Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35e9d50310dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А. В. Фомина
10 февраля 2022 г.

Рабочая программа дисциплины

Б1.О.14 Метрология и качество программного обеспечения

Направление подготовки
02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) подготовки
Программное и математическое обеспечение информационных технологий

Программа бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
Очная

Год набора 2021

Новокузнецк 2022

Оглавление

1 Цель дисциплины. 3
1.1 Формируемые компетенции 3
1.2 Индикаторы достижения компетенций 3
1.3 Знания, умения, навыки (ЗУВ) по дисциплине 3
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации. 4
3. Учебно-тематический план и содержание дисциплины. 4
3.1 Учебно-тематический план 4
3.2. Содержание занятий по видам учебной работы 5
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации. 8
5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины. 9
5.1 Учебная литература 9
5.2 Материально-техническое и программное обеспечение дисциплины. 9
5.3 Современные профессиональные базы данных и информационные справочные системы. 10
6 Иные сведения и (или) материалы. 10
6.1. Темы письменных учебных работ 11
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации 11

1 Цель дисциплины.

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП):

ОПК-2.

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1 и 2.

1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

Наименование вида компетенции *(универсальная, общепрофессиональная, профессиональная)*	Наименование категории (группы) компетенций	Код и название компетенции
общепрофессиональная	Теоретические и практические основы профессиональной деятельности	ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.

1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.	2.1 Решает задачу количественной оценки качества программного обеспечения 2.2 Применяет методы проектирования, разработки, и реализации программных продуктов 2.3 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	Б1.О.05 Дискретная математика Б1.О.08 Математические методы и программное обеспечение защиты информации Б1.О.11 Компьютерная графика Б1.О.14 Метрология и качество программного обеспечения Б1.О.19 Базы данных Б2.О.02(П) Технологическая (проектно-технологическая) практика Б3.01(Д) Подготовка к процедуре защиты и защита выпускной квалификационной работы

1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ОПК-2 Способен применять современный	2.1 Решает задачу количественной оценки	**Знать** – критерии качества программного обеспечения;

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.	качества программного обеспечения 2.3 Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения	**Уметь**: – решать задачу количественной оценки качества программного обеспечения **Владеть** – современным математическим аппаратом, инструментальными программными и аппаратными средствами измерений для оценки качества программного обеспечения.

2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 4 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения		
	ОФО	ОЗФО	ЗФО
1 Общая трудоемкость дисциплины	144		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	68		
Аудиторная работа (всего):	68		
в том числе:			
лекции	20		
практические занятия, семинары	48		
в интерактивной форме			
в электронной форме			
3 Самостоятельная работа обучающихся (всего)	76		
4 Промежуточная аттестация обучающегося	Зачет с оценкой – 8 семестр		

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план очной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость *(всего час.)*	Трудоемкость занятий (час.)			Формы текущего контроля и промежуточной аттестации успеваемости
			Аудиторн. занятия		СРС	
			лекц.	практ.		
Семестр 8						
1-2	Основные понятия и характеристики качества	18	*2*	*4*	*12*	Устный опрос

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость *(всего час.)*	Трудоемкость занятий (час.)			Формы текущего контроля и промежуточной аттестации успеваемости
			Аудиторн. занятия		СРС	
			лекц.	практ.		
	программного обеспечения					
3-5	Основы метрической теории программ	78	12	24	42	Индивидуальные задания
6-7	Стандартизация и сертификация программного обеспечения	30	4	14	12	Индивидуальные задания Доклад.
10-11	Концепция и модели управления качеством программного обеспечения	*18*	*2*	*6*	*10*	Доклад
	Промежуточная аттестация зачет с оценкой					зачет с оценкой
ИТОГО		**144**	**20**	**48**	**76**	**-**

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
Содержание лекционного курса		
1	**Основные понятия и характеристики качества программного обеспечения**	
1.1	Основные понятия и характеристики качества программного обеспечения	Понятие качества программного обеспечения. Дестабилизирующие факторы программного обеспечения. Метрики качества программных средств. Факторы, влияющие на качество программных средств. Внутреннее и внешнее качество. Качество в использовании. Особенности измерения и оценивания характеристик качества. Характеристика процесса измерений. Измерительные шкалы. Категории показателей.
2 Основы метрической теории программ		
2.1.	Алгоритмическая сложность. Метрики, основанные на лексическом анализе программ.	Общие сведения о программометрике. Понятие алгоритмической сложности. Свойства алгоритмической сложности. Метрики, основанные на лексическом анализе программ: Метрики Холстеда (вероятностная модель текста программы, измеряемые свойства программ, математическое ожидание длины текста программы, метрические характеристики программ, оптимизация количества и длины модулей в программе, количественная оценка работы программирования, оценка уровня языков программирования, метрика числа ошибок в программе, порядок расчета метрических характеристик программных средств)
2.2.	Метрики, основанные на лексическом анализе программ.	Метрики Джилба и Чепина.
2.3.	Метрики структурной сложности программ	Понятие структурной сложности программ. Критерии выделения маршрутов. Метрика Маккейба.
2.4.	Процедурно-	Процедурно-ориентированные метрики: метрики на

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	ориентированные метрики программных средств	основе функциональных указателей; метрики связности модулей; метрики сцепления модулей.
2.5.	Объектно-ориентированные метрики программных средств	Объектно-ориентированные метрики: общие характеристики объектно-ориентированных компонентов; набор метрик Мартина; набор метрик Чидамбера и Кемерера; комплексный набор метрик Лоренца и Кидда; набор метрик Абреу.
2.6.	Модели надежности	Модели надежности программных средств: прогнозные модели; оценочные модели (модели надежности Джелински-Моранды; статистическая модель Миллса; эвристическая модель); измерительные модели (модель Нельсона, модель Мусы)
3	**Стандартизация и сертификация программного обеспечения**	
3.1.	Стандартизация программного обеспечения	Роль стандартизации в управлении качеством. Виды стандартов обеспечения качества: основополагающий стандарт; стандарт на продукцию; стандарт на процесс; стандарт на методы контроля; стандарт на услугу, терминологический стандарт. Принципы стандартизации. Стандарты разработки информационных систем. Стандарты ISO. ГОСТ серии 24: требования к содержанию основополагающих документов, разрабатываемых на начальных стадиях подготовки информационных систем к проектированию. ГОСТ серии 34: методологическая основа создания информационных систем. Стандарты разработки программного обеспечения. Проблемы стандартизации в современных условиях
3.2.	Сертификация программного обеспечения	Назначение и цели сертификации. Правовое обеспечение сертификации. Метрологическая экспертиза ПО. Сертификация, ее объекты и цели. Законы, регламентирующие процедуру сертификации в России. Содержание процедуры сертификации. Схемы сертификации третьей стороной. Методики, позволяющие оценить показатели адекватности функционирования; надежности и своевременности представления информации; полноты, безошибочности, актуальности, защищенности от несанкционированного доступа и компьютерной вирусной инфекции, конфиденциальности информации.
4	**Концепция и модели управления качеством программного обеспечения**	
4.1.	Концепция управления качеством. Модели управления качеством. Организационно-технологические аспекты управления качеством.	Предшественники современной концепции управления качеством. Современная модель управления качеством. Стандарт управления качеством. Японские модели управления качеством. Европейский подход к управлению качеством. Российский опыт управления качеством. Нравственно надежный персонал. Управление качеством на этапах разработки. Организация труда. Модель совершенствования потенциальных возможностей: Capability Maturity Model; SPICE.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
Содержание практических занятий		
1	**Основные понятия и характеристики качества программного обеспечения**	
1.1	Основные понятия и характеристики качества программного обеспечения	Вводное занятие: Основные понятия и характеристики качества программного обеспечения
1.2.		Изучение нормативных документов, устанавливающих основные понятия и характеристики качества программного обеспечения
2 Основы метрической теории программ		
2.1.	Оценка ПС на основе метрик Холстеда	Формирование словаря программы, оптимизация количества и длины модулей в программе, оценка уровня языка программирования, оценка числа ошибок в программе.
2.2.		
2.3.	Оценка характеристик ПС на основе метрик Джилба.	Расчет абсолютной и относительной сложности ПС. Определение числа связей между модулями.
2.4.	Оценка характеристик ПС на основе метрик Чепина.	Расчет метрики Чепина для ПС. Оценка уровня сложности ПС на основе значения метрики Чепина.
2.5.	Оценка структурной сложности программ	Оценка сложности ПС на основе критериев выделения маршрутов.
2.6.		Оценка алгоритмической сложности на основе управляющих графов.
2.7	Оценка характеристик программ на основе процедурно-ориентированных метрик	Оценка уровня качества процедурно-ориентированных ПС: расчет метрики дефектов качества, количества функциональных указателей, производительности, качества, удельной стоимости, документированности.
2.8.		Анализ типов связности модулей ПС. Определение уровня сцепления модулей.
2.9.	Оценка характеристик программ на основе объектно-ориентированных метрик	Расчет метрик Мартина: центростремительного сцепления, центробежного сцепления и нестабильности. Расчет меры абстрактности. Расчет нормализованного расстояния до главной последовательности. Использование метрик Чидамбера и Кемерера для измерения сложности класса. Расчет взвешенных методов на класс (WMC). Расчет количества откликов на класс (RFS). Расчет количества методов на класс (NM). Расчет связности между классами (CBO). Расчет метрики отсутствия сцепления в методах (LCOM).
2.10		Использование метрик Лоренца и Кидда. Вычисление размера класса (CS), количества операций, переопределяемых подклассом (NOO), количества операций, добавленных подклассом (NOA), индекса специализации (SI), среднего размера операции (AOS), сложности операции (OC), среднего количества параметров на операцию (ANP), количества описаний сценариев (NSS), количества ключевых классов (NKC), количества подсистем (NSUB).
2.11		Использование метрик Абреу. Вычисление фактора закрытости метода (MHF), фактора закрытости свойства (AHF); фактора наследования метода (MIF); фактора наследования свойства (AIF), фактора полиморфизма (POF), фактора сцепления (COF).
2.12	Оценка надежности ПС	Применение модели Джелински-Моранды: определение количества ошибок в программе до начала тестирования, определение количества ошибок в программе, не устраненных после проведения тестирования. Применение статистической модели Миллса: определение количества ошибок в программе до начала тестирования, оценка степени отлаженности

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		программы. Применение эвристической модели для определения количества ошибок в программе до начала тестирования. Применение модели Нельсона для оценки ПС.
3. Стандартизация и сертификация программного обеспечения		
3.1	Решение задач по определению показателей качества ПС.	Задача по оценке надежности ПС. Задача по оценке надежности и корректности. Задача по оценке эффективности и универсальности
3.2	Применение моделей А.И.Костогрызова и В.В. Липаева для оценки работы ПС.	Применение модели бесприоритетного обслуживания для оценки работы ПС. Применение модели обслуживания с относительными приоритетами для оценки ПС.
3.3.		Применение модели обслуживания с абсолютными приоритетами для оценки работы ПС.
3.4.	Стандарты разработки информационных систем и ПО	Анализ нормативных документов
3.5.	Сертификация ПС	Подготовка документов на сертификацию ПС.
4	**Концепция и модели управления качеством программного обеспечения**	
4.1.	Стандарты управления качеством	Анализ нормативных документов
	Промежуточная аттестация – *зачет с оценкой*	

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблице 7.

Таблица 7 - Шкала и показатели оценивания результатов учебной работы обучающихся по видам в балльно-рейтинговой системе (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	Баллы
Семестр 2				
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	**80**	Индивидуальные задания (отчет о выполнении индивидуального задания) (6 работ)	**За одно индивидуальное задание от 5 до:** **5 баллов** (выполнено 51 - 65% заданий) **7 балла** (выполнено 66 - 85% заданий) **10 баллов** (выполнено 86 - 100% заданий)	31- 60
		Работа на практическом занятии (2 занятия по разделу 1.)	**За одно занятие от 2 до:** **2 баллов** (выполнено 51 - 65% заданий)3 **3 балла** (выполнено 66 - 85% заданий) **5 баллов** (выполнено 86 - 100% заданий)	4-10
		Подготовка доклада (2 доклада)	**За один доклад от 2 до:** **2 баллов** (выполнено 51 - 65% заданий)3 **3 балла** (выполнено 66 - 85% заданий) **5 баллов** (выполнено 86 - 100% заданий)	4-10
Итого по текущей работе в семестре				41 - 80
Промежуточная аттестация (зачет	20	Теоретический вопрос	**4 балла** (пороговое значение) **8 баллов** (максимальное значение)	4 - 8

с оценкой)		Решение задачи 1.	**3 балла** (пороговое значение) **6 баллов** (максимальное значение)	3 – 6
		Решение задачи 2.	**3 баллов** (пороговое значение) **6 баллов** (максимальное значение)	3 - 6
Итого по промежуточной аттестации в семестре (зачету с оценкой)				10 – 20 б.
Суммарная оценка по дисциплине в семестре: Сумма баллов текущей и промежуточной аттестации 51 – 100 б.				

В промежуточной аттестации оценка выставляется в ведомость в 100-балльной шкале и в буквенном эквиваленте (таблица 8)

Таблица 8 – Соотнесение 100-балльной шкалы и буквенного эквивалента оценки

Сумма набранных баллов	Уровни освоения дисциплины и компетенций	*Экзамен*		*Зачет*
		Оценка	*Буквенный эквивалент*	*Буквенный эквивалент*
86 - 100	Продвинутый	5	отлично	Зачтено
66 - 85	Повышенный	4	хорошо	
51 - 65	Пороговый	3	удовлетворительно	
0 - 50	Первый	2	неудовлетворительно	Не зачтено

5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

1. Черников, Б. В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. - Москва : ИД ФОРУМ: ИНФРА-М, 2012. - 240 с.: ил.; . - (Высшее образование). ISBN 978-5-8199-0499-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/256901. – Режим доступа: по подписке.

2. Черников, Б. В. Оценка качества программного обеспечения: Практикум: Учебное пособие / Б.В. Черников, Б.Е. Поклонов; Под ред. Б.В. Черникова - Москва : ИД ФОРУМ: НИЦ Инфра-М, 2012. - 400 с.: ил.; . - (Высшее образование). ISBN 978-5-8199-0516-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/315269. – Режим доступа: по подписке.

Дополнительная учебная литература

1. Кайгородцев, Г. И. Введение в курс метрической теории и метрологии программ/КайгородцевГ.И. - Новосибирск : НГТУ, 2016. - 192 с.: ISBN 978-5-7782-1648-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/549419. – Режим доступа: по подписке.

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях КГПИ КемГУ:

100 Учебная аудитория (мультимедийная) для проведения: -занятий лекционного типа; - групповых и индивидуальных консультаций. **Специализированная (учебная) мебель:** доска меловая, кафедра, столы, стулья, рабочее место для обучающегося с ОВЗ.	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19

Оборудование:*стационарное* - компьютер преподавателя, экран моторизированный, проектор, усилитель звука, колонки, микрофон преподавателя. **Используемое программное обеспечение:** Ubuntu Linux (свободно распространяемое ПО), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). **Интернет с обеспечением доступа в ЭИОС.**	
508 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения: - занятий семинарского (практического) типа; - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. **Специализированная (учебная) мебель:** доска меловая, кафедра, столы, стулья. **Оборудование для презентации учебного материала:** *стационарное* - компьютер преподавателя, проектор, экран. **Оборудование:** *стационарное* – компьютеры для обучающихся (18 шт.). **Используемое программное обеспечение:** MS Windows (Microsoft Imagine Premium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). **Интернет с обеспечением доступа в ЭИОС.**	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19

5.3 Современные профессиональные базы данных и информационные справочные системы.

Перечень СПБД и ИСС по дисциплине

1. Информационная система «Общероссийский математический портал», режим доступа : http://www.mathnet.ru/
2. Информационная система «Экспонента» - центр инженерных технологий и моделирования, режим доступа :http://www.exponenta.ru
3. База данных Science Direct (более 1500 журналов издательства Elsevier, среди них издания по математике и информатике), режим доступа :https://www.sciencedirect.com
4. Информационная система «Единое окно доступа к информационным ресурсам» http://window.edu.ru/catalog/
5. Базы данных и аналитические публикации на портале «Университетская информационная система Россия», режим доступа: https://uisrussia.msu.ru/
6. Официальный сайт Федерального агентства по техническому регулированию и метрологии (РОССТАНДАРТ). - Режим доступа: http://www.gost.ru//

6 Иные сведения и (или) материалы.

6.1. Темы письменных учебных работ

Таблица 9 - Темы письменных учебных работ

Раздел	Темы	Контрольные точки
Основы метрической теории программ	Метрики Холстеда. Метрики Джилба. Метрики Чепина	Отчет по индивидуальному заданию №1
	Оценка структурной сложности программ	Отчет по индивидуальному заданию №2
	Оценка характеристик программ на основе процедурно-ориентированных метрик. Оценка характеристик программ на основе объектно-ориентированных метрик	Отчет по индивидуальному заданию №3
	Оценка надежности ПС.	Отчет по индивидуальному заданию №4
Стандартизация и сертификация программного обеспечения	Применение модели бесприоритетного обслуживания для оценки работы ПС. Применение модели обслуживания с относительными приоритетами для оценки ПС.	Отчет по индивидуальному заданию №5
	Применение модели обслуживания с абсолютными приоритетами для оценки работы ПС.	Отчет по индивидуальному заданию №6

6.2. Примерные вопросы и задания / задачи для промежуточной аттестации

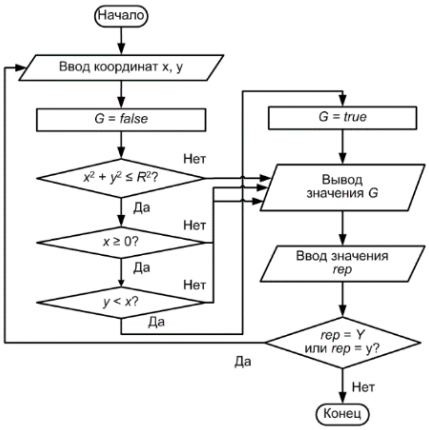
Таблица 10 - Типовые (примерные) контрольные вопросы и задания к зачету с оценкой

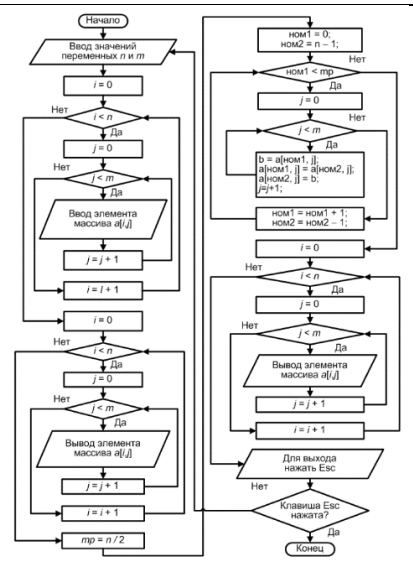
Разделы и темы	Примерные теоретические вопросы	Примерные практические задания и (или) задачи
Раздел 1. Основные понятия и характеристики качества программного обеспечения		
Тема 1.1 Основные понятия и характеристики качества программного обеспечения	1. Каковы основные понятия качества программного обеспечения? 2. Что представляют собой характеристики программы? 3. Как определить понятие системы обеспечения качества? 4. Какие объекты программных систем подвержены уязвимости? 5. Какова классификация дестабилизирующих факторов программного обеспечения? 6. Какие существуют внутренние	1. Определите понятие качества ПО, используя различные официальные источники. 2. К каким категориям относятся: качество исходного кода, объем программы, число простых (уникальных) операторов, число простых (уникальных) операндов, общее число операторов, общее число операндов, арифметическое выражение. 3. Определите измерительные шкалы для следующих показателей: использование моделей, структура программы, использование указателей, объем программы.

	источники угроз программного обеспечения? 7. Какие внешние факторы могут дестабилизировать работу программного обеспечения? 8. Какие общие факторы влияют па качество программного обеспечения? 9. Как определить понятия фактора качества, критерия качества, метрики? 10. Что представляют собой понятия оценочного элемента, показателя качества, базового значения показателя качества? 11. Какой смысл вкладывается в понятия сертификации, верификации и аттестации? 12. Как взаимно связаны факторы, влияющие на качество программного обеспечения? 13. Каковы функциональные и конструктивные характеристики качества программных средств? 14. Что такое внутреннее качество программных средств? 15. Как определить внешнее качество программных средств? 16. Что вкладывается в понятие качества при использовании программ-ных средств? 17. Как определить понятия системы измерений характеристик программного обеспечения, измерительной шкалы?	

	18. Каковы условия обеспечения объективности измерений? 19. Какие существуют категории групп показателей качества программных средств? 20. Как классифицируются типы измерительных шкал, какова их эмпирическая значимость?	
Раздел 2. Основы метрической теории программ		
Тема 2.1. Алгоритмическая сложность. Метрики, основанные на лексическом анализе программ.	1. Каковы задачи и область применения программометрики? 2. Что такое алгоритмическая сложность? Как определить ее верхнюю оценку? 3. Какими свойствами обладает алгоритмическая сложность? 4. Что устанавливает закон Ципфа? 5. Каковы измеряемые свойства программ? 6. Какие существуют классы несовершенств программных средств? 7. Что определяет соотношение Холстеда? 8. Какие элементы содержит словарь программы? 9. Как определить объем программы? 10. Что такое потенциальный объем программы? 11. Что характеризует уровень реализации программы? 12. Как оптимизировать количество и длину модулей в программе? 13. Как количественно оценить работу программирования?	1. Пусть общее число входных и выходных переменных 256. Используя соотношение Холстеда вычислите следующие характеристики: число модулей, число входных переменных одного модуля, длину модуля, объем модуля. 2. Пусть общее число входных и выходных переменных 256. Вычислите число иерархических уровней. 3. Пусть число простых операторов n1=28, число простых операндов n2=16, число всех операторов N1=116, число всех операндов N2=36, число входных и выходных параметров равно 10. Программа написана на языке С. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы. 4. Пусть число простых операторов n1=20, число простых операндов n2=12, число всех операторов N1=77, число всех операндов N2=47, число входных и выходных параметров равно 3. Программа написана на языке Ассемблер. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы. 5. Пусть число простых операторов n1=42, число простых операндов n2=23, число всех операторов N1=218, число всех

	14. Что устанавливает закон Хика? 15. Что определяет число Страуда? 16. Зачем используется коэффициент пересчета Кнута? 17. Как оценить уровень языка программирования? 18. Что устанавливает закон Миллера? 19. Каков порядок расчета метрических характеристик программных средств?	операндов N2=97, число входных и выходных параметров равно 13. Программа написана на языке Паскаль. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы. 6. Пусть число простых операторов n1=39, число простых операндов n2=24, число всех операторов N1=252, число всех операндов N2=95, число входных и выходных параметров равно 12. Программа написана на языке Фортран. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы. 7. Пусть число простых операторов n1=28, число простых операндов n2=27, число всех операторов N1=208, число всех операндов N2=93, число входных и выходных параметров равно 2. Программа написана на языке C#. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы.
Тема 2.2 Метрики, основанные на лексическом анализе программ	1. Какие метрики предлагаются Джилбом? 2. Какие показатели качества предлагает оценивать Джилб? 3. Как рассчитать начальную надежность программы? 4. Как оценить меру сложности программы по Чепину?	1. Пусть всего операторов и операций в программе L=263, условных операторов Lif=2, операторов цикла Lloop=4, число используемых модулей Lmod=4, число связей между модулями Nsv=3. Рассчитайте абсолютную и относительную сложность программы, относительное число связей, используя метрики Джилба. 2. Пусть всего операторов и операций в программе L=404, условных операторов Lif=1, операторов цикла Lloop=19. Программа состоит из одного модуля. Рассчитайте абсолютную и относительную сложность программы, используя метрики Джилба. 3. Пусть всего операторов и операций в программе L=308,

		условных операторов нет, операторов цикла Lloop=10. число используемых модулей Lmod=5, число связей между модулями Nsv=4. Рассчитайте абсолютную и относительную сложность программы, относительное число связей, используя метрики Джилба. 4. Пусть число переменных для расчета P=3, число модифицируемых переменных M=5, число переменных, используемых в управлении программой C=3, число неиспользуемых переменных T=0. Определите на основе метрики Чепина уровень сложности данного решения.
Тема 2.3. Метрики структурной сложности программ	1. Какой смысл вкладывается в понятие структурной сложности программ? 2. Что такое цикломатическое число? 3. Как определить сложность вычислительных маршрутов, маршрутов принятия логических решении и общую сложность программ? 4. Как оценить сложность программы по первому критерию выделения маршрутов? 5. Что такое полносвязный граф? 6. Как оценить сложность программы по второму критерию выделения маршрутов? 7. Каково назначение матрицы смежности и матрицы достижимости? 8. Как оценить сложность программы по третьему критерию выделения маршрутов? 9. Каково назначение метрики	1. На основании блок-схемы алгоритма составить управляющий граф и оценить алгоритмическую сложность программы с использованием метрики Маккейба.  2. На основании блок-схемы алгоритма составить управляющий граф и оценить алгоритмическую сложность программы с использованием метрики Маккейба.

	Маккейба?	
Тема 2.4. Процедурно-ориентированные метрики программных средств	1. Какие особенности оценки характерны для метода Альбрехта? 2. Как рассчитать количество функциональных указателей? 3. Какие оценочные элементы применяются для расчета количества функциональных указателей? 4. Как определить коэффициенты регулировки сложности и весовые коэффициенты важности при оценке качества на основе функциональных указателей? 5. Какие производные метрики оценки качества программ могут использоваться на основе функциональных указателей? 6. Какова связь оценок качества программ на основе функциональных указателей и оценок на основе анализа длины программы? 7. Что характеризует связность модулей программных средств? 8. Какие	1. Пусть количество внешних вводов f1=3, количество внешних выводов f2=4, количество внешних запросов f3=2, количество локальных внутренних логических файлов f4=1, количество внешних интерфейсных файлов f5=0. Определите значения коэффициентов регулировки сложности. Рассчитайте значение количества функциональных указателей FP. 2. Пусть количество внешних вводов f1=3, количество внешних выводов f2=5, количество внешних запросов f3=0, количество локальных внутренних логических файлов f4=0, количество внешних интерфейсных файлов f5=0. Определите значения коэффициентов регулировки сложности. Рассчитайте значение количества функциональных указателей FP. 3. Пусть количество внешних вводов f1=3, количество внешних выводов f2=5, количество внешних запросов f3=1, количество локальных внутренних логических файлов f4=2, количество внешних интерфейсных файлов f5=0. Определите значения коэффициентов регулировки

	существуют типы связности программных модулей? 9. Как определить тип связности модулей программных средств? 10. Какие существуют типы сцепления программных модулей? 11. Как охарактеризован, влияние сцепления модулей на качество программных средств? 10.	сложности. Рассчитайте значение количества функциональных указателей FP.
Тема 2.5. Объектно-ориентированные метрики программных средств	1. Какие специфические особенности объектно-ориентированных решений влияют на использование метрик? 2. Какие характеристики объектно-ориентрованных ПС необходимо учитывать при измерении? 3. Как используются метрики Мартина? 4. Как рассчитать проектные метрики? 5. Из каких метрик состоит комплексный набор Лоренца и Кидда? 6. Какой подход к оценке качества ПС предложил Абреу?	1. Пусть исходный текст программы включает три класса, центростремительное сцепление по теории Мартина Ca=0, центробежное сцепление по теории Мартина Ce=3. Код программы не содержит абстрактных классов. Рассчитайте метрику нестабильности, меру абстрактности, расстояние до главной последовательности, нормализованное расстояние до главной последовательности. 2. Пусть исходный текст программы включает 4 класса, центростремительное сцепление по теории Мартина Ca=0, центробежное сцепление по теории Мартина Ce=4. Код программы не содержит абстрактных классов. Рассчитайте метрику нестабильности, меру абстрактности, расстояние до главной последовательности, нормализованное расстояние до главной последовательности.
Тема 2.6. Модели надежности	1. Какие существуют прогнозные модели оценки надежности программ? 2. Каковы основные особенности модели надежности Джелински - Моранды? 3. Каковы основные особенности модели	1. В результате тестирования программы серией из четырех случайно выбранных из набора тестов обнаружено 2 ошибки. Ошибки обнаружены первым и третьим тестами. Требуется определить количество ошибок N в программе до начала тестирования, используя модель

	надежности Миллса? 4. Каковы основные особенности измерительной модели Нельсона? 5. Каковы основные особенности модели надежности Мусы?	Джелински-Моранды. 2. В результате тестирования программы серией из четырех случайно выбранных из набора тестов обнаружено 2 ошибки. Ошибки обнаружены первым и четвертым тестами. Все ошибки исправлены сразу после обнаружения. В предположении, что исправление ошибок не повлекло появление новых ошибок, требуется оценить количество оставшихся в программе ошибок, используя модель Джелински-Моранды. Результаты расчетов округлять в большую или меньшую сторону по стандартным правилам (например, если округлить число 2,3, то получим 2, а если округлить 2,5 или 2,6, то после округления получим 3). 3. В программу преднамеренно внесли (посеяли) 10 ошибок. В результате тестирования обнаружено 12 ошибок, из которых 10 ошибок были внесены преднамеренно. Все обнаруженные ошибки исправлены. До начала тестирования предполагалось, что программа содержит не более 4 ошибок. Требуется оценить количество ошибок до начала тестирования и степень отлаженности программы, используя модель Миллса.
Раздел 3. Стандартизация и сертификация программного обеспечения		
Тема 3.1. Стандартизация программного обеспечения	1. Как определить понятие сертификации? 2. Какие цели преследует сертификация продукции? 3. Что относится к объектам сертификации? 4. Какой правовой документ регламентирует процедуру сертификации? 5. Что включает в себя понятие правового регулирования?	1. Провести оценку надежности программного средства в фазе реализации, используя номенклатуру показателей и значения оценочных элементов из таблицы (таблица прилагается). При проведении расчетов считать, что критерии и метрики в пределах своего уровня имеют одинаковую важность (все метрики для одного и того же критерия и все критерии

<table>
<tr><td></td><td>6. Что является центральным органом по сертификации в России?
7. Кто относится к участникам сертификации?
8. Каков порядок проведения сертификации, установленный Госстандартом?
9. Какие действия осуществляются при проведении сертификации?
10. Каковы особенности сертификации программного обеспечения?</td><td>надежности имеют одинаковые коэффициенты важности, сумма значений которых на каждом уровне равна единице). Базовый показатель надежности по критерию устойчивости функционирования считать равным 0,6. а по критерию работоспособности 0,9.</td></tr>
<tr><td>Тема 3.2. Сертификация программного обеспечения</td><td>1. Как определить понятие сертификации?
2. Какие цели преследует сертификация продукции?
3. Что относится к объектам сертификации?
4. Какой правовой документ регламентирует процедуру сертификации?
5. Что включает в себя понятие правового регулирования?
6. Что является центральным органом по сертификации в России?
7. Кто относится к участникам сертификации?
8. Каков порядок проведения сертификации, установленный Росстандартом?
9. Какие действия осуществляются при проведении сертификации?
10. Каковы особенности сертификации программного обеспечения?</td><td>1. В информационной системе используется четыре типа запросов, которые обслуживаются по модели бесприоритетного обслуживания. Каждый из запросов запрашивает только один документ. Запросы поступают с интервалом в 1 минуту и имеют длительности обработки, представленные в таблице:
<table><tr><td>Запрос</td><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>Момент поступления</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Время обработки, мин</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>Постройте диаграмму поступления и исполнения запросов. Удовлетворяет ли такая информационная система критерию своевременности представления информации, если предельное среднее время реакции системы на обработку запроса установлено равным 6 минутам?</td></tr>
<tr><td colspan="3">Раздел 4. Концепция и модели управления качеством программного обеспечения</td></tr>
<tr><td>Тема 4.1. Концепция управления качеством. Модели управления качеством.</td><td>1. Что предусматривала система Ф. Тейлора?
2. Что понимал А.</td><td></td></tr>
</table>

Организационно-технологические аспекты управления качеством	Фсйгснбаум под Всеобщим контролем качества? 3. Какие принципы совершенствования качества предложил Э. Деминг? 4. Что понимает TQM под понятием процесса? 5. Как понимается соответствие стандарту ISO? 6. Какие принципы менеджмента качества лежат в основе стандарта ISO 9000? 7. Как выглядит модель процессного подхода в представлении стандарта ISO 9000? 8. Каково смысловое содержание системы менеджмента качества? 9. Какие элементы объединяет в себе японская четырехуровневая модель? 10. Какой смысл заложен в японской программе «пяти нулей»? 11. Какие принципы лежат в основе программы JIT? 12. Какие особенности характеризуют систему Канбан? 13. На каких положениях базируется модель EFQM? 14. Какие особенности оценки качества характерны для европейской модели? 15. Какие основные факторы характерны для процесса воспитания надежного персонала? 16. Какие уровни зрелости организации определяет модель CMM? 17. Какие дополнения разработаны к модели CMM?	

	18. Какие причины затрудняют применение модели CMM? 19. Каково принципиальное отличие стандарта SPICE от CMM? 20. Почему модель SPICE называют двумерной?	

Составитель (и): Решетникова Е. В., зав. кафедрой математики, физики и математического моделирования

(*фамилия, инициалы и должность преподавателя (ей)*)