МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
"Казанский (Приволжский) федеральный университет"
Высшая школа информационных технологий и интеллектуальных систем

УТВЕРЖДАЮ
Проректор по образовательной деятельности КФУ
Проф. Д.А. Таюрский
_____ «___»_____20__г.

подписано электронно-цифровой подписью

Программа дисциплины

Программная инженерия Б1.Б.14

Направление подготовки: 09.03.03 - Прикладная информатика
Профиль подготовки: не предусмотрено
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский

Автор(ы):

Кугуракова В.В.

Рецензент(ы):

Таланов М.О.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Хасьянов А. Ф.
Протокол заседания кафедры No ___ от "____" _____ 201__г

Учебно-методическая комиссия Высшей школы информационных технологий и интеллектуальных систем:
Протокол заседания УМК No ____ от "____" _____ 201__г

Регистрационный No 689514219

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) старший преподаватель, б/с Кугуракова В.В. Кафедра программной инженерии Высшая школа информационных технологий и интеллектуальных систем , vlada.kugurakova@gmail.com

1. Цели освоения дисциплины

В результате изучения курса 'Программная инженерия. Жизненный цикл разработки программного обеспечения' студент должен иметь представление о месте и роли программной инженерии в современном обществе, о ее взаимосвязи с другими дисциплинами; знать систему базовых понятий программной инженерии, методы организации работ по созданию и сопровождению программных средств, методы выполнения основных видов работ в рамках жизненного цикла ПС: анализа требований, проектирования и построения программного обеспечения, анализа свойств ПС и контроля их

качества; уметь применять на практике изученные методы организации работ по созданию и сопровождению ПС и методы выполнения основных видов работ в рамках жизненного цикла ПС: анализа требований, проектирования и построения программного обеспечения, анализа свойств ПС и контроля их качества.

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел основной профессиональной образовательной программы 09.03.03 'Прикладная информатика' и относится к базовой (общепрофессиональной) части. Осваивается на 3 курсе, в 5, 6 семестрах.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-1 (профессиональные компетенции)	ОПК-1 Способность использовать нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий
ПК-1 (профессиональные компетенции)	способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе
ПК-3 (профессиональные компетенции)	способность проектировать ИС в соответствии с профилем подготовки по видам обеспечения
ПК-4 (профессиональные компетенции)	способность документировать процессы создания информационных систем на стадиях жизненного цикла
ПК-6 (профессиональные компетенции)	способность собирать детальную информацию для формализации требований пользователей заказчика
ПК-16 (профессиональные компетенции)	способность осуществлять презентацию информационной системы и начальное обучение пользователей
ПК-19 (профессиональные компетенции)	способность принимать участие в реализации профессиональных коммуникаций в рамках проектных групп, обучать пользователей информационных систем

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-20 (профессиональные компетенции)	способность осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем
ПК-22 (профессиональные компетенции)	способность анализировать рынок программно-технических средств, информационных продуктов и услуг для создания и модификации информационных систем
ПК-24 (профессиональные компетенции)	способность готовить обзоры научной литературы и электронных информационно-образовательных ресурсов для профессиональной деятельности
ПК-7 (профессиональные компетенции)	способность проводить описание прикладных процессов и информационного обеспечения решения прикладных задач
ПК-9 (профессиональные компетенции)	способность составлять техническую документацию проектов автоматизации и информатизации прикладных процессов

В результате освоения дисциплины студент:

4. должен демонстрировать способность и готовность:

Выпускник, освоивший дисциплину:

1. должен знать:

основные процессы программной инженерии;

экономико-правовые основы разработки программных продуктов;

методы управления проектами программных средств;

профили стандартов открытых ИС;

задачи и методы обеспечения качества программных компонентов;

методы и средства оценки сложности алгоритмов;

модели и основные процессы жизненного цикла программных средств;

методы анализа прикладной области на различных уровнях;

основные способы использования, обобщения и анализа информации в области программной инженерии;

основные методы организации коллективной работы при решении задач в области программной инженерии.

2. должен уметь:

формулировать требования к создаваемым программным комплексам;

использовать международные и отечественные стандарты, разрабатывать программные приложения;

формировать архитектуру программных комплексов для информатизации предприятий;

применять методы анализа прикладной области на различных уровнях;

использовать нормативные правовые документы при создании программных комплексов.

3. должен владеть:

инструментальными средствами программной инженерии;

способами технико-экономического обоснования проектов программных средств;

навыками: использования нормативных правовых документов при разработке программных продуктов, анализа прикладной области на различных уровнях, документирования программных комплексов, стандартизации разработки программных средств, разработки программных комплексов для решения прикладных задач, навыками организации коллективной работы при решении задач в области программной инженерии.

4. должен демонстрировать способность и готовность:

Ставить и решать задачи комплексного анализа, связанные с созданием новых информационных технологий и информационных систем в экономике, с использованием базовых и специальных знаний, современных аналитических методов и моделей.

Разрабатывать новые и модернизировать уже существующие информационные технологии и системы (в экономике) в соответствии с техническим заданием.

Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 7 зачетных(ые) единиц(ы) 252 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 5 семестре; экзамен в 6 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи-ческие занятия	Лабора-торные работы	
1.	Тема 1. Введение. Предмет программной инженерии.	5		2	2	0	Лабораторные работы
2.	Тема 2. Жизненный цикл программного обеспечения.	5		4	8	0	Лабораторные работы
3.	Тема 3. Обзор методологий проектирования программных продуктов.	5		4	8	0	Лабораторные работы

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи-ческие занятия	Лабора-торные работы	
4.	Тема 4. Основные процессы программной инженерии.	5		4	8	0	Лабораторные работы
5.	Тема 5. Внедрение и сопровождение программных продуктов.	5		2	8	0	Лабораторные работы
6.	Тема 6. Экономика программной инженерии.	5		2	2	0	Лабораторные работы
7.	Тема 7. Методологии ведения проектов.	6		9	9	0	Письменная работа
8.	Тема 8. Проектная документация. Требования к проекту.	6		9	9	0	Письменная работа
9.	Тема 9. Принципы архитектуры и шаблоны проектирования.	6		9	9	0	Письменная работа
10.	Тема 10. Стандарт UML.	6		9	9	0	Контрольная работа
.	Тема . Итоговая форма контроля	5		0	0	0	Зачет
.	Тема . Итоговая форма контроля	6		0	0	0	Экзамен
	Итого			54	72	0	

4.2 Содержание дисциплины

Тема 1. Введение. Предмет программной инженерии.

лекционное занятие (2 часа(ов)):

Роль информационно-коммуникационных технологий в рыночной экономике. Задачи курса и порядок его изучения. Роль области знания "Программная инженерия" (Software Engneering) в современном мире. История становления и развития программной инженерии. Требования к подготовке специалиста: овладение управленческими, техническими и технологическими принципами организации.

практическое занятие (2 часа(ов)):

Обсуждение учебного проекта по группам студентов. Выбор проекта.

Тема 2. Жизненный цикл программного обеспечения.

лекционное занятие (4 часа(ов)):

Проблемы разработки сложного программного обеспечения (ПО). Общие принципы программной инженерии: абстракция и уточнение, модульность, повторное использование. Жизненный цикл ПО и процессы его разработки. Основные модели жизненного цикла: водопадная, итеративная, спиральная. Стандарты программной инженерии.

практическое занятие (8 часа(ов)):

Стандарты управления проектами. Обсуждение. Работа в команде.

Тема 3. Обзор методологий проектирования программных продуктов.

лекционное занятие (4 часа(ов)):

Каскадные и итеративные технологии. Критичность и масштабность программных проектов. Общие принципы гибких методологий разработки. Возможности и преимущества гибких методологий. Инженерные практики. Наиболее распространенные виды agile-методик. Методы внедрения agile. Технология экстремального программирования. SCRUM технология. Преимущества и недостатки технологий быстрой разработки программного обеспечения. Организация коллективной работы над проектом при использовании технологий быстрой разработки.

практическое занятие (8 часа(ов)):

Обсуждение agile-методик. Выполнение учебных проектов по командам.

Тема 4. Основные процессы программной инженерии.

лекционное занятие (4 часа(ов)):

Обзор процессов разработки программного обеспечения. Управление требованиями к программному обеспечению. Концептуальное и детальное проектирование программного обеспечения. Тестирование программного обеспечения. Сопровождение программного обеспечения. Конфигурационное управление. Документирование программного обеспечения.

практическое занятие (8 часа(ов)):

Обсуждение темы. Документирование учебного проекта.

Тема 5. Внедрение и сопровождение программных продуктов.

лекционное занятие (2 часа(ов)):

Планирование процесса внедрения программного продукта. Основные задачи, решаемые на этапе внедрения. Процесс устранения ошибок на этапе внедрения. Документирование программного обеспечения. Техническая поддержка пользователей на этапе сопровождения. Эволюция программного обеспечения. Наследуемые системы. Сопровождение программного обеспечения. Эволюция системной архитектуры. Реинжениринг программного обеспечения. Управление конфигурациями, изменениями, версиями и выпусками.

практическое занятие (8 часа(ов)):

Обсуждение темы. Версионность учебного проекта, документирование процесса..

Тема 6. Экономика программной инженерии.

лекционное занятие (2 часа(ов)):

Введение в экономику программной инженерии. Понятие экономики разработки программного обеспечения. Минимально жизнеспособный продукт (minimum viable product, MVP). Экономическая эффективность программного продукта. Факторы, влияющие на стоимость разработки программного обеспечения. Технологическая идея, программный код, программный продукт, софтверная компания.

практическое занятие (2 часа(ов)):

Обсуждение темы. Экономическое обоснование проекта.

Тема 7. Методологии ведения проектов.

лекционное занятие (9 часа(ов)):

Роли в проекте. Фазы проекта. Область деятельности, в ходе которой определяются и достигаются чёткие цели проекта при балансировании между объёмом работ, ресурсами (такими как деньги, труд, материалы, энергия, пространство и другими), временем, качеством и рисками. Ключевым фактором успеха проектного управления является наличие чёткого заранее определённого плана, минимизации рисков и отклонений от плана, эффективного управления изменениями (в отличие от процессного, функционального управления, управления уровнем услуг). Продуктами проекта могут быть продукция предприятия или организации (результаты научных и маркетинговых исследований, проектно-конструкторская и технологическая документация на новое изделие, разработанные для заказчика) и решение разных внутренних производственных задач (например, повышение качества продукции и эффективности организации труда, оптимизация финансовых потоков). Управление проектами является частью системы менеджмента предприятия. Альтернативные стандарты и школы иногда вкладывают в понятие управления проектами более широкий или более специфический смысл.

практическое занятие (9 часа(ов)):

Обсуждение темы. Разбор учебных кейсов.

Тема 8. Проектная документация. Требования к проекту.

лекционное занятие (9 часа(ов)):

Проектная документация. Требования к проекту. Существуют три основные характеристики, позволяющие количественно оценить полезность любого проекта для предприятия в целом (если проект не выполняется ради соблюдения установленных законом и иных обязательных требований к организации): - производительность, стоимость продукции и услуг, поставленных потребителям, за вычетом прямых расходов на приобретение товаров и услуг у сторонних поставщиков, за определенный период времени; - объем инвестиций, - все капитальные вложения и вложения средств в запасы на всех уровнях. В них входят любые затраты, срок амортизации которых превышает один финансовый год; - текущие расходы, - любые средства, расходуемые организацией для преобразования инвестиций в готовый продукт. Следовательно, любой проект, полезный для организации, должен отвечать хотя бы одному из следующих требований: - содействовать повышению производительности организации; - способствовать сокращению объемов инвестиций; - содействовать сокращению текущих расходов; - комплексно влиять на все три характеристики, обеспечивая заметное улучшение текущих и будущих основных показателей организации.

практическое занятие (9 часа(ов)):

Обсуждение темы. Разбор учебных кейсов.

Тема 9. Принципы архитектуры и шаблоны проектирования.

лекционное занятие (9 часа(ов)):

Принципы архитектуры и шаблоны проектирования. Шаблон проектирования или паттерн (англ. design pattern) в разработке программного обеспечения - повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться. Низкоуровневые шаблоны, учитывающие специфику конкретного языка программирования, называются идиомами. Это хорошие решения проектирования, характерные для конкретного языка или программной платформы, и потому не универсальные. На наивысшем уровне существуют архитектурные шаблоны, они охватывают собой архитектуру всей программной системы. Алгоритмы по своей сути также являются шаблонами, но не проектирования, а вычисления, так как решают вычислительные задачи.

практическое занятие (9 часа(ов)):

Обсуждение темы. Разбор учебных кейсов.

Тема 10. Стандарт UML.

лекционное занятие (9 часа(ов)):

UML (англ. Unified Modeling Language - унифицированный язык моделирования) - язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML является языком широкого профиля, это ? открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

практическое занятие (9 часа(ов)):

Обсуждение темы. Разбор учебных кейсов.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Введение. Предмет программной инженерии.	5		Выполнение лабораторной работы	8	Лабораторные работы
2.	Тема 2. Жизненный цикл программного обеспечения.	5		Выполнение лабораторной работы	8	Лабораторные работы
3.	Тема 3. Обзор методологий проектирования программных продуктов.	5		Выполнение лабораторной работы	10	Лабораторные работы
4.	Тема 4. Основные процессы программной инженерии.	5		Выполнение лабораторной работы	10	Лабораторные работы
5.	Тема 5. Внедрение и сопровождение программных продуктов.	5		Выполнение лабораторной работы	10	Лабораторные работы
6.	Тема 6. Экономика программной инженерии.	5		Выполнение лабораторной работы	8	Лабораторные работы
7.	Тема 7. Методологии ведения проектов.	6		подготовка к письменной работе	9	Письменная работа
8.	Тема 8. Проектная документация. Требования к проекту.	6		подготовка к письменной работе	9	Письменная работа

N	Раздел дисциплины	Се-местр	Неде-ля семе стра	Виды самостоятельной работы студентов	Трудо-емкость (в часах)	Формы контроля самосто-ятельной работы
9.	Тема 9. Принципы архитектуры и шаблоны проектирования.	6		подготовка к письменной работе	9	Письмен-ная работа
10.	Тема 10. Стандарт UML.	6		подготовка к контрольной работе	9	Контроль-ная работа
	Итого				90	

5. Образовательные технологии, включая интерактивные формы обучения

Обсуждение проекта в командах. Командная работа. Проектная деятельность

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

Тема 1. Введение. Предмет программной инженерии.

Лабо- ратор- ные ра- боты, примерные вопросы:

Разработка технического задания. 1. Подготовить описание предметной области - разработать краткое введение в предметную область; - выделить основные элементы предметной области и их взаимосвязи; - определить особенности и ограничения предметной области; - подготовить словарь используемых терминов и сокращений.

Тема 2. Жизненный цикл программного обеспечения.

Лабо- ратор- ные ра- боты, примерные вопросы:

2. Сформулировать цель создания системы - сформулировать цель создания системы ? как ответ на вопрос что за процесс в предметной области будет автоматизирован; - определить назначение системы, классифицировать существующие аналоги; - определить целевую аудиторию и ожидаемый уровень использования системы.

Тема 3. Обзор методологий проектирования программных продуктов.

Лабо- ратор- ные ра- боты, примерные вопросы:

3. Осуществить детализацию функций системы - осуществить изучение потребностей заказчика; - выделить категории пользователей; - определить функциональные требования для пользователей каждой категории; - подготовить описание функций системы.

Тема 4. Основные процессы программной инженерии.

Лабо- ратор- ные ра- боты, примерные вопросы:

4. Произвести анализ ограничений - провести анализ аппаратных особенностей и ограничений; - провести анализ топологии и особенностей развертывания; - определить технологические ограничения.

Тема 5. Внедрение и сопровождение программных продуктов.

Лабо- ратор- ные ра- боты, примерные вопросы:

Подготовить календарный план (документ А) - осуществить оценку сложности реализации подсистем; - разбить проект на работы, построить сетевой график; - осуществить оценку сроков и стоимости выполнения работ.

Тема 6. Экономика программной инженерии.

Лабо- ратор- ные ра- боты, примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

6. Выработать архитектурное решение - обосновать выбор технологической платформы; - если система должна реализовывать специфическую бизнес-логику, в которой обычно хорошо разбирается заказчик и плохо - исполнитель, эта логика должна быть задокументирована в техническом задании максимально подробно; - подготовить модульную структуру системы; - подготовить детализированное описание подсистем.

Тема 7. Методологии ведения проектов.

Письменная работа , примерные вопросы:

Необходимо создать описание архитектурных принципов для Вашего проекта. Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться. "Низкоуровневые" шаблоны, учитывающие специфику конкретного языка программирования, называются идиомами. Это хорошие решения проектирования, характерные для конкретного языка или программной платформы, и потому не универсальные. На наивысшем уровне существуют архитектурные шаблоны, они охватывают собой архитектуру всей программной системы. Алгоритмы по своей сути также являются шаблонами, но не проектирования, а вычисления, так как решают вычислительные задачи. В сравнении с полностью самостоятельным проектированием, шаблоны обладают рядом преимуществ. Основная польза от использования шаблонов состоит в снижении сложности разработки за счёт готовых абстракций для решения целого класса проблем. Шаблон даёт решению своё имя, что облегчает коммуникацию между разработчиками, позволяя ссылаться на известные шаблоны. Таким образом, за счёт шаблонов производится унификация деталей решений: модулей, элементов проекта, ? снижается количество ошибок. Применение шаблонов концептуально сродни использованию готовых библиотек кода. Правильно сформулированный шаблон проектирования позволяет, отыскав удачное решение, пользоваться им снова и снова. Набор шаблонов помогает разработчику выбрать возможный, наиболее подходящий вариант проектирования Выход Принципы архитектуры; Принципы реализации.

Тема 8. Проектная документация. Требования к проекту.

Письменная работа , примерные вопросы:

Сформировать совокупный список требований к системе - если система предполагает интерактивность в общении с пользователем, то определить функциональные требования (описывают в динамике сценарии взаимодействия посетителя с системой) и структуру данных; - выделить специфические требования (например, многоязычность, требования к дизайну экранов оператора); - сформировать прочие требования (например, какая документация должна быть предоставлена разработчиком); - сформировать итоговый список требований.

Тема 9. Принципы архитектуры и шаблоны проектирования.

Письменная работа , примерные вопросы:

Задача. Создать требования для вашей системы. Необходимо создать описание вашей системы в Use Case. Требования к программному обеспечению- совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе разработки требований к программному обеспечению, в результате анализа требований. Требования могут выражаться в виде текстовых утверждений и графических моделей. В классическом техническом подходе совокупность требований используется на стадии проектирования программного обеспечения (ПО). Требования также используются в процессе проверки ПО, так как тесты основываются на определённых требованиях. Этапу разработки требований может предшествовать технико-экономическое обоснование или концептуальная фаза анализа проекта. Фаза разработки требований может быть разбита на выявление требований (сбор, понимание, рассмотрение и выяснение потребностей заинтересованных лиц), анализ (проверка целостности и законченности), спецификация (документирование требований) и проверка правильности. На выходе: 1) Диаграмма Use cases (см ОбъединяющийUseCase), содержащая не менее 5-ти UC 2) Каждый UseCase с диаграммы должен иметь описание (см описание одного Use Case) 3) Для каждого Use Case, сделанного в прошлом задании нужно составить Activity Diagram.

Тема 10. Стандарт UML.

Контрольная работа , примерные вопросы:

Задача: Создать HLD вашего проекта. High-Level Design описывает предлагаемое решение на уровне архитектуры, не вдаваясь в подробности спецификаций, конфигураций и т.п. Верно будет дать определение HLD как концепту. HLD входит в 'стандартный' набор документации, но зачем же нужно тратить время и деньги на документирование того что 'впринципе и так понятно'. В ходе подготовки и согласования HLD Заказчик приводит видение Исполнителя в соответствие своим потребностям. Исполнитель, в свою очередь, уточняет объем, сроки и стоимость работ - эта информация необходима для управления проектом. Таким образом, наличие 'правильного' HLD снижает риски проекта для обеих сторон. Можно использовать https://www.draw.io или visual-paradigm.com Что должно быть как результат: Диаграмма компонентов+Описание; Диаграмма последовательности на каждый созданный Use Case (не менее 5-ти) в рамках компонентов; Диаграмма БД+Описание.

Итоговая форма контроля

зачет (в 5 семестре)

Итоговая форма контроля

экзамен (в 6 семестре)

Примерные вопросы к итоговой форме контроля

Вопросы к зачету

1. Раскрыть варианты использования. Актёров (интерфейсов) не менее 3-х. Логика функционирования программного средства должна быть полностью описана. В результате должен быть сформирован список актёров (интерфейсов) и их функций (вариантов использования) в виде уровневого списка или таблицы.

2. Сформировать диаграмму вариантов использования, полностью описывающую функциональное назначение программного средства согласно варианту, назначенному преподавателем.

3. Сформировать диаграмму кооперации.

4. Разработать диаграмму классов. Классы в диаграмме должны иметь не менее 2-х видов ассоциаций. Все отношения должны быть поименованы. Должны присутствовать классы с атрибутами и классы с операциями. Один или более классов должны быть представлены в виде объектов.

5. Сформировать диаграмму деятельности с дорожками для самого сложного варианта использования.

6. Сформировать диаграмму последовательности для варианта использования, для которого не строилась диаграмма деятельности.

7. Построить диаграмму деятельности

8. Построить диаграмму состояний

9. Разработать диаграмму компонентов программного средства.

10. Разработать диаграммы развертывания программного средства.

Вопросы к экзамену:

1. Сопоставьте фазы жизненного цикла и их порядок. 1 - самая первая фаза.

2. Абстракция данных и иерархия

Соотнесите принцип проектирования и его описание.

Пусть q(x) является свойством, верным относительно объектов x некоторого типа T. Тогда q(y) также должно быть верным для объектов y типа S, где S является подтипом типа T.

3. Пожалуйста выберите Гибкие методологии разработки ПО

4. Что из перечисленного ниже фазами жизненного ПО не являются?

Vision

Requirements specification

End of live (Вывод из эксплуатации)

Sale (Продажа)
Deployment
Maintenance
Support

5.Какие принципы НЕ характеры для Waterfall.
Team of equals
Team responsibility
Requirements are not set in stone
Customers are team members
Preliminary Design, Detailed Design, Coding and Unit Testing, Integration, and Testing

6. UML diagrams
Какие типы UML диаграмм не включены в UML 2.0?
Collaboration
Deployment diagram
Profile diagram
Position diagram
Timing diagram

7. При agile-подходе часто пренебрегают созданием плана (?дорожной карты?) развития продукта, равно как и управлением требованиями, в процессе которого и формируется такая ?карта?.

8. У вас есть компания по производству продукта по защите безопасности. Является ли свойство устойчивости вашего продукта к DDoS атакам функциональным требованием?

9. Расскажите про Activity diagram

10. Расскажите про Sequence diagram

7.1. Основная литература:

Трояновский Владимир Михайлович Программная инженерия информационно-управляющих систем в свете прикладной теории случайных процессов : учеб. пособие / В.М. Трояновский. - М. : ИНФРА-М, 2018. - 325 с. Режим доступа: http://znanium.com/bookread2.php?book=941137

Исаченко Олег Вячеславович Программное обеспечение компьютерных сетей: Учебное пособие / О.В. Исаченко. - М.: НИЦ ИНФРА-М, 2014. - 117 с. Режим доступа:

http://znanium.com/bookread2.php?book=424039

Хорев Павел Борисович Программно-аппаратная защита информации: Учебное пособие / Хорев П.Б., - 2-е изд., испр. и доп. - М.:Форум, НИЦ ИНФРА-М, 2015. - 352 с. Режим доступа: http://znanium.com/bookread2.php?book=489084

7.2. Дополнительная литература:

Бубнов Алексей Алексеевич Введение в программную инженерию : Учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. - М.: КУРС: ИНФРА-М, 2017. - 336 с. Режим доступа: http://znanium.com/bookread2.php?book=850951

Медведев М. А. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017. - 64 с. Режим доступа: http://znanium.com/bookread2.php?book=948428

Гагарина Лариса Геннадьевна Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учеб. пос. / Т.И. Немцова и др.; Под ред. Л.Г. Гагариной - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015 - 496 с. Режим доступа: http://znanium.com/bookread2.php?book=472870

7.3. Интернет-ресурсы:

Сайт Software Engineering Institute в составе Carnegie Mellon University, содержащий информацию по предмету Software Engineering - http://www.sei.cmu.edu/

Сайт международного института бизнес-анализа, содержащий материалы по бизнес анализу - http://iiba.org/

Сайт российского представительства Project Management Institute с информацией о стандарте управления проектами по PMBOK - http://www.pmi.ru

Сайт, содержащий информацию по тестированию программного обеспечения - http://software-testing.ru/

Сообщество, в рамках которого обсуждаются любые вопросы, связанные с разработкой ПО - http://www.infoq.com/

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Программная инженерия" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB,audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудованием имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен студентам. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, УМК, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) нового поколения.

-

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 09.03.03 "Прикладная информатика" и профилю подготовки не предусмотрено .

Автор(ы):
Кугуракова В.В. _____
"__" _____ 201 __ г.

Рецензент(ы):
Таланов М.О. _____
"__" _____ 201 __ г.