Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 23.05.2024 11:08:47
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____/ Д.Г. Демидов /
« 15 » февраля 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программная инженерия»

Направление подготовки/специальность
09.03.01 Информатика и вычислительная техника

Профиль/специализация
Программное обеспечение информационных систем

Квалификация
бакалавр

Формы обучения
заочная

Москва, 2024 г.

Разработчик(и):

К.т.н., доцент. / В.С. Ноздрин /

Согласовано:
Заведующий кафедрой «Прикладной информатики»,
К.э.н, доцент / С.В. Суворов /

Содержание

Оглавление

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 7
3.5 Тематика курсовых проектов (курсовых работ) 7
4 Учебно-методическое и информационное обеспечение 7
4.1 Нормативные документы и ГОСТы 7
4.2 Основная литература 7
4.3 Дополнительная литература 7
4.4 Электронные образовательные ресурсы 7
4.5 Лицензионное и свободно распространяемое программное обеспечение 8
4.6 Современные профессиональные базы данных и информационные справочные системы 8
5 Материально-техническое обеспечение 8
5.1 Требования к оборудованию и помещению для занятий 8
5.2 Требования к программному обеспечению 8
6 Методические рекомендации 8
6.1 Методические рекомендации для преподавателя по организации обучения 8
6.2 Методические указания для обучающихся по освоению дисциплины 9
7 Фонд оценочных средств 9
7.1 Методы контроля и оценивания результатов обучения 9
7.2 Шкала и критерии оценивания результатов обучения 9
7.3 Оценочные средства 13

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цели дисциплины:

– 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.

Задачи дисциплины

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

Планируемые результаты обучения

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

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

Обучение по дисциплине «Программная инженерия» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ПК-4. Способен разрабатывать алгоритмы решения поставленных задач в соответствии с требованиями.	ПК-4.1. Знать: Архитектура, устройство и принцип функционирования вычислительных систем. Основы современных систем управления базами данных. Теория баз данных. Современные принципы построения интерфейсов пользователя. Современные методики тестирования эргономики пользовательских интерфейсов. Современные стандарты взаимодействия компонентов распределенных приложений. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ИР. Методы и средства проектирования баз данных. Методы и средства проектирования программных интерфейсов. ПК-4.2. Уметь: Вырабатывать варианты реализации требований. Производить оценку и обоснование рекомендуемых решений. Использовать программные продукты для графического отображения алгоритмов. Применять методы и средства проектирования ИР, структур данных, баз данных, программных интерфейсов.

	ПК-4.3. Владеть: Составление формализованных описаний решений поставленных задач в соответствии с требованиями принятых в организации нормативных документов. Разработка алгоритмов решения поставленных задач в соответствии с требованиями принятых в организации нормативных документов. Проектирование структур данных. Проектирование баз данных. Проектирование интерфейсов.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1 «Дисциплины (модули) Специальная подготовка».

Дисциплина «Программная инженерия» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

- Объектно-ориентированное программирование;
- Теория вычислительных процессов и языков программирования;
- Структуры и алгоритмы обработки данных.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет _6_ зачетных(е) единиц(ы) (_216_ часов).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.3 Заочная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			8	
1	**Аудиторные занятия**	24	24	
	В том числе:			
1.1	Лекции	8	8	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	16	16	
2	**Самостоятельная работа**	192	192	
	В том числе:			
2.1	СРС			
2.2	Курсовая работа	192	192	
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен	экзамен	экзамен	
	Итого:	216	216	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.2 Заочная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Раздел 1.						
1.1	Тема 1. Введение. Программная инженерия	54	2		4		48
1.2	Тема 2. Инженерия качества	54	2		4		48
1.3	Тема 3. Программометрика	54	2		4		48
1.4	Тема 4. Тестирование программных систем	54	2		4		48
Итого		**216**	**8**		**16**		**192**

3.3 Содержание дисциплины

Раздел 1.

Тема 1. Введение. Программная инженерия

Ее цели и задачи. Программная документация, виды программных документов. Требования к ней. Стандарты документирования. ЕСПД. ГОСТ. Техническое задание на программный продукт.

Тема 2. Инженерия качества

Классификация моделей надежности. Данные для оценивания надежности программных систем. Качество программного обеспечения. Характеристики качества: функциональность, надежность, практичность, эффективность, сопровождаемость, мобильность.

Тема 3. Программометрика

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.

Тема 4. 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. Понятие «белого» и «черного» ящика при тестировании. Тестирование производительности, надежности, безопасности, интеграции, стресс-тестирование.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

3.4.2 Лабораторные занятия

Лабораторная работа №1 «Стандарты документирования».

Лабораторная работа №2 «Стандарты жизненного цикла в программной инженерии и их назначение».

Лабораторная работа №3 «Технико-экономическое обоснование проекта программной системы».

Лабораторная работа №4 «Выявление, анализ и документирование требований к программной системе».

Лабораторная работа №5 «Стандарты качества программного обеспечения».

Лабораторная работа №6 «Методы и средства отладки программных систем».

Лабораторная работа №7 «Методы тестирования программных систем».

Лабораторная работа №8 «Управление программными дефектами».

3.5 Тематика курсовых проектов (курсовых работ)

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

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. ФГОС 09.03.01 Информатика и вычислительная техника Приказ Минобрнауки России от 19.09.2017 N 929 (ред. от 08.02.2021) https://fgos.ru/fgos/fgos-09-03-01-informatika-i-vychislitelnaya-tehnika-929

4.2 Основная литература

1. Современные технологии разработки программного обеспечения : учебно-методическое пособие / составитель Н. А. Федькова. — Брянск : Брянский ГАУ, 2022. — 58 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/305087 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.
2. Программная инженерия: курс лекций. Часть III: Направление подготовки 09.03.03 Прикладная информатика; Профиль подготовки «Прикладная информатика в экономике». Квалификация выпускника – бакалавр Северо-Кавказский федеральный университет 130 стр., 20220 год https://reader.lanbook.com/book/306992#63
.

4.3 Дополнительная литература

1. Шуваев, А. В. Программная инженерия : учебное пособие / А. В. Шуваев. — Ставрополь : СтГАУ, 2023. — 84 с. — Текст : электронный // Лань :

электронно-библиотечная система. — URL: https://e.lanbook.com/book/245891 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.
2. Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 196 с. — ISBN 978-5-8114-9323-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/189470 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.

4.4 Электронные образовательные ресурсы

https://online.mospolytech.ru/enrol/index.php?id=12024 Программная инженерия

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Операционная система, Windows 11 (или ниже) - Microsoft Open License
2. Офисные приложения, Microsoft Office 2013(или ниже) - Microsoft Open License

4.6 Современные профессиональные базы данных и информационные справочные системы

1.не предусмотрено

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

5.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

Microsoft Windows.
Веб-браузер, Chrome.
ПО, предоставленное преподавателем.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

самоконтроль и самооценка студента;

контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

уровень освоения студентом учебного материала;

умения студента использовать теоретические знания при выполнении практических задач;

сформированность компетенций;

оформление материала в соответствии с требованиями..

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

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

Лабораторные работы, экзамен.

7.2 Шкала и критерии оценивания результатов обучения

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

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ПК-4. Способен разрабатывать алгоритмы решения поставленных задач в соответствии с требованиями.				
ПК-4.1. Знать: Архитектура,	Обучающийся демонстрируе	Обучающийся демонстрирует	Обучающийся демонстрирует	Обучающийся демонстрирует

устройство и принцип функционирования вычислительных систем. Основы современных систем управления базами данных. Теория баз данных. Современные принципы построения интерфейсов пользователя. Современные методики тестирования эргономики пользовательских интерфейсов. Современные стандарты взаимодействия компонентов распределенных приложений. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ИР. Методы и средства проектирования баз данных. Методы и средства проектирования программных интерфейсов. ПК-4.2. Уметь: Вырабатывать варианты реализации требований. Производить оценку и обоснование	т полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

рекомендуемых решений. Использовать программные продукты для графического отображения алгоритмов. Применять методы и средства проектирования ИР, структур данных, баз данных, программных интерфейсов. ПК-4.3. Владеть: Составление формализованных описаний решений поставленных задач в соответствии с требованиями принятых в организации нормативных документов. Разработка алгоритмов решения поставленных задач в соответствии с требованиями принятых в организации нормативных документов. Проектирование структур данных. Проектирование баз данных. Проектирование интерфейсов.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

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

Шкала оценивания	**Описание**
Отлично	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при переносе знаний и умений на новые, нестандартные задачи.
Хорошо	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. В то же время при ответе допускает несущественные погрешности, задачи решает с недочетами, не влияющими на общий ход решения.
Удовлетворительно	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. Но показывает неглубокие знания, при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, в решении задач могут содержаться грубые ошибки. Для получения правильного ответа требуются уточняющие вопросы.

Неудовлетворительно	Не выполнены обязательные **условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины, ИЛИ студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями.

7.3 Оценочные средства

7.3.1 Текущий контроль

Подготовка и защита лабораторных работ.

Защита курсового проекта.

7.3.2 Промежуточная аттестация

Список вопросов на экзамен

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.