

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ

Руководитель направления

доц., к.т.н., доц.

(должность, уч. степень, звание)

А.А. Ключарев

(инициалы, фамилия)



(подпись)

«13» мая 2019 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ


«Метрология программного обеспечения»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности	Проектирование программных систем
Форма обучения	заочная

Лист согласования рабочей программы дисциплины

Программу составил (а)

профессор, д.т.н., доцент
(должность, уч. степень, звание)

 10 мая 2019 г.
(подпись, дата)

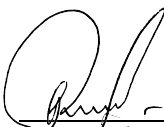
Е.В. Копкин
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«13» мая 2019 г, протокол № 08-2018/19

Заведующий кафедрой № 43

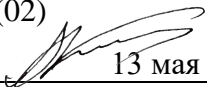
д.т.н., проф.
(уч. степень, звание)

 13 мая 2019 г.
(подпись, дата)

М.Ю. Охтилев
(инициалы, фамилия)

Ответственный за ОП ВО 09.03.04(02)

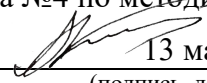
доц., к.т.н., доц.
(должность, уч. степень, звание)

 13 мая 2019 г.
(подпись, дата)

А.А. Ключарев
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н., доц.
(должность, уч. степень, звание)

 13 мая 2019 г.
(подпись, дата)

А.А. Ключарев
(инициалы, фамилия)

Аннотация

Дисциплина «Метрология программного обеспечения» входит в образовательную программу высшего образования по направлению подготовки/ специальности 09.03.04 «Программная инженерия» направленности «Проектирование программных систем». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-4 «Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью»

Содержание дисциплины охватывает круг вопросов, связанных с задачами оценивания метрических характеристик качества программного обеспечения на различных этапах его жизненного цикла, от разработки спецификаций до завершения отладки и тестирования программного продукта.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме зачета.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели Целями преподавания дисциплины являются изучение основных положений метрологии программного обеспечения, принципов построения, проектирования и использования средств для измерения характеристик и параметров программ, программных систем и комплексов, методов оценки эффективности программного обеспечения, методов оценки корректности программ, а также теоретическая и практическая подготовка обучающихся в области измерения основных метрических характеристик алгоритмов и процессов обработки данных по тексту программ; оценивания основных параметров будущего программного обеспечения (временных и объёмных, структурных, надёжных); оценивания сложности программ и программных комплексов на различных этапах их жизненного цикла.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.3.1 знает основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы ОПК-4.У.2 умеет применять стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы ОПК-4.В.3 имеет навыки составления технической документации на различных этапах жизненного цикла информационной системы

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- Информатика
- Основы программирования
- Объектно-ориентированное программирование
- Теория вероятностей
- Структуры и алгоритмы обработки данных
- Проектирование программных систем
- Управление качеством ПО

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

- Функциональное и логическое программирование
- Web-технологии
- Программирование мобильных устройств
- Программирование встроенных приложений

- Теория вычислительных процессов
- Управление программными проектами
- Разработка и анализ требований

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№7
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	3/ 108	3/ 108
Аудиторные занятия, всего час.	12	12
в том числе:		
лекции (Л), (час)	4	4
практические/семинарские занятия (ПЗ), (час)		
лабораторные работы (ЛР), (час)	8	8
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)		
Самостоятельная работа, всего (час)	96	96
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Зачет	Зачет

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 7					
Раздел 1. Модели и характеристики качества программного обеспечения	2		4		48
Раздел 2. Сложность и надежность программного обеспечения	2		4		48
Итого в семестре:	4		8		96
Итого	4	0	8	0	96

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
Раздел 1. Модели и характеристики качества программного обеспечения	Основные понятия качества программного обеспечения. Международная стандартизация в области качества программного обеспечения. Модели качества программного обеспечения. Характеристики и

	подхарактеристики качества программного обеспечения.
Раздел 2. Сложность и надежность программного обеспечения	Сложность программного обеспечения. Метрики сложности программного обеспечения. Основные понятия надежности программного обеспечения. Показатели надежности программного обеспечения. Модели надежности программного обеспечения.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего				

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела дисциплины
Семестр 7			
1	Составление вариантов тестирования и метрика покрытия требований	2	1
2	Оценка качества программного обеспечения	2	1
3	Определение сложности программного кода на основе лексического анализа	2	2
4	Оценка алгоритмической сложности программного кода	2	2
Всего		8	

4.5. Курсовое проектирование/ выполнение курсовой работы

Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 7, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	76	76

Курсовое проектирование (КП, КР)		
Расчетно-графические задания (РГЗ)		
Выполнение реферата (Р)		
Подготовка к текущему контролю успеваемости (ТКУ)	4	4
Домашнее задание (ДЗ)		
Контрольные работы заочников (КРЗ)	8	8
Подготовка к промежуточной аттестации (ПА)	8	8
Всего:	96	96

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр	Библиографическая ссылка / URL адрес	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004.4 Л 61	Тестирование компонентов и комплексов программ : учебник / В. В. Липаев ; РАН. Ин-т системного программирования. - М. : Синтег, 2010. - 392 с.	Студенческий отдел - 23
	Сертификация программных средств: учебник / В. В. Липаев; М.: СИНТЕГ, 2010. – 348 с. URL адрес: https://computer-museum.ru/books/lipaev/lip_sertifikacia.pdf	

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
http://internet-law.ru/gosts/gost/52164	ГОСТ Р ИСО 9000-2015 Системы менеджмента качества. Основные положения и словарь.
http://docs.cntd.ru/document/1200121069	ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов.
http://standartgost.ru/g/ГОСТ_Р_ИСО/МЭК	ГОСТ Р ИСО/МЭК 25021-2014 Информационные технологии. Системная и

_25021-2014	программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Элементы показателя качества
http://docs.cntd.ru/document/1200111327	ГОСТ Р ИСО/МЭК 25040-2014 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Процесс оценки
http://www.internet-law.ru/gosts/gost/19239/	ГОСТ 19781-90 Обеспечение систем обработки информации программное. Термины и определения.

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1	MATLAB 6.5/7.0
2	Microsoft Word или OpenOffice Writer

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Вычислительная лаборатория	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Зачет	Список вопросов; Тесты.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	<ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	<ul style="list-style-type: none"> – обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена
	Учебным планом не предусмотрено

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. Зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета
1.	Основные понятия качества программного обеспечения. Стандарты качества.

2.	Модели качества программного обеспечения.
3.	Характеристики и подхарактеристики качества при использовании.
4.	Характеристики качества продукта. Функциональная пригодность.
5.	Характеристики качества продукта. Уровень производительности.
6.	Характеристики качества продукта. Удобство использования.
7.	Характеристики качества продукта. Совместимость, переносимость.
8.	Характеристики качества продукта. Надежность.
9.	Характеристики качества продукта. Защищенность.
10.	Характеристики качества продукта. Сопровождаемость.
11.	Метрики размера программ.
12.	Метрики сложности потока управления программ
13.	Статистическая сложность программных модулей. Метрики Холстеда.
14.	Структурная сложность программных модулей. Метрика Мак-Кейба.
15.	Структурная сложность программных модулей. Матрица смежности.
16.	Структурная сложность программных модулей. Матрица достижимости.
17.	Структурная сложность комплексов программ.
18.	Вычислительная сложность функционирования комплексов программ.
19.	Корректность программ и программных модулей.
20.	Программные эталоны и методы проверки корректности программ.
21.	Верификация программ.
22.	Надежность программного обеспечения. Основные понятия.
23.	Показатели надежности программного обеспечения.
24.	Классификация моделей надежности программного обеспечения.
25.	Модели надежности программного обеспечения. Модель Шумана.
26.	Модели надежности программного обеспечения. Модель La Padula.
27.	Модели надежности программного обеспечения. Модель Джелинского-Моранды.
28.	Модели надежности программного обеспечения. Модель Шика-Волвертона.
29.	Модели надежности программного обеспечения. Модель Муса.
30.	Модели надежности программного обеспечения. Модель Миллса.
31.	Модели надежности программного обеспечения. Модель Липова.
32.	Модели надежности программного обеспечения. Простая интуитивная модель.
33.	Модели надежности программного обеспечения. Модель Коркорэна.
34.	Модели надежности программного обеспечения. Модель Нельсона.

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
1.	Что понимают под метрикой программного обеспечения? Приведите две метрики, одна из которых позволяет оценить покрытие требований, другая — тестовое покрытие на основе анализа потока управления (можно указать наименование, формулу с расшифровкой буквенных обозначений или метод оценки)
2.	Что представляет собой такая характеристика как доверие? Если в приложении присутствуют грамматические ошибки в наименовании элементов меню, то каким образом это повлияет на данную характеристику? (поясните ответ)
3.	По какому критерию метрик Холстеда можно сделать вывод о наличии возможных недостатков программного кода? Если для некоторого листинга количество операторов $N_1 = 499$, операндов $N_2 = 326$, расчетная длина $\tilde{N} = 454$, то какой можно сделать вывод относительно данного кода, согласно Холстеду? Поясните ответ и укажите значение критерия, на основе которого вы пришли к такому выводу
4.	Что показывает метрика Джилба? Какие из представленных операторов на языке C++ (<i>for; return; while; if-else; goto; && ; ? ; inline</i>) будут учитываться в абсолютной сложности программы (CL) при расчете метрики Джилба?
5.	Что представляет собой тестовое покрытие? Возможно ли 100% тестовое покрытие для современного ПО (поясните ответ) и какие подходы используют для оценки и измерения тестового покрытия? Какой из подходов связан с матрицей соответствия требований?
6.	<p>Что собой представляет характеристика функциональной пригодности? Каким образом ограничения дизайна и реализации влияют на характеристику функциональной пригодности приложения? (поясните ответ)</p> <p>Пример ограничений дизайна и реализации:</p> <ul style="list-style-type: none"> • Все программные коды управляющей логики системы должны быть написаны на языке PHP • Весь код PHP должен соответствовать стандарту PHP 7.4 • Весь код PHP должен быть написан в соответствии со стандартом PSR-2 • Документация системы по дизайну, коду и сопровождению должна соответствовать принятому внутреннему стандарту Some Standard, версия 2.1
7.	Выполните расчет цикломатической сложности для заданного графа и постройте матрицы смежности и достижимости
8.	Перечислите три основных подхода к оценке и измерению тестового покрытия. Какой из трех подходов связан с метрикой Маккейба?
9.	Поясните, какие ошибки были допущены при составлении заданного тест кейса
10.	Что представляет собой такая характеристика как анализируемость? Приведите не менее трех примеров того, что упрощает анализ программного обеспечения и соответственно увеличивает численное значение данной характеристики
11.	Приведите формулу расчета тестового покрытия для требований (расшифруйте буквенные обозначения). Если в результате расчета, тестовое покрытие для требований равно 100%, означает ли это, что исходный код приложения полностью протестирован? (поясните ответ)
12.	Что представляет собой такая характеристика как модифицируемость? Приведите не менее трех примеров того, что упрощает модификацию программного обеспечения и соответственно увеличивает численное значение данной характеристики
13.	Что представляют собой основной и альтернативный потоки? Приведите пример какой-либо задачи (функционального требования) для основного потока и альтернативного.

14.	Что представляет собой такая характеристика как модульность? Приведите не менее трех характеристик на которые влияет наличие модульной структуры приложения и поясните, как именно численные значения данных характеристик связаны с модульной структурой.																
15.	Какие из нижеперечисленных утверждений для метрики тестового покрытия не являются верными (приведите корректные формулировки или пояснения): 1) Тестовое покрытие представляет собой плотность покрытия тестами требований или исполняемого кода; 2) Современное программное обеспечение обладает такой сложностью, что для проведения 100% тестового покрытия используют специальные инструменты и/или техники тест дизайна; 3) Матрица соответствия требований может включать в себя как функциональные, так и нефункциональные требования; 4) В формуле расчета тестового покрытия для требований учитываются общее количество требований и количество тестов, которые проверяют данные требования.																
16.	Что представляет собой такая характеристика как тестируемость? Приведите не менее трех примеров того, что упрощает тестирование программного обеспечения и соответственно увеличивает численное значение данной характеристики.																
17.	<p>Что представляет собой тест кейс? Приведите подробный пример тест кейса для задачи: оформление возврата товара (интернет-магазин)</p> <table border="1"> <tr> <td>ID и наименование:</td> <td>(заполнить)</td> </tr> <tr> <td>Ссылка на требование:</td> <td>FR-07</td> </tr> <tr> <td>Дата создания:</td> <td>(заполнить)</td> </tr> <tr> <td>Автор:</td> <td>(заполнить)</td> </tr> <tr> <td>Приоритет:</td> <td>Высокий</td> </tr> <tr> <td colspan="2">Предусловия: необходимо выполнить заказ и покупку товара</td> </tr> <tr> <td>Шаги теста:</td> <td>Ожидаемый результат:</td> </tr> <tr> <td>(заполнить)</td> <td>(заполнить)</td> </tr> </table>	ID и наименование:	(заполнить)	Ссылка на требование:	FR-07	Дата создания:	(заполнить)	Автор:	(заполнить)	Приоритет:	Высокий	Предусловия: необходимо выполнить заказ и покупку товара		Шаги теста:	Ожидаемый результат:	(заполнить)	(заполнить)
ID и наименование:	(заполнить)																
Ссылка на требование:	FR-07																
Дата создания:	(заполнить)																
Автор:	(заполнить)																
Приоритет:	Высокий																
Предусловия: необходимо выполнить заказ и покупку товара																	
Шаги теста:	Ожидаемый результат:																
(заполнить)	(заполнить)																
18.	Что представляет собой такая характеристика как гибкость? Приведите пример того, как (произвольный) продукт может быть использован для достижения дополнительных целей, которые не были предусмотрены заранее																
19.	С помощью какого артефакта можно проверить покрытие требований и соответственно убедиться, что каждому требованию соответствует определённый тест? Приведете пример данного артефакта или опишите принцип использования. Если количество тестов в данном артефакте больше или меньше количества требований, о чем свидетельствуют указанные ситуации?																
20.	Что представляет собой такая характеристика как возможность многократного использования? Каким образом многократное использование (активов) может снижать вероятность возникновения ошибок в проектах?																
21.	Что представляет собой матрица соответствия требований и как она связана с оценкой тестового покрытия на основе анализа потока управления? Приведите основную формулу (с расшифровкой буквенных обозначений) для оценки тестового покрытия на основе анализа потока управления.																
22.	Что представляет собой такая характеристика как устанавливаемость? Если для некоторого продукта версии 1.42 вышла версия 2.0, которая прекратила поддержку и соответственно установку на определённые операционные системы																

	(которые поддерживала версия 1.42), то на какую из характеристик модели качества продукта это окажет влияние? (поясните ответ)								
23.	<p>Для некоторого листинга программы <i>A</i> известны следующие характеристики:</p> <ul style="list-style-type: none"> • (по Холстеду) Количество операндов $N_2 = 195$, расчетная длина $\tilde{N} = 322$; • (по Джилбу) Абсолютная сложность программы $CL = 17$, относительная сложность программы $cl \approx 0,058$. <p>Какой можно сделать вывод о наличии недостатков данного кода, согласно Холстеду? Поясните ответ и укажите точное значение критерия, на основе которого вы пришли к такому выводу</p>								
24.	Что показывает метрика Джилба? Для представленного листинга программы укажите абсолютную сложность (CL) и номера строк операторов (например, 17 — <code>while</code>), которые учитывает данная характеристика (расчет самой метрики Джилба выполнять не нужно).								
25.	Что понимают под метрикой ПО? Приведите две метрики, одна из которых позволяет оценить покрытие требований (формула с расшифровкой буквенных обозначений или метод оценки), другая — тестовое покрытие на основе анализа потока управления (наименование метрики)								
26.	Для некоторого приложения <i>A</i> дана матрица соответствия требования (в виде таблицы). Каково тестовое покрытие требований для заданного приложения? Поясните, как сильно изменится значение тестового покрытия, если некоторые тест кейсы (заданные преподавателем) будут признаны неактуальными и, соответственно, удалены (поясните ответ)?								
27.	Постройте граф по заданной матрице смежности и поясните, какая проблема существует в программном коде, информация о путях исполнения которого содержится в представленной матрице (необходимо указать проблемные пути или узлы)								
28.	<p>Какие из нижеперечисленных утверждений для модели качества продукта не являются верными (приведите корректные формулировки или пояснения):</p> <ol style="list-style-type: none"> 1) Оценка уровня производительности продукта осуществляется в том числе и с учетом использованных материалов по типу бумаги для печати; 2) Существование предполагает способность продукта совместно функционировать с другими независимыми продуктами в общей среде с разделением общих ресурсов и с минимальным отрицательным влиянием на любой другой продукт; 3) Восстанавливаемость продукта может определяться через время, в течении которого система находится в нерабочем состоянии; 4) Износ программного обеспечения и, соответственно, проблемы с надежностью связаны с недостатками требований, которые имеют место при разработке продукта или в результате изменений условий использования. 								
29.	В таблице (задается преподавателем) представлены фрагменты кода <i>A</i> , <i>B</i> , <i>C</i> и <i>D</i> с управляющими операторами, подсчитайте значение абсолютной сложности программы отдельно для циклов ($CL_{цикл}$) в отношении данных фрагментов кода (в случае неверного ответа хотя бы для одного из фрагментов, правильный ответ для всех остальных фрагментов оценивается в 1 балл).								
30.	<p>Какова абсолютная сложность (CL) для некоторого листинга программы <i>A</i> по Джилбу, если известны следующие операторы и количество их повторений (таблица ниже)? Укажите, какие именно из представленных операторов, учитываются в данной метрике</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Оператор</th> <th>Количество повторений</th> <th>Оператор</th> <th>Количество повторений</th> </tr> </thead> <tbody> <tr> <td>for</td> <td>4</td> <td>goto</td> <td>1</td> </tr> </tbody> </table>	Оператор	Количество повторений	Оператор	Количество повторений	for	4	goto	1
Оператор	Количество повторений	Оператор	Количество повторений						
for	4	goto	1						

		return	1	&&	11
		while	2	? :	2
		if-else	8	inline	2
31.	Выделите линейно-независимые маршруты и циклы для заданного графа (начальный и конечные узлы отмечены цветом):				
32.	Что представляет собой такая характеристика как возможность многократного использования? Каким образом многократное использование (активов) может снижать вероятность возникновения ошибок в проектах?				

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Оценка качества программных средств (по ГОСТ 28195-89)
2	Оценка качества при использовании (по ГОСТ Р ИСО/МЭК 25010-2015)
3	Оценка качества продукта (по ГОСТ Р ИСО/МЭК 25010-2015)
4	Разработка элементов показателей качества (по ГОСТ Р ИСО/МЭК 25021-2014)

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

Лекция имеет четкую и строгую структуру, как правило, состоит из трех частей: вступления (введения), изложения и заключения.

Вступление (введение) определяет тему, план и цель лекции. В нем излагается метод лекции и ее актуальность, основная идея (проблема, центральный вопрос), связь данной лекции с предыдущими и последующими занятиями, ее основные вопросы. Введение должно быть кратким и целенаправленным.

В основной части лекции, в которой реализуется научное содержание темы, все главные узловые вопросы, приводится вся система доказательств.

Число вопросов в лекции, как правило, колеблется от двух до четырех. Иногда отдельные вопросы делятся на подвопросы, облегчающие изложение и усвоение материала.

Заключение имеет целью обобщить в кратких формулировках основные идеи лекции, логически завершая ее как целостное творение.

Материалы лекций содержатся в учебном пособии «Метрология программного обеспечения [Текст]: Учебное пособие / Е.В. Копкин, В.А. Каргин, Е.В. Павлов: СПб ГУАП, 2017. – 145 с.», который выкладывается студентам в личный кабинет.

11.2. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Лабораторная работа №1. «Составление вариантов тестирования и метрика покрытия требований»

Целью данной работы является изучение способов тестирования функций приложения и оценка плотности покрытия тестами функциональных требований.

Задание на лабораторную работу:

- 1) Разработать консольное приложение в соответствии с индивидуальным вариантом задания и представленной концепцией продукта.
- 2) Составить тест кейсы для тестирования функций данного приложения и построить матрицу соответствия требований для анализа тестового покрытия.
- 3) Выполнить тестирование приложения по составленным тест кейсам.

Концепция продукта и требования, которым должно удовлетворять разрабатываемое приложение.

Концепция продукта (бизнес-правило):

В связи с предстоящим переходом на программное обеспечение отечественного производства, необходимо разработать современное ПО, отвечающее требованиям формата эксплуатации и хранения данных в рамках стратегии развития информационного общества в Российской Федерации на 2020-2030 годы. В разрабатываемом ПО должна быть предусмотрена возможность добавления и/или обновления соответствующих данных.

Внешние требования к интерфейсу:

В диалоговых сообщениях с пользователем разрешается использовать русский (по умолчанию), старославянский или церковнославянский языки.

После выполнения действия пользователя (выбор пункта меню) и перед выводом информации должна производиться очистка экрана консоли.

Требования к сопровождению:

Каждую функцию (метод или фрагмент программного кода), отвечающую за реализацию того или иного функционального требования, необходимо сопроводить соответствующим комментарием, поясняющим её назначение.

Хранение наиболее приоритетных данных, которыми оперирует приложение, должно осуществляться посредством внешних файлов (количество создаваемых файлов не ограничено).

Ограничения проекта:

Все программные коды управляющей логики приложения должны быть написаны на языке C++ (желательное, но не обязательное требование)

Рекомендации:

Для реализации приложения рекомендуется использовать бесплатную IDE Visual Studio Community: <https://visualstudio.microsoft.com/ru/vs/community/>

Студент вправе выбрать любые функциональные требования для реализации при условии, что данные требования (функции) соответствуют назначению приложения. В таблице Л1.1.1 представлен список возможных требований к ПО; серым маркером выделены те требования, реализация которых является *обязательной*.

Таблица Л1.1.1 — Возможные функциональные требования

FR-01	Вывод данных из файла (обязательное требование)
FR-02	Добавление данных в файл (обязательное требование)
FR-03	Удаление данных из файла (обязательное требование)
FR-04	Редактирование данных в файле
FR-05	Математические расчеты (расчет стоимости услуг, топлива в зависимости от маршрута, различная статистика, etc.)
FR-06	Сортировка данных в файле (или при выводе на экран консоли)
FR-N	Иные требования, которые не противоречат назначению ПО

В таблице Л1.1.2 представлен список возможных функциональных требований на примере варианта «Эко-приложение для отслеживания незаконных свалок» (функциональные требования студент должен сформулировать самостоятельно).

Таблица Л1.1.2 — Пример функциональных требований

FR-01	Вывод списка незаконных свалок (адрес, дата обнаружения, статус)
-------	--

FR-02	Добавить информацию о незаконной свалке (адрес, дата, статус — активна / закрыта)
FR-03	Удалить информацию о незаконной свалке
FR-04	Вывод статистики (общее количество незаконных свалок, данные за определённый месяц, сколько дней существует незаконная свалка с момента её обнаружения)
FR-05	Вывести адреса пунктов переработки отходов (с указанием вида перерабатываемых отходов)
FR-06	Добавить адрес пункта переработки отходов
FR-07	Удалить пункт переработки отходов

Лабораторная работа №2. «Оценка качества программного обеспечения»

Целью данной работы является изучение методологии оценки качества программного обеспечения.

Задание на лабораторную работу:

Выполнить оценку качества разработанного приложения в соответствии со стандартом ГОСТ Р ИСО / МЭК 25010-2015.

Указать обоснование (пояснение) выполненных оценок качества и привести инфографику для каждой характеристики качества в виде лепестковой диаграммы.

Если количество включаемых подхарактеристик меньше трёх, то графическое представление основной характеристики приводить не нужно.

Для выполнения соответствующей оценки качества в рамках стандарта ГОСТ Р ИСО / МЭК 25010-2015 необходимо взять разработанное приложение из ЛР 1.

Рекомендации:

Для выполнения графической составляющей работы (лепестковая диаграмма) вы можете воспользоваться готовыми диаграммами из примера отчета, изменив только исходные данные.

При изучении теоретического материала в части оценки качества, обратите внимание, что многие характеристики и подхарактеристики связаны между собой и влияют друг на друга. Например, все подхарактеристики надежности в совокупности определяют степень надежности программного продукта. Таким образом, оценка той или иной характеристики должна осуществляться как *среднее арифметическое* совокупности её подхарактеристик.

Количественную оценку характеристик качества необходимо выполнить в рамках следующей интервальной шкалы:

- 1-40% (низкая оценка характеристики или свойства)
- 41-60% (средняя оценка...)
- 61-100% (высокая оценка...)

Если приложение не обладает искомой характеристикой или её оценка не представляется возможной, например, в приложении может отсутствовать пользовательский интерфейс, и оно является прослойкой между API конкретного сервера и приложением пользователя, в этом случае для оценки свойств, характеризующих пользовательский интерфейс, выбираются косвенные критерии или указывается наименьшая из оценок.

Лабораторная работа №3. «Определение сложности программного кода на основе лексического анализа»

Целью данной работы является оценка сложности программного кода на основе лексического анализа с использованием метрик Холстеда и Джилба.

Задание на лабораторную работу:

1) Составить словарь операторов и операндов программного кода приложения (или его фрагмента) и произвести расчёт метрик Холстеда и Джилба.

2) Выполнить анализ полученных результатов.

В рамках данной работы разрешается ограничить исходный код программы в соответствии с требованиями, представленными ниже:

Разрешается ограничить исходный код программы (из ЛР 1) и использовать для расчётов метрик Холстеда и Джилба фрагмент кода (или модуль) размером 160-220 строк (из которых не более 15% пустых)

Также можно опустить перечисление номеров строк для тех операторов или разделителей, количество которых соизмеримо с количеством строк оцениваемого кода, например, это справедливо в отношении точки с запятой.

Номера строк должны совпадать с тем фрагментом кода, который будет представлен в приложении к отчету. В самом приложении необходимо привести только фактически анализируемый фрагмент кода, а не весь листинг приложения.

Рекомендации:

Поиск операторов и операндов в коде программы можно осуществить с помощью сочетаний клавиш CTRL + F или воспользоваться [Notepad++](#) (либо любым другим редактором, который способен подсвечивать выделенные лексемы заданного языка программирования).

При составлении словаря операторов для интерпретации конструкций языка C++ может быть полезна следующая ссылка: [Справочное руководство по C++](#)

Для подсчета общего количества операторов и операндов можно скопировать таблицу в Excel и воспользоваться функцией суммирования значений заданных ячеек: =СУММ(диапазон).

Лабораторная работа №4. «Оценка алгоритмической сложности программного кода»

Целью данной работы является анализ графа потока управления и оценка алгоритмической сложности программного кода на основе метрики Маккейба.

Задание на лабораторную работу:

Начертить схему алгоритма программного кода приложения (или его фрагмента), построить граф потока управления, выделить линейно-независимые маршруты и циклы и выполнить расчёт цикломатического числа Маккейба.

Отразить в выводах результаты и проанализировать корректность расчета цикломатической сложности.

Схему алгоритма программного кода можно сгенерировать с помощью <https://code2flow.com/app> (для экспорта изображения — SHARE) или начертить самостоятельно (предпочтительнее), используя:

- Веб-сервис <https://www.diagrams.net/>
- Приложение <https://www.softwareideas.net/en/download>

Обратите внимание, что в случае генерации схем алгоритмов (например, при использовании code2flow.com) некоторые элементы программного кода могут быть отображены некорректно, следовательно, оценка цикломатической сложности будет неверна. Кроме того, по заданию вам необходимо обозначить номера элементов блок-схемы (узлы графа), а их в случае генерации схемы будет довольно много.

Также разрешается выполнить данную работу, включая построение графа потока управления, от руки и вложить скан или фотографию в отчёт при условии, что элементы изображения будут визуально различимы. В равной степени это требование относится и в отношении экспортированных изображений схем алгоритмов — если изображение не

помещается на одной странице, его необходимо разбить на несколько страниц так, чтобы при его анализе не возникало никаких затруднений.

Рекомендации:

Для графического представления графа потока управления подойдёт любой графический редактор, однако *самый быстрый и аккуратный способ представить граф* — это автоматическая визуализация с помощью языка описания графов DOT (Graph Description Language). В языке DOT от вас потребуется указать направления переходов между узлами, что намного быстрее, нежели чертить граф самостоятельно. Кроме того, автоматически построенные графы удобнее для выделения линейно-независимых маршрутов и циклов.

Рекомендуемый пакет для визуализации графов:

https://graphviz.gitlab.io/pages/Download/Download_windows.html

Основные команды в редакторе GraphViz (GV Edit):

- F5 — построение графа
- SHIFT + F5 — экспорт изображения

В редакторе GraphViz присутствует один серьёзный недостаток: в случае возникновения синтаксической ошибки, периодически (это происходит не так часто) для продолжения работы и построения графа, требуется перезапуск программы.

Документация по DOT: <https://graphviz.gitlab.io/pages/pdf/dotguide.pdf>.

Структура и форма отчета о лабораторной работе

Лабораторная работа №1. «Составление вариантов тестирования и метрика покрытия требований»

Отчет о ЛР №1 должен включать в себя следующие структурные элементы:

- 1) Титульный лист
- 2) Цель работы
- 3) Задание на лабораторную работу
- 4) Спецификация программных требований
- 5) Тест кейсы и фактические результаты тестирования (Alt + PrtSc — для скриншота экрана консоли); матрица соответствия требований
- 6) Выводы по работе
- 7) Список использованных источников (представлен в шаблоне)
- 8) Приложение А (исходный код программы)
- 9) Приложение Б (содержимое внешних файлов)

Лабораторная работа №2. «Оценка качества программного обеспечения»

Отчет о ЛР №2 должен включать в себя следующие структурные элементы:

- 1) Титульный лист
- 2) Цель работы
- 3) Задание на лабораторную работу
- 4) Оценка качества при использовании
- 5) Оценка качества продукта
- 6) Выводы по работе, которые должны включать в себя графическое представление характеристик качества в виде лепестковых диаграмм
- 7) Список использованных источников (представлен в шаблоне)
- 8) Приложение А (исходный код программы)

Лабораторная работа №3. «Определение сложности программного кода на основе лексического анализа»

Отчет о ЛР №3 должен включать в себя следующие структурные элементы:

- 1) Титульный лист
- 2) Цель работы
- 3) Задание на лабораторную работу
- 4) Словарь операторов и операндов анализируемого кода
- 5) Оценка сложности анализируемого кода
- 6) Выводы по работе (включая анализ результатов)
- 7) Список использованных источников (представлен в шаблоне)
- 8) Приложение А (анализируемый фрагмент кода)

Лабораторная работа №4. «Оценка алгоритмической сложности программного кода»

Отчет о ЛР №4 должен включать в себя следующие структурные элементы:

- 1) Титульный лист
- 2) Цель работы
- 3) Задание на лабораторную работу
- 4) Схема алгоритма (*с обозначением возможных вершин графа*)
- 5) Граф потока управления и расчет цикломатической сложности
- 6) Линейно-независимые маршруты и циклы
- 7) Выводы по работе (включая анализ расчета цикломатической сложности)
- 8) Список использованных источников (представлен в шаблоне)
- 9) Приложение А (анализируемый фрагмент кода)

Требования к оформлению отчета о лабораторной работе

Лабораторная работа №1. «Составление вариантов тестирования и метрика покрытия требований»

- 1) Выбрать индивидуальный вариант задания (ссылка от преподавателя);
- 2) Составить перечень из 7-ми профильных функциональных требований. Профильное требование подразумевает, что оно имеет непосредственное отношение к разрабатываемому ПО и определяет его назначение;
- 3) Согласовать перечень функциональных требований с преподавателем (данный пункт не является обязательным, если сформулированные требования в полной мере соответствуют представленным примерам в методических указаниях и шаблоне выполненной работы);
- 4) Разработать соответствующее приложение;
- 5) Составить не менее 10 тест кейсов к функциональным требованиям и построить матрицу соответствия требований;
- 6) Выполнить тестирование программы по составленным тест кейсам;
- 7) Написать отчет о работе.

Лабораторная работа №2. «Оценка качества программного обеспечения»

- 1) Выполнить оценку характеристик качества для разработанного приложения в соответствии со стандартом ГОСТ Р ИСО / МЭК 25010-2015;
- 2) Написать отчет о работе.

Лабораторная работа №3. «Определение сложности программного кода на основе лексического анализа»

- 1) Составить словарь операторов и операндов программного кода приложения (или его фрагмента);

- 2) Выполнить расчет по метрикам Холстеда и Джилба;
- 3) Проанализировать полученные результаты;
- 4) Написать отчет о работе.

Лабораторная работа №4. «Оценка алгоритмической сложности программного кода»

- 1) Начертить (предпочтительнее) или сгенерировать схему алгоритма;
- 2) Построить граф потока управления;
- 3) Выделить линейно-независимые маршруты и циклы;
- 4) Выполнить расчёт цикломатической сложности Маккейба;
- 5) Произвести оценку корректности выполненного расчета цикломатической сложности программного кода (или его фрагмента);
- 6) Написать отчет о работе.

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости осуществляется путем выставления баллов за выполненные лабораторные работы в соответствии с данными, приведенными в таблице 20. Полученные баллы учитываются при проведении промежуточной аттестации.

Таблица 20 — Классификация критериев оценки лабораторных работ

Критерий	Баллы
ЛР выполнена вне сроков сдачи	10
ЛР выполнена (с незначительными замечаниями)	12
ЛР выполнена (без замечаний, либо замечания устранены)	15

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».

После выполнения лабораторных работ обучающиеся выполняют тестовые задания, каждое из которых включает 6 вопросов (заданий). Максимальное число баллов за выполнение тестового задания – 24 балла.

Зачет проводится в письменной форме. Билет содержит 2 теоретических вопроса.

Для получения оценки «зачтено» обучающийся должен набрать минимум 55 баллов за выполненные лабораторные работы и тестовое задание, а также правильно ответить на вопросы билета.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой