Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 24.10.2023 11:54:08
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета
информационных технологий
/Д. Г. Демидов/

"28" 04 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программная инженерия»

Направление подготовки
09.03.01 Информатика и вычислительная техника
Образовательная программа (профиль подготовки)
«Программное обеспечение информационных систем»

Квалификация (степень) выпускника
бакалавр

Форма обучения
заочная

Москва 2022

Программа дисциплины «Программная инженерия» составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению **09.03.01 Информатика и вычислительная техника** и профилю подготовки **«Программное обеспечение информационных систем».**

Программу составил



_____/А.В. Осипов/

Программа дисциплины утверждена на заседании кафедры «Прикладная информатика»
«28» августа 2022 г. протокол № 1

Заведующий кафедрой
доцент, к.э.н.



/С. В. Суворов/

Программа согласована с руководителем образовательной программы по направлению подготовки **09.03.01 Информатика и вычислительная техника** по профилю подготовки **«Программное обеспечение информационных систем».**



_____ /С. В. Суворов/
«____» августа 2022 г.

Программа утверждена на заседании учебно-методической комиссии факультета Информационных технологий

Председатель комиссии  / Д. Г. Демидов/

«____»_____2022 г. Протокол:

1. Цели освоения дисциплины

К **основным целям** освоения дисциплины «Программная инженерия» следует отнести:

– формирование знаний о современных принципах, методах и средствах разработки программных систем;

– подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра по направлению, в том числе формирование умений по выявлению необходимых усовершенствований и разработке нового программного обеспечения.

К **основным задачам** освоения дисциплины «Программная инженерия» следует отнести:

– освоение методологии, анализа и выбора принципов и методов создания сложного программного обеспечения.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Программная инженерия» относится к числу профессиональных учебных дисциплин вариативной части базового цикла (Б1) основной образовательной программы бакалавриата. Дисциплина «Программная инженерия» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

В базовой части базового цикла (Б1):

- информатика;
- программирование;
- базы данных;
- вычислительные системы, сети, телекоммуникации;
- структуры и алгоритмы компьютерной обработки данных.

В вариативной части базового цикла (Б1):

- современные интернет-технологии;
- методы хранения и обработки информации.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины (модуля) у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций:

Код компетенции	**В результате освоения образовательной программы обучающийся должен обладать**	**Перечень планируемых результатов обучения по дисциплине**
ПК-2	способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение.	**знать:** • основы разработки, внедрения и адаптации прикладного программного обеспечения. **уметь:** • использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения. **владеть:** • методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения.
ПК-10	способностью принимать участие во внедрении, адаптации и настройке информационных систем.	**знать:** • теоретические и практические подходы к внедрению, адаптации и настройке информационных систем. **уметь:** • организовывать внедрение, адаптацию и настройку информационных систем. **владеть:** • методами внедрения, адаптации и настройки информационных систем.
ПК-14	способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.	**знать:** • правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач. **уметь:** • использовать стандарты документи-

		рования процессов ведения базы данных и поддержки информационного обеспечения решения прикладных задач. **владеть:** • методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единиц, т.е. 180 академических часов (из них 156 часов – самостоятельная работа студентов).

На четвертом курсе в восьмом семестре выделяется 5 зачетные единицы, т.е. 180 академических часов (из них 156 часов – самостоятельная работа студентов).

Разделы дисциплины «Программная инженерия» изучаются на четвертом курсе в восьмом семестре.

Восьмой семестр: лекции – 1 час в неделю (8 часов), лабораторные работы – 2 часа в неделю (16 часов), форма контроля – экзамен.

Содержание разделов дисциплины «Программная инженерия» отражено ниже. Структура и содержание разделов дисциплины «Программная инженерия» по срокам и видам работы отражены в приложении 1.

Содержание разделов дисциплины

Пятый семестр

Введение. Программная инженерия

Предмет, задачи и содержание дисциплины. Структура курса, его место и роль в подготовке специалиста, связь с другими дисциплинами. Программная инженерия. Ее цели и задачи. Категории специалистов, применяющих методы программной инженерии в своей практической деятельности. Отличия программной инженерии от других инженерных дисциплин. Модели процесса разработки программного обеспечения. Программная документация, виды программных документов. Требования к ней. Стандарты документирования. ЕСПД. ГОСТ. Техническое задание на программный продукт. Программы и методики испытаний программного продукта. Руководство оператора программного продукта.

Эволюция программного обеспечения

Программная система и ее подсистемы. Архитектура программной системы. Стадии и этапы разработки программных систем и программной документации. Основные стандарты программной инженерии и их краткая характеристика. Эволюция программного обеспечения. Краткая характеристика реинженерии, реверсной инженерии, рефакторинга. Жизненный цикл программной системы. Стандарты жизненного цикла в программной инженерии и их назначение. Перечень и содержание процессов жизненного цикла. Виды моделей жизненного цикла. Сравнительная оценка модели процессов жизненного цикла стандарта 12207 и областей-процессов ядра знаний SWEBOK.

Проект программной системы

Проект программной системы, программа проектов и портфель проектов. Состав и содержание документов системного проекта программной системы. Выработка проекта контракта с заказчиком на обеспечение жизненного цикла программной системы. Назначение, содержание и состав плана обеспечения жизненного цикла и распределения ресурсов проекта программной системы. Технико-экономическое обоснование проекта программной системы. Разработка документов технико-экономического обоснования проекта программной системы на базе экспертных оценок модели COCOMO.

Риски программных систем и требования к ним

Риски в жизненном цикле программных систем. Идентификация рисков. Риски при формировании требований к программным системам. Качественный и количественный анализ рисков. Требования к программным системам. Инженерия требований, выявление, анализ, спецификация, аттестация и управление требованиями. Выявление, анализ и документирование требований к программной системе. Версии требований. Трассировка требований. Программные средства управления требованиями.

Инженерия качества

Классификация моделей надежности. Данные для оценивания надежности программных систем. Качество программного обеспечения. Характеристики качества: функциональность, надежность, практичность, эффективность, сопровождаемость, мобильность. Стандарты качества программного обеспечения. Оценка качества. Выработка требований к характеристикам качества программных систем. Классификация программных систем по стандарту ISO 12182. Анализ и

сравнение требований к их характеристикам качества. Динамические и статические методы обеспечения качества программного обеспечения. Верификация. Статический анализ программного обеспечения и обнаружение ошибок. Рефакторинг и модификация программ.

Программометрика

Объектно-ориентированные метрики: метрики Чидамбера-Кемерера, Лоренца и Кида, MOOD. Метрики сложности и размера программ. Задачи, методы и средства аудита программного кода. Отладка программных систем. Методы и средства отладки. Автоматизация отладки программ. CASE-системы. Методы верификации, кодирования и тестирования программных систем. Их краткая характеристика. Автоматизация верификации.

Тестирование программных систем

Тестирование программных систем: основные концепции тестирования, уровни, техники, метрики тестирования. Управление процессом тестирования. Классификация тестов для проверки программных систем. Организация работ для проведения тестирования. Задачи группы тестировщиков. Анализ и оценивание корректности программ по их покрытию тестами. Функциональное, структурное, динамическое, статическое, модульное тестирование программных систем. Тестирование программных систем. Разбиение на классы эквивалентности. Анализ граничных условий. Отрицательное тестирование. Понятие «белого» и «черного» ящика при тестировании. Тестирование производительности, надежности, безопасности, интеграции, стресс-тестирование. Тестирование приложений с графическим пользовательским интерфейсом. Ключевые факторы, влияющие на проектирование интерфейсов. Тестирование конформности (соответствия) реализации программных систем исходным стандартам.

Программные дефекты

Дефект. Управление дефектами. Возможности автоматического выявления дефектов в программе. Общие особенности дефектов, ошибок и рисков в программных системах. Причины и свойства дефектов, ошибок и модификаций в программных системах.

Шестой семестр

Сопровождение программных систем

Основные концепции, ключевые вопросы и техники сопровождения программных систем. Организация и методы сопровождения программных систем.

Задачи и процессы переноса программ и данных на иные платформы. Управление конфигурацией, идентификацией, контролем, учетом статуса, версиями программного проекта. Этапы и процедуры при управлении конфигурацией программных средств. Технологическое обеспечение при сопровождении и управлении конфигурацией программных средств.

Интеграции программ и данных

Цели и задачи интеграции программ и данных в программной инженерии. Краткая характеристика технологий интеграции программ и данных: COM, CORBA, .NET и др. Проблемы преобразования данных при замене одной базы данных на другую. Методы переноса данных. Цели и задачи внесения изменений в программные системы в процессе их сопровождения. Методы внесения изменений. Основные проблемы, возникающие при внесении изменений.

Методологии проектирования программных систем

Групповая разработка. Организация коллектива разработчиков. Организация коллектива разработчиков в компании Microsoft. Метод MSF и его основные модели. Управление проектом и рисками в MSF. Цели и задачи проектной группы в MSF. Agile-методология как пример гибкой методологии разработки. Рекомендации Agile по управлению продуктом, командой, контрактами, рисками. Инженерные практики. Контроль и обеспечение качества. Анализ требований. Масштабирование Agile. Гибкие методологии проектирования. Экстремальное программирование (XP). Методология SCRUM. Водопадная модель проектирования и ее разновидности. Спиральная модель проектирования. Ее особенности. Общие черты инкрементной и эволюционной модели.

Проектирование программных систем с использованием UML

Возможности Rational Rose. Проектирование программных систем с использованием UML: назначение UML, его базовые понятия. UML как средство документирования. Диаграммы классов в UML и моделирование предметной области. Ассоциации. Обобщение. Зависимости. Диаграммы последовательности и взаимодействия компонент в UML. Декомпозиция системы средствами UML. Структурный метод проектирования программных систем. Структуры наглядного проектирования языка UML. Унифицированный процесс RUP (Rational Unified Process) компании Rational Software как итеративный метод разработки программного обеспечения: фазы, визуальное моделирование.

Новые тенденции программирования

Пути развития компонентного программирования. Основные понятия и этапы жизненного цикла компонентного программирования. Основные элементы аспектно-ориентированного программирования. Объекты генерирующего программирования и краткая их характеристика.

Прототипирование программных систем

Быстрая разработка приложений. Итерационно-инкрементальные подходы. Методологии разработки программного обеспечения, базирующиеся на модели CMMI. Задачи и особенности объектно-ориентированного проектирования программных систем. Основные понятия и модели объектно-ориентированного проектирования программных систем.

Управление ресурсами

Управление ресурсами в жизненном цикле программных систем. Основные ресурсы обеспечения жизненного цикла программных систем: обеспечение функциональной пригодности, конструктивных характеристик качества, имитации внешней среды для тестирования и испытаний, сопровождения и мониторинга. Ресурсы специалистов для обеспечения жизненного цикла программных систем. Паттерн «Оценка кандидата». Оценка и отбор кандидатов по резюме. Проведение собеседования. Формирование команды. Лидерство и управление. «Правильные» люди. Мотивация. Эффективное взаимодействие.

Сертификация программных продуктов

Процессы сертификации в жизненном цикле программных средств. Организация сертификации программных продуктов. Документирование процессов и результатов сертификации программных продуктов.

5. Образовательные технологии

Методика преподавания дисциплины «Программная инженерия» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– подготовка к выполнению лабораторных работ в лабораториях вуза;

– обсуждение и защита рефератов по дисциплине;

– подготовка, представление и обсуждение презентаций на семинарских занятиях;

– организация и проведение текущего контроля знаний студентов в форме бланкового тестирования;

– проведение интерактивных занятий по процедуре подготовки к интернет-тестированию;

– использование интерактивных форм текущего контроля в форме аудиторного и внеаудиторного интернет-тестирования;

– проведение мастер-классов экспертов и специалистов по программированию.

Удельный вес занятий, проводимых в интерактивных формах, определен главной целью образовательной программы, особенностью контингента обучающихся и содержанием дисциплины «Программная инженерия» и в целом по дисциплине составляет 50% аудиторных занятий. Занятия лекционного типа составляют 25% от объема аудиторных занятий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

В шестом семестре

- реферат по теме: «Применение методов программной инженерии для разработки программы студента» (индивидуально для каждого обучающегося);
- подготовка к выполнению лабораторных работ и их защита.

Оценочные средства текущего контроля успеваемости включают контрольные вопросы и задания в форме бланкового и (или) компьютерного тестирования, для контроля освоения обучающимися разделов дисциплины, защита рефератов.

Образцы тестовых заданий, контрольных вопросов и заданий для проведения текущего контроля приведены в приложении 2.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-2	способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение.
ПК-10	способностью принимать участие во внедрении, адаптации и настройке информационных систем.
ПК-14	способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины (модуля), описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ПК-2 - способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: основы разработки, внедрения и адаптации прикладного программного обеспечения.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: основы разработки, внедрения и адаптации прикладного	Обучающийся демонстрирует неполное соответствие следующих знаний: основы разработки, внедрения и адаптации	Обучающийся демонстрирует частичное соответствие следующих знаний: основы разработки, внедрения и адаптации прикладного	Обучающийся демонстрирует полное соответствие следующих знаний: основы разработки, внедрения и адаптации

	программного обеспечения.	прикладного программного обеспечения. Допускаются значительные ошибки, проявляется недостаточность знаний по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	программного обеспечения, но допускаются незначительные ошибки, неточности, затруднения при работе с программным обеспечением.	прикладного программного обеспечения, свободно оперирует приобретенными знаниями.
уметь: использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения.	Обучающийся не умеет или в недостаточной степени умеет использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения.	Обучающийся демонстрирует неполное соответствие следующих умений: использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих умений: использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при операциях по работе с программным обеспечением, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие следующих умений: использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.

владеть: методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения.	Обучающийся не владеет или в недостаточной степени владеет методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения.	Обучающийся владеет методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения в неполном объеме, допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду показателей, Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения, свободно применяет полученные навыки в ситуациях повышенной сложности.
ПК-10 - способностью принимать участие во внедрении, адаптации и настройке информационных систем				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: теоретические и практические подходы к внедрению, адаптации и настройке информационных систем.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: теоретические и практические подходы к внедрению, адаптации и настройке информационных систем.	Обучающийся демонстрирует неполное соответствие следующих знаний: теоретические и практические подходы к внедрению, адаптации и настройке информационных систем. Допускаются значительные ошибки, проявляется недостаточность знаний по ряду показателей, обучающийся испытывает	Обучающийся демонстрирует частичное соответствие следующих знаний: теоретические и практические подходы к внедрению, адаптации и настройке информационных систем, но допускаются незначительные ошибки, неточности, затруднения при работе с программным обеспечением.	Обучающийся демонстрирует полное соответствие следующих знаний: теоретические и практические подходы к внедрению, адаптации и настройке информационных систем, свободно оперирует приобретенными знаниями.

		значительные затруднения при оперировании знаниями при их переносе на новые ситуации.		
уметь: организовывать внедрение, адаптацию и настройку информационных систем.	Обучающийся не умеет или в недостаточной степени умеет организовывать внедрение, адаптацию и настройку информационных систем.	Обучающийся демонстрирует неполное соответствие следующих умений: организовывать внедрение, адаптацию и настройку информационных систем. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих умений: организовывать внедрение, адаптацию и настройку информационных систем. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при операциях по работе с программным обеспечением, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие следующих умений: организовывать внедрение, адаптацию и настройку информационных систем. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: методами внедрения, адаптации и настройки информационных систем.	Обучающийся не владеет или в недостаточной степени владеет методами внедрения, адаптации и настройки информационных систем.	Обучающийся владеет методами внедрения, адаптации и настройки информационных систем в неполном объеме, допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду показателей, Обучающийся испытывает	Обучающийся частично владеет методами внедрения, адаптации и настройки информационных систем, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые,	Обучающийся в полном объеме владеет методами внедрения, адаптации и настройки информационных систем, свободно применяет полученные навыки в ситуациях повышенной сложности.

		значительные затруднения при применении навыков в новых ситуациях.	нестандартные ситуации.	
ПК-14 - способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач.	Обучающийся демонстрирует неполное соответствие следующих знаний: правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач. Допускаются значительные ошибки, проявляется недостаточность знаний по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний: правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач, но допускаются незначительные ошибки, неточности, затруднения при работе с программным обеспечением.	Обучающийся демонстрирует полное соответствие следующих знаний: правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач, свободно оперирует приобретенными знаниями.
уметь: использовать стандарты документирования процессов ведения базы данных и поддержки информационного обеспечения решения	Обучающийся не умеет или в недостаточной степени умеет использовать стандарты документирования процессов ведения базы данных и поддержки	Обучающийся демонстрирует неполное соответствие следующих умений: использовать стандарты документирования процессов ведения	Обучающийся демонстрирует частичное соответствие следующих умений: использовать стандарты документирования процессов ведения	Обучающийся демонстрирует полное соответствие следующих умений: использовать стандарты документирования процессов ведения

прикладных задач.	информационного обеспечения решения прикладных задач.	базы данных и поддержки информационного обеспечения решения прикладных задач. Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	базы данных и поддержки информационного обеспечения решения прикладных задач. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при операциях по работе с программным обеспечением, переносе умений на новые, нестандартные ситуации.	базы данных и поддержки информационного обеспечения решения прикладных задач. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач.	Обучающийся не владеет или в недостаточной степени владеет методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач.	Обучающийся владеет методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач в неполном объеме, допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду показателей, Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме методами ведения базы данных и поддержки информационного обеспечения решения прикладных задач, свободно применяет полученные навыки в ситуациях повышенной сложности.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: зачет

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Программная инженерия»: выполнили лабораторные работы, выступили с докладом по теме реферата.

Шкала оценивания	**Описание**
Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Форма промежуточной аттестации: экзамен

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Программная инженерия»: выполнили лабораторные работы, выступили с докладом по теме реферата.

Шкала оценивания	**Описание**
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Фонды оценочных средств представлены в приложении 2 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Перерва А., Еранов С., Иванова В., Сергеев С. Путь IT-менеджера. Управление проектной средой и IT-проектами. – СПб.: Питер, 2016.
2. Иванова В., Перерва А. Путь аналитика. Практическое руководство IT-специалиста. 2 изд. – СПб.: Питер, 2015.
3. Иванова Г.С. Технология программирования. – Москва: Кнорус, 2013 г.
4. Орлов С.А., Цилькер Б. Я. Технологии разработки программного обеспечения. – СПб.: Питер, 2012 г.

б) дополнительная литература:

1. Андрианов С.Н. Краткий конспект лекций по курсу «Программная инженерия»: Учеб. пособие/ – СПб.: СПбГУ, 2007.
2. Архипенков С. Лекции по управлению программными проектами: Учеб. пособие/ – СПб.: СПбГУ, 2010.

в) программное обеспечение и интернет-ресурсы:

1. Пакет Open Server версии 5.2.6 ultimate.
2. Язык программирования Java версии 8.1.
3. Интегрированная среда разработки IDE NetBeans 8.1.

8. Материально-техническое обеспечение дисциплины

Компьютерные классы кафедры «Прикладная информатика», оснащенные компьютерами с установленным программным обеспечением в соответствии с п. 7 рабочей программы

9. Методические рекомендации для самостоятельной работы студентов

Обучение по дисциплине «Программная инженерия» предполагает изучение курса на аудиторных занятиях и самостоятельной работы обучающихся. Виды самостоятельной работы обучающегося, порядок и сроки ее выполнения:

- подготовка к лекциям, лабораторным работам и практическим занятиям с использованием конспекта лекций, материалов лабораторных занятий и учебно-методического и информационного обеспечения дисциплины (в течение 5 и 6 семестров в соответствии с расписанием занятий);
- оформление отчетов по выполненным лабораторным работам и теоретическая подготовка к их сдаче (в течение 5 и 6 семестров в соответствии с расписанием занятий).

Перечень вопросов для проведения текущего контроля и промежуточной аттестации – в соответствии с тематикой разделов дисциплины.

С целью обеспечения успешного обучения обучающийся должен готовиться к

лекции, которая является важнейшей формой организации учебного процесса, поскольку:

- знакомит с новым учебным материалом;
- разъясняет учебные элементы, трудные для понимания;
- систематизирует учебный материал;
- ориентирует в учебном процессе.

Подготовка к занятиям лекционного типа заключается в следующем:

- внимательном чтении материала предыдущей лекции;
- предварительное ознакомление с темой предстоящей лекции (по тематическому плану, по информации лектора);
- ознакомление с учебным материалом по учебнику и учебным пособиям;
- уяснение места изучаемой темы в своей профессиональной подготовке;
- подготовке возможных вопросов, которые студент задаст лектору на лекции.

Подготовка к написанию реферата исходит из того, что реферат – это самостоятельная учебно-исследовательская работа обучающегося, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее. Содержание материала должно быть логичным, изложение материала носит проблемно-поисковый характер.

Этапы работы над рефератом.

1. Формулирование темы. Тема должна быть не только актуальной по своему значению, но и оригинальной, интересной по содержанию.
2. Подбор и изучение основных источников по теме.
3. Составление библиографии.
4. Обработка и систематизация информации.
5. Разработка плана реферата.
6. Оформление реферата в виде презентации.
7. Публичное выступление с результатами исследования на занятии.

Содержание работы должно отражать.

1. Знание современного состояния проблемы.
2. Обоснование выбранной темы.
3. Использование известных результатов и фактов.
4. Полноту цитируемой литературы, ссылки на работы ученых, занимающихся данной проблемой.
5. Актуальность поставленной проблемы.
6. Материал, подтверждающий научное, либо практическое значение в настоящее время.

Типовая структура реферата.

1. Титульный лист.
2. План (простой или развернутый).
3. Введение.
4. Основная часть.
5. Заключение.
6. Список литературы.

Реферат может рассматриваться как одна из форм контрольных работ. Объем реферата не должен превышать 10 страниц.

Представление реферата осуществляется в форме доклада с предъявлением презентации.

Приложение 2 к
рабочей программе

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Направление подготовки: 09.03.01 Информатика и вычислительная техника

ОП (профиль): «Программное обеспечение информационных систем»

Форма обучения: заочная

Вид профессиональной деятельности:
- Проектная*;*
- производственно-технологическая*;*
- аналитическая*;*

Кафедра: Прикладной информатики

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

«ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Состав: 1. Паспорт фонда оценочных средств

2. Описание оценочных средств:

2.1 Показатели уровня сформированности компетенций.
2.2 Пример тем рефератов.
2.3 Пример комплекта разноуровневых задач и заданий.
2.4 Пример вопросов устного опроса.

Москва, 2022 год

Паспорт фонда оценочных средств

Фонд оценочных средств включает в себя:

1. Показатели уровня сформированности компетенций.
2. Пример тем рефератов.
3. Пример комплекта разноуровневых задач и заданий.
4. Пример вопросов устного опроса.

ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Программная инженерия					
ФГОС ВО **09.03.01 Информатика и вычислительная техника**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **профессиональные компетенции**:					
КОМПЕТЕНЦИИ		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства**	**Степени уровней освоения компетенций**
ИН-ДЕКС	**ФОРМУЛИРОВ-КА**				
ПК-2	***способностью*** *разрабатывать, внедрять и адаптировать прикладное программное обеспечение.*	**Знать:** - основы разработки, внедрения и адаптации прикладного программного обеспечения. **Уметь:** - использовать рекомендации программной инженерии для разработки, внедрения и адаптации прикладного программного обеспечения. **Владеть*:*** - методами и средствами разработки, внедрения и адаптации прикладного программного обеспечения.	лекции, самостоятельная работа, лабораторные работы.	УО, РЗЗ, Р	**Базовый уровень** - студент способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение для решения задач научной и проектно-технологической деятельности в стандартных учебных ситуациях. **Повышенный уровень** - студент способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение для решения задач научной и проектно-технологической деятельности в практической деятельности.

ПК-10	***способностью*** *принимать участие во внедрении, адаптации и настройке информационных систем.*	**Знать:** - теоретические и практические подходы к внедрению, адаптации и настройке информационных систем. **Уметь:** - организовывать внедрение, адаптацию и настройку информационных систем. **Владеть:** - методами внедрения, адаптации и настройки информационных систем.	лекции, самостоятельная работа, лабораторные работы.	УО, РЗЗ, Р	**Базовый уровень** - студент способен принимать участие во внедрении, адаптации и настройке информационных систем в стандартных учебных ситуациях. **Повышенный уровень** - студент способен принимать участие во внедрении, адаптации и настройке информационных систем в практической деятельности.
ПК-14	***способностью*** *осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.*	**Знать:** - правила ведения базы данных и поддержки информационного обеспечения решения прикладных задач. **Уметь:** - использовать стандарты документирования процессов ведения базы данных и поддержки информационного обеспечения решения прикладных задач. **Владеть*:*** - методами ведения базы данных	лекции, самостоятельная работа, лабораторные работы.	УО, РЗЗ, Р	**Базовый уровень** - студент способен осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач в стандартных учебных ситуациях. **Повышенный уровень** - студент способен осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач в практической деятельности.

		и поддержки информационного обеспечения решения прикладных задач.			

Пример тем рефератов

1. Техническое задание программы, разработанной студентом в семестре.
2. Программа и методика испытаний программы, разработанной студентом в семестре.
3. Руководство оператора программы, разработанной студентом в семестре.
4. Основные стандарты программной инженерии и их краткая характеристика.
5. Виды моделей жизненного цикла программ.
6. Архитектура программной системы.
7. Качество программного обеспечения.
8. Верификация программного обеспечения.
9. Организация и методы сопровождения программных систем.
10. Применение методов программной инженерии для разработки программы студента (индивидуально для каждого обучающегося).

Пример комплекта разноуровневых задач и заданий

Требуется составить техническое задание, программу и методику испытаний и руководство оператора на разработку электронного портала учебного заведения, предусмотрев реализацию основных функций электронного портала. В качестве примера подражания рекомендуется рассмотреть электронный портал Московского политехнического университета.

Пример вопросов устного опроса

1. Программная инженерия. Ее цели и задачи.
2. Категории специалистов, применяющих методы программной инженерии в своей практической деятельности.
3. Отличия программной инженерии от других инженерных дисциплин.
4. Модели процесса разработки программного обеспечения.
5. Эволюция программного обеспечения.
6. Краткая характеристика реинженерии, реверсной инженерии, рефакторинга.
7. Жизненный цикл программной системы.
8. Стандарты жизненного цикла программ в программной инженерии и их назначение.
9. Перечень и содержание процессов жизненного цикла программ.
10. Программная система и ее подсистемы.
11. Стадии и этапы разработки программных систем и программной документации.
12. Проект программной системы, программа проектов и портфель проектов.
13. Выработка проекта контракта с заказчиком на обеспечение жизненного цикла программной системы.
14. Назначение, содержание и состав плана обеспечения жизненного цикла и распределения ресурсов проекта программной системы.
15. Технико-экономическое обоснование проекта программной системы.
16. Разработка документов технико-экономического обоснования проекта программной системы на базе экспертных оценок модели COCOMO.
17. Риски в жизненном цикле программных систем.
18. Идентификация рисков.
19. Риски при формировании требований к программным системам.
20. Качественный и количественный анализ рисков.
21. Требования к программным системам.
22. Инженерия требований, выявление, анализ, спецификация, аттестация и управление требованиями.
23. Выявление, анализ и документирование требований к программной системе.
24. Версии требований.
25. Трассировка требований.
26. Программные средства управления требованиями.

27. Инженерия качества.
28. Классификация моделей надежности.
29. Данные для оценивания надежности программных систем.
30. Характеристики качества: функциональность, надежность, практичность, эффективность, сопровождаемость, мобильность.
31. Стандарты качества программного обеспечения.
32. Оценка качества программного обеспечения.
33. Выработка требований к характеристикам качества программных систем.
34. Классификация программных систем по стандарту ISO 12182.
35. Анализ и сравнение требований к их характеристикам качества.
36. Динамические и статические методы обеспечения качества программного обеспечения.
37. Статический анализ программного обеспечения и обнаружение ошибок.
38. Рефакторинг и модификация программ.
39. Программометрика. Объектно-ориентированные метрики: метрики Чидамбера-Кемерера, Лоренца и Кида, MOOD.
40. Метрики сложности и размера программ.
41. Задачи, методы и средства аудита программного кода.
42. Отладка программных систем. Методы и средства отладки.
43. Автоматизация отладки программ. CASE-системы.
44. Методы верификации, кодирования и тестирования программных систем. Их краткая характеристика.
45. Автоматизация верификации программного обеспечения.
46. Тестирование программных систем: основные концепции тестирования, уровни, техники, метрики тестирования.
47. Управление процессом тестирования программного обеспечения.
48. Классификация тестов для проверки программных систем.
49. Организация работ для проведения тестирования.
50. Задачи группы тестировщиков программного обеспечения.
51. Анализ и оценивание корректности программ по их покрытию тестами.
52. Функциональное, структурное, динамическое, статическое, модульное тестирование программных систем.
53. Тестирование программных систем. Разбиение на классы эквивалентности. Анализ граничных условий.
54. Отрицательное тестирование программных систем.

55. Понятие «белого» и «черного» ящика при тестировании программных систем.
56. Тестирование производительности, надежности, безопасности, интеграции, стресс-тестирование.
57. Тестирование приложений с графическим пользовательским интерфейсом.
58. Ключевые факторы, влияющие на проектирование интерфейсов.
59. Дефект. Управление дефектами. Возможности автоматического выявления дефектов в программе.
60. Общие особенности дефектов, ошибок и рисков в программных системах.
61. Причины и свойства дефектов, ошибок и модификаций в программных системах.
62. Тестирование конформности (соответствия) реализации программных систем исходным стандартам.
63. Сопровождение программных систем. Основные концепции, ключевые вопросы и техники сопровождения.
64. Задачи и процессы переноса программ и данных на иные платформы.
65. Программная документация, виды программных документов. Требования к ней.
66. Стандарты документирования. ЕСПД.
67. ГОСТ. Техническое задание на программный продукт.
68. ГОСТ. Программы и методики испытаний программного продукта.
69. ГОСТ. Руководство оператора программного продукта.
70. Управление конфигурацией, идентификацией, контролем, учетом статуса, версиями программного проекта.
71. Этапы и процедуры при управлении конфигурацией программных средств.
72. Технологическое обеспечение при сопровождении и управлении конфигурацией программных средств.
73. Цели и задачи интеграции программ и данных в программной инженерии.
74. Краткая характеристика технологий интеграции программ и данных: COM, CORBA, .NET и др.
75. Проблемы преобразования данных при замене одной базы данных на другую. Методы переноса данных.
76. Цели и задачи внесения изменений в программные системы в процессе их сопровождения.
77. Методы внесения изменений. Основные проблемы, возникающие при внесении изменений.

78. Групповая разработка. Организация коллектива разработчиков. Организация коллектива разработчиков в компании Microsoft.
79. Метод MSF и его основные модели. Управление проектом и рисками в MSF. Цели и задачи проектной группы в MSF.
80. Возможности Rational Rose. Проектирование программных систем с использованием UML: назначение UML, его базовые понятия.
81. UML как средство документирования.
82. Диаграммы классов в UML и моделирование предметной области. Ассоциации. Обобщение. Зависимости.
83. Диаграммы последовательности и взаимодействия компонент в UML.
84. Декомпозиция системы средствами UML.
85. Структурный метод проектирования программных систем. Структуры наглядного проектирования языка UML.
86. Унифицированный процесс RUP (Rational Unified Process) компании Rational Software как итеративный метод разработки программного обеспечения: фазы, визуальное моделирование.
87. Agile-методология как пример гибкой методологии разработки.
88. Рекомендации Agile по управлению продуктом, командой, контрактами, рисками. Инженерные практики. Контроль и обеспечение качества. Анализ требований. Масштабирование Agile.
89. Гибкие методологии проектирования. Экстремальное программирование (XP).
90. Методология SCRUM.
91. Водопадная модель проектирования и ее разновидности.
92. Спиральная модель проектирования. Ее особенности.
93. Общие черты инкрементной и эволюционной модели. Примеры.
94. Сравнительная оценка модели процессов жизненного цикла стандарта 12207 и областей-процессов ядра знаний SWEBOK.
95. Пути развития компонентного программирования. Основные понятия и этапы жизненного цикла компонентного программирования.
96. Основные элементы аспектно-ориентированного программирования.
97. Объекты генерирующего программирования и краткая их характеристика.
98. Прототипирование. Быстрая разработка приложений.
99. Методологии разработки программного обеспечения, базирующиеся на модели CMMI.

100. Задачи и особенности объектно-ориентированного проектирования программных систем. Основные понятия и модели объектно-ориентированного проектирования программных систем.
101. Управление ресурсами в жизненном цикле программных систем.
102. Основные ресурсы обеспечения жизненного цикла программных систем: обеспечение функциональной пригодности, конструктивных характеристик качества, имитации внешней среды для тестирования и испытаний, сопровождения и мониторинга.
103. Ресурсы специалистов для обеспечения жизненного цикла программных систем. Паттерн «Оценка кандидата».
104. Оценка и отбор кандидатов по резюме. Проведение собеседования.
105. Формирование команды. Лидерство и управление. «Правильные» люди. Мотивация. Эффективное взаимодействие. Примеры.
106. Процессы сертификации в жизненном цикле программных средств.
107. Организация сертификации программных продуктов.
108. Документирование процессов и результатов сертификации программных продуктов.

Приложение 1 к
рабочей программе

Структура и содержание дисциплины «Программная инженерия» по направлению подготовки 09.03.01 Информатика и вычислительная техника (бакалавр)

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	РГР	Реферат	К/р	Э	З
	Пятый семестр														
1.1	**Введение. Программная инженерия.**	8	1	1											
1.2	*Лабораторная работа «*Стандарты документирования».	8	1,2			2	20								
1.3	**Эволюция программного обеспечения.**	8	2	1											
1.4	*Лабораторная работа «Стандарты жизненного цикла в программной инженерии и их назначение*».	8	3,4			2	20								
1.5	**Проект программной системы.**	8	3	1											
1.6	*Лабораторная работа* «Технико-экономическое обоснование проекта программной системы».	8	5,6			2	20								
1.7	**Риски программных систем и**	8	4	1											

	требования к ним.											+			
1.8	*Лабораторная работа* «Выявление, анализ и документирование требований к программной системе».	8	7,8			2	20					+			
1.9	**Инженерия качества.**	8	5	1								+			
1.10	*Лабораторная работа* «Стандарты качества программного обеспечения».	8	9,10			2	20					+			
1.11	**Программометрика.**	8	6	1								+			
1.12	Лабораторная работа «Методы и средства отладки программных систем».	8	11,12			2	20					+			
1.13	**Тестирование программных систем.**	8	7	1								+			
1.14	*Лабораторная работа* «Методы тестирования программных систем».	8	13,14			2	20					+			
1.15	**Программные дефекты.**	8	8	1								+			
1.16	*Лабораторная работа* «Управление программными дефектами».	8	15,16			2	16					+			
	Форма аттестации														З
	Всего часов по дисциплине в восьмом семестре			8		16	156					+			