МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ
Директор ИЭиАС
_____ В.Р. Храмшин

13.02.2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

МЕТРОЛОГИЯ И СТАНДАРТИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	3
Семестр	6

Магнитогорск
2024 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования
25.01.2024, протокол № 5

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС
13.02.2024 г. протокол № 4

Председатель _____ В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры ВТиП, канд. техн. наук _____ Н.С. Сибилева

Рецензент:
Директор НИИ «Промбезопасность» , д-р. техн. наук _____ М.Ю. Наркевич

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины «Метрология и стандартизация программного обеспечения» является ознакомление студентов с современными методами и средствами оценки надежности программного обеспечения, правовыми основами стандартизации, правилами и проведением сертификации программного обеспечения; овладение необходимым и достаточным уровнем общекультурных и профессиональных компетенций в соответствии с требованиями ФГОС ВО по направлению 09.03.01 "Информатика и вычислительная техника".

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Метрология и стандартизация программного обеспечения входит в обязательную часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Программирование

Информатика

Объектно-ориентированное программирование

Структуры и модели данных

Графический дизайн интерфейсов

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Метрология и стандартизация программного обеспечения» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;	
ОПК-4.1	Применяет стандарты, участвует в разработке норм и правил, технической документации на различных этапах жизненного цикла информационных систем

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 акад. часов, в том числе:

– контактная работа – 72 акад. часов:

– аудиторная – 68 акад. часов;

– внеаудиторная – 4 акад. часов;

– самостоятельная работа – 36,3 акад. часов;

– в форме практической подготовки – 0 акад. час;

– подготовка к экзамену – 35,7 акад. час

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Раздел 1. Основные понятия метрологии программных средств. Характеристики качества программных								
1.1 Определения понятия метрологии. История развития метрологии. Определения качества программных средств. Внутренние и внешние дестабилизирующие факторы при разработке программных средств.	6	4				1. Подготовка к лабораторному занятию. 2. Самостоятельное изучение учебной литературы.	Беседа-обсуждение	ОПК-4.1
1.2 Ошибки при разработке программных средств. Метрики качества программных средств.		2	2			1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы. 3. Самостоятельное изучение учебной литературы.	Устный опрос	ОПК-4.1
Итого по разделу		6	2					
2. Раздел 2. Метрическая теория программ								
2.1 Основные задачи метрической теории программ. Классификация моделей оценки характеристик программных средств	6	4	2			1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы.	1. Проверка индивидуальных заданий. 2. Устный опрос.	ОПК-4.1
2.2 Метрики лексической сложности программных средств. Метрики структурной сложности программных средств		4	2		11,3	1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы.	1. Проверка индивидуальных заданий 2. Устный опрос.	ОПК-4.1

2.3 Метрики процедурной сложности программных средств. Метрики объектной сложности программных средств.		4	6		10	1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы.	1. Проверка индивидуальных заданий 2. Устный опрос.	ОПК-4.1
Итого по разделу		12	10		21,3			
3. Раздел 3. Стандартизация программных средств								
3.1 История становления стандартизации. Объекты и составляющие стандартизации. Цели и задачи стандартизации.	6	2	2		10	1. Подготовка к лабораторному занятию.	1. Устный опрос.	ОПК-4.1
3.2 Документы в области стандартизации. Функции стандартизации. Информационное обеспечение стандартизации. Программа РФ по стандартизации в сфере информатизаци.		2	6		5	1. Подготовка к лабораторному занятию.	1. Устный опрос.	ОПК-4.1
3.3 Стандарты документирования программных средств. Проблемы современной стандартизации.		3	4			1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы.	1. Проверка индивидуальных заданий 2. Устный опрос.	ОПК-4.1
Итого по разделу		7	12		15			
4. Раздел 4. Сертификация программных средств								
4.1 Основные понятия сертификации. Цели подтверждения соответствия. Знаки подтверждения соответствия.	6	4	6			1. Подготовка к лабораторному занятию. 2. Выполнение лабораторной работы.	1. Проверка индивидуальных заданий 2. Устный опрос.	ОПК-4.1
4.2 Схемы проведения процедуры сертификации. Состояние и перспективы развития сертификации. Проблемы и направления совершенствования системы подтверждения		5	4			1. Изучение учебной литературы.	Устный опрос	ОПК-4.1
Итого по разделу		9	10					
Итого за семестр		34	34		36,3		экзамен	
Итого по дисциплине		34	34		36,3		экзамен	

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

В ходе проведения лекционных занятий предусматривается:

- использование электронного демонстрационного материала по темам, требующим иллюстрации работы программных продуктов: MS Word, MS Excel, MS Visio;
- организация дискуссий при обсуждении эффективности работы рассматриваемых алгоритмов.

В ходе проведения всех лабораторных занятий предусматривается использование средств вычислительной техники при выполнении индивидуальных заданий .

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе личностно значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее за-планированными ошибками), лекция-беседа, лекция-дискуссия, лекция-пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы с знаниями в различных предметных областях.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения / Т. Н. Ананьева, Г. Н. Исаев, Н. Г. Новикова. — Москва : ИНФРА-М, 2019. — 232 с. [Режим доступа https://new.znanium.com]. - (Высшее образование: Бакалавриат). - www.dx.doi.org/10.12737/18657.

2. Черников, Б. В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. - Москва : ИД ФОРУМ: ИНФРА-М, 2015. - 240 с. - (Высшее образование). - https://new.znanium.com/catalog/product/256901.

б) Дополнительная литература:

1. Схиртладзе, А.Г. Метрология, стандартизация и сертификация в 2 томах: учебник для академического бакалавриата. 5-е изд., доп., перераб. / А.Г. Схиртладзе, А.Г. Схиртлад-зе– М.: Издательство Юрайт, 2015. – 831 с.

2. Сергеев, А.Г. Метрология, стандартизация и сертификация : учебник и практикум для академического бакалавриата. 2-е изд., доп., перераб./ А.Г. Сергеев, В.В. Терегеря. – М. : Издательство Юрайт, 2015. – 838 с.Лифиц, И.М. Стандартизация, метрология и подтверждение соответствия. 11-е изд., доп. перераб./ И.М. Лифиц. – М.: Издательство Юрайт, 2016. – 411 с.

3. Райкова, Е.Ю. Стандартизация, подтверждение соответствия, метрология : учебник для прикладного бакалавриата / Е.Ю. Райкова. 1-е изд. – М.: Издательство Юрайт, 2016. – 349 с.

4. Визуализация результатов научной деятельности : учеб. пособие / О.С. Логунова и др. – Магнитогорск, 2015. – 85 с.

5. Жуков, В.К. Метрология. Теория измерений: учебное пособие для бакалавриата и магистратуры / В.К. Жуков. – М.: Издательство Юрайт, 2016. – 414 с.

в) Методические указания:

1. Логунова, О.С. Стандартизация и метрология программного обеспечения: методические указания к выполнению курсовой работы по дисциплине «Метрология, стандартизация, сертификация» для студентов, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника» / О.С. Логунова, Е.А. Ильина, Н.С. Сибилева, А.Ю. Миков. Магнитогорск: Магнитогорский государственный технический университет им. Г.И. Носова, 2016. 24 с.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
Anaconda Python	свободно распространяемое ПО	бессрочно
Borland Turbo C++	№112301 от 23.11.2005	бессрочно
Eclipse	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
NetBeans	свободно распространяемое ПО	бессрочно
Oracle Open JDK	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка

Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.

3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.

6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372

Приложение 1

Учебно-методическое обеспечение самостоятельной работы обучающихся

Лабораторная работа №1 «Оценка характеристик программы на основе лексического анализа»

Необходимо определить значения метрик Холстеда и осуществить оценку качества разработанного исходного кода программы на любом известном Вам языке программирования согласно варианту. Свести полученные значения в таблицу. Предусмотреть ввод необходимых значений с клавиатуры, генерацию случайных значений величин и вывод полученных решений на экран.

Варианты заданий:

1. Составить программу, которая позволяет для матрицы размерностью m*n вычислить произведение положительных, сумму отрицательных и количество нулевых элементов.
2. Двумерный массив C содержит пять строк и столбцов, одномерный массив A содержит 5 элементов. Вычислить произведение матрицы на вектор.
3. Дана матрица B. Для каждого столбца с четным номером вычислить и напечатать сумму квадратов элементов этого столбца, а для каждого столбца с нечетным номером вычислить произведение элементов.
4. Найти минимальный элемент j-го столбца матрицы A размером n*n, для которого сумма абсолютных значений элементов максимальна (если таких столбцов несколько, взять первый из них).
5. Дан двумерный массив A. Каждый элемент массива, стоящий выше главной диагонали, заменить его квадратом, а ниже диагонали – кубом его значения. Элементы главной диагонали оставить без изменения.
6. Задан двумерный массив B. Найти в нём максимальный элемент и разделить на него каждый элемент массива B.
7. Заданы матрица n*n и число k<n. Строку с максимальным по модулю элементом в k-м столбце переставить с k-й строкой.
8. Дана матрица B. Для каждого столбца матрицы вычислить и напечатать разность между квадратом суммы и суммой квадратов элементов этого столбца.
9. Заданы двумерный массив 5*5 и число K. Разделить элементы K-й строки на диагональный элемент, расположенный в данной строке.
10. Определить и напечатать среднее значение элементов двухмерного массива размером n*m. Найти индексы элемента, наиболее близкого по значению к среднему.
11. Составить программу нахождения минимального положительного элемента матрицы размерностью m*n, а также значение индексов этого элемента.
12. Составить программу для определения количества отрицательных, положительных и равных нулю элементов матрицы R(m*n).
13. Составить программу вычисления суммы и произведения элементов квадратной матрицы A размерностью n*n, расположенных ниже главной диагонали.
14. Составить программу для преобразования квадратной матрицы, состоящего в симметричной относительно главной диагонали перестановки ее элементов.
15. Дана действительная матрица размера m*n. Найти значение наибольшего по модулю элемента матрицы, а также индексы всех элементов с найденными значениями модуля.
16. Дана действительная матрица размера m*n. Найти максимальные элементы по строкам и их сумму.
17. Задана квадратная матрица размером k*k. Составить программу вычисления суммы и произведения элементов матрицы, находящихся на главной диагонали.

Ответ на задание оформить в виде отчета, содержащего:

1. Исходный код программы (с нумерацией строк кода).

2. Расчеты характеристик качества кода. Расчеты представить в виде формул. Исходные данные для расчета характеристик представить в виде таблицы операторов и операций, таблицы операндов и таблицы входных и выходных переменных. Образец таблицы представлен в таблице 1.1. Итоговые показатели свести в таблицу, образец которой представлен в таблице 1.2.

Таблица 1 – Операнды программы

№	Операнд	Номера строк	Количество повторений
1	System	1	1
	…	…	
		Всего	

3. Итоговую таблицу показателей.

Таблица 2 – Итоговая таблица показателей

№	Характеристика	Обозначение и формула для вычисления	Значение
1	Число уникальных операторов и операций	n_1	
2	Число уникальных операндов	n_2	
3	Общее число всех операторов и операций	N_1	
4	Общее число всех операндов	N_2	
5	Число входных и выходных переменных (параметров)	n_2^*	
6	Словарь программы	$n = n_1 + n_2$	
7	Длина реализации программы	$N = N_1 + N_2$	
8	Объем программы	$V = (N_1 + N_2)*log_2(n_1 + n_2)$	
9	Потенциальный объем программы	$V^* = (n_2^*+2)*log_2(n_2^*+2)$	
10	Уровень реализации программы	$L=V^*/V$	
11	Уровень реализации языка	$\lambda=L*V^*$	
12		Основные выводы	

Лабораторная работа №2 «Оценка структурной сложности программы»

Задание. Руководствуясь приведенными в таблице 2.1 вариантами задач, необходимо выполнить следующие задания:

1. Разработать и реализовать алгоритм решения задачи.
2. Построить граф потока управления.
3. Обозначить возможные маршруты тестирования в соответствии с первым и вторым критериями.
4. Сформировать матрицы смежности и достижимости.
5. Определить значение цикломатического числа, характеризующего цикломатическую сложность программы.
6. Оформить выводы на основании полученных результатов.

Таблица 2.1 – Варианты решения задачи

№ варианта	Условие задачи
1	Необходимо определить, является ли введенное с клавиатуры число простым, а также определить количество цифр, составляющих это число.
2	Определить максимальное натуральное число, не превышающее *R*, которое без остатка делится на *P*. *R* и *P* необходимо вводить с клавиатуры.
3	Вывести на экран все двухзначные натуральные числа из диапазона от *C* до *D*, значение которых кратно 17. При отсутствии подобных чисел вывести сообщение «Необходимых чисел нет».

№ варианта	Условие задачи
4	Вывести на экран все натуральные числа из диапазона от *C* до *D*, сумма цифр которых равна *Q*. При отсутствии подобных чисел вывести сообщение «Необходимых чисел нет».
5	Для натурального числа, заданного с клавиатуры получить новое число, дописав к нему цифру *s* (также введенную с клавиатуры) в начале и в конце.
6	Для натурального числа, заданного с клавиатуры получить новое число, которое будет являться суммой первой и последней цифр заданного натурального числа.
7	Найти в промежутке от *C* до *D* минимальное натуральное число, имеющее наибольшее количество делителей.
8	Найти произведение всех цифр введенного с клавиатуры *n*-значного числа.
9	Для натурального числа, заданного с клавиатуры получить новое число, в котором будут изменены местами первая и последняя цифры исходного числа.
10	Вывести на экран все натуральные числа из диапазона от *C* до *D*, в записи которых цифра 6 встречается *K* раз. При отсутствии подобных чисел вывести сообщение «Необходимых чисел нет».
11	Вывести на экран все натуральные шестизначные числа из диапазона от *C* до *D*, у которых совпадают суммы трех младших и трех старших цифр. При отсутствии подобных чисел вывести сообщение «Необходимых чисел нет».
12	Для введенного с клавиатуры натурального числа найти наибольшую и наименьшую его цифры.
13	Дано натуральное число *k*. Среди всех *m*-значных чисел найти те, сумма цифр которых равна заданному натуральному числу.
14	Найти первый отрицательный элемент последовательности $cos(tg(x_i))$, если x_i – запрашивается программой, а $x_{i+1}=x_i+0,4$
15	Найти квадрат суммы всех цифр введенного с клавиатуры *n*-значного числа.
16	Дано целое число в диапазоне от 1 до 100000. Проверить, присутствуют ли одновременно в записи числа цифры 2 и 7. При отсутствии подобных чисел вывести сообщение «Необходимых цифр нет».
17	Дано целое число в диапазоне от 1 до 100000. Проверить, присутствуют ли в записи числа четные цифры. При отсутствии подобных цифр вывести сообщение «Необходимых цифр нет».

Результаты оформить в виде отчета, содержащего:

1. Исходный код решения задачи (с нумерацией строк).
2. Разработанный алгоритм решения задачи, оформленный согласно ГОСТ 19.701-90.
3. Граф потока управления (с выделенными ветвлениями).
4. Возможные маршруты тестирования в соответствии с первым и вторым критериями.
5. Матрицы смежности и достижимости для каждого случая.
6. Расчет значения цикломатического числа для каждого критерия.
7. Выводы на основании полученных результатов.

Лабораторная работа № 3 «Оценка функциональной сложности программы»

Задание. Необходимо разработать программу согласно варианту из *таблицы 3.6* на любом известном Вам языке программирования и оформить отчет по выполненной работе.

В программе предусмотреть ввод необходимых значений с клавиатуры, и, самое важное, декомпозицию программного кода на функции в зависимости от их назначения. Например, отдельная функция для вывода данных в приемлемом для задачи виде, функция сортировки данных, функция случайной генерации значений и т.д. и т.п.

Количество дополнительных функций (помимо основной) должно быть не менее четырех.

После разработки программы необходимо сопроводить ее блок-схемой реализованного алгоритма и произвести анализ качества программного кода с помощью следующих метрик:

1. Метрики на основе функциональных указателей

1.1 Определить количество функциональных указателей, указанных в таблице 3.1, и свести полученные значения в таблицу 3.2:

Таблица 3.1 – Виды оценочных элементов

Оценочный элемент	**Расшифровка**
j_1	Количество внешних вводов данных пользователем
j_2	Количество внешних выводов данных
j_3	Количество внешних запросов
j_4	Количество локальных внутренних логических файлов
j_5	Количество внешних интерфейсных файлов

Таблица 3.2 – Типовая таблица для расчета количества функциональных указателей

Характеристика	Количество с учетом сложности			Итого
	Низкая	Средняя	Высокая	
Внешние вводы	*u*3 =*	*u*4 =*	*u*5 =*	
Внешние выводы	*u*4 =*	*u*5 =*	*u*7 =*	
Внешние запросы	*u*3 =*	*u*4 =*	*u*6 =*	
Внутренние логические файлы	*u*7 =*	*u*10 =*	*u*15 =*	
Внутренние интерфейсные файлы	*u*5 =*	*u*7 =*	*u*10 =*	
Общее количество				

В таблице 3.2 за *u* принято количество конкретных показателей для каждой из характеристик. При этом, тип сложности обращений к данным определяется следующим образом, как показано в таблице 3.3:

Таблица 3.3 – Сложность данных в зависимости от их количества

Количество элементов данных	**Низкая сложность**	**Средняя сложность**	**Высокая сложность**
Внешние вводы	1-4	5-19	>19
Внешние выводы	1-4	5-19	>19
Внешние запросы	1-4	5-19	>19
Внутренние логические файлы	1-19	20-50	>50
Внутренние интерфейсные файлы	1-19	20-50	>50

1.2 На основе рассчитанного в таблице 3.2 значения функциональных указателей, необходимо определить коэффициенты регулировки сложности, дав ответы на вопросы из таблицы 3.4 согласно критериям из таблицы 5.

Таблица 3.4 – Вопросы для расчета функциональных показателей

№	**Вопрос**
1	Какое влияние имеет наличие средств передачи данных?
2	Какое влияние имеет распределенная обработка данных?
3	Какое влияние имеет распространенность используемой аппаратной платформы?
4	Какое влияние имеет критичность к требованиям производительности и ограничению времени ответа?
5	Какое влияние имеет частота транзакций?
6	Какое влияние имеет ввод данных в режиме реального времени?
7	Какое влияние имеет эффективность работы конечного пользователя?

№	Вопрос
8	Какое влияние имеет оперативное обновление локальных файлов в режиме реального времени?
9	Какое влияние имеет скорость обработки данных (вычислений)?
10	Какое влияние имеет количество и категории пользователей?
11	Какое влияние имеет легкость инсталляции?
12	Какое влияние имеет легкость эксплуатации?
13	Какое влияние имеет разнообразие условий применения?
14	Какое влияние имеет простота внесения изменений?

Таблица 3.5 – Критерии ответов на вопросы

№ *k*	**Варианты ответов на поставленные вопросы**
1	Случайное влияние
2	Небольшое, эпизодическое влияние
3	Среднее влияние
4	Важное, значительное влияние
5	Основное, существенное влияние

1.3 На основании полученных значений рассчитать метрики FP.

1.4 Сделать выводы

2. Метрики связности модулей

2.1 Для каждого реализованного модуля (функции) согласно методике определения типа связности рассчитать значение силы связности.

2.2 Сделать выводы.

3. Метрики сцепления модулей

3.1 Для каждого реализованного модуля (функции) согласно типу модуля по шкале сцепления, определить значение силы сцепления.

3.2 Сделать выводы.

4. Общие выводы по работе

Варианты решения задач:

Таблица 3.6 – Варианты решения задачи

№ варианта	Условие задачи
1	Реализовать программу, которая должна преобразовывать введенную с клавиатуры строку восьмеричных цифр в эквивалентное ей целое десятичное число
2	Реализовать программу, которая должна рассчитать, сколько раз в заданной с клавиатуры строке встретился указанный с клавиатуры символ
3	Реализовать программу, которая должна в прямоугольной матрице m*n, значения элементов которой находятся в промежутке (-50; 40) находить сумму элементов i-строки и j-столбца.
4	Реализовать программу, которая должна по введенной строке, содержащей цифры и буквы, сформировать новую строку, состоящую только из букв изначальной строки.
5	Реализовать программу, которая должна осуществлять зеркальное отражение введенной с клавиатуры строки
6	Реализовать программу, которая должна осуществлять поиск минимального и максимального значений элементов из числа тех, которые не встречаются в массиве. Диапазон значений задан в промежутке от 12 до 56.
7	Реализовать программу, которая должна осуществлять поиск частоты встречаемости каждого из элементов массива, а затем выводить в порядке возрастания частоты встречаемости и элемент массива, и частоту его встречаемости. Элементы массива должны находиться в промежутке от 19 до 78.
8	Реализовать программу, которая должна подсчитывать количество

№ варианта	Условие задачи
	отрицательных элементов в массиве и выводить их в порядке убывания. Диапазон значений находится в промежутке от -34 до 20.
9	Реализовать программу, которая должна во введенной пользователем строке осуществить замену всех знаков «,» на знак «-» начиная с заданной позиции, а затем подсчитать количество произведенных замен.
10	Реализовать программу, которая должна сжать введенную с клавиатуры строку, удалив из нее все встречающиеся пробелы
11	Реализовать программу, которая должна преобразовать введенную строку двоичных цифр в соответствующее десятичное число.
12	Реализовать программу, которая должна осуществить замену каждого третьего символа в заданной пользователем строке на символ, также введенный с клавиатуры
13	Реализовать программу, которая должна находить все значения из заданного пользователем диапазона, которые не встречаются в качестве значений элементов массива *p*.
14	Реализовать программу, которая должна определить количество простых чисел в заданном диапазоне и вывести их на экран. В случае, если простых чисел в указанном диапазоне нет, предложить расширить диапазон.
15	Реализовать программу, которая должна преобразовать введенную строку шестнадцатеричных цифр в соответствующее десятичное число.
16	Реализовать программу, которая должна найти произведение и сумму всех цифр введенного с клавиатуры *n*-значного числа.
17	Реализовать программу, которая должна найти максимальный элемент в одномерном массиве и заменить им каждый четный элемент массива.

Лабораторная работа №4 «Оценка качества программы с помощью объектно-ориентированных метрик. Часть 1»

Задание. На основании условий задачи, представленных в таблице 4.1, необходимо выполнить следующие пункты:

1. Разработать блок-схемы алгоритмов каждого из методов класса (2 шт.).

2. Разработать блок-схему общей работы программного продукта.

3. Запрограммировать задание согласно варианту.

4. Оценить качество программного продукта с помощью метрик ***Мартина*** (центростремительное сцепление, центробежное сцепление, нестабильность, абстрактность, расстояние до главной последовательности, нормализованное расстояние до главной последовательности).

5. Сделать содержательные выводы.

6. Оформить отчет о проделанной работе и загрузить его на образовательный портал.

Таблица 4.1 – Варианты условий задачи для лабораторной работы

№	Задание
1	Создать класс «Заправочная станция». В классе необходимо определить следующие поля: идентификатор АЗС, наименование АЗС, месторасположение АЗС. Идентификатор АЗС имеет следующий формат: ССС-ЦЦЦ (С – символ, Ц – цифра). Необходимо предусмотреть: вывод информации об АЗС в заданном пользователем регионе,

№	Задание
	вывод информации о всех АЗС, таким образом, чтобы первые цифры в идентификаторах АЗС образовывали неубывающую последовательность.
2	Создать класс «Заправочная колонка». В классе необходимо определить следующие поля: идентификатор колонки, производитель колонки, наименование АЗС, доступное количество литров бензина. Идентификатор АЗС имеет следующий формат: ЦЦЦС (С – символ, Ц – цифра). Необходимо предусмотреть: вывод информации о всех колонках заданного производителя по убыванию значения второй цифры в идентификаторе колонки, отображение доступного количества литров бензина в колонках производителя «Некут».
3	Создать класс «Заказы автосервиса». В классе необходимо предусмотреть следующие поля: фамилия клиента, номерной знак автомобиля, название станции, адрес станции. Номерной знак автомобиля имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). На основе имеющегося класса определить поле «Сумма заказа на обслуживание». После чего определить и вывести информацию о клиентах станции, которые имеют минимальную и максимальную сумму заказов, а также вывести информацию о клиентах, обслуживающихся на заданной станции.
4	Создать класс «Заказы эвакуатора». В классе необходимо предусмотреть следующие поля: эвакуируемый автомобиль, грузоподъемность эвакуатора, телефон для вызова эвакуатора, тип эвакуатора (грузовой, с лебедкой, с частичной погрузкой и т.д.). Поле «эвакуируемый автомобиль» имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). Вывести информацию (в порядке возрастания грузоподъемности) об эвакуаторах, которые эвакуировали автомобиль с введенным с клавиатуры номером. Найти информацию о наиболее редко вызываемом эвакуаторе, основываясь на данных из поля «Телефон для вызова».
5	Создать класс «Автозапчасти». В классе необходимо предусмотреть следующие поля: стоимость автозапчасти, доступное на складе количество, производитель, габариты (строка из 11 символов). Поле «Габариты» имеет следующий формат: ЦЦЦхЦЦЦхЦЦЦ (Ц-цифра, х-символ-разделитель). Найти информацию о автозапчастях определенного производителя и вывести данные в порядке возрастания второго блока цифр в поле «Габариты». Вывести информацию о самой дорогой запчасти на складе, габариты которой не превышают значения, введенного с клавиатуры.
6	Создать класс «Дорожно-транспортное происшествие». В классе необходимо предусмотреть следующие поля: место происшествия (строка до 60 символов), информация о страховой компании первого участника, информация о страховой компании второго участника, сумма причиненного ущерба, обстоятельства. Вывести информацию о происшествиях (в порядке убывания суммы ущерба), сумма причиненного ущерба в которых превышает число, введенное с клавиатуры. Найти информацию о происшествиях, обстоятельства которых совпадают с тем, которое введено с клавиатуры.
7	Создать класс «Штраф». В классе необходимо предусмотреть следующие поля: фамилия нарушителя, дата события, тип нарушения, сумма штрафа. Вывести информацию о штрафах в порядке возрастания их стоимости, тип нарушения штрафов задается с клавиатуры. Вывести информацию о трех самых частых типах нарушений, стоимость которых превышает число, заданное с клавиатуры.
8	Создать класс «Автомобиль». В классе необходимо предусмотреть следующие поля: марка автомобиля, владелец автомобиля, год выпуска автомобиля, регистрационный номер автомобиля. Регистрационный номер автомобиля имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). Вывести информацию об автомобилях в порядке возрастания года выпуска, чей год выпуска меньше заданного. Вывести информацию об автомобилях заданной марки в порядке убывания последней цифры в номере автомобиля.
9	Создать класс «Заправочная станция». В классе необходимо определить следующие поля: идентификатор АЗС, наименование АЗС, владелец АЗС. Идентификатор АЗС имеет следующий формат: СЦЦССЦЦ (С – символ, Ц – цифра). Необходимо предусмотреть: вывод информации о владельце АЗС по заданному названию АЗС, корректировку информации о владельце АЗС в случае, если ее наименование «Морпзаг».
10	Создать класс «Заправочная колонка». В классе необходимо определить следующие поля: идентификатор колонки, адрес установки колонки, наименование АЗС, доступное количество

№	Задание
	литров бензина. Идентификатор АЗС имеет следующий формат: СССЦЦ (С – символ, Ц – цифра). Необходимо предусмотреть: вывод информации об адресах установки колонок заданного наименования АЗС в алфавитном порядке идентификаторов колонок, отображение доступного количества литров бензина в колонках, наименование АЗС при этом «МагниткаОйл».
11	Создать класс «Заказы автосервиса». В классе необходимо предусмотреть следующие поля: фамилия клиента, номерной знак автомобиля, название станции, адрес станции. Номерной знак автомобиля имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). Определить и вывести информацию о клиентах станции, номера автомобилей которых содержат последовательность цифр, введенную с клавиатуры, а также вывести информацию о станциях, на которых обслуживался определенный клиент.
12	Создать класс «Заказы эвакуатора». В классе необходимо предусмотреть следующие поля: эвакуируемый автомобиль, грузоподъемность эвакуатора, телефон для вызова эвакуатора, тип эвакуатора (грузовой, с лебедкой, с частичной погрузкой и т.д.). Поле «эвакуируемый автомобиль» имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). Найти информацию об автомобилях, когда либо эвакуированных эвакуатором м заданной с клавиатуры грузоподъемностью и вывести данные в порядке убывания последовательности цифр в номере автомобиля. Определить грузоподъемность и телефон для вызова эвакуатора на основании типа эвакуатора, введенного с клавиатуры.
13	Создать класс «Автозапчасти». В классе необходимо предусмотреть следующие поля: стоимость автозапчасти, доступное на складе количество, производитель, габариты. Поле «Габариты» имеет следующий формат: ЦЦЦхЦЦЦхЦЦЦ (Ц-цифра, х-символ-разделитель). Найти минимальную и максимальную по габаритам запчасть производителя, введенного с клавиатуры. Узнать, сколько запчастей определенного производителя присутствует на складе.
14	Создать класс «Штраф». В классе необходимо предусмотреть следующие поля: фамилия нарушителя, дата события, тип нарушения, сумма штрафа. Найти информацию о штрафах определенного нарушителя и отобразить ее в порядке убывания стоимости штрафа. Найти нарушения с суммой штрафа более 500 рублей, совершенных в период, начало и конец которого введены с клавиатуры.
15	Создать класс «Дорожно-транспортное происшествие». В классе необходимо предусмотреть следующие поля: место происшествия, информация о страховой компании первого участника, информация о страховой компании второго участника, сумма причиненного ущерба, обстоятельства. Вывести информацию о происшествиях в порядке убывания суммы ущерба, при условии, что страховые компании обоих участников совпадают. Найти место происшествия, в котором совершено наибольшее количество происшествий.
16	Создать класс «Автомобиль». В классе необходимо предусмотреть следующие поля: марка автомобиля, владелец автомобиля, год выпуска автомобиля, регистрационный номер автомобиля. Регистрационный номер автомобиля имеет следующий формат: СЦЦЦСС (С – символ, Ц – цифра). Вывести информацию об автомобилях в порядке убывания года выпуска, у которых в номере присутствует заданная последовательность цифр. Вывести информацию об автомобилях заданной марки в порядке возрастания года выпуска.

Лабораторная работа №5 «Оценка качества программы с помощью объектно-ориентированных метрик. Часть 2»

Взяв за основу Ваш вариант лабораторной работы №4, необходимо дополнить/переработать функционал программы таким образом, чтобы у Вас появились классы-наследники, или классы-родители со своими методами, использование которых позволит получить расширенную информацию об исследуемой задаче.

Задание. Осуществить анализ разработанной программы с помощью метрик **Лоренца и Кидда** (набор из 10 метрик); построить блок-схему алгоритма работы программы; привести

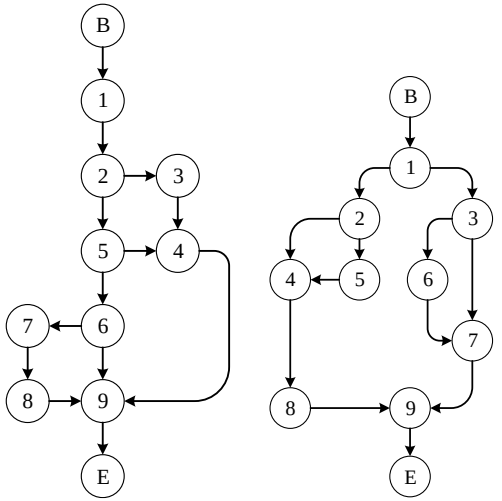
содержательные выводы касательно качества Вашей программы; оформить отчет о проделанной работе и загрузить его на образовательный портал.

Приложение 2

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ОПК-4: Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью		
ОПК-4.1	Применяет стандарты, участвует в разработке норм и правил, технической документации на различных этапах жизненного цикла информационных систем	*Перечень теоретических вопросов:* 1. Определение понятия «метрология». Виды метрологии. 2. Единство измерений. Функции измерений. 3. Основные составляющие метрологии. 4. Понятие и составляющие качества программного обеспечения. 5. Внутренние дестабилизирующие факторы, влияющие на качество программного обеспечения 6. Внешние дестабилизирующие факторы, влияющие на качество программного обеспечения 7. Понятия фактора качества, критерия качества, оценочного элемента, показателя качества, метрики качества 8. Внутреннее и внешнее качество программного обеспечения. 9. Внешние метрики, их назначение и особенности 10. Внутренние метрики, их назначение и особенности 11. Метрики качества в использовании, их связь с другими характеристиками ПО 12. Классификация моделей оценки характеристик ПО 13. Краткая характеристика и классификация метрик, основанных на лексическом анализе программы 14. Краткая характеристика и классификация метрик структурной сложности программы 15. Понятие графа потока управления, виды маршрутов в графе 16. Краткая характеристика и классификация

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		процедурно-ориентированных метрик. 17. Краткая характеристика и классификация объектно-ориентированных метрик *Примерные практические задания:* Согласно представленных на рисунках графах потока управления необходимо обозначить возможные маршруты тестирования в соответствии с первым и вторым критериями структурной сложности, определить значение цикломатического числа и сформировать оценку качества структурной сложности программного продукта на основании полученных результатов. 

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Метрология и стандартизация программного обеспечения» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по дисциплине проводится по результатам отчетности на практических занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.