет преимущества методики автоматической генерации случайных данных, главное из которых — возможность сгенерировать большой объём данных для тестирования.</p> <p>ПК-1: Выделяет недостатки методики автоматической генерации случайных данных — сложность выявления проблем, связанных с краевыми случаями. Определяет условия, при</p>

	<p>которых автоматическая генерация может быть полезна — для нагрузочного тестирования, в случаях наличия сложных зависимостей в тестовых данных, обязательно в комбинации с построением тест-кейсов для краевых случаев вручную. При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.</p>
Хорошо	<p>ОПК-3: Понимает отличия между методиками ручного построения тестовых данных и автоматической их генерации. ПК-1: Выделяет преимущества и недостатки методики автоматической генерации случайных данных. Затрудняется с определением условий, при которых автоматическая генерация может быть полезна, а также области её применимости. При ответе на вопрос рассуждает уверенно, но не всегда аргументированно, примеры приводит не всегда.</p>
Удовлетворительно	<p>ОПК-3: Понимает отличия между методиками ручного построения тестовых данных и автоматической их генерации. ПК-1: Неточно или неверно выделяет преимущества и недостатки методики автоматической генерации случайных данных. Не может определить условия, при которых автоматическая генерация может быть полезна, а также область её применимости. При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.</p>
Неудовлетворительно	<p>ОПК-3: Не видит отличий между методиками ручного построения тестовых данных и автоматической их генерации. ПК-1: Не понимает или плохо понимает суть поставленного вопроса. При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.</p>

Пример задания по теме «Процесс разработки и модели жизненного цикла программного обеспечения»:

Рассмотрите спиральную модель разработки и сопоставьте её с каскадной и итеративной моделями.

Критерии оценивания:

Оценка	Критерии
Отлично	<p>ОПК-3: Формулирует основные положения спиральной модели разработки. Выделяет её основные отличия от каскадной модели (повторяемость циклов разработки) и итеративной модели (большую значимость планирования в процессе разработки, строгую этапность внутри каждого витка).</p> <p>ПК-1: Отмечает и обосновывает существенность анализа рисков в спиральной модели разработки и её явную ориентацию на менеджеров проектов. При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.</p>
Хорошо	<p>ОПК-3: Формулирует основные положения спиральной модели</p>

	<p>разработки. Выделяет её основные отличия от каскадной модели и итеративной модели.</p> <p>ПК-1: Затрудняется с обоснованием существенности анализа рисков в спиральной модели разработки и не отмечает её явную ориентацию на менеджеров проектов. При ответе на вопрос рассуждает уверенно, но не всегда аргументированно, примеры приводит не всегда.</p>
Удовлетворительно	<p>ОПК-3: Формулирует основные положения спиральной модели разработки. Неточно или неверно выделяет её основные отличия от каскадной модели и итеративной модели. Не может выделить характерные черты спиральной модели и область её применимости.</p> <p>ПК-1: При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.</p>
Неудовлетворительно	<p>ОПК-3: Не может сформулировать основные положения спиральной модели разработки или формулирует их с грубыми ошибками.</p> <p>ПК-1: При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.</p>

Список вопросов к зачету:

1. Качества программного обеспечения. Удобство, надёжность, повторная применимость, переносимость, интероперабельность, продуктивность, прозрачность. Влияние качеств друг на друга.
2. Принципы программной инженерии. Выявление желательных качеств разрабатываемого ПО. Соответствие принципов программной инженерии и качеств в конкретном проекте. Методологии и технологии, реализующие принципы инженерии.
3. Спецификации программного обеспечения. Стандарты спецификации. Общее описание программного продукта. Пользовательские сценарии. Функциональные и нефункциональные требования. Анализ требований.
4. Верификация программного обеспечения. Задачи и цели процесса верификации. Типы процессов тестирования и верификации.
5. Процесс разработки и модели жизненного цикла программного обеспечения.
6. Управление программной инженерией.

Пример теста на зачете:

1. Какое качество продукта определяется остаточной долей ошибок в нём и степенью их серьёзности?
 - корректность
 - **надёжность**
 - устойчивость
 - верифицируемость
2. Способом обеспечения интероперабельности продукта является
 - проектирование с учётом изменений
 - **стандартизация и опубликование интерфейсов**
 - использование каскадной модели разработки
 - использование виртуальной машины
3. Какой принцип инженерии подразумевает применение математических методов как основного инструмента при решении всех задач разработки?
 - **формальность**
 - инкрементность
 - модульность
 - абстракция
 - общность
4. Выберите два утверждения, верно описывающие понятие прототипирования
 - Код прототипа после внесения необходимых исправлений включается в код основного проекта
 - **Прототипирование предназначено для получения ответа на некоторый вопрос проектирования**
 - **Прототип может реализовываться на другом языке, нежели основной проект**
 - Прототип должен содержать все необходимые проверки корректности входных данных
5. Выберите понятие, обозначающее определение параметров состояния и реализацию поведения объектов классов, изменяющих это состояние
 - абстракция
 - **инкапсуляция**
 - полиморфизм
 - проектирование
 - бифуркация
6. Выберите два утверждения, характерные для итеративной модели разработки ПО
 - Процесс разработки заранее разложен на фазы, которые не зависят от особенностей реализуемого продукта и не меняются в ходе разработки
 - **Используется промежуточная поставка частично готового продукта заказчику**
 - Системное тестирование выполняется однократно в ходе реализации проекта
 - **В ходе разработки активно используется прототипирование**
 - Модель непригодна для разработки приложений с графическим интерфейсом
7. Выберите два недостатка итеративной модели разработки

- преувеличивается значение документации в ходе разработки
- используется неадекватная модель сопровождения
- **проект может быть труднопрогнозируемым в части результирующей функциональности**
- **предъявляются повышенные требования к заказчику**

8. Выберите из списка операционные спецификации

- **сеть Петри**
- диаграмма потоков данных
- **диаграмма последовательности**
- интерфейс класса
- диаграмма, показывающая время выполнения каждого метода класса

9. Диаграммы какого типа включают такие элементы, как хранилища данных, потоки данных, функции и устройства ввода/вывода?

- диаграммы состояний
- диаграммы последовательности
- диаграммы деятельности
- **диаграммы DFD**

10. Выберите утверждения, описывающие методику аудита кода программы

- **Методика ориентирована на обнаружение ошибок**
- **Методика ориентирована на устранение ошибок**
- Текст программы и документация раздается участникам заранее
- **Список типов возможных ошибок составляется заранее**
- При необходимости производится запуск и тестирование разработанной программы

11. Выберите тип тестирования, направленный на выяснение того, что код, функционировавший корректно ранее, продолжает функционировать корректно и после внесения в него изменений

- тестирование методом прозрачного ящика
- тестирование методом чёрного ящика
- **регрессионное тестирование**
- модульное тестирование
- системное тестирование

12. Укажите минимальное количество тестов, необходимое для покрытия всех путей графа потока управления следующего фрагмента кода

```
if(!var) otherVar = 12;
std::cout << otherVar << std::endl;
```

Ответ: 2.

Правильные ответы выделены жирным шрифтом.

Критерии оценивания:

За каждый выбранный правильный ответ начисляется количество баллов равное единице, делённое на общее количество правильных ответов на вопрос. За каждый выбранный неправильный ответ у студента снимается количество баллов, равное 0.5

балла. Если суммарный балл студента за вопрос оказывается отрицательным, вместо него засчитывается 0.

Баллы	Оценка	Степень сформированности ОПК-3	Степень сформированности ПК-1
0-3	Неудовлетворительно	не сформирована	не сформирована
4-6	Удовлетворительно	пороговый уровень	пороговый уровень
7-9	Хорошо	продвинутый уровень	продвинутый уровень
10-12	Отлично	высокий уровень	высокий уровень

Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех лабораторных работ и зачетного теста на оценку не ниже удовлетворительно.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования,

описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ОПК-3	Задания для самостоятельной работы Тест	1 – 7	<p>Знать:</p> <ul style="list-style-type: none"> • основные качества программного обеспечения и принципы программной инженерии; • основные характеристики и особенности различных этапов жизненного цикла программного обеспечения; • основные методы управления программной инженерией. <p>Уметь:</p> <ul style="list-style-type: none"> • анализировать требования к программным системам на предмет выявления желательных качеств этих систем; • анализировать спецификации программного обеспечения различного уровня с целью выявления недостатков и их исправления; • выполнять декомпозицию, 	<p>Знать основные качества продукта и процесса., качества и типы программных систем., классификацию и качества спецификаций программного обеспечения, способы представления операционных и описательных спецификаций, приемы, методики, подходы к верификации, этапы процесса разработки и модели жизненного цикла программного обеспечения, понятие и функции управления.</p> <p>Уметь определять качества продукта и процесса, качества и типы программных систем, использовать</p>	<p>Знать основные качества продукта и процесса., качества и типы программных систем., классификацию и качества спецификаций программного обеспечения, способы представления операционных и описательных спецификаций, определение, приемы, методики, подходы к верификации, этапы процесса разработки и модели жизненного цикла программного обеспечения, понятие и функции управления.</p> <p>Уметь определять качества продукта и процесса, качества и типы программных систем, анализировать</p>	<p>Знать основные качества продукта и процесса., качества и типы программных систем., классификацию и качества спецификаций программного обеспечения, способы представления операционных и описательных спецификаций, определение, приемы, методики, подходы к верификации, этапы процесса разработки и модели жизненного цикла программного обеспечения, понятие и функции управления.</p> <p>Уметь определять качества продукта и процесса, качества и типы программных систем, анализировать влияние различных качеств друг на друга, анализировать требования к программным системам на предмет</p>

			<p>проектирование, тестирование, поддержку программных систем на различных уровнях абстракции;</p> <p>Владеть навыками:</p> <ul style="list-style-type: none"> • проектирования, тестирования, поддержки программных систем на различных уровнях абстракции; • перепроектирования компонентов существующих программных систем в соответствии с требованиями спецификаций. 	<p>спецификации программного обеспечения, осуществлять верификацию и тестирование программного обеспечения, реализовывать основные этапы процесса разработки и моделей жизненного цикла программного обеспечения.</p>	<p>влияние различных качеств друг на друга, использовать и разрабатывать спецификации программного обеспечения, осуществлять верификацию и тестирование программного обеспечения, реализовывать все этапы процесса разработки и моделей жизненного цикла программного обеспечения, определять производительность процесса разработки и оценку затрат, осуществлять управление проектом.</p>	<p>выявления желательных качеств этих систем; использовать и разрабатывать спецификации программного обеспечения, анализировать спецификации программного обеспечения различного уровня с целью выявления недостатков и их исправления; осуществлять верификацию и тестирование программного обеспечения, реализовывать все этапы процесса разработки и моделей жизненного цикла программного обеспечения, определять производительность процесса разработки и оценку затрат, выполнять декомпозицию, проектирование, тестирование, поддержку программных систем, осуществлять управление проектом.</p>
Профессиональные компетенции						
ПК-1	<p>Задания для самостоятельной работы</p> <p>Тест</p>	1-13	<p>Знать: методологию разработки программного обеспечения;</p> <p>Уметь: выбирать подходящие методы разработки программного обеспечения;</p>	<p>Знать основные методы разработки программного обеспечения</p> <p>Уметь работать с технической документацией</p> <p>Знать и применять методики</p>	<p>Знать возможности и область применимости методологии разработки программного обеспечения.</p> <p>Уметь планировать разработку ПО. Знать и применять</p>	<p>Знать возможности и область применимости методологии разработки программного обеспечения.</p> <p>Уметь планировать разработку ПО.</p> <p>Знать различные архитектурные шаблоны.</p>

			<p>находить и работать с новыми технологиями. Владеть навыками: работы с документацией программного обеспечения.</p>	<p>выполнения конкретных задач Использовать систему управления версиями.</p>	<p>оптимальные методы выполнения конкретных задач Использовать разные системы управления версиями. Знать сценарии решения типичных задач разработки с использованием систем контроля версий. Работать с технической документацией на английском языке</p>	<p>Знать и применять оптимальные методы выполнения конкретных задач Использовать разные системы управления версиями. Знать достоинства и недостатки систем контроля версий Знать сценарии решения большинства задач разработки с использованием систем контроля версий. Работать с технической документацией на английском языке</p>
--	--	--	--	--	---	--

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы

формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;

- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за самостоятельные работы и экзаменационную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за самостоятельные работы и экзаменационную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за самостоятельные работы и экзаменационную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Программная инженерия»

Методические указания для студентов по освоению дисциплины

Теоретические занятия по дисциплине «Программная инженерия» проводятся в форме интерактивных лекций с использованием мультимедиа-технологий. Лекции предполагают наличие дискуссий по поводу тех или иных вопросов программной инженерии осуществляемых в результате соответствующего предложения преподавателя. Их наличие отражает специфику предмета: значительную нечёткость и многоплановость условий применения принципов программной инженерии.

Практическое применение полученных знаний отрабатывается и во время семинарских занятий, ориентированных помимо закрепления лекционного материала на разбор различных модельных ситуаций, характерных для современной программной индустрии. Рассмотрение конкретных вопросов производится на примере разработки системы для поддержания работы dataflow-сети и передачи контекстной информации между агентами dataflow-сети. Техническая документация и применённые методы проектирования модельных агентов, разработки сценариев использования, спецификации и реализации изменений в SIB позволяют рассмотреть большинство вопросов курса в рамках работ над одной системой. Дополнительный эффект может быть достигнут посредством организации встреч студентов с профессионалами в области программной инженерии и проведении мастер-классов.

Основной формой практической работы студентов по усвоению данного курса являются семинарские занятия. Все необходимые задания и вспомогательные учебные материалы предоставляются студентам в электронном виде посредством системы управления обучением LMS Moodle. На семинарских занятиях также производится контроль текущих знаний студентов в форме опросов.

Окончательная аттестация осуществляется в форме экзамена, состоящего из двух частей. Первая часть осуществляет проверку знание базовых определений и понятий с помощью компьютерного тестирования, обеспечивающего быстрый и объективный контроль знаний студентов. Вторая часть экзамена представляет собой письменную работу, которая включает в себе как проверку владения концепциями программной инженерии, так и решение задач, относящихся к модельным ситуациям, типичным для современной индустрии программного обеспечения.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт

меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](#) (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.