

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

Кафедра «Космические технологии»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

Б1.В.07 «Обеспечение качества и надежности программных систем»

Направление подготовки - 09.03.01 «Информатика и вычислительная
техника»

ОПОП академического бакалавриата
«Системный инжиниринг и космические информационные технологии»

Квалификация (степень) выпускника - бакалавр
Форма обучения - очная

Рязань 2025

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Промежуточный контроль по дисциплине осуществляется путем проведения Зачета. Форма проведения Зачета – тестирование и выполнение практических заданий. При необходимости, проводится теоретическая беседа с обучаемым для уточнения оценки. Выполнение заданий на практических занятиях в течение семестра и заданий на самостоятельную работу является обязательным условием для допуска к Зачету.

2. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
Тема 1. Основы теории надежности технических систем	ПК-1.4 ПК-1.5	экзамен
Тема 2. Обеспечение надежности программных систем		
Тема 3. Обеспечение качества программных систем	ПК-1.4	экзамен

3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированности каждой компетенции в рамках освоения данной дисциплины оцениваются по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

3.1. Описание критериев и шкалы оценивания промежуточной аттестации

а) описание критериев и шкалы оценивания тестирования:

За каждый тестовый вопрос назначается максимально 1 балл в соответствии со следующим правилом:

- 1 балл – ответ на тестовый вопрос полностью правильный;
- 0,5 балла – отчет на тестовый вопрос частично правильный (выбраны не все правильные варианты, указаны частично верные варианты);
- 0 баллов – ответ на тестовый вопрос полностью не верный.

б) описание критериев и шкалы оценивания решения практического задания:

Шкала оценивания	Критерий
------------------	----------

Шкала оценивания	Критерий
10 баллов (эталонный уровень)	Задание выполнено верно, полностью самостоятельно, без дополнительных наводящих вопросов преподавателя
7 балла (продвинутый уровень)	Задание выполнено верно, но имеются технические неточности
4 балла (пороговый уровень)	Задание выполнено верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задание не выполнено

На Зачет выносятся 40 тестовых вопросов и 2 практических задания. Максимально студент может набрать 60 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «не удовлетворительно».

Шкала оценивания	Критерий	
отлично (эталонный уровень)	55 – 60 баллов	Обязательным условием является выполнение всех предусмотренных в течении семестра заданий (на практических занятиях и при самостоятельной работе)
хорошо (продвинутый уровень)	50 – 54 баллов	
удовлетворительно (пороговый уровень)	35 – 49 баллов	
не удовлетворительно	0 – 34 баллов	Студент не выполнил всех предусмотренных в течении семестра текущих заданий (на практических занятиях и при самостоятельной работе)

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-1. Способен управлять работами по созданию (модификации) и сопровождению информационных ресурсов	ПК-1.4. Осуществляет тестирование ИР с точки зрения пользовательского удобства на основании данных о поведении пользователей.
	ПК-1.5. Организует работы по обеспечению безопасной работы информационных ресурсов.

а) типовые тестовые вопросы:

- Укажите содержательное определение термина «Качество программного обеспечения»:
 - Качество программного обеспечения - это комплекс характеристик программного продукта, определяющих его способность выполнять возложенные на него функции.**
 - Качество программного обеспечения – это комплекс взаимосвязанных программ для решения определённой проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции.
 - Качество программного обеспечения – это комплекс программ, баз данных, файлов, а также описывающих их документов, составляющих систему для решения группы связанных задач на одном или нескольких взаимодействующих компьюте-

рах.

2. Укажите содержательное определение термина «Модель качества программного обеспечения»:

- **Модель качества программного обеспечения - это набор характеристик и отношений между ними, которые обеспечивают основу для определения требований качества и оценки качества программного обеспечения.**
- Модель качества программного обеспечения – это критерии, которые определяют, насколько программное обеспечение соответствует потребностям или ожиданиям его пользователей.
- **Модель качества программного обеспечения – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций.**

3. Укажите известные метрики лексического анализа для оценки характеристик программ:

- **Метрики Холстеда.**
- Метрики Мартина.
- **Метрики Джилба.**
- **Метрики Чепина.**

4. Укажите сущности и метрические характеристики присущие метрикам Холстеда:

- **Число простых (уникальных) операторов, появляющихся в программе.**
- Число появлений в программе каждого оператора.
- **Число простых (уникальных) операндов, появляющихся в программе.**
- Число появлений в программе каждого операнда.
- **Общее число всех операторов, появляющихся в программе.**
- **Общее число всех операндов, появляющихся в программе.**
- **Число появлений в программе каждого уникального оператора.**
- **Число появлений в программе каждого уникального операнда.**

5. Что можно определить методом Холстеда при анализе характеристик программы?

- **Словарь.**
- Маршруты принятия логических решений.
- **Длину реализации программы.**
- **Длину программы.**

6. Какие элементы программ относят к словарю операторов при проведении статистических исследований текстов программ по метрикам Холстеда:

- **Имена арифметических и логических операций.**
- **Присваивания.**
- Указатели адреса начала массива.
- **Условные и безусловные переходы.**
- **Разделители.**
- **Скобки (парные).**
- **Имена процедур и функций.**
- **Выражения типа BEGIN ...END, IF...THEN...ELSE, и другие подобные.**

7. Метрики Джилба позволяют определять следующие характеристики программы:

- Отношение числа модулей программы к числу связей между модулями
- **Логическую сложность программы.**
- **Абсолютную сложность программы, характеризуемую количеством операторов условий.**
- **Относительную сложность программы, определяющую насыщенность программы операторами условия.**
- Отношение общего числа операторов программы к числу ненормальных выходов из множества операторов.

8. Метрики Чепина позволяют определять следующие характеристики программы:

- **Меру сложности понимания программы на основе входных и выходных данных.**
- Меру размера программы на этапах ее разработки.
- Меру логической сложности программы.

9. Согласно метрикам Чепина все множество переменных анализируемой программы, составляющих список ввода-вывода, разбивается на следующие функциональные группы:

- **Р - вводимые переменные для расчетов и для обеспечения вывода.**
- **I - имена арифметических и логических операций.**
- **М - модифицируемые, или создаваемые внутри программы, переменные.**
- **С - переменные, участвующие в управлении работой программного модуля.**
- **Т - не используемые в программе (так называемые паразитные) переменные.**

10. Укажите значимое выражение для формального определения метрики Чепина:

- $Q = a_1 * P + a_2 * M + a_3 * C + a_4 * T + a_5 * I$
- **$Q = a_1 * P + a_2 * M + a_3 * C + a_4 * T$**
- $Q = a_1 * T + a_2 * C + a_3 * P + a_4 * M$

11. Какими показателями осуществляется оценка структурной сложности программ:

- **Количеством взаимодействующих компонентов.**
- Многообразием поведения программы.
- **Числом связей между компонентами.**
- **Сложностью взаимодействия компонентов.**
- **Числом маршрутов исполнения программы и их сложностью.**

12. При оценке структурной сложности программ рекомендуется всю совокупность маршрутов исполнения программного модуля разделить на следующие группы:

- **Вычислительные маршруты.**
- **Маршруты принятия логических решений.**
- Маршруты оптимизации.

13. При оценке структурной сложности программ рекомендуются использовать следующие критерии выделения маршрутов:

- **Критерий, предполагающий, что граф потока управления программой должен быть проверен по минимальному набору маршрутов, проходящих через каждый оператор ветвления по каждой дуге.**
- Критерий выделения маршрутов исполнения программы, основанный на процедуре минимизации проверки всех возможных маршрутов ее выполнения и экспертной оценке структурной сложности.

- **Критерий, основанный на анализе базовых маршрутов в программе, которые формируются и оцениваются на основе цикломатического числа графа потока управления программы.**
- **Критерий выделения маршрутов, основанный на формировании полного состава базовых структур графа потока управления программой, заключающийся в анализе каждого из реальных ациклических маршрутов исходного графа программы и каждого цикла, достижимого из всех этих маршрутов.**

14. На каких принципах или стратегии построена метрика Маккейба для оценки структурной сложности программ:

- **Построена не на принципах анализа лексических характеристик программы, а на результатах анализа потока управления от одного оператора к другому.**
- **Построена на основе стратегии проверки корректности программных средств, получившей название основного маршрута тестирования Маккейба.**
- Построена на принципах анализа лексических характеристик программы и на результатах дальнейшего анализа потока управления от одного оператора к другому.

15. Цикломатическая сложность программы это:

- **Структурная мера сложности программы, применяемая для измерения качества программного обеспечения и основанная на методах статистического анализа кода программы.**
- Счетное число линейно независимых маршрутов, проходящих через программный код.
- **Структурная мера сложности программы равная цикломатическому числу графа программы, увеличенному на единицу.**

16. Оценка уровня качества процедурно-ориентированных программных средств с использованием расчетных методов производится на основе применения актуальной метрики дефектов качества (укажите вариант):

- $DQ = \text{Количество_Строк_Программы (тыс.строк)} \setminus \text{Количество_Ошибок (единиц)}$, при этом предполагается соблюдение нормы $0 \leq DQ \leq 1$.
- **$DQ = \text{Количество_Ошибок (единиц)} \setminus \text{Количество_Строк_Программы (тыс.строк)}$, при этом предполагается соблюдение нормы $0 \leq DQ \leq 1$.**
- $DQ = \text{Количество_Ошибок (единиц)} \setminus \text{Количество_Строк_Программы (тыс.строк)}$, при этом предполагается соблюдение нормы $1 \leq DQ \leq 10$.

17. Укажите сущности и метрические характеристики присущие метрикам оценки качества процедурно-ориентированных программных средств с использованием метрик связности и сцепления модулей:

- **Уровень качества программного средства прямо пропорционален силе связности (прочности) и обратно пропорционален силе сцепления модулей.**
- Уровень качества программного средства прямо пропорционален силе сцепления модулей и обратно пропорционален силе связности (прочности).
- **Для обеспечения необходимого уровня надежности программы разработка качественного программного средства должна предполагать максимизацию связности и минимизацию сцепления модулей.**

18. Укажите различные типы связности модулей, учитываемые при оценке уровня качества процедурно-ориентированных программных средств:

- **Связность по совпадению.**

- **Логическая связность.**
- Связность по данным.
- **Временная связность.**
- **Процедурная связность.**
- **Информационная связность.**
- **Функциональная связность.**

19. Укажите различные типы сцепления модулей, учитываемые при оценке уровня качества процедурно-ориентированных программных средств:

- Сцепление по логике.
- **Сцепление по данным.**
- **Сцепление по образцу.**
- **Сцепление по управлению.**
- **Сцепление по внешним ссылкам.**
- **Сцепление по общей области.**
- **Сцепление по содержанию.**

20. Укажите метрические сущности, присущие метрикам Мартина для оценки характеристик программы при объектно-ориентированном программировании:

- I (нестабильность) – расчетная метрика, определяемая в соответствии с выражением $I = Ce \setminus (Ca + Ce)$. При $I=1$ означает максимально стабильную категорию, при $I=0$ указывает максимально нестабильную категорию.
- Ca (центростремительное сцепление) – метрика, определяющая количество классов вне конкретной категории, которые зависят от классов внутри нее.
- Ce (центробежное сцепление) – метрика, оценивающая количество классов внутри конкретной категории, которые зависят от классов вне нее.
- I (нестабильность) – расчетная метрика, определяемая в соответствии с выражением $I = Ce \setminus (Ca + Ce)$. При $I=0$ означает максимально стабильную категорию, при $I=1$ указывает максимально нестабильную категорию.

21. Укажите характеристики и формулу метрики Мартина, определяющей меру A – абстрактность, которая позволяет оценить абстрактность категории:

- $A = nAll \setminus nA$, где nA – количество абстрактных классов в категории; $nAll$ – общее количество классов в категории.
- $A = nA \setminus nAll$, где nA – количество абстрактных классов в категории; $nAll$ – общее количество классов в категории.
- $A = nA \setminus nAll$, где $nAll$ – количество абстрактных классов в категории; nA – общее количество классов в категории.

22. Перечислите актуальные проектные метрики Чидамбера-Кемера, основанные на анализе методов класса, дерева наследования и других характеристик объектно-ориентированной среды:

- Взвешенные методы на класс.
- **Глубина дерева наследования.**
- **Количество потомков.**
- **Связность между классами объектов.**
- **Количество откликов на класс.**
- **Отсутствие сцепления в методах.**

23. Укажите актуальную классификацию метрик Лоренца-Кидда, имеющей практическую направленность на использование в промышленной объектно-ориентированной разработке программного обеспечения:

- **Метрики размера, основанные на подсчете свойств и операций для отдельных классов, а также их средних значений для всей объектно-ориентированной системы.**
- Метрики размера, основанные на подсчете свойств и операций для отдельных классов, а также их абсолютных значений для всей объектно-ориентированной системы.
- **Метрики наследования, учитывающие способы повторного использования операций в иерархии классов.**
- **Внутренние метрики, отвечающие на вопросы связности и кодирования.**
- **Внешние метрики, изучающие сцепление и повторное использование.**

24. Укажите основные цели метрик Абреу при оценке качества объектно-ориентированного программирования:

- **Учет базовых механизмов объектно-ориентированного подхода к созданию ПО (инкапсуляция, наследование, полиморфизм, посылка сообщений).**
- Независимость от уровня профессионализма разработчиков при использовании базовых механизмов объектно-ориентированного подхода к созданию ПО.
- **Формальное определение такого набора метрик, который позволит исключить или значительно снизить субъективность определения показателей качества ПО.**
- **Независимость от размера оцениваемого программного продукта.**
- **Независимость от языка программирования, который использовался при создании оцениваемого ПО.**

25. Укажите показатели качества программного продукта, которые включает в себя набор метрик Абреу:

- **Фактор закрытости метода.**
- Фактор инвариантности к языкам программирования.
- **Фактор закрытости свойства.**
- **Фактор наследования метода.**
- **Фактор наследования свойства фактор полиморфизма.**

26. Укажите на соблюдении каких условий основывается модель Джелински-Моранды при оценке надежности программных средств:

Линейная зависимость плотности вероятности интервалов времени между появлением ошибок в программе.

- **Интенсивность ошибок в программе линейно зависит от количества оставшихся ошибок на любом случайном интервале.**
- **Каждый тест находит только одну ошибку в программе.**
- **После каждого появления ошибка в программе устраняется и не вносится новая ошибка.**
- **Экспоненциальная зависимость плотности вероятности интервалов времени между появлением ошибок в программе.**
- **После каждого появления ошибка в программе устраняется и может вноситься новая ошибка.**

27. Укажите особенности и условия применимости статистической модели Миллса при оценке надежности программных средств:

- Модель позволяет оценить только количество ошибок до начала тестирования программы.
- Для применения модели до начала тестирования в программу преднамеренно вносятся ошибки.
- Считается, что обнаружение преднамеренно внесенных и так называемых собственных ошибок программы равновероятно.
- Модель позволяет оценить не только количество ошибок до начала тестирования, но и степень отлаженности программы.

28. Укажите особенности применимости эвристической модели оценки надежности программных средств:

- Эвристическая модель позволяет оценить количество ошибок N до начала тестирования по результатам тестирования программы двумя независимыми группами тестирующих.
- Эвристическая модель может хорошо работать при «перекрестном» тестировании программ несколькими группами тестирующих.
- Для применения эвристической модели применяется следующее выражение:
 $N = N_1 * N_2 \setminus N_{1,2}$, где N_1 – количество ошибок, обнаруженных первой группой тестирующих; N_2 – количество ошибок, обнаруженных второй группой тестирующих; $N_{1,2}$ – количество ошибок, обнаруженные первой и второй группами тестирующих.
- Для применения эвристической модели применяется следующее выражение:
 $N = N_{1,2} \setminus N_1 * N_2$, где N_1 – количество ошибок, обнаруженных первой группой тестирующих; N_2 – количество ошибок, обнаруженных второй группой тестирующих; $N_{1,2}$ – количество ошибок, обнаруженные первой и второй группами тестирующих.

29. Процесс оценки качества программного обеспечения (по ГОСТ 28195, ГОСТ 9126, ГОСТ 25000) осуществляется для каждой фазы его жизненного цикла и включает:

- **Выбор совокупности (номенклатуры) показателей качества оцениваемого программного средства.**
- **Определение значений показателей качества оцениваемого программного средства.**
- Оптимизация значений показателей качества оцениваемого программного средства.
- **Сравнение полученных значений показателей качества оцениваемого программного средства с базовыми значениями показателей качества.**

30. В соответствии с ГОСТ весь период жизненного цикла программного обеспечения делится на следующие временные промежутки (этапы) или фазы (укажите актуальные этапы в произвольной последовательности):

- Маркетинговые исследования – этап анализа рынка ПО.
- Анализ – этап определения требований к ПО, спецификация требований и формирования технического задания на проектирование ПО.
- Проектирование – этап разработки технического проекта.
- Реализация – этап разработки ПО, средств тестирования и документации.
- Тестирование – этап испытания ПО и устранение недостатков.
- Изготовление – этап преобразования ПО в форму, готовую для поставки; завершение формирования документации.

- **Внедрение** – этап подтверждения стабильной работы ПО; ввод в стадию активного применения.
- **Эксплуатация** – этап применения ПО по назначению.
- **Утилизация** – этап вывода ПО из стадии активного применения.

31. В соответствии с ГОСТ оценка качества ПО на всех фазах жизненного цикла осуществляется на четырехуровневой системе показателей (укажите уровни):

- Уровень системы показателей качества в соответствии ГОСТ.
- Уровень комплексных показателей качества.
- **Уровень факторов (характеристик) качества.**
- **Уровень критериев (подхарактеристик) качества.**
- **Уровень метрик критериев качества.**
- **Уровень атрибутов метрик критериев качества.**

32. Согласно ГОСТ фактор «Надежность» представляет собой интегральную оценку, которой соответствует несколько критериев качества (комплексных показателей второго уровня), укажите эти критерии:

- Модифицируемость.
- **Работоспособность.**
- **Устойчивость функционирования.**
- Удобство эксплуатации.
- Логическая корректность.

33. Согласно ГОСТ фактор «Сопровождаемость» представляет собой интегральную оценку, которой соответствует несколько следующих критериев качества (комплексных показателей второго уровня):

- Согласованность.
- **Структурность.**
- **Простота конструкции.**
- **Наглядность.**
- **Повторяемость.**
- Мобильность.

34. Согласно ГОСТ фактор «Удобство применения» представляет собой интегральную оценку, которой соответствует несколько следующих критериев качества (комплексных показателей второго уровня):

- Модифицируемость.
- **Легкость освоения.**
- **Доступность эксплуатационных документов.**
- **Удобство эксплуатации.**
- Проверенность.

35. Согласно ГОСТ фактор «Эффективность» представляет собой интегральную оценку, которой соответствует несколько следующих критериев качества (комплексных показателей второго уровня):

- Работоспособность.
- **Уровень автоматизации.**
- **Временная эффективность.**

- **Ресурсоемкость.**
- Доступность.

36. Согласно ГОСТ фактор «Универсальность» представляет собой интегральную оценку, которой соответствует несколько следующих критериев качества (комплексных показателей второго уровня):

- Наглядность.
- **Гибкость.**
- **Мобильность.**
- **Модифицируемость.**
- Ресурсоемкость.

37. Согласно ГОСТ фактор «Корректность» представляет собой интегральную оценку, которой соответствует несколько следующих критериев качества (комплексных показателей второго уровня):

- Структурность.
- **Полнота реализации.**
- **Согласованность.**
- **Проверенность.**
- **Логическая корректность.**
- Мобильность.

38. Укажите содержательное определение термина «Надежность изделия»:

- **Под надежностью изделия понимают свойство изделия выполнять заданные функции, сохраняя свои эксплуатационные показатели в определенных пределах в течение требуемого промежутка времени или требуемой наработки при соблюдении режимов эксплуатации, правил технического обслуживания, хранения и транспортировки.**
- Под надежностью изделия понимают свойство изделия длительно сохранять работоспособность до предельного состояния с необходимыми перерывами для профилактического обслуживания.
- Под надежностью изделия понимают свойство изделия поддерживать свои эксплуатационные показатели в течение и после срока хранения и транспортировки, установленного технической документацией.
- Под надежностью изделия понимают состояние изделия, при котором оно способно выполнять заданные функции с параметрами, установленным требованиями технической документации.

39. Укажите согласно ГОСТ шесть единичных показателей надежности ЭВС:

- Коэффициент Готовности.
- **Вероятность безотказной работы.**
- **Гамма-процентная наработка до первого отказа.**
- **Интенсивность отказов.**
- **Средняя наработка до первого отказа.**
- **Средняя наработка на отказ.**
- **Параметр потока отказов.**

40. Укажите согласно ГОСТ три основных комплексных показателей надежности ЭВС:

- Коэффициент средней наработки на отказ.

- Коэффициент готовности.
- Коэффициент технического использования.
- Коэффициент оперативной готовности.

б) типовые практические задания:

Практическое задание 1. Метрики Холстеда

Для исходного заданного текста программы определить значения метрик Холстеда, на основе которых дать оценку качества исходной программы.

Практическое задание 2. Метрики Джилба

В предлагаемых задачах необходимо разработать программу, реализующую алгоритм решения по приведенному условию, а затем оценить характеристики разработанной программы на основе лексического анализа текста и применения метрик Джилба.

Задача 1. Определить число, образованное k старшими цифрами введенного с клавиатуры натурального числа. Исходное число и значение k вводятся с клавиатуры. Пример: для числа 456771 и $k = 2$ искомое число равно 45.

Задача 2. Функция $F(x, y)$ задана следующим образом:

$$F(x, y) = \begin{cases} x - y, & \text{если } x \leq y; \\ x + y, & \text{если } x > y. \end{cases}$$

Вывести на экран в виде таблицы значения функции $F(x, y)$ для значений аргументов $x = 0,5$ — $0,7$ с шагом $0,1$ и $y = 0,2$ — $1,0$ с шагом $0,2$.

Практическое задание 3. Метрики Чепина

В задачах, предлагаемых для самостоятельного решения, необходимо выполнить следующее:

-Разработать программу, реализующую заданный алгоритм (рекомендуется использовать язык программирования C#, C++);

-Сформировать таблицы по образцу, приведённому в рассмотренных задачах;

- На основе лексического анализа исходного текста программы определить метрики Чепина;

- Провести анализ полученных результатов, сформировав содержательные выводы.

Задача 1. В целочисленной матрице A , размером $N \times M$ заменить элементы главной диагонали на номера столбцов (числа N и M задаются с клавиатуры в диапазоне от 3 до 10). Первоначальное заполнение матрицы осуществить при помощи датчика случайных чисел в диапазоне от -10 до 10. Исходную и видоизменённую матрицу вывести на экран.

Задача 2. В целочисленной матрице A размером $N \times M$ заменить элементы главной диагонали на значения элементов одномерного массива B (числа N и M задаются с клавиатуры в диапазоне от 3 до 10). Первоначальное заполнение массивов A и B осуществить при помощи датчика случайных чисел в диапазоне от -10 до 10. Исходную и видоизменённую матрицу вывести на экран.

Задача 3. В целочисленной матрице A размером $N \times M$ заменить элементы столбца, расположенного по центру, на значения элементов одномерного массива B (числа N и M задаются с клавиатуры в диапазоне от 3 до 10). Первоначальное заполнение массивов A и B осуществить при помощи датчика случайных чисел в диапазоне от -5 до 15. Исходную и видоизменённую матрицы, а также массив B вывести на экран.

Практическое задание 4. Оценка структурной (алгоритмической) сложности программы с использованием Метрик Маккейба

В предлагаемых для самостоятельного решения задачах необходимо выполнить следующее с целью оценки алгоритмической сложности:

- разработать алгоритм решения задачи;

- построить граф потока управления;
- сформировать маршруты тестирования в соответствии с критериями метрик Маккейба;
- определить значение цикломатического числа, характеризующего структурную сложность программ;
- сформировать матрицы смежности и достижимости;
- провести анализ полученных результатов, сформировав содержательные выводы.

Задача 1. Вывести на экран все натуральные числа из диапазона от А до В, сумма цифр которых равна S. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В и заданная сумма цифр S вводятся с клавиатуры.

Задача 2. Вывести на экран все натуральные трехзначные и пятизначные числа из диапазона от А до В, значение которых кратно 13. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В вводятся с клавиатуры.

Задача 3. Вывести на экран все натуральные числа из диапазона от А до В, у которых совпадают старшая и младшая цифры. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона А и В вводятся с клавиатуры.

Практическое задание 5. Оценка надежности ПС

Для заданного варианта задачи (текста программных) выполнить оценку надежности программных средств с использованием модели Миллса.

Практическое задание 6. Для заданного варианта задачи (текста программы) выполнить расчет надежности программного обеспечения с использованием модели Нельсона по результатам испытаний.

Практическое задание 7. Для заданного типового варианта задачи (программного средства) выполнить расчет значений характеристик качества программных средств с использованием модели качества, регламентированной в ГОСТ 28195, ГОСТ 9126.

Практическое задание 8. Решить задачу определения вероятности безотказной работы электронного устройства для случая последовательной структурной схемы включения его элементов.

Практическое задание 9. Решить задачу определения вероятности безотказной работы электронного устройства для случая параллельной структурной схемы включения его элементов.

Практическое задание 10. Решить задачу определения надежности электронного устройства для случая, когда необходимо использовать метод преобразования структурной схемы устройства по базовому элементу.

Практическое задание 11. Найти среднюю наработку до первого отказа и вероятность безотказной работы устройства, для заданного варианта структурной модели надежности, при его двукратном резервировании и заданных значениях вероятности безотказной работы его элементов.

Практическое задание 12. Решить типовые задачи по теории надежности электронных устройств

Задача 1. Аппаратура связи состоит из 2000 элементов, средняя интенсивность отказов которых $\lambda_{\text{ср}} = 0,33 \cdot 10^{-5}$ 1/час. Необходимо определить вероятность безотказной работы аппаратуры в течении $t = 200$ час и среднее время безотказной работы аппаратуры.

Задача 2. Невосстанавливаемая в процессе работы электронная машина состоит из 200000 элементов, средняя интенсивность отказов которых $\lambda_{\text{ср}} = 0,2 \cdot 10^{-6}$ 1/час. Требуется определить вероятность безотказной работы электронной машины в течении $t = 24$ часа и среднее время безотказной работы электронной машины.

Задача 3. Система управления состоит из 6000 элементов, средняя интенсивность отказов которых $\lambda_{\text{ср}} = 0,16 \cdot 10^{-6}$ 1/час. Необходимо определить вероятность безотказной работы в течении $t = 50$ час и среднее время безотказной работы.

Задача 4. Прибор состоит из $n = 5$ узлов. Надежность узлов характеризуется вероятностью безотказной работы в течение времени t , которая равна: $P_1(t)=0,98$; $P_2(t)=0,99$; $P_3(t)=0,998$; $P_4(t)=0,975$; $P_5(t)=0,985$. Необходимо определить вероятность безотказной работы прибора.

Задача 5. Система состоит из пяти приборов, среднее время безотказной работы которых равно: $mt_1=83$ час; $mt_2=220$ час; $mt_3=280$ час; $mt_4=400$ час; $mt_5=700$ час. Для приборов справедлив экспоненциальный закон надежности. Требуется найти среднее время безотказной работы системы.

Задача 6. Прибор состоит из пяти блоков. Вероятность безотказной работы каждого блока в течение времени $t = 50$ час равна: $P_1(50)=0,98$; $P_2(50)=0,99$; $P_3(50)=0,998$; $P_4(50)=0,975$; $P_5(50)=0,985$. Справедлив экспоненциальный закон надежности. Требуется найти среднее время безотказной работы прибора.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СОГЛАСОВАНО

ФГБОУ ВО "РГРТУ", РГРТУ, Гусев Сергей Игоревич,
Проректор по научной работе и инновациям

24.07.25 11:58 (MSK)

Простая подпись