

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244e728abc3661ab35e9d50210dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А.В. Фомина
«08» февраля 2024 г.

Рабочая программа дисциплины

К.М.08.04 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 2023

Новокузнецк 2024

Оглавление

1 Цель дисциплины	3
Формируемые компетенции, индикаторы достижения компетенций, знания, умения, навыки	3
Место дисциплины.....	3
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.	3
3. Учебно-тематический план и содержание дисциплины.....	4
3.1 Учебно-тематический план	4
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.....	5
5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.	6
5.1 Учебная литература	6
5.2 Материально-техническое и программное обеспечение дисциплины.....	6
5.3 Современные профессиональные базы данных и информационные справочные системы.....	7
6 Иные сведения и (или) материалы.....	8
6.1. Темы письменных учебных работ	8
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации	8

1 Цель дисциплины.

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): *ОПК-2*.

Формируемые компетенции, индикаторы достижения компетенций, знания, умения, навыки

Таблица 1 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
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	Знать – критерии 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.

Место дисциплины

Дисциплина включена в модуль «Модуль современных информационных технологий» ОПОП ВО, часть, формируемая участниками образовательных отношений. Дисциплина осваивается на 4 курсе в 8 семестре.

2 Объём и трудоёмкость дисциплины по видам учебных занятий.

Формы промежуточной аттестации.

Таблица 2 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объем часов по формам обучения		
	ОФО	ОЗФО	ЗФО
1 Общая трудоемкость дисциплины	72		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	68		
Аудиторная работа (всего):	68		
в том числе:			
лекции	20		
практические занятия, семинары			
практикумы			
лабораторные работы	48		
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	4		
4 Промежуточная аттестация обучающегося	Зачет - 8 семестр		

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 3 - 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 программного обеспечения	9	2	4	1	Устный опрос
3-5	Основы метрической теории программ	39	12	24	1	Индивидуальные задания
6-7	Стандартизация и сертификация программного обеспечения	15	4	14	1	Индивидуальные задания Доклад.
10-11	Концепция и модели управления качеством программного обеспечения	9	2	6	1	Доклад
	Промежуточная аттестация зачет с оценкой					зачет
ИТОГО		72	20	48	4	-

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 4.

Таблица 4 - Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	Баллы
Семестр 8				
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 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 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
		Решение задачи 1.	3 балла (пороговое значение)	3 – 6

		6 баллов (максимальное значение)	
	Решение задачи 2.	3 баллов (пороговое значение)	3 - 6
		6 баллов (максимальное значение)	
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 5)

Таблица 5 – 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/Кайгородцев Г.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 ФГБОУ ВО «КемГУ»:

100 Учебная аудитория (мультимедийная) для проведения: -занятий лекционного типа; - групповых и индивидуальных консультаций.	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19
--	---

<p>Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья, рабочее место для обучающегося с ОВЗ.</p> <p>Оборудование: <i>стационарное</i> - компьютер преподавателя, экран моторизированный, проектор, усилитель звука, колонки, микрофон преподавателя.</p> <p>Используемое программное обеспечение: Ubuntu Linux (свободно распространяемое ПО), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО).</p> <p>Интернет с обеспечением доступа в ЭИОС.</p>	
<p>508 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения:</p> <ul style="list-style-type: none"> - занятий семинарского (практического) типа; - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. <p>Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья.</p> <p>Оборудование для презентации учебного материала: <i>стационарное</i> - компьютер преподавателя, проектор, экран.</p> <p>Оборудование: <i>стационарное</i> – компьютеры для обучающихся (18 шт.).</p> <p>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).</p> <p>Интернет с обеспечением доступа в ЭИОС.</p>	<p>654079, Кемеровская область, г. Новокузнецк, пр-кт Metallургов, д. 19</p>

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

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 беспriorитетного 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

Семестр 8

Таблица 7 - Примерные теоретические вопросы и практические задания / задачи к зачету

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания и (или) задачи
<i>Раздел 1. Основные понятия и характеристики качества программного обеспечения</i>		
Тема 1.1 Основные понятия и характеристики качества программного обеспечения	<ol style="list-style-type: none"> Каковы основные понятия качества программного обеспечения? Что представляют собой характеристики программы? Как определить понятие системы обеспечения качества? Какие объекты программных систем подвержены уязвимости? Какова классификация дестабилизирующих факторов программного обеспечения? 	<ol style="list-style-type: none"> Определите понятие качества ПО, используя различные официальные источники. 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. Определите измерительные шкалы для следующих показателей: использование моделей, структура программы, использование указателей,

	<p>6. Какие существуют внутренние источники угроз программного обеспечения?</p> <p>7. Какие внешние факторы могут дестабилизировать работу программного обеспечения?</p> <p>8. Какие общие факторы влияют на качество программного обеспечения?</p> <p>9. Как определить понятия фактора качества, критерия качества, метрики?</p> <p>10. Что представляют собой понятия оценочного элемента, показателя качества, базового значения показателя качества?</p> <p>11. Какой смысл вкладывается в понятия сертификации, верификации и аттестации?</p> <p>12. Как взаимосвязаны факторы, влияющие на качество программного обеспечения?</p> <p>13. Каковы функциональные и конструктивные характеристики качества программных средств?</p> <p>14. Что такое внутреннее качество программных средств?</p> <p>15. Как определить внешнее качество программных средств?</p> <p>16. Что вкладывается в понятие качества при использовании программных средств?</p> <p>17. Как определить понятия системы измерений характеристик программного обеспечения, измерительной шкалы?</p> <p>18. Каковы условия обеспечения объективности измерений?</p>	<p>объем программы.</p>
--	---	-------------------------

	<p>19. Какие существуют категории групп показателей качества программных средств?</p> <p>20. Как классифицируются типы измерительных шкал, какова их эмпирическая значимость?</p>	
<i>Раздел 2. Основы метрической теории программ</i>		
<p>Тема 2.1. Алгоритмическая сложность. Метрики, основанные на лексическом анализе программ.</p>	<ol style="list-style-type: none"> 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? 14. Что устанавливает закон Хика? 15. Что определяет число Страуда? 	<ol style="list-style-type: none"> 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 $n_1=28$, число простых операндов $n_2=16$, число всех операторов $N_1=116$, число всех операндов $N_2=36$, 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 $n_1=20$, число простых операндов $n_2=12$, число всех операторов $N_1=77$, число всех операндов $N_2=47$, 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 $n_1=42$, число простых операндов $n_2=23$, число всех операторов $N_1=218$, число всех операндов $N_2=97$, число входных и выходных параметров равно 13. Программа написана на языке Паскаль. Рассчитайте длину и

	<p>16. Зачем используется коэффициент пересчета Кнута?</p> <p>17. Как оценить уровень языка программирования?</p> <p>18. Что устанавливает закон Миллера?</p> <p>19. Каков порядок расчета метрических характеристик программных средств?</p>	<p>объем программы по Джилбу. Определите уровень реализации программы.</p> <p>6. Пусть число простых операторов $n_1=39$, число простых операндов $n_2=24$, число всех операторов $N_1=252$, число всех операндов $N_2=95$, число входных и выходных параметров равно 12. Программа написана на языке Фортран. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы.</p> <p>7. Пусть число простых операторов $n_1=28$, число простых операндов $n_2=27$, число всех операторов $N_1=208$, число всех операндов $N_2=93$, число входных и выходных параметров равно 2. Программа написана на языке С#. Рассчитайте длину и объем программы по Джилбу. Определите уровень реализации программы.</p>
<p>Тема 2.2 Метрики, основанные на лексическом анализе программ</p>	<p>1. Какие метрики предлагаются Джилбом?</p> <p>2. Какие показатели качества предлагает оценивать Джилб?</p> <p>3. Как рассчитать начальную надежность программы?</p> <p>4. Как оценить меру сложности программы по Чепину?</p>	<p>1. Пусть всего операторов и операций в программе $L=263$, условных операторов $L_{if}=2$, операторов цикла $L_{loop}=4$, число используемых модулей $L_{mod}=4$, число связей между модулями $N_{sv}=3$. Рассчитайте абсолютную и относительную сложность программы, относительное число связей, используя метрики Джилба.</p> <p>2. Пусть всего операторов и операций в программе $L=404$, условных операторов $L_{if}=1$, операторов цикла $L_{loop}=19$. Программа состоит из одного модуля. Рассчитайте абсолютную и относительную сложность программы, используя метрики Джилба.</p> <p>3. Пусть всего операторов и операций в программе $L=308$, условных операторов нет, операторов цикла $L_{loop}=10$. число используемых модулей $L_{mod}=5$, число связей между модулями $N_{sv}=4$. Рассчитайте абсолютную и</p>

		<p>относительную сложность программы, относительное число связей, используя метрики Джилба.</p> <p>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.</p>
<p>Тема 2.3. Метрики структурной сложности программ</p>	<ol style="list-style-type: none"> 1. Какой смысл вкладывается в понятие структурной сложности программ? 2. Что такое цикломатическое число? 3. Как определить сложность вычислительных маршрутов, маршрутов принятия логических решений и общую сложность программ? 4. Как оценить сложность программы по первому критерию выделения маршрутов? 5. Что такое полностью связный 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 Маккейба? 	<ol style="list-style-type: none"> 1. На основании блок-схемы алгоритма составить управляющий граф и оценить алгоритмическую сложность программы с использованием метрики Маккейба. <pre> graph TD Start([Начало]) --> Input[/Ввод координат x, y/] Input --> GFalse[G = false] GFalse --> D1{x^2 + y^2 <= R^2?} D1 -- Нет --> GTrue1[G = true] D1 -- Да --> D2{x >= 0?} D2 -- Нет --> GTrue1 D2 -- Да --> D3{y < x?} D3 -- Нет --> GTrue1 D3 -- Да --> InputGer[/Ввод значения ger/] InputGer --> D4{ger = Y или ger = y?} D4 -- Да --> D1 D4 -- Нет --> End([Конец]) GTrue1 --> Output[/Вывод значения G/] </pre> <ol style="list-style-type: none"> 2. На основании блок-схемы алгоритма составить управляющий граф и оценить алгоритмическую сложность программы с использованием метрики Маккейба.

<p>Тема 2.4. Процедурно-ориентированные метрики программных средств</p>	<ol style="list-style-type: none"> Какие особенности оценки характерны для метода Альбрехта? Как рассчитать количество функциональных указателей? Какие оценочные элементы применяются для расчета количества функциональных указателей? Как определить коэффициенты регуляции сложности и весовые коэффициенты важности при оценке качества на основе функциональных указателей? Какие производные метрики оценки качества программ могут использоваться на основе функциональных указателей? Какова связь оценок качества программ на основе функциональных указателей и оценок на основе анализа длины программы? Что характеризует связность модулей программных средств? Какие существуют типы связности 	<ol style="list-style-type: none"> 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 регуляции сложности. Рассчитайте значение количества функциональных указателей FP. 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 регуляции сложности. Рассчитайте значение количества функциональных указателей FP. 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 регуляции сложности. Рассчитайте значение

	<p>программных модулей?</p> <p>9. Как определить тип связности модулей программных средств?</p> <p>10. Какие существуют типы сцепления программных модулей?</p> <p>11. Как охарактеризован, влияние сцепления модулей на качество программных средств?</p> <p>10.</p>	<p>количества функциональных указателей FP.</p>
<p>Тема 2.5. Объектно-ориентированные метрики программных средств</p>	<p>1. Какие специфические особенности объектно-ориентированных решений влияют на использование метрик?</p> <p>2. Какие характеристики объектно-ориентированных ПС необходимо учитывать при измерении?</p> <p>3. Как используются метрики Мартина?</p> <p>4. Как рассчитать проектные метрики?</p> <p>5. Из каких метрик состоит комплексный набор Лоренца и Кидда?</p> <p>6. Какой подход к оценке качества ПС предложил Абреу?</p>	<p>1. Пусть исходный текст программы включает три класса, центростремительное сцепление по теории Мартина $C_a=0$, центробежное сцепление по теории Мартина $C_e=3$. Код программы не содержит абстрактных классов. Рассчитайте метрику нестабильности, меру абстрактности, расстояние до главной последовательности, нормализованное расстояние до главной последовательности.</p> <p>2. Пусть исходный текст программы включает 4 класса, центростремительное сцепление по теории Мартина $C_a=0$, центробежное сцепление по теории Мартина $C_e=4$. Код программы не содержит абстрактных классов. Рассчитайте метрику нестабильности, меру абстрактности, расстояние до главной последовательности, нормализованное расстояние до главной последовательности.</p>
<p>Тема 2.6. Модели надежности</p>	<p>1. Какие существуют прогнозные модели оценки надежности программ?</p> <p>2. Каковы основные особенности модели надежности Джелински - Моранды?</p> <p>3. Каковы основные особенности модели надежности Миллса?</p> <p>4. Каковы основные особенности измерительной</p>	<p>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 Джелински-Моранды.</p> <p>2. В результате тестирования программы серией из четырех</p>

	<p>модели Нельсона?</p> <p>5. Каковы основные особенности модели надежности Мусы?</p>	<p>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).</p> <p>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.</p>
<i>Раздел 3. Стандартизация и сертификация программного обеспечения</i>		
<p>Тема 3.1. Стандартизация программного обеспечения</p>	<p>1. Как определить понятие сертификации?</p> <p>2. Какие цели преследует сертификация продукции?</p> <p>3. Что относится к объектам сертификации?</p> <p>4. Какой правовой документ регламентирует процедуру сертификации?</p> <p>5. Что включает в себя понятие правового регулирования?</p> <p>6. Что является центральным органом по сертификации в России?</p> <p>7. Кто относится к</p>	<p>1. Провести оценку надежности программного средства в фазе реализации, используя номенклатуру показателей и значения оценочных элементов из таблицы (таблица прилагается).</p> <p>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 надежности имеют одинаковые коэффициенты важности, сумма значений которых на каждом уровне равна единице). Базовый показатель надежности по</p>

	<p>участникам сертификации?</p> <p>8. Каков порядок проведения сертификации, установленный Госстандартом?</p> <p>9. Какие действия осуществляются при проведении сертификации?</p> <p>10. Каковы особенности сертификации программного обеспечения?</p>	<p>критерию устойчивости функционирования считать равным 0,6. а по критерию работоспособности 0,9.</p>															
<p>Тема 3.2. Сертификация программного обеспечения</p>	<p>1. Как определить понятие сертификации?</p> <p>2. Какие цели преследует сертификация продукции?</p> <p>3. Что относится к объектам сертификации?</p> <p>4. Какой правовой документ регламентирует процедуру сертификации?</p> <p>5. Что включает в себя понятие правового регулирования?</p> <p>6. Что является центральным органом по сертификации в России?</p> <p>7. Кто относится к участникам сертификации?</p> <p>8. Каков порядок проведения сертификации, установленный Росстандартом?</p> <p>9. Какие действия осуществляются при проведении сертификации?</p> <p>10. Каковы особенности сертификации программного обеспечения?</p>	<p>1. В информационной системе используется четыре типа запросов, которые обслуживаются по модели беспriorитетного обслуживания. Каждый из запросов запрашивает только один документ. Запросы поступают с интервалом в 1 минуту и имеют длительности обработки, представленные в таблице:</p> <table border="1" data-bbox="1098 958 1501 1099"> <thead> <tr> <th>Запрос</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>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?</p>	Запрос	A	B	C	D	Момент поступления	0	1	2	3	Время обработки, мин	1	2	3	4
Запрос	A	B	C	D													
Момент поступления	0	1	2	3													
Время обработки, мин	1	2	3	4													
<i>Раздел 4. Концепция и модели управления качеством программного обеспечения</i>																	
<p>Тема 4.1. Концепция управления качеством. Модели управления качеством. Организационно-технологические аспекты управления качеством</p>	<p>1. Что предусматривала система Ф. Тейлора?</p> <p>2. Что понимал А. Фэйгенбаум под Всеобщим контролем качества?</p> <p>3. Какие принципы совершенствования качества предложил Э. Деминг?</p> <p>4. Что понимает</p>																

	<p>TQM под понятием процесса?</p> <p>5. Как понимается соответствие стандарту ISO?</p> <p>6. Какие принципы менеджмента качества лежат в основе стандарта ISO 9000?</p> <p>7. Как выглядит модель процессного подхода в представлении стандарта ISO 9000?</p> <p>8. Каково смысловое содержание системы менеджмента качества?</p> <p>9. Какие элементы объединяет в себе японская четырехуровневая модель?</p> <p>10. Какой смысл заложен в японской программе «пяти нулей»?</p> <p>11. Какие принципы лежат в основе программы JIT?</p> <p>12. Какие особенности характеризуют систему Канбан?</p> <p>13. На каких положениях базируется модель EFQM?</p> <p>14. Какие особенности оценки качества характерны для европейской модели?</p> <p>15. Какие основные факторы характерны для процесса воспитания надежного персонала?</p> <p>16. Какие уровни зрелости организации определяет модель CMM?</p> <p>17. Какие дополнения разработаны к модели CMM?</p> <p>18. Какие причины затрудняют применение модели CMM?</p> <p>19. Каково принципиальное отличие стандарта SPICE от CMM?</p> <p>20. Почему модель SPICE называют</p>	
--	--	--

	двумерной?	
--	------------	--

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

(фамилия, инициалы и должность преподавателя (ей))