



1920

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного  
учреждения высшего образования

«Кубанский государственный университет» в г. Геленджике

УТВЕРЖДАЮ  
Проректор по работе с филиалами  
А.А. Евдокимов  
\_\_\_\_\_ 2021 г.

**Рабочая программа дисциплины**

**МДК.01.02 «Поддержка и тестирование модулей программного обеспечения»**

**09.02.07 «Информационные системы и программирование»**


Рабочая программа МДК.01.02 «Поддержка и тестирование модулей программного обеспечения» разработана на основе федерального государственного образовательного стандарта среднего профессионального образования (ФГОС СПО) по специальности «Информационные системы и программирование», утвержденного приказом Министерства образования и науки от 9 декабря 2016 года № 1547 (зарегистрирован Министерством юстиции Российской Федерации 26 декабря 2016г., регистрационный №44936) (далее – ФГОС СПО).

Дисциплина	МДК.01.02 «Поддержка и тестирование модулей программного обеспечения»
Форма обучения	очная
Учебный год	2021-2022
2,3 курсы	4, 5 семестр
Лекции	110 час.
Практические занятия	62 час.
Консультации	4 час.
Экзамен	6 час.
форма итогового контроля	дифференцированный зачёт, экзамен

Составитель: преподаватель  Поддубная Е.А.

Утверждена на заседании предметной (цикловой) комиссии профессиональных дисциплин программирования в компьютерных системах  
Протокол № 10 от 20 мая 2021 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности **09.02.07 «Информационные системы и программирование»** и **09.02.03 «Программирование в компьютерных системах»**

  
\_\_\_\_\_ Л.А. Благова  
подпись

Рецензенты:

Директор ООО «Современные  
информационные технологии»



\_\_\_\_\_ А.В.Сметанин

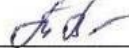
Системный администратор  
ЗАО «Геленджикский дельфинград»




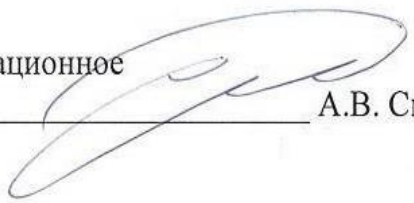
\_\_\_\_\_ Т.П. Кривошеенко

ЛИСТ  
согласования рабочей учебной программы по дисциплине  
ЕН.02 «Дискретная математика с элементами математической логики»  
Специальность среднего профессионального образования:  
**09.02.07 «Информационные системы и программирование»**

СОГЛАСОВАНО:

Заместитель директора по УР филиала \_\_\_\_\_  Т.А. Резуненко

Заведующая сектором библиотеки филиала \_\_\_\_\_  Л.Г. Соколова

Инженер-электроник (программно-информационное  
обеспечение образовательной программы) \_\_\_\_\_  А.В. Сметанин

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	4
1.1. Область применения программы .....	4
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена .....	4
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины .....	4
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций) .....	5
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	9
2.1. Объем учебной дисциплины и виды учебной работы .....	9
2.2. Структура дисциплины: .....	9
2.3. Тематический план и содержание учебной дисциплины .....	10
2.4. Содержание разделов дисциплины .....	12
2.4.1. Занятия лекционного типа .....	12
2.4.2. Занятия семинарского типа .....	12
2.4.3. Практические (лабораторные) занятия .....	12
2.4.4. Содержание самостоятельной работы .....	13
2.4.5. <i>Содержание учебно-методического обеспечения для самостоятельной работы</i>	
3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ .....	16
3.1. Образовательные технологии при проведении лекция .....	16
3.2. Образовательные технологии при проведении практических занятий .....	16
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ .....	18
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине .....	18
4.2. Информационное обеспечение реализации программы .....	18
5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	
5.1. Основная литература .....	21
5.2. Дополнительная литература.....	22
5.3. Периодические издания.....	23
5.4. Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины .....	
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .....	25
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ .....	30
7.1. Паспорт фонда оценочных средств .....	30
7.2. Критерии оценки знаний .....	30
7.3. Оценочные средства для проведения текущей аттестации .....	31
7.4. Оценочные средства для проведения промежуточной аттестации .....	32
7.4.1. Примерные вопросы для проведения промежуточной аттестации .....	32
7.4.2. Примерные задачи для проведения промежуточной аттестации .....	40
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	51

## **1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **МДК.01.02 «Поддержка и тестирование программных модулей»**

#### **Область применения программы**

Рабочая программа МДК.01.02 «Поддержка и тестирование программных модулей» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07. «Информационные системы и программирование»

**Место учебной дисциплины в структуре программы подготовки специалистов среднего звена**

Учебная дисциплина МДК.01.02 «Поддержка и тестирование программных модулей» принадлежит профессиональному модулю ПМ.01. «Разработка модулей программного обеспечения для компьютерных систем». Она обеспечивает профессиональный уровень подготовки специалиста и соответствует развитию их профессионально значимых качеств.

Для освоения дисциплины обучающиеся используют знания, умения и навыки, сформированные при изучении МДК.01.01 «Разработка программных модулей» и ОП.04 «Основы алгоритмизации и программирования».

#### **Цель и планируемые результаты освоения дисциплины:**

Для основного вида деятельности «**Разработка модулей программного обеспечения для компьютерных систем**» осваиваются частично ПК.1.3. «Выполнять отладку программных модулей с использованием специализированных программных средств», для основного вида деятельности «**Осуществление интеграции программных модулей**» осваиваются частично компетенции ПК 2.1. «Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент», ПК 2.3. «Выполнять отладку программного модуля с использованием специализированных программных средств», ПК «Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения», ПК 4.2. Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем

Максимальная учебная нагрузка обучающегося - 182 часа, в том числе

- обязательная аудиторная нагрузка - 172 часа;
- самостоятельная работа обучающегося - не предусмотрено
- консультации - 4 часа;
- экзамен - 6 часов

#### **Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)**

В результате освоения учебной дисциплины обучающийся должен

уметь:

- *применять логические операции, формулы логики, законы алгебры логики;*

– *формулировать задачи логического характера и применять средства*

математической логики для их решения.

В результате освоения учебной дисциплины обучающийся должен

знать:

- основные принципы математической логики, теории множеств и теории алгоритмов;
- формулы алгебры высказываний;
- методы минимизации алгебраических преобразований;
- основы языка и алгебры предикатов;
- основные принципы теории множеств.

В результате освоения учебной дисциплины обучающийся осваивает элементы общих компетенций.

Перечень общих компетенция

Код компетенции	Формулировка компетенции	Знания, умения <sup>4</sup>
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	<p><b>Умения:</b> распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> <p><b>Знания:</b> актуальный профессиональный и социальный контекст, в котором приходится работать и жить; Основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения</p>

		задач; порядок оценки результатов решения задач профессиональной деятельности
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	<p><b>Умения:</b> определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</p> <p><b>Знания:</b> номенклатура информационных источников, Применяемых в профессиональной деятельности; Приемы структурирования информации; формат оформления результатов поиска информации</p>

ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие.	<p><b>Умения:</b> определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования</p> <p><b>Знания:</b> содержание актуальной нормативно-правовой документации; современная научная и профессиональная профессионального развития и самообразования</p>
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, Руководством, клиентами и	<p><b>Умения:</b> организовывать работу коллектива и команды; Взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности</p> <p><b>Знания:</b> психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности</p>
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и	<p><b>Умения:</b> грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе</p> <p><b>Знания:</b> особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.</p>



	культурного контекста.	
ОК 09	Использовать информационные технологии в профессиональной деятельности	<p><b>Умения:</b> применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение</p> <p><b>Знания:</b> современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности</p>

#### Перечень профессиональных компетенций

В результате освоения учебной дисциплины обучающийся осваивает элементы профессиональных компетенций: ПК 1.3, 2.1, 2.3, 2.4, 4.2.

Перечень профессиональных компетенций, элементы которых формируются в рамках учебной дисциплины

Код	Наименование видов деятельности и профессиональных компетенций
ОВД 1	Разработка модулей программного обеспечения для компьютерных систем
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ОВД 2	Осуществление интеграции программных модулей
ПК 2.1	Выполнять тестирование программных модулей
ПК 2.3.	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.4	Осуществлять рефакторинг и оптимизацию программного кода
ОВД 4	Сопровождение и обслуживание программного обеспечения компьютерных систем
ПК 4.2.	Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем
ПК 5.1.	Собирать исходные данные для разработки проектной документации на информационную систему

В результате освоения МДК 01.02 студент должен:

Иметь практический опыт	Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем. Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на
-------------------------	--

	<p>предмет соответствия стандартам кодирования. Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования</p> <p>Разрабатывать тестовые наборы (пакеты) для программного модуля.</p> <p>Разрабатывать тестовые сценарии программного средства.</p> <p>Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. Измерять эксплуатационные характеристики программного обеспечения компьютерных систем на соответствие требованиям</p>
уметь	<p>Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства. Анализировать проектную и техническую документацию.</p> <p>Использовать специализированные графические средства построения анализа архитектуры программных продуктов. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Определять источники и приемники данных. Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace). Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выявлять ошибки в системных компонентах на основе спецификаций. Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Использовать инструментальные средства отладки программных продуктов. Определять источники и приемники данных. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Выполнять отладку, используя методы и инструменты условной компиляции. Выявлять ошибки в системных компонентах на основе спецификаций.</p> <p>Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций. Измерять и анализировать эксплуатационные характеристики качества программного обеспечения</p>
знать	<p>Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и варианты интеграционных решений. Современные технологии и инструменты</p>

	<p>интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев ошибок при интеграции приложений. Методы отладочных классов. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Графические средств проектирования архитектуры программных продуктов. Методы организации работы в команде разработчиков. Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков. Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков. Основные методы и средства эффективного анализа функционирования программного обеспечения. Основные принципы контроля конфигурации и поддержки целостности конфигурации ПО</p>
--	---

В результате освоения учебной дисциплины обучающийся осваивает элементы основных видов деятельности:

Основной вид деятельности	Требования к знаниям, умениям, практическим действиям
ОВД 1. «Разработка модулей программного обеспечения для компьютерных систем»	Указаны выше
ОВД 2. «Осуществление интеграции программных модулей»	Указаны выше

ОВД 4 «Сопровождение и обслуживание программного обеспечения компьютерных систем»	Указаны выше
---	--------------

## Технологии формирования ОК

**ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.**

Предоставить обучающемуся возможность на практических занятиях для самостоятельного выбора способа решения задачи

**ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.**

Предоставить обучающемуся возможность на практических занятиях для самостоятельного выбора способа подготовки информации для решения задачи

**ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.**

Предоставить обучающемуся возможность на лекциях и практических занятиях для выражать свою заинтересованность в обучении и реализовывать ее

**ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.**

Предоставить обучающемуся возможность на практических занятиях, на учебной и производственной практиках для решения задач в коллективе, помогать в случае затруднений

**ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.**

Предоставить обучающемуся возможность на лекциях практических занятиях для общения на государственном языке

**ОК 09. Использовать информационные технологии в профессиональной деятельности.**

Предоставить обучающимся возможности на практических занятиях выполнять задания средствами ИТ. Предоставлять студентам возможность самостоятельно осуществлять поиск, анализ и оценку информации при выполнении практической и самостоятельной работы

Изучать профессиональные стандарты на государственном языке (русском) и знакомить обучающихся с международными стандартами на английском языке

### **Технологии формирования ПК**

Изучение теории ручного и автоматизированного тестирования и закрепления ее на опыте

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### Объем учебной дисциплины и виды учебной работы

<b>Вид учебной работы</b>	<b>Объем в часах</b>
<b>Объем образовательной программы</b>	<b>182</b>
в том числе:	
теоретическое обучение (+ текущая аттестация – зачёт)	110
практические занятия	62
консультации	4
<b>Промежуточная аттестация (экзамен)</b>	<b>6</b>

### Структура дисциплины

Наименование разделов и тем	Всего	Количество аудиторных часов		Самостоятельная работа обучающегося (час)
		Теоретическое обучение	Практические и лабораторные занятия	
<b>Раздел 1. Отладка и тестирование программного обеспечения</b>	<b>118</b>	<b>68</b>	<b>50</b>	-
Тема 1.1. Понятие разработки ПО	8	6	2	-
<b>Тема 1.2. Понятие тестирования</b>	8	6	2	-
Тема 1.3. Тестирование требований	14	6	8	-
Тема 1.4. Классификация тестирования	26	14	12	
Тема 1.5. Отчёты о тестировании	20	12	8	
Тема 1.6. Методы отладки и тестирования	18	12	6	
Тема 1.7. Автоматизированное тестирование	24	12	12	
<b>Раздел 2. Документирование</b>	<b>54</b>	<b>42</b>	<b>12</b>	
Тема 2.1. Документирование программного обеспечения в соответствии с Единой системой программной документации	<b>12</b>	8	4	
Тема 2.2. Документирование основных процессов ЖЦ ПО	<b>16</b>	12	4	

Тема 2.3. Документирование вспомогательных процессов ЖЦ ПО	<b>14</b>	<b>12</b>	<b>2</b>	
Тема 2.4. Документирование сертификации	<b>12</b>	<b>10</b>	<b>2</b>	
<b>Всего по дисциплине</b>	<b>172</b>	<b>110</b>	<b>62</b>	

**Тематический план и содержание учебной дисциплины МДК.01.02 «Поддержка и тестирование модулей программного обеспечения»**

<i>МДК.01.02 Поддержка и тестирование программных модулей</i>				
<i>Раздел 1. Отладка и тестирование программного обеспечения</i>				
Тема 1.1. Понятие разработки ПО	<b>Содержание учебного материала</b>	<b>Уровень освоения</b>	<b>58</b>	
	1.Понятие программного обеспечения и его классификация	2	1	
	2.Жизненный цикл ПО.	2	2	
	3.Этапы разработки ПО	2	3	
	<b>Практические занятия</b>			
	1.Проверочная работа по теме 1.1.	3	4	
Тема 1.2. Понятие тестирования	<b>Содержание учебного материала</b>			
	4.Тестирование как часть процесса верификации программного обеспечения.	2	5	
	5. Обязанности тестировщика	2	6	
	6. Жизненный цикл тестирования	2	7	
	<b>Практические занятия</b>			
	2.Проверочная работа по теории	3	8	
Тема 1.3. Тестирование требований	<b>Содержание учебного материала</b>			
	7.Понятие требований и их качество (уровни и типы)	2	9	
	8. Техника тестирования требований	2	10	
	9.Типичные ошибки при анализе и тестировании требований и причины их появления	2	11	
	<b>Практические занятия</b>			
		3.Проверочная работа по теории	3	12
		4.Составление требований к программе	3	13
	5.Проверка составленных требований	3	14	
	6.Анализ выявленных ошибок	3	15	
Тема 1.4. Классификация тестирования	<b>Содержание учебного материала</b>			
	10.Классификация тестирования	2	16	
	11.Виды тестирования (по запуску кода на исполнение, по доступу к коду и архитектуре	2	17	



	приложения, по степени автоматизации, по уровню тестирования)		
	12.Виды тестирования (уровню функционального тестирования, по принципам работы с приложением, по природе приложения, по привлечению конечных пользователей)	2	18
	13. Виды тестирования (по степени формализации, по целям и задачам, по техникам и подходам, по хронологии)	2	19
	14.Виды тестирования (по знанию системы, по исполнителям, по времени проведения, производительности)	2	20
	15. Альтернативные и дополнительные классификации	2	21
	16.Классификация по принадлежности к методам «белого» и «чёрного» ящикам	2	22
	<b>Практические занятия</b>		
	7. Проверочная работа	3	23
	8.Тестирование программы методом «белого ящика»	3	24
	9.Перекрёстная проверка результатов тестирования	3	25
	10.Тестирование программы методом «чёрного» ящика.	3	26
	11.Перекрёстная проверка результатов тестирования	3	27
	<b>Содержание учебного материала</b>		
	17. Виды ошибок (ошибки (синтаксические, компоновки, выполнения, определения данных, накопления погрешностей, проектирования), дефекты, сбои, отказы и т.д.)	2	28
	18.Отчёт о дефектах, его атрибуты и ЖЦ	2	29
	19. Инструментальные средства управления отчётами о дефектах	2	30
	20. Свойства качественных отчётов и логика их создания. Типичные ошибки при написании отчёта о дефектах	2	31
	21. Планирование тестирования (тест-план)	2	32
	22. Оценка трудозатрат на планирование и тестирование	2	33
	<b>Практические занятия</b>		
	12.Проверочная работа	3	34
	13.Составление проекта отчёта о дефектах	3	35
	14.Перекрёстный анализ составленных отчётов	3	36
	15. Составление тест-плана	3	37
	16. Оценка трудозатрат на планирование и тестирование по составленным планам	3	38
Тема 1.5. Отчёты о тестировании	<b>Содержание учебного материала</b>		
Тема 1.6.	<b>Содержание учебного материала</b>		

Методы отладки и тестирования	<b>23.Методы отладки:</b> (ручного тестирования, индукции, дедукции, обратного прослеживания)	2	39
	24.Методы тестирования (позитивные и негативные тест-кейсы, классы эквивалентности и граничные условия)	2	40
	25.Методы тестирования (доменное тестирование и комбинация параметров)	2	41
	26.Методы тестирования (парное тестирование и поиск комбинаций, исследовательское тестирование)	2	42
	27. Регрессионное тестирование и его виды	2	43
	28. Поиск причин возникновения дефектов	2	44
	<b>Практические занятия</b>		
	17.Проверочная работа по теории методов тестирования	3	45
	18.Отладка программы всеми известными методами и составление отчёта о результатах	3	46
	19.Тестирование одной программы 9-ю методами и составление отчёта о результатах	3	47
Тема 1.7. Автоматизированное тестирование	<b>Содержание учебного материала</b>		
	29.Автоматизация тестирования, её задачи, преимущества и ограничения. Пирамида тестирования.	2	48
	30.Этапы автоматизированного тестирования	2	49
	31.Разработка тест-кейсов. Сценарный тест	2	50
	32.Виды автоматизированного тестирования (регрессионное, кроссбраузерное и кроссплатформенное, локации и производительности)	2	51
	33.Технологии автоматизации тестирования, требования к автоматизированному тесту.	2	52
	Дифференцированный зачёт		
	34. Бесплатные инструменты автоматизированного тестирования (SilkTest, HP Quick Test Professional, Test Complete, Selenium и т.д.)	2	53
	<b>Практические занятия</b>		
	20.Проверочная работа по теории автоматизированного тестирования		54
	21.Разработка тест-кейсов и сценариев тестирования для программы	3	55
	22. Ознакомление с инструментами автоматизированного тестирования	3	56
	23.Изучение интерфейса инструмента автоматизированного тестирования	3	57
	24.Выполнение автоматизированного тестирования программы и составление отчета	3	58
	25.Защита отчёта о результатах автоматизированного тестирования	3	59

<b>Раздел 2. Документирование</b>			
Тема 2.1. Документиро- вание программного обеспечения в соответствии с Единой системой программной документации	<b>Содержание учебного материала</b>		
	35.Понятие стандартизации	2	60
	36. Структура ЕСПД	2	61
	37 Структура стандарта в ЕСПД.	2	62
	38.Стандарты документирование сложных ПС (международные ISO 9294, 12182, 12207, 15910, 9127; IEEE1063 -1987)	2	63
	<b>Практические занятия</b>		
	26. Проверочная работа по теории документирования по ЕСПД	3	64
27.Проверочная работа по теории документирования по международным стандартам	3	65	
Тема 2.2. Документи- рование основных процессов ЖЦ ПО	<b>Содержание учебного материала</b>		
	39.Документирование предварительных требований заказчика, обследование и описание системы, целей разработки ПС, концепция и основные предложения по созданию базовой версии, предварительный план разработки базовой версии,	2	66
	40. Техническое задание	2	67
	41. Документирование проектирования и выбора характеристик качества ПС, план обеспечения качества ПС	2	68
	42. Спецификация требований к системе и комплексу программ	2	69
	43. Пояснительная записка к предварительному или детальному проекту программного средства.	2	70
	44.Документирование процесса разработки и программирования компонент программных средств	2	71
	<b>Практические занятия</b>		
	28.Составление проекта разработки программного средства	3	72
	29.Защита проекта разработки программного средства	3	73
Тема 2.3. Документи- рование вспомогатель- ных процессов	<b>Содержание учебного материала</b>		
	45.Руководство программистам	2	746
	46.Обеспечение верификации и тестирования ПС, план, исходные данные и отчёты о результатах. Акт приёма ПС в промышленную эксплуатацию	2	75
	47.Документирование верификации, отчёт о результатах разработки, акт завершения	2	76

ЖЦ ПО	работ по проекту, акт приёмки ПС в промышленную эксплуатацию		
	48.Документирование сопровождения и конфигурационного управления версиями ПС (описание среды ЖЦ и конфигурации ПС, отчеты пользователей о выявленных дефектах и предложения по их корректировке, описание корректировок и результатов их тестирования, протоколы о выпуске новой версии ПС, отчет о результатах эксплуатации снятой с сопровождения версии)	2	77
	49.Документирование эксплуатации ПС (инструкция по формированию и ведению информации БД, Паспорт на ПС, руководство по обучению специалистов применению ПС)	2	78
	50.Требования к формированию Пользовательской документации (руководство системного администратора, руководство пользователя	2	79
	<b>Практические занятия</b>		
	30.Составление руководств системного администратора и пользователя	3	80
Тема 2.4. Документиро- вание сертификации	Содержание учебного материала		
	51.Сущность сертификации	2	81
	52. Проведение сертификации	2	82
	53Правовые и организационно-методические основы сертификаци		83
	54. Деятельность ИСО и МЭК в области сертификации.		84
	55 Документирование сертификации		85
	<b>Практические занятия</b>		
	31.Оформление документации на программные средства с использованием инструментальных средств	3	86
	<b>Всего по дисциплине</b>	172	

**Содержание разделов дисциплины**  
**Занятия лекционного типа**

Номер раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
<b>Раздел 1. Отладка и тестирование программного</b>			
	Тема 1.1. Понятие разработки ПО	1. Понятие программного обеспечения и его классификация	Т, У
		2. Жизненный цикл ПО.	Т, У
		3. Этапы разработки ПО	Т, У
	Тема 1.2. Понятие тестирования	4. Тестирование как часть процесса верификации программного обеспечения.	Т, У
		5. Обязанности тестировщика	Т, У
		6. Жизненный цикл тестирования	Т, У
	Тема 1.3. Тестирование требований	7. Понятие требований и их качество (уровни и типы)	Т, У
		8. Техника тестирования требований	Т, У
		9. Типичные ошибки при анализе и тестировании требований и причины их появления	Т, У
	Тема 1.4. Классификация тестирования	10. Классификация тестирования	Т, У
		11. Виды тестирования (по запуску кода на исполнение, по доступу к коду и архитектуре приложения, по степени автоматизации, по уровню тестирования)	Т, У
		12. Виды тестирования (уровню функционального тестирования, по принципам работы с приложением, по природе приложения, по привлечению конечных пользователей)	Т, У
		13. Виды тестирования (по степени формализации, по целям и задачам, по техникам и подходам, по хронологии)	Т, У
		14. Виды тестирования (по знанию системы, по исполнителям, по времени проведения, по производительности)	Т, У
		15. Альтернативные и дополнительные классификации	Т, У
		16. Классификация по принадлежности к методам «белого» и «чёрного» ящикам	Т, У
		17. Виды ошибок (ошибки (синтаксические, компоновки, выполнения, определения данных, накопления погрешностей, проектирования), дефекты, сбои, отказы и т.д.)	Т, У
	Тема 1.5. Отчёты о тестировании	18. Отчёт о дефектах, его атрибуты и ЖЦ	Т, У
		19. Инструментальные средства управления отчётами о дефектах	Т, У
		20. Свойства качественных отчётов и логика их создания. Типичные ошибки при написании отчёта о дефектах	Т, У
		21. Планирование тестирования (тест-план)	Т, У

		22. Оценка трудозатрат на планирование и тестирование	Т, У
Тема 1.6. Методы отладки и тестирования		23. Методы отладки: (ручного тестирования, индукции, дедукции, обратного прослеживания)	Т, У
		24. Методы тестирования (позитивные и негативные тест-кейсы, классы эквивалентности и граничные условия)	Т, У
		25. Методы тестирования (доменное тестирование и комбинация параметров)	Т, У
		26. Методы тестирования (попарное тестирование и поиск комбинаций, исследовательское тестирование)	Т, У
		27. Регрессионное тестирование и его виды	Т, У
		28. Поиск причин возникновения дефектов	Т, У
Тема 1.7. Автоматизированное тестирование		29. Автоматизация тестирования, её задачи, преимущества и ограничения. Пирамида тестирования.	Т, У
		30. Этапы автоматизированного тестирования	Т, У
		31. Разработка тест-кейсов. Сценарный тест	Т, У
		32. Виды автоматизированного тестирования (регрессионное, кроссбраузерное и кроссплатформенное, локации и производительности)	Т, У
		33. Технологии автоматизации тестирования, требования к автоматизированному тесту. Дифференцированный зачёт	
		34. Бесплатные инструменты автоматизированного тестирования (SilkTest, HP Quick Test Professional, Test Complete, Selenium и т.д.)	Т, У
<b>2Раздел 2. Документирование</b>			
Тема 2.1. Документирование программного обеспечения в соответствии с ЕСПД		35. Понятие стандартизации	Т, У
		36. Структура ЕСПД	Т, У
		37 Структура стандарта в ЕСПД.	Т, У
		38. Стандарты документирование сложных ПС (международные ISO 9294, 12182, 12207, 15910, 9127; IEEE 1063 -1987)	Т, У
Тема 2.2. Документирование основных процессов ЖЦ ПО		39. Документирование предварительных требований заказчика, обследование и описание системы, целей разработки ПС, концепция и основные предложения по созданию базовой версии, предварительный план разработки базовой версии,	Т, У
		40. Техническое задание	Т, У
		41. Документирование проектирования и выбора характеристик качества ПС, план обеспечения качества ПС	Т, У
		42. Спецификация требований к системе и комплексу программ	Т, У
		43. Пояснительная записка к предварительному или детальному проекту программного средства.	Т, У

		44.Документирование процесса разработки и программирования компонент программных средств	Т, У
Тема 2.3. Документирование вспомогательных процессов ЖЦ ПО		45.Руководство программистам	Т, У
		46.Обеспечение верификации и тестирования ПС, план, исходные данные и отчёты о результатах. Акт приёма ПС в промышленную эксплуатацию	Т, У
		47.Документирование верификации, отчёт о результатах разработки, акт завершения работ по проекту, акт приёмки ПС в промышленную эксплуатацию	Т, У
		48.Документирование сопровождения и конфигурационного управления версиями ПС (описание среды ЖЦ и конфигурации ПС, отчеты пользователей о выявленных дефектах и предложения по их корректировке, описание корректировок и результатов их тестирования, протоколы о выпуске новой версии ПС, отчет о результатах эксплуатации снятой с сопровождения версии)	Т, У
		49.Документирование эксплуатации ПС (инструкция по формированию и ведению информации БД, Паспорт на ПС, руководство по обучению специалистов применению ПС)	Т, У
		50.Требования к формированию Пользовательской документации (руководство системного администратора, руководство пользователя	Т, У
Тема 2.4. Документирование сертификации		51.Сущность сертификации	Т, У
		52. Проведение сертификации	Т, У
		53. Правовые и организационно-методические основы сертификации	Т, У
		54. Деятельность ИСО и МЭК в области сертификации.	Т, У
		55 Документирование сертификации	Т, У
Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа			

### Занятия семинарского типа

Не предусмотрено

### Практические занятия

Номер раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	<b>Раздел 1. Отладка и тестирование программного обеспечения</b>		
	Тема 1.1. Понятие разработки ПО	1.Проверочная работа по теме 1.1	
	Тема 1.2. Понятие	1.Проверочная работа по теме 1.2	

	тестирования		
	Тема 1.3. Тестирование требований	3.Проверочная работа по теории	ПР, У
		4.Составление требований к программе	ПР, У
		5.Проверка составленных требований	ПР, У
		6.Анализ выявленных ошибок	ПР, У
	Тема 1.4. Классификация тестирования	8.Тестирование программы методом «белого ящика»	ПР, У
		9.Перекрёстная проверка результатов тестирования	ПР, У
		10.Тестирование программы методом «чёрного» ящика.	ПР, У
		11.Перекрёстная проверка результатов тестирования	ПР, У
	Тема 1.5. Отчёты о тестировании	13.Составление проекта отчёта о дефектах	ПР, У
		14.Перекрёстный анализ составленных отчётов	ПР, У
		15. Составление тест-плана	ПР, У
		16. Оценка трудозатрат на планирование и тестирование по составленным планам	ПР, У
	Тема 1.6. Методы отладки и тестирования	17.Проверочная работа по теории методов тестирования	ПР, У
		18.Отладка программы всеми известными методами и составление отчёта о результатах	ПР, У
		19.Тестирование одной программы 9-ю методами и составление отчёта о результатах	ПР, У
	Тема 1.7. Автоматизированное тестирование	20.Проверочная работа по теории автоматизированного тестирования	ПР, У
		21.Разработка тест-кейсов и сценариев тестирования для программы	ПР, У
		22. Ознакомление с инструментами автоматизированного тестирования	ПР, У
		23.Изучение интерфейса инструмента автоматизированного тестирования	ПР, У
		24.Выполнение автоматизированного тестирования программы и составление отчета	ПР, У
		25.Защита отчёта о результатах автоматизированного тестирования	ПР, У
<b>Раздел 2. Документирование</b>			
	Тема 2.1. Документирование программного обеспечения в соответствии с ЕСПД	26. Проверочная работа по теории документирования по ЕСПД	ПР, У
		27.Проверочная работа по теории документирования по международным стандартам	ПР, У
	Тема 2.2. Документирование основных процессов ЖЦ ПО	28.Составление проекта разработки программного средства	ПР, У
		29.Защита проекта разработки программного средства	ПР, У
	Тема 2.3. Документирование вспомогательных процессов ЖЦ ПО	30.Составление руководств системного администратора и пользователя	ПР, У



	Тема 2.4. Документирование сертификации	31.Оформление документации на программные средства с использованием инструментальных средств	ПР, У
		Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа	

## Содержание самостоятельной работы

Подготовка к зачёту и к экзамену

### Теоретические вопросы:

1. Понятие тестирования. Принципы, виды и методы тестирования программных продуктов
2. Принцип построения тестового набора данных и составления отладочных заданий.
3. Оформление протокола тестирования.
4. Структурное тестирование.
5. Пошаговое и монолитное тестирование.
6. Оценочное тестирование. Виды и принципы проведения оценочного тестирования.
7. Нисходящее и восходящее тестирование. Критерии формирования тестовых наборов
8. Системное и функциональное тестирование.
9. Определение количества ошибок в ПП и числа необходимых тестов
10. Тестирование программного продукта методом «белого ящика»
11. Тестирование программного продукта методом «чёрного ящика»
12. Понятие отладки программных продуктов.
13. Принципы отладки программных продуктов.
14. Классификация ошибок. Локализация ошибок
15. Методы отладки программного продукта
16. Методы ручного тестирования
17. Метод обратного прослеживания
18. Метод индукции. Метод дедукции.
19. Инструментальные средства отладки ПП
20. Системное программирование, системное ПО.
21. Формализация задачи и разработка алгоритма.
22. Жизненный цикл ПО. Основные этапы разработки ПО.
23. Модели жизненного цикла программного средства.
24. Основные понятия структурного программирования.
25. Модуль. Структура модуля.
26. Списки. Объявление списка, инициализация списка, печать
27. Стеки. Объявление стека, инициализация стека. Добавление элемента в стек.
28. Очереди. Объявление, инициализация очереди. Добавление элемента в очередь.
29. Создание и заполнение внешнего файла, чтение данных из внешнего файла.
30. Текстовые файлы.
31. Структура и способы описания языков программирования высокого уровня.
32. Подпрограмма – процедура. Подпрограмма- функция.
33. Формальные и фактические параметры.
34. Локальные и глобальные переменные.
35. Разработка программного продукта с использованием подпрограммы-процедуры.
36. Модульное программирование.
37. Методы разработки программных модулей.

38. Осуществление разработки кода программного модуля на современных языках программирования
39. Реализация процедур и функций работы с бинарным деревом.
40. Разработка программного продукта с использованием модуля.
41. Объектно-ориентированное проектирование.
42. Документирование результатов анализа и проектирования.
43. Основы языка UML (Unified Modeling Language).
44. Создание абстрактных типов данных. Диаграмма объекта.
45. Принципы объектно-ориентированного анализа: абстрагирование, инкапсуляция, наследование, полиморфизм, модульность, сохраняемость, параллелизм
46. Структура программы на языке C++. Проект.
47. Компиляция программы и сборка исполняемого модуля.
48. Размещение программы и данных в памяти.
49. Структура исполняемого модуля.
50. Стандартная библиотека функций языка C++.
51. Компиляция программы и сборка исполняемого модуля.
52. Размещение программы и данных в памяти.
53. Виртуальные функции и абстрактные базовые классы.
54. Множественное наследование.
55. Ассоциативные массивы.
56. Объекты-функции и предикаты.
57. Цикл разработки прикладного программного обеспечения: концептуализация, анализ, проектирование, кодирование, тестирование, эволюция, сопровождение.
58. Критерии оценки качества программы.
59. Средства и инструменты разработки программного обеспечения.
60. Разработка кода программного продукта на основе готовой спецификации на уровне модуля.
61. Ознакомление с технологией тестирования программных продуктов
62. Выполнение отладки и тестирования программы на уровне модуля
63. Использование инструментальных средств на этапе отладки программного продукта
64. Тестирование программного модуля по определенному сценарию
65. Использование инструментальных средств автоматизации процесса оформления документации.
66. Создание документации к программам. Системы автоматического создания документации. Использование комментариев в программах.
67. Создание собственных модулей. Выкладка их в общий репозиторий на PyPi. Создание инсталляционных пакетов.
68. Тестирование приложений. Тестирование черного и белого ящика.

#### **Практические задания:**

1. Дан массив A из n целых чисел. Найти сумму максимального и минимального элемента в массиве. (Поиск максимума и минимума реализовать с помощью подпрограмм-функций).
2. Дан файл целых чисел. Выбрать наибольшее из чисел, принадлежащее интервалу [a,b]. Концы интервала a и b вводятся с клавиатуры.
3. Дан текстовый файл F1. Переписать его содержимое в файл F2, сохраняя строчную структуру и удаляя пустые строки.

4. Даны две символьные строки S1 и S2, содержащие только строчные латинские буквы. Построить строку S3, в которую войдут только общие символы S1 и S2 в алфавитном порядке и без повторений.
5. Дан файл целых чисел. Определить, сколько раз в нем повторяется максимальное значение.
6. Дан файл целых чисел. Определить, сколько раз в нем повторяется максимальное значение.
7. По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка, соединяющего две точки. (длина отрезка=  $\text{sgrt}(\text{sgr}(x_2-x_1)+\text{sgr}(y_2-y_1))$ ), где (x1,y1)- координаты одной точки, (x2,y2)-координаты второй точки отрезка).
8. Дан файл целых чисел F1. Создать два новых файла F2 и F3 из положительных и отрицательных чисел соответственно
9. Даны два файла целых чисел. Определить, в каком из них больше положительных, отрицательных и нулевых значений.
10. Составить рекурсивную подпрограмму вычисления N!
11. Дана вещественная матрица размера m\*n. Найти значение наибольшего по модулю элемента матрицы и указать его местоположение в матрице.
12. Определить среднее арифметическое чисел, хранящихся в файле Note.txt.
13. Дан список L, из N целых чисел. Удалить первое вхождение максимального элемента в списке.
14. Дан список L, из N целых чисел. Удалить первое вхождение минимального элемента в списке.
15. Дан текстовый файл Note.txt. Определить длину самой длинной строки этого файла.
16. Разработать и произвести отладку программы: Найти сумму бесконечного ряда. Суммировать до тех пор, пока сумма не станет больше заданного  $p > 0$ . Вывести эти числа.
17. Разработать и произвести отладку программы для определения N!-M!.  $N! = 1*2*3*4*.....*n$
18. Разработать и произвести отладку программы: Вычислить сумму квадратов всех целых чисел, пока сумма квадратов меньше заданного числа A. Вывести эти числа.
19. Разработать и произвести отладку программы: Произведение первых четных чисел равно P, сколько сомножителей взято.
20. Разработать и произвести отладку программы: Определить все двузначные числа, сумма квадратов цифр которых кратны числу 15.
21. Разработать и произвести отладку программы: Даны два одномерных массива одинаковой длины. Получить третий массив такой же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов.
22. Разработать и произвести отладку программы: дан одномерный массив чисел. Определите сумму элементов, принадлежащих промежутку от A до B (A и B водить с клавиатуры).
23. Разработать и произвести отладку программы определения количества элементов массива, больших среднего арифметического всех его элементов.
24. Разработать и произвести отладку программы: Дан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка [-10,10]. Из элементов массива P сформировать массив M той же размерности по правилу: если номер четный, то  $M_i = i * P_i$ , если нечетный, то  $M_i = -P_i$ . Исходный и скорректированный массив вывести на экран.
25. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка [-30,30]. Из элементов массива P сформировать массив M из четных чисел. Исходный и скорректированный массивы вывести на экран.
26. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка [-10,10]. Из элементов массива P сформировать массив M той же размерности по возрастанию. Исходный и скорректированный массивы вывести на экран.
27. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке убывания.

28. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке возрастания
29. Разработать и произвести отладку программы для решения квадратного уравнения.
30. Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.
31. Разработать и произвести отладку программы для вычисления делителей натурального числа  $N$ . Вывести сами делители, их количество.
32. Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа  $N$ . Вывести эти цифры и сумму.
33. Создать и отладить приложение для решения квадратного уравнения.
34. Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.
35. Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.
36. Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.
37. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; заменить отрицательные числа на 0, положительные – на 1.
38. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по убыванию.
39. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по возрастанию
40. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; поменять местами два элемента массива с номерами  $k_1$  и  $k_2$ .
41. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет минимальный и максимальный элементы массива.
42. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет сумму всех элементов и количество положительных элементов.

### 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

#### Образовательные технологии при проведении лекций

Для реализации компетентного подхода предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

Преподавание дисциплины организовано по модульно-блочному принципу.

В процессе преподавания применяются образовательные технологии развития абстрактного, логического и критического мышления. Обязательны компьютерные практикумы по разделам дисциплины, тестирование, тематические презентации, интерактивные технологии.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное мультимедийное оборудование и интернет

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
1	Тема 1.1. Понятие разработки ПО	МБТ, ИКТ	8*
2	Тема 1.2. Понятие тестирования	МБТ, ИКТ	8*
3	Тема 1.3. Тестирование требований	МБТ, ИКТ	14*
4	Тема 1.4. Классификация тестирования	МБТ, ИКТ	26*
5	Тема 1.5. Отчёты о тестировании	МБТ, ИКТ	20*
6	Тема 1.6. Методы отладки и тестирования	МБТ, ИКТ	18*
7	Тема 1.7. Автоматизированное тестирование	МБТ, ИКТ	24*
8	Тема 2.1. Документирование программного обеспечения в соответствии с Единой системой программной документации	МБТ, ИКТ	12*
9	Тема 2.2. Документирование основных процессов ЖЦ ПО	МБТ, ИКТ	16*
10	Тема 2.3. Документирование вспомогательных процессов ЖЦ ПО	МБТ, ИКТ	14*
11	Тема 2.4. Документирование сертификации	МБТ, ИКТ	12*
<b>Итого по курсу</b>			<b>172*</b>
в том числе интерактивное обучение при необходимости*			

#### Образовательные технологии при проведении практических занятий (лабораторных работ)

№	Тема занятия	Виды технологий	Часы
1	Проверочная работа по теме 1.1	ИКТ	
2	Проверочная работа по теме 1.2	ИКТ	
3	Проверочная работа по теории	ИКТ	

4	Составление требований к программе	ИКТ	
5	Проверка составленных требований	ИКТ	
6	Анализ выявленных ошибок	ИКТ	
7	Тестирование программы методом «белого ящика»	ИКТ	
8	Перекрёстная проверка результатов тестирования	ИКТ	
9	Тестирование программы методом «чёрного» ящика.	ИКТ	
10	Перекрёстная проверка результатов тестирования	ИКТ	
11	Составление проекта отчёта о дефектах	ИКТ	
12	Перекрёстный анализ составленных отчётов	ИКТ	
13	Составление тест-плана	ИКТ	
14	Оценка трудозатрат на планирование и тестирование по составленным планам	ИКТ	
15	Проверочная работа по теории методов тестирования	ИКТ	
16	Отладка программы всеми известными методами и составление отчёта о результатах	ИКТ	
17	Тестирование одной программы 9-ю методами и составление отчёта о результатах	ИКТ	
18	Проверочная работа по теории автоматизированного тестирования	ИКТ	
19	Разработка тест-кейсов и сценариев тестирования для программы	ИКТ	
20	Ознакомление с инструментами автоматизированного тестирования	ИКТ	
21	Изучение интерфейса инструмента автоматизированного тестирования	ИКТ	
22	Выполнение автоматизированного тестирования программы и составление отчета	ИКТ	
23	Защита отчёта о результатах автоматизированного тестирования	ИКТ	
24	Проверочная работа по теории документирования по ЕСПД	ИКТ	
25	Проверочная работа по теории документирования по международным стандартам	ИКТ	
26	Составление проекта разработки программного средства	ИКТ	
27	Защита проекта разработки программного средства	ИКТ	
28	Составление руководств системного администратора и пользователя	ИКТ	
29	Оформление документации на программные средства с использованием инструментальных средств	ИКТ	
	Примечание: Т – тестирование, Р – написание реферата, У – устный опрос, КР – контрольная работа		

#### **4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.01.02 «Поддержка и тестирование модулей программного обеспечения»**

##### **Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Кабинет для преподавания дисциплины *«Основы разработки баз данных»*, оснащенный оборудованием и техническими средствами обучения:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- необходимая для проведения практических занятий методическая и справочная литература (в т.ч. в электронном виде).
- компьютеры;
- выход в интернет;
- мультимедийный проектор, экран;
- мультимедийные презентации.

##### **Информационное обеспечение реализации программы**

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе

##### **Основная литература**

1. **Белугина, С. В.** Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие для СПО / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 312 с. — ISBN 978-5-8114-9817-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200390> — Режим доступа: для авториз. пользователей.
2. **Гагарина, Л.Г.** Технология разработки программного обеспечения : учебное пособие для студентов вузов, обучающихся по направлениям подготовки "Информатика и вычислительная техника" / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул ; под ред. Л. Г. Гагариной. - Москва : ФОРУМ : ИНФРА-М, 2017. - 399 с. : ил. - (Высшее образование). - Библиогр.: с. 388-391. - ISBN 978-5-8199-0342-1. - ISBN 978-5-16-003193-4 : Текст непосредственный
3. **Гниденко, И. Г.** Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492496>
4. **Федорова, Г. Н.** **Разработка модулей программного обеспечения для компьютерных систем** : учебник для студентов учреждений среднего профессионального образования / Г. Н. Федорова. - 4-е изд., перераб. - Москва : Академия, 2020. - 383 с. : ил. -

(Профессиональное образование. ТОП-50). - Словарь терминов: с. 372-377. - Библиогр.: с. 378. - ISBN 978-5-4468-8692-0 . - Текст : непосредственный. (25)

### Дополнительная литература

1. Акопов, А. С. Компьютерное моделирование : учебник и практикум для среднего профессионального образования / А. С. Акопов. — Москва : Издательство Юрайт, 2022. — 389 с. — (Профессиональное образование). — ISBN 978-5-534-10712-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/495518>
2. Богатырев, В. А. Информационные системы и технологии. Теория надежности : учебное пособие для вузов / В. А. Богатырев. — Москва : Издательство Юрайт, 2022. — 318 с. — (Высшее образование). — ISBN 978-5-534-00475-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490026>
3. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. — Москва : Издательство Юрайт, 2022. — 258 с. — (Профессиональное образование). — ISBN 978-5-534-03173-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491568>
4. Советов, Б. Я. Информационные технологии : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский. — 7-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 327 с. — (Профессиональное образование). — ISBN 978-5-534-06399-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489604>

### Периодические издания

1. Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
  2. Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
  3. Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>
  4. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
  5. Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
  6. Computerword Россия.- URL: <http://dlib.eastview.com/browse/publication/64081>
  7. Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
  8. Информационно-управляющие системы.- URL: <http://dlib.eastview.com/browse/publication/71235>
  9. Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
  10. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
  11. Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>
- Прикладная информатика.- URL: [http://elibrary.ru/title\\_about.asp?id=25599](http://elibrary.ru/title_about.asp?id=25599)

### Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «Университетская библиотека ONLINE»: сайт. – URL:<http://biblioclub.ru>
2. ЭБС Издательства «Лань»: сайт. – URL:<http://e.lanbook.com>
3. ЭБС «Юрайт»: сайт. –URL:<https://urait.ru/>
4. ЭБС «BOOK.ru»: сайт. – URL: <https://www.book.ru>
5. ЭБС «ZNANIUM.COM»: сайт. – URL: <https://www.znanium.com>



6. Базы данных компании «Ист Вью»: сайт . –URL: <http://dlib.eastview.com>
7. Научная электронная библиотека «eLibrary.ru»: сайт. – URL: <http://elibrary.ru/>
8. Электронная библиотека "Издательского дома "Гребенников". - URL: <http://www.grebennikon.ru/>
9. Университетская информационная система РОССИЯ (УИС Россия). - URL: <http://uisrussia.msu.ru/>
10. "Лекториум ТВ" - видеолекции ведущих лекторов России. - URL: <http://www.lektorium.tv/>
11. База учебных планов, учебно-методических комплексов, публикаций и конференций КубГУ. - URL: <http://docspace.kubsu.ru/>

## 5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

### МДК.01.02 «Поддержка и тестирование модулей программного обеспечения»

#### Критерии оценивания практических работ

Раздел модуля 2. Технологии тестирования программных модулей		
<p>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Оценка <b>«отлично»</b> - выполнена отладка модуля (Дополнительно для квалификаций <b>"Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования</b>); с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«хорошо»</b> - выполнена отладка модуля (Дополнительно для квалификаций <b>"Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования</b>); сохранены и представлены результаты отладки.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

<p>ПК 1.4 Выполнять тестирование программных модулей</p>	<p>Оценка «<b>отлично</b>» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами. <b>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий"</b>: выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия, сделан вывод о достаточности тестового пакета.</p> <p>Оценка «<b>хорошо</b>» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования. <b>Дополнительно</b></p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля.</p> <p><b>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий"</b>: оценке тестового по-</p>
--	---	---

	<p>для квалификации "<b>Специалист по тестированию в области информационных технологий</b>": выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнено тестирование модуля и оформлены результаты тестирования. <b>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий"</b>: выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия с некоторыми погрешностями.</p>	<p>крытия.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Оценка <b>«отлично»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>Оценка <b>«хорошо»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p> <p>Оценка <b>«удовлетворительно»</b> - определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

Отметка «5» ставится, если:

- работа выполнена полностью, в логических рассуждениях и обосновании решения нет пробелов и ошибок;
- в решении нет ошибок (возможна одна неточность, описка, не являющаяся следствием незнания или непонимания учебного материала).

Отметка «4» ставится, если:

- работа выполнена полностью, но в обосновании шагов решения недостаточны;
- допущена 1-2 ошибки или 1 ошибка и два-три недочета в выкладках.
- Отметка «3» ставится, если:
- допущены 3 ошибки или 2 ошибки и более двух-трех недочетов в выкладках, но обучающийся владеет обязательными умениями по проверяемой теме.
- Отметка «2» ставится, если:
- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными умениями по данной теме в полной мере

### – 5.2. Критерии оценивания конспектов

- Отметка «5» ставится, если:
- работа содержит полные ответы на все теоретические вопросы для составления конспекта;

Отметка «4» ставится, если:

- работа содержит неполный ответ хотя бы на один теоретический вопрос для составления конспекта;

Отметка «3» ставится, если:

- работа содержит неполные ответы на 2 теоретических вопроса для составления конспекта.

Отметка «2» ставится, если:

- работа содержит неполные ответы на 2 и более теоретических вопроса для составления конспекта.

### Критерии оценивания презентаций

Оценка	5	4	3	2
Содержание	Работа полностью завершена	Почти полностью сделаны наиболее важные компоненты	Не все важнейшие компоненты работы выполнены	Работа сделана фрагментарно и с помощью преподавателя

		работы		
	Работа демонстрирует глубокое понимание описываемых процессов	Работа демонстрирует понимание основных моментов, хотя некоторые детали не уточняются	Работа демонстрирует понимание, но неполное	Работа демонстрирует минимальное понимание
	Грамотно используется научная лексика	Научная лексика используется, но иногда не корректно	Научная терминология или используется мало или используется некорректно.	Минимум научных терминов
<b>Грамотность</b>	Нет ошибок: ни грамматических, ни синтаксических	Минимальное количество ошибок	Есть ошибки, мешающие восприятию	Много ошибок, делающих материал трудночитаемым
<b>Дизайн</b>	Имеются постоянные элементы дизайна. Дизайн подчеркивает содержание	Имеются постоянные элементы дизайна. Дизайн соответствует содержанию.	Нет постоянных элементов дизайна. Дизайн может и не соответствовать содержанию.	Элементы дизайна мешают содержанию, накладываясь на него.
	Все параметры шрифта хорошо подобраны (текст хорошо читается)	Параметры шрифта подобраны. Шрифт читаем	Параметры шрифта недостаточно хорошо подобраны, могут мешать восприятию	Параметры не подобраны. Делают текст трудночитаемым

## **6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Обучающиеся для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку, как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа является важнейшей формой учебно-познавательного процесса. Цель заданий для самостоятельной работы – закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Обучающийся должен изучить список нормативно-правовых актов и экономической литературы, рекомендуемый по учебной дисциплине; уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания обучающийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Обучение осуществляется по модульно-блочной технологии (лекции, практики) с включением инновационных элементов.

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;
- запись ведётся очень быстро, чётко, по возможности короткими выражениями;
- не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить. Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;

- имена, даты, названия, выводы, определения записываются точно;
- следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине «Стандартизация, сертификация и техническое документирование» проводятся в основном по схеме:

- устный опрос по теории в начале занятия (обсуждение теоретических проблемных вопросов по теме);
- работа в группах по разрешению различных ситуаций по теме занятия;
- решение практических задач индивидуально;
- подведение итогов занятия (или рефлексия);
- индивидуальные задания для подготовки к следующим практическим занятиям.

Цель практического занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

- вербальные (преобладающим методом должно быть объяснение);
- практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

- библиотечные фонды филиала КубГУ в г. Геленджике;
- электронная библиотечная система «Университетская библиотека онлайн»;
- электронная библиотечная система Издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения.

Просмотр книги начинается с титульного листа, следующего после обложки. На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание.

Прочитав предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

- пометки, замечания, выделение главного;
- план, тезисы, выписки, цитаты;
- конспект, рабочая запись, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике.

Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей



и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка.

Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи - записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7-15 раз;
- конспект может быть, как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:

- прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;
- на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;
- записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;
- конспектирование ведётся не с целью иметь определённые записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;
- после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обращаться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

- конспектируя книгу большого объёма, запись следует вести в общей тетради;
- на каждой странице слева оставляют поля шириной 25-30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;
- каждая страница тетради нумеруется;
- для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;
- при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять

общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.

- не следует сокращать имена и названия, кроме очень часто повторяющихся;
- в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Для написания реферата необходимо выбрать тему, согласовать ее с преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

## 7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

### Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Основы математической логики	ОК 1, 2,4,5,9,10	Тестирование, анализ выполнения практических работ
2.	Понятие множества	ОК 1, 2,4,5,9,10	Тестирование, анализ выполнения практических работ
3.	Понятие и основные свойства предиката Элементы теории графов	ОК 1, 2,4,5,9,10	Тестирование, анализ выполнения практических работ
4.	Элементы теории алгоритмов. Основы математической логики	ОК 1, 2,4,5,9,10	Тестирование, анализ выполнения практических работ
5.	Элементы теории алгоритмов	ОК 1, 2,4,5,9,10	Тестирование, анализ выполнения практических работ, зачёт

#### Критерии оценки знаний

Контроль и оценка результатов освоения модуля осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения студентами индивидуальных самостоятельных заданий.

**Тест.** Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов (не менее 50%).

#### Критерии оценки знаний студентов в целом по дисциплине:

**«отлично»** - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы модуля и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

**«хорошо»** - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы модуля, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

### **Оценочные средств для проведения текущей аттестации**

Дайте ответы на вопросы

**1. Тестирование программного обеспечения (Software Testing)** - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. [IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004] В более широком смысле, тестирование - это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

**2. Верификация (Verification)** - это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа [IEEE]. Т.е. выполняются ли наши цели, сроки, задачи по разработке проекта, определенные в начале текущей фазы.

**3. Валидация (Validation)** - это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе [BS7925-1].

**4. Тест дизайн (Test Design)** - это этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест кейсы), в соответствии с определёнными ранее критериями качества и целями тестирования.

**5. Тестовый случай (Test Case)** - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

**6. Баг/Дефект Репорт (Bug Report)** - это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

**7. Тестовое Покрытие (Test Coverage)** - это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

**8. Время Прохождения Тест Кейса (Test Case Pass Time)** - это время от начала прохождения шагов тест кейса до получения результата теста.

**9. Цели тестирования** ( Обнаружение дефектов, Повышение уверенности в уровне качества, Предоставление информации для принятия решений, Предотвращение дефектов)

**10. Принципы тестирования:** (Тестирование показывает наличие дефектов, Исчерпывающее тестирование невозможно, Раннее тестирование, Скопление дефектов, Повторные прогоны тестов находят меньше ошибок, заблуждение об отсутствии дефектов, **Тестирование зависит от контекста, Заблуждение об отсутствии ошибок**)

**11. Цели тестирования** (Увеличение приемлемого уровня пользовательского доверия в том, что программа функционирует корректно во всех необходимых обстоятельствах: корректное поведение; определяется из требований к продукту; уровень доверия; наглядность, уровень остаточного обнаружения дефектов, требования к надёжности, что бы это всё ни значило, необходимые обстоятельства - требование реального окружения; реалистичная среда тестирования (схожие наборы данных и т.п.).

**12. Уровни восприятия тестирования в компании.**

Уровень 0 - (Не отличает некорректное поведение и ошибки в программе. Не учитывает требования надежности и безопасности)

Уровень 1 - предназначение – показать корректность ПО

Уровень 2 - Демонстрация ошибок

Уровень 3 - Тестирование может показать наличие ошибок

**13. Риски использования ПО** (Последствия незначительные, последствия катастрофические)

**14. Участники тестирования, их роль, квалификация и обязанности**

1. Проектирование тестов – На основании формальных критериев – На основании знаний предметной области, опыта и экспертизы

2. Автоматизация тестов – Знание средств, скриптов

3. Исполнение тестов – Нет специальных требований к квалификации

4. Анализ результатов – Знания предметной области)

**15. Мониторинг прогресса и контроль тестирования (ISTQB)** (Целью мониторинга тестирования является предоставление результата и обзора процесса тестирования. Информация отслеживается вручную или автоматически и может быть использована для измерения критериев выхода, таких как покрытие. Метрики также могут быть использованы для оценки прогресса тестирования по сравнению с запланированным расписанием и бюджетом. *Контроль тестирования* описывает любые направляющие или корректирующие действия, принятые как результат по полученной и собранной информации и значениям метрик. Контроль тестирования может затрагивать любые действия по тестированию, а также воздействовать на другие действия и задачи жизненного цикла ПО)

**16. Тестовый случай, тестовый сценарий и тестовое покрытие.**

(*Тестовый случай (Test Case* - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части. Под тест кейсом понимается структура вида: Action > Expected Result > Test Result

*Тестовый сценарий* - последовательность тестовых случаев; состоит из набора входных значений, предусловий выполнения, ожидаемых результатов и постусловий, определяемых для покрытия определенных тестовых условий ( или тестового условия) или целей (цели) тестирования.

*Тестовое Покрытие* - это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.)

### Тема 1.3. Тестирование требований

#### ЛАБОРАТОРНАЯ РАБОТА №3

#### Тестирование требований

*Цель работы:* изучить критерии качества требований, выполнить тестирование спецификации требований.

#### *Теоретические сведения*

Требования обеспечивают основу для последующего тестирования - процесса анализа программного средства на предмет соответствия зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика. От качества сформированных требований зависит качество программного обеспечения, т.к. требования к программному продукту являются базой для генерации тестов и обнаружения дефектов, представляющих собой любое отклонение от спецификации.

В связи с вышеизложенным при разработке программного обеспечения тестирование необходимо выполнять уже на стадии разработки спецификации (рисунок 1).



**Рисунок 1 - Жизненный цикл проекта**

Тестирование требований является важным этапом разработки продукта, так как это приводит к:

- 1) снижению риска получить продукт, не отвечающий ожиданиям заказчика или нуждам конечных пользователей;
- 2) снижению затрат на разработку и тестирование продукта;
- 3) сокращению сроков сдачи готового продукта;
- 4) налаживанию взаимопонимания при создании продукта между всеми вовлеченными исполнителями.

Выделяют 9 критериев качества требований:

- 1) корректность;
- 2) недвусмысленность;

- 3) полнота;
- 4) непротиворечивость;
- 5) упорядоченность по важности и стабильности;
- 6) проверяемость;
- 7) модифицируемость;
- 8) трассируемость;

**Корректные требования.** Набор требований к программному обеспечению является корректным тогда и только тогда, когда каждое требование, сформулированное в нем, представляет нечто, требуемое от создаваемой системы. Данная формулировка отражена на рисунке 2.

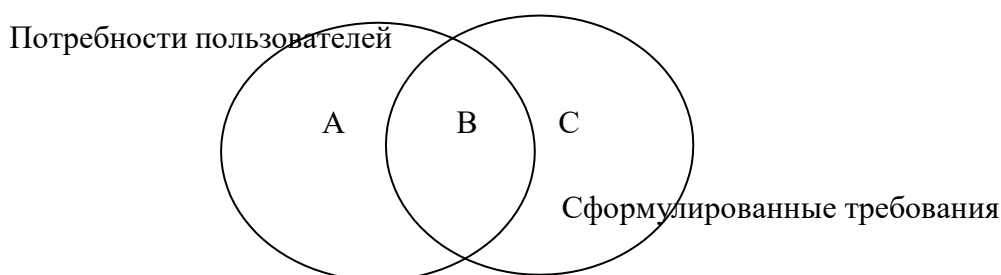


Рисунок 2 - Множества потребностей и требований

Если левый круг (*область A*) представляет множество потребностей пользователя, а правый (*область C*) — требования, то корректные требования будут находиться в области пересечения кругов, *область B*.

**Недвусмысленные требования.** Требование является недвусмысленным тогда и только тогда, когда его можно однозначно интерпретировать. Хотя главным свойством любого требования по праву считается корректность, неоднозначность зачастую представляет собой более сложную проблему. Если формулировка требований может по-разному интерпретироваться разработчиками, пользователями и другими участниками проекта, вполне может оказаться, что построенная система будет полностью отличаться от того, что представлял себе пользователь.

**Полнота набора требований.** Набор требований является полным тогда и только тогда, когда он описывает все важные требования, интересующие пользователя, в том числе требования, связанные с функциональными возможностями, производительностью, ограничениями проектирования, атрибутами или внешними интерфейсами. Полный набор требований должен также задавать требуемый ответ программы на всевозможные классы ввода — как правильные, так и неправильные — во всевозможных ситуациях. Помимо этого, он должен содержать полные ссылки и подписи всех рисунков, таблиц и диаграмм набора требований, а также определения всех терминов и единиц измерения.

**Непротиворечивость набора требований.** Множество требований является внутренне непротиворечивым, когда ни одно его подмножество, состоящее из отдельных требований, не противоречит другим подмножествам. Конфликты могут

иметь различную форму и проявляться на различных уровнях детализации; если набор требований был написан достаточно формально и поддерживается соответствующими автоматическими средствами, конфликт иногда удастся обнаружить посредством механического анализа. Но, скорее всего, разработчикам вместе с другими участниками проекта придется провести проверку множества требований вручную, чтобы удалить все потенциальные конфликты.

**Упорядочение требований по их важности и стабильности.** В высококачественном наборе требований разработчики, клиенты и другие заинтересованные лица упорядочивают отдельные требования по их важности для клиента и стабильности. Этот процесс упорядочения особенно важен для управления масштабом. Если ресурсы недостаточны, чтобы в пределах выделенного времени и бюджета реализовать все требования, очень полезно знать, какие требования являются не столь уж обязательными, а какие пользователь считает критическими.

**Проверяемые требования.** Требование в целом является проверяемым, когда каждое из составляющих его элементарных требований является проверяемым, т.е. когда можно протестировать каждое из них и выяснить, действительно ли они выполняются.

**Модифицируемый набор требований.** Множество требований является модифицируемым, когда его структура и стиль таковы, что любое изменение требований можно произвести просто, полно и согласованно, не нарушая существующей структуры и стиля всего множества. Для этого требуется, чтобы пакет требований имел минимальную избыточность и был хорошо организован, с соответствующим содержанием, индексом и возможностью перекрестных ссылок.

**Трассируемые требования.** Требование в целом является трассируемым, когда ясно происхождение каждого из составляющих его элементарных требований и существует механизм, который делает возможным обращение к этому требованию при дальнейших действиях по разработке. На практике это обычно означает, что каждое требование имеет уникальный номер или идентификатор. Возможность трассировки имеет огромное значение. Разработчики могут использовать ее как для достижения лучшего понимания проекта, так и для обеспечения более высокой степени уверенности, что все требования выполняются данной реализацией.

#### **Существуют различные методы тестирования требований:**

1. Метод просмотра (универсальный метод, выполняется бизнес-аналитиком или тестировщиком):
  - Ознакомление с требованиями.
  - Проверка требований по критериям качества.
  - Оформление дефектов.
  - Оформление отчета.
2. Метод экспертизы (выполняется при участии команды из бизнес-аналитиков, представителей заказчика, разработчиков, лояльных пользователей, тестировщиков):
  - Планирование.
  - Обзорная встреча.
  - Подготовка.
  - Совещание.
  - Переработка.



- Завершающий этап.
- 3. Метод составления вариантов тестирования (выполняется тестировщиком). Варианты тестирования занимают промежуточную позицию между User Case и Test Case, помимо использования для тестирования требований в дальнейшем легко расширяются до Test Cases и составляют основу тестовой документации.

*Порядок выполнения работы*

1. Получить задание у преподавателя.
2. Протестировать спецификацию методом просмотра.
3. Оформить отчет и защитить лабораторную работу.

*Содержание отчета*

1. Цель работы.
2. Краткие теоретические сведения.
3. Отчет по тестированию спецификации.
4. Выводы по работе.

*Контрольные вопросы*

1. Как выглядит жизненный цикл проекта?
2. Какие выделяют критерии качества?
3. Какие требования считаются проверяемыми?
4. Какие требования считаются модифицируемыми?
5. Какие требования считаются корректными?
6. Какие требования считаются недвусмысленными?
7. Какие требования считаются полными?
8. Какие требования считаются непротиворечивыми?
9. Какие требования считаются упорядоченными по важности и стабильности?
10. Какие требования считаются трассируемыми?
11. Какие существуют методы тестирования требований?
12. Какие этапы включает в себя метод просмотра при тестировании требований?

#### Тема 1.4.

#### Классификация тестирования

*Цель работы:* изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

*Теоретические сведения*

Тестирование - процесс, направленный на оценку корректности, полноты и качества разработанного программного обеспечения.

Тестирование можно классифицировать по очень большому количеству признаков. Далее приведен обобщенный список видов тестирования по различным основаниям.

**Типы тестов по покрытию (по глубине)**

***Smoke test*** — тестирование системы для определения корректной работы базовых функций программы в целом, без углубления в детали. При проведении теста определяется пригодность сборки для дальнейшего тестирования.

***Minimal Acceptance Test (MAT, Positive test)***: тестирование системы или ее части

только на валидных данных (валидные данные - это данные, которые необходимо использовать для корректной работы модуля/функции). При тестировании проверяется правильная работа всех функций и модулей с валидными данными.

Для крупных и сложных приложений используется ограниченный набор сценариев и функций.

**Acceptance Test (AT):** полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях. Вид теста, направленный на подтверждение того, что приложение может использоваться по назначению при любых условиях.

Тест на этом уровне покрывает все возможные сценарии тестирования: проверку работоспособности модулей при вводе корректных значений; проверку при вводе некорректных значений; использование форматов данных отличных от тех, которые указаны в требованиях; проверку исключительных ситуаций, сообщений об ошибках; тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности; тестирование граничных значений интервалов; сценарии не предусмотренные спецификацией и т.д.

**Тестовые активности (типы тестов по покрытию (по ширине)):**

**Defect Validation** - проверка результата исправления дефектов. Включает в себя проверку на воспроизводимость дефектов, которые были исправлены в новой сборке продукта, а также проверку того, что исправление не повлияло на ранее работавшую функциональность

**New Feature Test (NFT, AT of NF)**- определение качества поставленной на тестирование новой функциональности, которая ранее не тестировалась. Данный тип тестирования включает в себя: проведение полного теста (AT) непосредственно новой функциональности; тестирование новой функциональности на соответствие документации; проверку всевозможных взаимодействий ранее реализованной функциональности с новыми модулями и функциями.

**Regression testing (регрессионное тестирование)** - проводится с целью оценки качества ранее реализованной функциональности. Включает в себя проверку стабильности ранее реализованной функциональности после внесения изменений, например добавления новой функциональности, исправление дефектов, оптимизация кода, разворачивание приложения на новом окружении. Регрессионное тестирование может быть проведено на уровне Smoke, MAT или AT.

**Типы тестов по знанию коду**

**Черный ящик** - тестирование системы, функциональное или нефункциональное, без знания внутренней структуры и компонентов системы. У тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним.

**Белый ящик** - тестирование основанное на анализе внутренней структуры компонентов или системы. У тестировщика есть доступ к внутренней структуре и коду приложения.

**Серый ящик** - комбинация методов белого и черного ящика, состоящая в том, что к части кода архитектуры у тестировщика есть, а к части кода - нет.

### **Типы тестов по степени автоматизации**

**Ручное** - тестирование, в котором тест-кейсы выполняются тестировщиком вручную без использования средств автоматизации.

**Автоматизированное** - набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тест-кейсы частично или полностью выполняет специальное инструментальное средство.

### **Типы тестов по изолированности компонентов**

**Unit/component (модульное)** - тестирование отдельных компонентов (модулей) программного обеспечения.

**Integration (интеграционное)** - тестируется взаимодействие между интегрированными компонентами или системами.

**System (системное)** - тестируется работоспособность системы в целом с целью проверки того, что она соответствует установленным требованиям.

### **Типы тестов по подготовленности.**

**Интуитивное тестирование** выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.

**Исследовательское тестирование** - метод проектирования тестовых сценариев во время выполнения этих сценариев. Тестировщик совершает проверки, продумывает их, придумывает новые проверки, часто использует для этого полученную информацию.

**Тестирование по документации** - тестирование по подготовленным тестовым сценариям, руководству по осуществлению тестов.

### **Типы тестов по месту и времени проведения**

**User Acceptance Testing (UAT) (приемочное тестирование)** - формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приемки и дать возможность пользователям, заказчикам или иным авторизованным лицам определить, принимать систему.

**Alpha Testing (альфа-тестирование)** - моделируемое или действительное функциональное тестирование, выполняется в организации, разрабатывающей продукт, но не проектной командой (это может быть независимая команда тестировщиков, потенциальные пользователи, заказчики). Альфа тестирование часто применяется к коробочному программному обеспечению в качестве внутреннего приемочного тестирования.

**Beta Testing (бета-тестирование)** - эксплуатационное тестирование потенциальными или существующими клиентами/заказчиками на внешней стороне (в среде, где продукт будет использоваться) никак связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы. Бета-тестирование часто проводится как форма внешнего приемочного тестирования готового программного обеспечения для того, чтобы получить отзывы рынка.

### **Типы тестов по объекту тестирования**

**Functional testing (функциональное тестирование)** - это тестирование, основанное на анализе спецификации, функциональности компонента или системы. Функциональным можно назвать любой вид тестирования, который согласно

требованиям проверяет правильную работу.

**Safety testing (тестирование безопасности)** - тестирование программного продукта с целью определить его безопасность (безопасность - способность программного продукта при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде.

**Security testing (тестирование защищенности)** - это тестирование с целью оценить защищенность программного продукта. Тестирование защищенности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение.

**Compatibility testing (тестирование совместимости)** - процесс тестирования для определения возможности взаимодействия программного продукта, проверка работоспособности приложения в различных средах (браузеры и их версии, операционные системы, их типа, версии и разрядность)

Виды тестов:

*J* кросс-браузерное тестирование (различные браузеры или версии браузеров)

*J* кросс-платформенное тестирование (различные операционные системы или версии операционных систем)

**Нефункциональное тестирование** - это проверка характеристик программы. Иначе говоря, когда проверяется не именно правильность работы, а какие-либо свойства (внешний вид и удобство пользования, скорость работы и т.п.).

#### **1. Тестирование пользовательского интерфейса (GUI)**

тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя.

*J* навигация

*J* цвета, графика, оформление

*J* содержание выводимой информации

*J* поведение курсора и горячие клавиши

*J* отображение различного количества данных (нет данных, минимальное и максимальное количество)

*J* изменение размеров окна или разрешения экрана

#### **2. Тестирование удобства использования (Usability Testing)** - тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях эксплуатации.

*J* визуальное оформление

*J* навигация

*J* логичность

#### **3. Тестирование доступности (Accessibility testing)** - тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.

#### **4. Тестирование интернационализации** - тестирование способности продукта работать в локализованных средах (способность изменять элементы интерфейса в зависимости от длины и направления текста, менять сортировки/форматы под различные локали и т.д.). (Максим Черняк).

Интернационализация - это процесс, упрощающий дальнейшую адаптацию

продукта к языковым и культурным особенностям региона, отличного от того, в котором разрабатывался продукт. Это адаптация продукта для потенциального использования практически в любом месте, Интернационализация производится на начальных этапах разработки, в то время как локализация — для каждого целевого языка.

**5. Тестирование локализации (Localization testing)** - тестирование, проводимое с целью проверить качество перевода продукта с одного языка на другой.

**6. Тестирование производительности или нагрузочное тестирование** - процесс тестирования с целью определения производительности программного продукта.

Виды тестов:

*J* нагрузочное тестирование (Performance and Load testing) - вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например количестве параллельных пользователей и/или операций, а также определения какую нагрузку может выдержать компонент или система;

*J* объемное тестирование (Volume testing) - позволяет получить оценку производительности при увеличении объемов данных в базе данных приложения;

*J* тестирование стабильности и надежности (Stability / Reliability testing) - позволяет проверять работоспособность приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

*Z* стрессовое тестирование (Stress testing) - вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

**7. Тестирование требований (Requirements testing)** - проверка требований на соответствие основным характеристикам качества.

**8. Тестирование прототипа (Prototype testing)** - метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.

**9. Тестирование установки (Installability testing) и лицензирования** - процесс тестирования устанавливаемости программного продукта.

Виды тестов:

*J* формальный тест программы установки приложения (проверка пользовательского интерфейса, навигации, удобства пользования, соответствия общепринятым стандартам оформления);

*J* функциональный тест программы установки;

*J* тестирование механизма лицензирования и функций защиты от пиратства;

*J* проверка стабильности приложения после установки.

**10. Тестирование на отказ и восстановление (Failover and Recovery Testing)** - тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении.

**Тестирование программного продукта включает следующие этапы:**

1. Изучение и анализ предмета тестирования.
2. Планирование тестирования.

### 3. Исполнение тестирования.

**Изучение и анализ предмета тестирования** начинается еще до утверждения спецификации и продолжается на стадии разработки (кодирования) программного обеспечения. Конечной целью этапа изучения и анализ предмета тестирования является получение ответов на два вопроса:

- какие функциональности предстоит протестировать,
- как эти функциональности работают.

**Планирование тестирования** происходит на стадии разработки (кодирования) программного обеспечения. На стадии планирования тестирования перед тестирующим стоит задача поиска компромисса между объемом тестирования, который возможен в теории, и объемом тестирования, который возможен на практике. На данной стадии необходимо ответить на вопрос: как будем тестировать? Результатом планирования тестирования является тестовая документация.

**Выполнение тестирования** происходит на стадии тестирования и представляет собой практический поиск дефектов с использованием тестовой документации, составленной ранее.

**Для всех программных продуктов выполняют следующие типы тестов и их композиции.**

**Для первого билда** рекомендуется проводить Smoke+AT готовой функциональности: поверхностное тестирование (Smoke Test) выполняется для определения пригодности сборки для дальнейшего тестирования; полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях (Acceptance Test, AT) позволяет обнаружить дефекты и внести запись о них в багтрекинг-систему.

**Для последующих билдов композиции тестов могут быть следующими:**

- Если не была добавлена новая функциональность, то: DV+MAT. Т.е., выполняется проверка исправления дефектов программистом (Defect Validation, DV), а также проверка работоспособности остальной функциональности после исправления дефектов на позитивных сценариях (Minimal Acceptance Test, MAT).
- Если была добавлена новая функциональность, то: Smoke+DV+NFT+Regression Test. В частности, выполняется поверхностное тестирование (Smoke Test), проверка исправления дефектов программистом (Defect Validation, DV), тестирование новых функциональностей (New Feature Testing, NFT), проверка старых функциональностей, т.е. регрессионное тестирование (Regression Test).

- Если была добавлена новая функциональность, то возможен также вариант: DV+NFT+Resression test, т.е. без выполнения Smoke Test.

В зависимости от типа и специфики приложения (web, desktop, mobile) выполняют специализированные тесты (например, кроссбраузерное или кроссплатформенное тестирование, тестирование локализации интернационализации и др.).

*Порядок выполнения работы*

1. Получить задание у преподавателя.
2. Выполнить генерацию тестов различных видов для конкретного объекта реального мира (пример приведен на рисунке 1).

### 3. Спланировать тестовые активности для следующих задач:

Поставлен на тестирование модуль 1, модуль 2, модуль 3.

Проведены исправления (fix) для заведенных дефектов, доставлена новая функциональность - модуль 4.

Заказчик решил расширять рынки сбыта и просит осуществить поддержку для Великобритании (кроме уже существующей Беларуси).

Заказчик хочет убедиться, что ПО держит нагрузку в 2000 пользователей.

Оформить

## Тема 1.5. Отчёты о тестировании

### Документирование результатов тестирования

*Цель работы:* составить итоговый отчет о результатах тестирования web-приложения.

#### Теоретические сведения

Итоговый отчет можно разделить на части с соответствующей информацией:

- Приветствие.
- Общая информация (Common Information).
- Тестовое окружение (Test Platform).
- Рекомендации QA (QA Recommendations).
- Детализированная информация (Detailed Information).
- Окончание содержимого.

#### Приветствие

Свое письмо с отчетом необходимо начать с приветствия всех адресатов.

Если по каким-либо причинам произошла задержка данных отчета, либо не весь запланированный функционал был проверен, то эту информацию необходимо предоставить в начале письма. Следует извиниться за задержку и указать адекватные причины произошедшего. Также в самом начале письма следует указывать, если были какие-то внешние факторы, препятствующие проверке какой-то части функционала.

Если во время тестирования не произошло никаких форс-мажорных обстоятельств, то достаточно обычного вежливого приветствия и далее уже переход к следующим пунктам.

#### Общая информация (Common Information)

В данной части отчета описывается, какие виды тестов проводились. Зачастую указываются модули, которые тестировались или функционал. Стоит удостовериться, не забыта ли какая-то часть функционала, особенно это актуально, когда нужно собрать итоговый отчет, соединив в себе данные о разных видах тестов и функционале.

#### Тестовое окружение (Test Platform)

Как правило, в этой части указываются:

- Название проекта.
- Номер сборки.
- Ссылка на проект (сборку). Необходимо убедиться, что зайдя по этой ссылке вы действительно попадаете на проект или можете установить приложение.

При указании данных в этой части отчета нужно быть очень внимательным, т.к. неправильная ссылка на сборку или неверный номер сборки не дают достоверной информации всем заинтересованным людям, а также затрудняют работу человеку,

собирающему финальный отчет.

### **Рекомендации QA (QA Recommendations)**

Данная часть отчета является наиболее важной, т.к. здесь отражается общее состояние сборки. Здесь показывается аналитическая работа тестировщика, его рекомендации по улучшению функционала, наиболее слабые места и наиболее критичные дефекты, динамика изменения качества проекта.

В этом разделе должна быть информация о следующем:

- Указан функционал (часть функционала), который заблокирован для проверки. Даны пояснения почему этот функционал не проверен (указаны наиболее критичные дефекты).
- Произведен анализ качества проверенного функционала. Следует указать, улучшилось оно или ухудшилось по сравнению с предыдущей версией, какое качество на сегодняшний момент, какие факторы повлияли на выставление именно такого качества сборки.
- Если качество сборки ухудшилось, то обязательно должны быть указаны регрессионные места.
- Наиболее нестабильные части функционала следует выделить и указать причину, по которой они таковыми являются.
- Даны рекомендации по тому функционалу и дефектам, скорейшее исправление которых является наиболее приоритетным.
- Список наиболее критичных для сборки дефектов, с указанием названия и их критичности.
- Для отчета уровня Smoke обязательно указать весь нестабильный функционал.

Если сборка является релизной или предрелизной, то любое ухудшение качества является критичным и важно об этом сообщить менеджеру как можно раньше. Помимо всего вышеуказанного для релизных и предрелизных сборок в отчете о качестве продукта важно указывать следующее:

- Дана информация о всех проблемах, характерных сборке. Проведен анализ, насколько оставшиеся проблемы являются критичными для конечного пользователя.
- Указаны дефекты, которые следует исправить, чтобы качество конечной сборки было выше.

### **Детализированная информация (Detailed Information)**

В данной части отчета описывается более подробная информация о проверенных частях функционала, устанавливается качество каждой проверенной части функционала(модуля) в отдельности. В зависимости от типа проводимых тестов, эта часть отчета будет отличаться.

#### ***Smoke***

При оценке качества функционала на уровне Smoke теста, оно может быть либо Приемлемым, либо Неприемлемым. Качество сборки зависит от нескольких факторов:

- Если это релизная или предрелизная сборка, то для выставления Приемлемого качества на уровне Smoke не должно быть найдено функциональных дефектов.
- Наличие нового функционала. Новый функционал, который впервые поставляется на тестирование, не должен содержать дефектов уровня Smoke для выставления Приемлемого качества всей сборки.
- Чтобы установить сборке Приемлемое качество, не должно быть дефектов уровня Smoke у того функционала, по которому планируется проводить полные тесты.



- Все наиболее важные части функционала обрабатывают корректно, тогда качество всего функционала на уровне Smoke может быть оценено, как Приемлемое.

В части о детализированной информации качества сборки следует более подробно описать проблемы, которые были найдены во время теста.

### **DV**

В этой части отчета указывается качество о проведении валидации дефектов.

Здесь должна быть следующая информация:

- Общее количество всех дефектов, поступивших на проверку.
- Количество неисправленных дефектов и их процент от общего количества.
- Список дефектов, которые не были проверены и причины, по которым этого не было сделано.
- Наглядная таблица с неисправленными дефектами.

По вышеуказанным результатам выставляется качество теста. Если процент неисправленных дефектов < 10%, то качество Приемлемое, если > 10%, то качество Неприемлемое.

### **NFT**

При проведении полного теста нового функционала качество отдельно проверенного функционала может быть: Высокое, Среднее, Низкое.

В отчете следует отдельно указывать информацию о качестве каждой части нового функционала. В этой части отчета должна быть следующая информация:

- Дана общая оценка реализации нового функционала (сгруппированная по качеству).
- Подробная (детальная) информация о качестве каждой из частей новой функциональности.
- Проведен анализ каждой из новых функций в отдельности.
- Даны ясные пояснения о выставлении соответствующего качества.
- Даны рекомендации по улучшению качества (какие проблемы следует исправить).
- Показана таблица с новыми функциями (название), их качеством, статусом функции из CQ.

### **AT, MAT, Regression**

Если проводились тесты указанных уровней, то в первую очередь при написании отчета нужно анализировать динамику изменения качества проверенной функциональности в сравнении с более ранними версиями сборки.

Также как и у предыдущего вида тестов, качество этих может быть: Высокое, Среднее, Низкое.

Для указанных видов тестов в данной части отчета должна быть описана информация следующего характера:

- Дана сравнительная характеристика каждой из частей функционала в сравнении с предыдущими версиями сборки.
- Подробная (детальная) информация о качестве каждой из частей проверенной функциональности.
- Даны ясные пояснения о выставлении соответствующего качества каждой функции в отдельности.
- Даны рекомендации по улучшению качества (какие проблемы следует

исправить).

### **Окончание содержимого**

В завершении содержимое отчета должно включать в себя информацию следующего характера:

- Ссылка на тест-план.
- Ссылка на документ feature matrix (если таковой имеется).
- Ссылка на документ со статистикой (если таковой имеется).
- Общее количество всех новых дефектов.
- Подпись высылающего отчет.

Данные ссылки должны быть корректными, необходимо проверить достоверную ли информацию получает пользователь, открывший ссылку. Следует обращать особое внимание на подпись, удостоверьтесь, что указана именно ваша подпись либо какая-то универсальная для определенного проекта подпись.

### **Порядок выполнения работы**

1. Получить задание у преподавателя.
2. Составить итоговый отчет по результатам тестирования web- приложения.
3. Оформить отчет и защитить лабораторную работу.

### **Содержание отчета**

1. Цель работы.
2. Краткие теоретические сведения.
3. Итоговый отчет о результатах тестирования web-приложения.
4. Выводы по работе.

### **Контрольные вопросы**

1. Какая структура итогового отчета о результатах тестирования?
2. Что содержится в разделе Приветствие?
3. Что содержится в разделе Общая информация?
4. Что содержится в разделе Тестовое окружение?
5. Что содержится в разделе Рекомендации QA?
6. Что содержится в разделе Детализированная информация?
7. Что содержится в разделе Окончание содержимого?

## **Тема 1.6. Методы отладки и тестирования**

### **Документирование результатов тестирования**

*Цель работы:* составить итоговый отчет о результатах тестирования web-приложения.

### **Теоретические сведения**

Итоговый отчет можно разделить на части с соответствующей информацией:

- Приветствие.
- Общая информация (Common Information).
- Тестовое окружение (Test Platform).
- Рекомендации QA (QA Recommendations).
- Детализированная информация (Detailed Information).
- Окончание содержимого.

## **Приветствие**

Свое письмо с отчетом необходимо начать с приветствия всех адресатов.

Если по каким-либо причинам произошла задержка данных отчета, либо не весь запланированный функционал был проверен, то эту информацию необходимо предоставить в начале письма. Следует извиниться за задержку и указать адекватные причины произошедшего. Также в самом начале письма следует указывать, если были какие-то внешние факторы, препятствующие проверке какой-то части функционала.

Если во время тестирования не произошло никаких форс-мажорных обстоятельств, то достаточно обычного вежливого приветствия и далее уже переход к следующим пунктам.

## **Общая информация (Common Information)**

В данной части отчета описывается, какие виды тестов проводились. Зачастую указываются модули, которые тестировались или функционал. Стоит удостовериться, не забыта ли какая-то часть функционала, особенно это актуально, когда нужно собрать итоговый отчет, соединив в себе данные о разных видах тестов и функционале.

## **Тестовое окружение (Test Platform)**

Как правило, в этой части указываются:

- Название проекта.
- Номер сборки.
- Ссылка на проект (сборку). Необходимо убедиться, что зайдя по этой ссылке вы действительно попадаете на проект или можете установить приложение.

При указании данных в этой части отчета нужно быть очень внимательным, т.к. неправильная ссылка на сборку или неверный номер сборки не дают достоверной информации всем заинтересованным людям, а также затрудняют работу человеку, собирающему финальный отчет.

## **Рекомендации QA (QA Recommendations)**

Данная часть отчета является наиболее важной, т.к. здесь отражается общее состояние сборки. Здесь показывается аналитическая работа тестировщика, его рекомендации по улучшению функционала, наиболее слабые места и наиболее критичные дефекты, динамика изменения качества проекта.

В этом разделе должна быть информация о следующем:

- Указан функционал (часть функционала), который заблокирован для проверки. Даны пояснения почему этот функционал не проверен (указаны наиболее критичные дефекты).
- Произведен анализ качества проверенного функционала. Следует указать, улучшилось оно или ухудшилось по сравнению с предыдущей версией, какое качество на сегодняшний момент, какие факторы повлияли на выставление именно такого качества сборки.
- Если качество сборки ухудшилось, то обязательно должны быть указаны регрессионные места.
- Наиболее нестабильные части функционала следует выделить и указать причину, по которой они таковыми являются.
- Даны рекомендации по тому функционалу и дефектам, скорейшее исправление которых является наиболее приоритетным.
- Список наиболее критичных для сборки дефектов, с указанием названия и их критичности.
- Для отчета уровня Smoke обязательно указать весь нестабильный функционал.

Если сборка является релизной или предрелизной, то любое ухудшение качества является критичным и важно об этом сообщить менеджеру как можно раньше. Помимо всего вышеуказанного для релизных и предрелизных сборок в отчете о качестве продукта важно указывать следующее:

- Дана информация о всех проблемах, характерных сборке. Проведен анализ, насколько оставшиеся проблемы являются критичными для конечного пользователя.
- Указаны дефекты, которые следует исправить, чтобы качество конечной сборки было выше.

#### **Детализированная информация (Detailed Information)**

В данной части отчета описывается более подробная информация о проверенных частях функционала, устанавливается качество каждой проверенной части функционала(модуля) в отдельности. В зависимости от типа проводимых тестов, эта часть отчета будет отличаться.

#### ***Smoke***

При оценке качества функционала на уровне Smoke теста, оно может быть либо Приемлемым, либо Неприемлемым. Качество сборки зависит от нескольких факторов:

- Если это релизная или предрелизная сборка, то для выставления Приемлемого качества на уровне Smoke не должно быть найдено функциональных дефектов.
- Наличие нового функционала. Новый функционал, который впервые поставляется на тестирование, не должен содержать дефектов уровня Smoke для выставления Приемлемого качества всей сборки.
- Чтобы установить сборке Приемлемое качество, не должно быть дефектов уровня Smoke у того функционала, по которому планируется проводить полные тесты.
- Все наиболее важные части функционала отработывают корректно, тогда качество всего функционала на уровне Smoke может быть оценено, как Приемлемое.

В части о детализированной информации качества сборки следует более подробно описать проблемы, которые были найдены во время теста.

#### **DV**

В этой части отчета указывается качество о проведении валидации дефектов.

Здесь должна быть следующая информация:

- Общее количество всех дефектов, поступивших на проверку.
- Количество неисправленных дефектов и их процент от общего количества.
- Список дефектов, которые не были проверены и причины, по которым этого не было сделано.
- Наглядная таблица с неисправленными дефектами.

По вышеуказанным результатам выставляется качество теста. Если процент неисправленных дефектов < 10%, то качество Приемлемое, если > 10%, то качество Неприемлемое.

#### **NFT**

При проведении полного теста нового функционала качество отдельно проверенного функционала может быть: Высокое, Среднее, Низкое.

В отчете следует отдельно указывать информацию о качестве каждой части нового функционала. В этой части отчета должна быть следующая информация:

- Дана общая оценка реализации нового функционала (сгруппированная по

качеству).

- Подробная (детальная) информация о качестве каждой из частей новой функциональности.
- Проведен анализ каждой из новых функций в отдельности.
- Даны ясные пояснения о выставлении соответствующего качества.
- Даны рекомендации по улучшению качества (какие проблемы следует исправить).
- Показана таблица с новыми функциями (название), их качеством, статусом функции из CQ.

### ***AT, MAT, Regression***

Если проводились тесты указанных уровней, то в первую очередь при написании отчета нужно анализировать динамику изменения качества проверенной функциональности в сравнении с более ранними версиями сборки.

Также как и у предыдущего вида тестов, качество этих может быть: Высокое, Среднее, Низкое.

Для указанных видов тестов в данной части отчета должна быть описана информация следующего характера:

- Дана сравнительная характеристика каждой из частей функционала в сравнении с предыдущими версиями сборки.
- Подробная (детальная) информация о качестве каждой из частей проверенной функциональности.
- Даны ясные пояснения о выставлении соответствующего качества каждой функции в отдельности.
- Даны рекомендации по улучшению качества (какие проблемы следует исправить).

### **Окончание содержимого**

В завершении содержимое отчета должно включать в себя информацию следующего характера:

- Ссылка на тест-план.
- Ссылка на документ feature matrix (если таковой имеется).
- Ссылка на документ со статистикой (если таковой имеется).
- Общее количество всех новых дефектов.

## Тема 1.7.

Автоматизи-рованное тестирование Тема 1.7.

- Автоматизи-рованное тестирование Подпись высылающего отчет.

Данные ссылки должны быть корректными, необходимо проверить достоверную ли информацию получает пользователь, открывший ссылку. Следует обращать особое внимание на подпись, удостоверьтесь, что указана именно ваша подпись либо какая-то универсальная для определенного проекта подпись.

### **Порядок выполнения работы**

4. Получить задание у преподавателя.
5. Составить итоговый отчет по результатам тестирования web- приложения.
6. Оформить отчет и защитить лабораторную работу.

## Содержание отчета

5. Цель работы.
6. Краткие теоретические сведения.
7. Итоговый отчет о результатах тестирования web-приложения.
8. Выводы по работе.

### Контрольные вопросы

8. Какая структура итогового отчета о результатах тестирования?
9. Что содержится в разделе Приветствие?
10. Что содержится в разделе Общая информация?
11. Что содержится в разделе Тестовое окружение?
12. Что содержится в разделе Рекомендации QA?
13. Что содержится в разделе Детализированная информация?
14. Что содержится в разделе Окончание содержимого?

## Тема 1.7. Автоматизированное тестирование

Ответить на вопросы (с примерными ответами)

### 1. Что такое автоматизированное тестирование?

Автоматизированное тестирование или автоматизация тестирования – это метод тестирования программного обеспечения, который выполняется с использованием специальных программных средств, которые, в свою очередь необходимы для выполнения набора тестовых примеров. Напротив, ручное тестирование выполняется человеком, сидящим перед компьютером и тщательно выполняющим каждый шаг теста «руками».

Программное обеспечение для автоматизации тестирования также может вводить тестовые данные в тестовую среду, сравнивать ожидаемые и фактические результаты и создавать подробные отчеты о тестах. Как правило, автоматизация тестирования требует значительных вложений денег и ресурсов.

### 2. Зачем нужна автоматизация?

Это хороший способ повысить эффективность, а также увеличить охват и скорость тестирования программного обеспечения, когда вам нужно повторять одни и те же тестовые сценарии.

- Ручное тестирование всех рабочих процессов, полей и негативных сценариев требует больше времени и денег (при определенных условиях).
- Сложно тестировать многоязычные сайты вручную.
- Не требует вмешательства человека. Запускаете и переходите к другим задачам.
- Увеличивает скорость выполнения тестов.
- Помогает увеличить охват тестированием.
- Ручное тестирование может наскучить, и следствиями станут потеря вовлеченности и появление ошибок.

### 3. Какие тестовые случаи стоит автоматизировать?

Для увеличения рентабельности инвестиций в автоматизацию тестовые случаи для автоматизации можно выбрать по следующим критериям:

- Высокие риски и сбои недопустимы – крайне актуально для банковской сферы.
- Тестовые сценарии, которые регулярно повторяются.
- Тестовые сценарии, которые очень сложны и утомительны для выполнения вручную.
- Тестовые примеры, отнимающие много времени.

Следующая категория тестовых случаев не подходит для автоматизации:

- Новые тестовые примеры, которые не выполнялись вручную хотя бы один раз.
- Сценарии тестирования, требования к которым часто меняются.
- Тестовые примеры, которые выполняются на разовой основе.

#### **4. Процесс автоматизированного тестирования:**

В процессе автоматизации выполняются следующие шаги:

- А) Выбор тестового инструмента**
- Б) Определение объема автоматизации**
- В) Планирование, дизайн и разработка**
- Г) Выполнение теста**
- Д) Техническое обслуживание**

#### **5. Выбор инструмента тестирования.**

Выбор средства тестирования во многом зависит от технологии, на которой построено тестируемое приложение. Например, QTP не поддерживает Informatica. Таким образом, QTP нельзя использовать для тестирования приложений Informatica. Хорошая идея – провести Proof of Concept of Tool (демонстрация практической осуществимости) на AUT.

**Определяем объем автоматизации** Объем автоматизации – это область тестируемого приложения, которая будет автоматизирована. Его помогают определить следующие пункты:

- Функции, важные для бизнеса
- Сценарии с большим объемом данных
- Общие функции приложений
- Техническая осуществимость
- Частота повторного использования бизнес-компонентов
- Сложность тестовых случаев
- Возможность использовать одни и те же тестовые сценарии для кросс-браузерного тестирования

**6. Планирование, проектирование и разработка.** На этом этапе вы создаете стратегию и план автоматизации, которые содержат следующие детали:

- Выбранные инструменты автоматизации
- Конструкция каркаса и его особенности
- Входящие и выходящие за рамки элементы автоматизации
- Подготовка стендов автоматизации
- График и временная шкала сценариев и выполнения
- Результаты тестирования автоматизации

## **7.Выполнение теста.**

На этом этапе выполняются сценарии автоматизации. Сценариям необходимо ввести тестовые данные, прежде чем они будут запущены. После выполнения они предоставляют подробные отчеты об испытаниях

Выполнение может быть выполнено с использованием инструмента автоматизации напрямую или с помощью инструмента управления тестированием, который вызовет инструмент автоматизации.

Пример: Центр качества – это инструмент управления тестированием, который, в свою очередь, вызывает QTP для выполнения сценариев автоматизации. Скрипты могут выполняться на одной машине или на группе машин. Для экономии времени тестирование можно проводить ночью.

## **8.Обслуживание автоматизированного тестирования**

Этот этап автоматизированного тестирования проводится для проверки того, как работают новые функции, добавленные в программное обеспечение: нормально или нет. Сопровождение в автотестировании выполняется, когда добавляются новые сценарии автоматизации, и их необходимо проверять и поддерживать, чтобы повышать эффективность сценариев автоматизации с каждым последующим циклом выпуска.

## **9.Платформа для автоматизации**

Фреймворк – это набор руководств по автоматизации, которые:

- поддерживают последовательность тестирования;
- улучшают структурирование теста;
- позволяют использовать минимальное количество кода;
- уменьшают затраты на обслуживания кода;
- повышают удобство повторного использования;
- дают возможность нетехническим тестировщикам участвовать в кодировании тестов;
- помогают сократить срок обучения использованию инструмента;
- включают данные везде, где это необходимо.

Для автоматизации тестирования программного обеспечения используют четыре типа фреймворков:

1. платформа автоматизации на основе данных;
2. фреймворк автоматизации на основе ключевых слов;
3. модульная платформа автоматизации;
4. гибридная среда автоматизации.

## **10.Рекомендации для эффективной автоматизации тестирования**

Чтобы получить максимальную рентабельность инвестиций в автоматизацию, соблюдайте следующие правила:

- Объем автоматизации необходимо детально определить до начала проекта. Это позволит убедиться, что ожидания от автоматизации будут оправданы.
- Определите правильный инструмент автоматизации: инструмент не должен выбираться на основании его популярности, он должен соответствовать требованиям автоматизации на конкретном проекте.
- Выберите подходящий фреймворк.
- Стандарты создания сценариев. При написании сценариев для автоматизации необходимо соблюдать стандарты. Вот некоторые из них:



- создайте единые скрипты, комментарии и отступы кода;
- разработайте правила наименования тестовых сценариев;
- прикладывайте необходимые документы, если, например, сложно понять прохождение тестового сценария без скриншота и/или спецификации.
- Определите метрики и следите за ними. Успех автоматизации нельзя определить лишь путем сравнения затраченных усилий, на тот или иной вид тестирования. Вот основные показатели:
  - процент обнаруженных дефектов;
  - время, необходимое для тестирования автоматизации выпуска каждого нового цикла;
  - минимальное время требуемое для выпуска;
  - индекс удовлетворенности клиентов;
  - улучшение производительности.

Приведенные выше рекомендации, если их соблюдать, позволят качественно выполнить автоматизацию тестирования.

### 11. Преимущества автоматизации тестирования

- На 70% быстрее, чем при ручном тестировании.
- Более широкий тестовый охват функций приложения.
- Надежные в результаты.
- Обеспечивает согласованность тестовых моделей.
- Экономит время и деньги.
- Повышает точность.
- Позволяет исполнять процесс тестирования без вмешательства человека.
- Повышает эффективность .
- увеличивает скорость исполнения тестирования.
- Повторно использует тестовые скрипты.
- Позволяет тестировать часто и тщательно.
- Большой цикл выполнения может быть достигнут за счет автоматизации.
- Сокращает время выхода продукта на рынок

### 12. Типы автоматизированного тестирования

- Смоук тестирование
- [Модульное тестирование](#)
- [Интеграционное тестирование](#)
- Функциональное тестирование
- Проверка ключевых слов
- Регрессионное тестирование
- Тестирование на основе данных
- Тестирование черного ящика

### 13. Как выбрать инструмент автоматизации?

Выбор подходящего инструмента может оказаться сложной задачей. Следующие критерии помогут вам выбрать лучший инструмент для ваших требований:

- поддержка окружающей среды;
- легкость использования;

- тестирование базы данных;
- идентификация объекта;
- тестирование изображений;
- тестирование восстановления после ошибок;
- отображение объектов;
- используемый язык сценариев;
- поддержка различных типов тестирования, в том числе функционального, тестового управления, мобильного и т. д.;
- поддержка нескольких фреймворков тестирования;
- легко отлаживать сценарии программного обеспечения автоматизации;
- умение распознавать предметы в любой среде;
- обширные отчеты об испытаниях и их результаты;
- минимизация затрат на обучение выбранным инструментам.

Выбор инструмента – одна из самых серьезных проблем, которую необходимо решить, прежде чем приступить непосредственно к автоматизации. Во-первых, определите требования, изучите различные инструменты и их возможности, установите ожидания от инструмента и сделайте Proof Of Concept.

#### **14. Инструменты автоматизации тестирования**

На рынке доступно множество инструментов для функционального и регрессионного тестирования. В следующих выпусках расскажем в пользу чего делают выбор наши практикующие специалисты.

##### **Ranorex Studio**

Это универсальный инструмент для автоматизации функциональных тестов пользовательского интерфейса, регрессионных тестов, тестов на основе данных и многого другого. Ranorex Studio включает простой в использовании интерфейс для автоматизации тестирования веб-приложений, настольных и мобильных приложений.

##### **Особенности:**

- Функциональный пользовательский интерфейс и сквозное тестирование на ПК, в Интернете и на мобильных устройствах
- Кроссбраузерное тестирование
- SAP, ERP, Delphi и унаследованные приложения.
- iOS и Android
- Запускайте тесты локально или удаленно, параллельно или распределяйте в Selenium Grid
- Надежная отчетность

##### **Testim**

«Самый быстрый путь к отказоустойчивым сквозным тестам – без кода, с кодированием или и тем, и другим. Testim позволяет создавать удивительно стабильные тесты без кода, которые используют наш ИИ, а также гибкость для экспорта тестов в виде кода. Такие клиенты, как Microsoft, NetApp, Wix и JFrog, ежемесячно проводят миллионы тестов на Testim. **Особенности**

- Вы можете использовать современный JavaScript API от Testim и свою IDE для отладки, настройки или рефакторинга тестов.
- Храните тесты в своей системе управления версиями, чтобы синхронизировать их с ветвями и запускать тесты при каждой фиксации.
- Интеграция с популярными инструментами»

## **21 Labs**

«Это сложная самообучающаяся платформа автоматизации тестирования и аналитики для приложений iOS и Android.

### **Особенности:**

- Быстрая и интеллектуальная разработка – создание с помощью ИИ дает пользователям возможность создавать автоматизированные функциональные тесты и тесты пользовательского интерфейса за считанные минуты.
- Результаты, которым вы доверяете – бесшовная система алгоритмических локаторов обеспечивает стабильные результаты во всех средах.
- Устранение проблем с обслуживанием и нестабильных результатов – самообучающееся обслуживание автоматически обновляет тесты и гарантирует, что ваша команда может сосредоточиться на разработке новых функций, полагаясь на результаты тестов.
- Выпускайте с уверенностью – производственная интеграция закрывает цикл обратной связи и анализирует фактическое покрытие. Используйте данные при выпуске.
- Полностью SaaS, не требует установки или устройств для создания или выполнения тестов. Предлагает беспрепятственный доступ к десяткам устройств».

## **Selenium**

Это инструмент тестирования программного обеспечения, используемый для регрессионного тестирования. Это инструмент тестирования с открытым исходным кодом, который предоставляет возможность воспроизведения и записи для регрессионного тестирования. Селен IDE поддерживает только Mozilla Firefox веб – браузер

### **Особенности:**

- Он обеспечивает возможность экспорта записанного скрипта на других языках, таких как Java, Ruby, RSpec, Python, C# и т. д.
- Его можно использовать с такими фреймворками, как JUnit и TestNG.
- Он может выполнять несколько тестов одновременно Автозаполнение для общих команд Selenium
- Пошаговые тесты
- Идентифицирует элемент с помощью идентификатора, имени, X-пути и т. Д. Храните тесты как Ruby Script, HTML и любой другой формат
- Он предоставляет возможность утверждать заголовок для каждой страницы
- Он поддерживает файл selenium user-extensions.js
- Это позволяет вставлять комментарии в середину скрипта для лучшего понимания и отладки.

## **QTP (MicroFocus UFT)**

Широко используется для функционального и регрессионного тестирования, он касается всех основных программных приложений и сред. Чтобы упростить создание и обслуживание тестов, в нем используется концепция тестирования, управляемого ключевыми словами. Это позволяет тестировщику создавать тестовые примеры прямо из приложения.

### **Особенности:**

- Нетехническому человеку проще адаптироваться и создавать рабочие тестовые примеры.
- Он быстрее устраняет дефекты, тщательно документируя и воспроизводя дефекты для разработчика.
- Сверните создание тестов и документацию по тестам на одном сайте
- Параметризация проще, чем в WinRunner
- QTP поддерживает среду разработки .NET
- У него лучший механизм идентификации объекта
- Он может улучшить существующие сценарии QTP без доступности «Тестируемого приложения», используя активный экран.

### **Rational Functional Tester**

Это объектно-ориентированный инструмент автоматизированного функционального тестирования, способный выполнять автоматическое функциональное, регрессионное тестирование, тестирование на основе данных и тестирование графического интерфейса. Основные особенности этого инструмента:

#### **Особенности:**

- Поддерживает широкий спектр протоколов и приложений, таких как Java, HTML, NET, Windows, SAP, Visual Basic и т. д.
- Может записывать и воспроизводить действия по запросу
- Он хорошо интегрируется с инструментами управления исходным кодом, такими как Rational Clear Case и Rational Team Concert. Он позволяет разработчикам создавать скрипт, связанный с ключевыми словами, чтобы его можно было использовать повторно. Редактор Eclipse Java Developer Toolkit
- Помогает команде кодировать тестовые сценарии на Java с помощью Eclipse.
- Поддерживает настраиваемые элементы управления через прокси SDK (Java / .Net)
- Поддерживает управление версиями, чтобы обеспечить параллельную разработку тестовых сценариев и одновременное использование географически распределенной командой.

### **Watir**

Это программное обеспечение с открытым исходным кодом для регрессионного тестирования. Это позволяет вам писать тесты, которые легко читать и поддерживать. Watir поддерживает только Internet Explorer в Windows, а веб-драйвер Watir поддерживает Chrome, Firefox, IE, Opera и т. д.

#### **Особенности:**

- Он поддерживает несколько браузеров на разных платформах.
- Вместо того, чтобы использовать собственный сценарий поставщика, он использует полнофункциональный современный язык сценариев Ruby.
- Он поддерживает ваше веб-приложение независимо от того, на чем оно разработано.

### **SilkTest**

Silk Test предназначен для выполнения функционального и регрессионного тестирования. Для приложений электронного бизнеса шелковый тест является ведущим продуктом для функционального тестирования. Это продукт поглощения Segue Software компанией Borland в 2006 году. Это объектно-ориентированный язык, как и C ++. Он

использует концепцию объекта, классов и наследования. Его основная особенность включает

**Особенности:**

- Он состоит из всех файлов исходных скриптов.
- Он преобразует команды сценария в команды графического интерфейса. На одном компьютере команды могут выполняться на удаленном или хост-компьютере.
- Чтобы идентифицировать движение мыши вместе с нажатиями клавиш, можно запустить Silktest. Он может использовать как методы воспроизведения и записи, так и методы описательного программирования для получения диалогов.
- Он определяет все элементы управления и окна тестируемого приложения как объекты и определяет все атрибуты и свойства каждого окна.

## **Раздел 2. Документирование**

Тема 2.1. Документирование программного обеспечения в соответствии с ЕСПД

### **Проверочная работа по теме 2.1.**

#### **Единая система программной документации (ЕСПД)**

**Цель работы:** Изучение требований к оформлению программной документации.

**План работы:**

1 Изучение ГОСТов, входящих в ЕСПД (Единую систему программной документации); Изучение основополагающих стандартов: Гост 19.101-77 «ЕСПД. Виды программ и программных документов»; Гост 19.103-77 «ЕСПД. Обозначение программ и программных документов»; Гост 19.104-78 СТ СЭВ 2088-80) «ЕСПД. Основные надписи». Гост 19.106-78 СТ СЭВ 2088-80) «ЕСПД. Требования к программным документам, выполненным печатным способом».

2 Изучение ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению»

3 Знакомство с ГОСТ ИСО/МЭК 12207-95

4 Изучение ГОСТ (СТ СЭВ) 19.101-77 (1626-79). ЕСПД. Виды программ и программных документов.

5 Изучение ГОСТ 19.102-77. ЕСПД. Стадии разработки.

6 Изучение ГОСТ 19.781-90 Обеспечение систем обработки информации программное. Термины и определения.

7 Изучение **ГОСТ Р ИСО/МЭК 9294-93** Информационная технология. Руководство по управлению документированием программного обеспечения.

8 Изучить ГОСТ Р ИСО 9127-94 Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов.

**Содержание отчета:**

В рабочей тетради должны быть:

- 1 Наименование и цель работы.
- 2 Порядок выполнения работы.
- 3 Ответы на контрольные вопросы.

Основу отечественной нормативной базы в области документирования программ составляет комплекс стандартов Единой системы программной документации (ЕСПД). Стандарты ЕСПД упорядочивают процесс документирования программных систем.

Стандарты ЕСПД охватывают ту часть документации, которая создается в процессе разработки программы, и связаны с документированием функциональных характеристик.

Кроме ЕСПД в официальной нормативной базе РФ в области документирования программ и в смежных областях есть ряд перспективных стандартов, например:

- международный стандарт **ISO/IEC 12207: 1995-08-01** на организацию жизненного цикла продуктов программного обеспечения;
- стандарты комплекса ГОСТ 34 на создание и развитие автоматизированных систем.

**Стандарты ЕСПД подразделяют на группы, приведенные в таблице:**

Код группы	Наименование группы
	Общие положения
	Основополагающие стандарты
	Правила выполнения документации разработки
	Правила выполнения документации изготовления
	Правила выполнения документации сопровождения
	Правила выполнения эксплуатационной документации
	Правила обращения программной документации
	Резервные группы
	Прочие стандарты

Обозначение стандарта ЕСПД строят по классификационному признаку:

Обозначение стандарта ЕСПД должно состоять из:

1. числа 19 (присвоенных классу стандартов ЕСПД);
2. одной цифры (после точки), обозначающей код классификационной группы стандартов, указанной в таблице;
3. двузначного числа (после тире), указывающего год регистрации стандарта.
4. ОСТ 19.001-77. ЕСПД. Общие положения.
5. ГОСТ 19.003-80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические.
6. ГОСТ 19.005-85. ЕСПД. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.
7. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов.
8. ГОСТ 19.102-77. ЕСПД. Стадии разработки.
9. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов.
10. ГОСТ 19.104-78. ЕСПД. Основные надписи.
11. ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам.
12. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом.

13. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.
14. ГОСТ 19.202-78. ЕСПД. Спецификация. Требования к содержанию и оформлению.
15. ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению.
16. ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению.
17. ГОСТ 19.402-78. ЕСПД. Описание программы.
18. ГОСТ 19.403-79. ЕСПД. Ведомость держателей подлинников.
19. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
20. ГОСТ 19.501-78. ЕСПД. Формуляр. Требования к содержанию и оформлению.
21. ГОСТ 19.502-78. ЕСПД. Описание применения. Требования к содержанию и оформлению.
22. ГОСТ 19.503-79. ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
23. ГОСТ 19.504-79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению.
24. ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению.
25. ГОСТ 19.506-79. ЕСПД. Описание языка. Требования к содержанию и оформлению.
26. ГОСТ 19.507-79. ЕСПД. Ведомость эксплуатационных документов.
27. ГОСТ 19.508-79. ЕСПД. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.
28. ГОСТ 19.601-78. ЕСПД. Общие правила дублирования, учета и хранения.
29. ГОСТ 19.602-78. ЕСПД. Правила дублирования, учета и хранения программных документов, выполненных печатным способом.
30. ГОСТ 19.603-78. ЕСПД. Общие правила внесения изменений.
31. ГОСТ 19.604-78. ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом.
32. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
33. ГОСТ 19.781-90. Обеспечение систем обработки информации программное.

**Контрольные вопросы:**

1 Какие документы относят к программным? Какие сведения они должны содержать в соответствии с ГОСТ 19.103-77 «Обозначение программ и программных документов»?

2 Что представляет собой состав и структура программного документа в соответствии с ГОСТ 19.105-78?

3 В какой последовательности располагают материалы программного документа, выполненным печатным способом, согласно ГОСТ 19.106-78\* (СТ СЭВ 2088-80)?

4 Что устанавливает ГОСТ 19.104-78\*(СТ СЭВ 2088-80) «ОСНОВНЫЕ НАДПИСИ»?

5 Какие структурные данные входят в состав основных надписей листа утверждения и титульного листа в программных документах в соответствии с ГОСТ 19.104-78\* (СТ СЭВ 2088-80)?

6 Что устанавливает ГОСТ (СТ СЭВ) 19.101-77 (1626-79). ЕСПД. Виды программ и программных документов)?

#### **ЗАДАНИЕ 2**

Изучить **ГОСТ 19.102-77. ЕСПД. Стадии разработки.**

Из **ГОСТ 19.102-77 законспектировать Стадии разработки программы, этапы и содержание работ.**

Первым стандартом, который можно использовать при формировании технических заданий на программирование является **ГОСТ (СТ СЭВ) 19.201-78 (1626-79). ЕСПД. (Техническое задание. Требование к содержанию и оформлению).**

Ответить на вопрос: Какие разделы должно содержать Техническое задание согласно ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению»?

Вторым стандартом является **ГОСТ (СТ СЭВ) 19.101-77 (1626-79). ЕСПД. Виды программ и программных документов).** Изучить этот стандарт и ответить на вопрос:

Что устанавливает этот стандарт для вычислительных машин, комплексов и систем независимо от их назначения и области применения?

#### **ЗАДАНИЕ 3**

Изучить **ГОСТ Р ИСО/МЭК 9294-93** Информационная технология. Руководство по управлению документированием программного обеспечения.

Ответить на вопрос: Что является Целью стандарта ГОСТ Р ИСО/МЭК 9294-93 и что он устанавливает?

Изучить **ГОСТ Р ИСО/МЭК 8631-94** Информационная технология. Программные конструктивы и условные обозначения для их представления.

Ответить на вопрос: Что этот ГОСТ описывает?

#### **ЗАДАНИЕ 4**

Изучить **ГОСТ Р ИСО 9127-94** Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов. **Ответить на вопрос:** Что представляет собой информация на упаковке для потребительских программных пакетов?

#### **ЗАДАНИЕ 5**

Выписать в тетрадь, что конкретно устанавливают следующие ГОСТы (*Стандарты для наведения порядка в программном хозяйстве*)

**ГОСТ 19.202-78** ЕСПД. Спецификация

**ГОСТ 19.401-78** ЕСПД. Текст программы

**ГОСТ 19.403-79** ЕСПД. Ведомость держателей подлинников

**ГОСТ 19.501-78** ЕСПД. Формуляр

**ГОСТ 19.507-79** ЕСПД. Ведомость эксплуатационных документов

#### **Новые Государственные стандарты РФ (ГОСТ Р)**

В РФ действует ряд стандартов в части документирования ПС, разработанных на основе прямого применения международных стандартов ИСО. Это — самые «свежие» по времени принятия стандарты. Некоторые из них напрямую адресованы руководителям проекта или директорам информационных служб. Вместе с тем они неоправданно мало известны в среде профессионалов. Вот их представление.

Стандарт полностью соответствует международному стандарту ИСО/МЭК ТО 9294:1990 и устанавливает рекомендации по эффективному управлению



документированием ПС для руководителей, отвечающих за их создание. Целью стандарта является оказание помощи в определении стратегии документирования ПС; выборе стандартов по документированию; выборе процедур документирования; определении необходимых ресурсов; составлении планов документирования

**ГОСТ Р ИСО/МЭК 9126-93.** Информационная технология.  
Оценка программной продукции. Характеристики качества и руководства по их применению.

Стандарт полностью соответствует международному стандарту ИСО/МЭК 9126:1991. В его контексте под характеристикой качества понимается «набор свойств (атрибутов) программной продукции, по которым ее качество описывается и оценивается». Стандарт определяет шесть комплексных характеристик, которые с минимальным дублированием описывают качество ПС (ПО, программной продукции): функциональные возможности; надежность; практичность; эффективность; сопровождаемость; мобильность. Эти характеристики образуют основу для дальнейшего уточнения и описания качества ПС.

**ГОСТ Р ИСО/МЭК 9127-94.** Системы обработки информации.  
Документация пользователя и информация на упаковке для потребительских программных пакетов

Стандарт полностью соответствует международному стандарту ИСО 9127:1989. В контексте настоящего стандарта под потребительским программным пакетом (ПП) понимается «программная продукция, спроектированная и продаваемая для выполнения определенных функций; программа и соответствующая ей документация, упакованные для продажи как единое целое». Под документацией пользователя понимается документация, которая обеспечивает конечного пользователя информацией по установке и эксплуатации ПП. Под информацией на упаковке понимают информацию, воспроизводимую на внешней упаковке ПП. Ее целью является предоставление потенциальным покупателям первичных сведений о ПП.

**ГОСТ Р ИСО/МЭК 8631-94** Информационная технология.  
Программные конструктивы и условные обозначения для их представления. Описывает представление процедурных алгоритмов.

## **ЗАДАНИЕ 6:**

### **Знакомство с Международным стандартом ISO/IEC 12207: 1995-08-01**

Ответить на вопросы:

Что определяет стандарт ISO/IEC 12207: 1995-08-01? Что он охватывает и описывает?

Что понимается под термином СИСТЕМА по ISO/IEC 12207: 1995-08-01?

Что такое Модель жизненного цикла (ЖЦ) ПО по ISO/IEC 12207: 1995-08-01? Опишите 5 основных процессов ЖЦ ПО.

Перечислите 8 вспомогательных процессов, которые поддерживают реализацию другого процесса, будучи неотъемлемой частью всего ЖЦ программного изделия, и обеспечивают должное качество проекта ПО.

## ТЕМЫ РЕФЕРАТОВ

1. Метрология ПО. Основные понятия и определения.
2. Основные понятия метрической теории программ.
3. Метрики: интервальные, порядковые и категорийные шкалы
4. Системы показателей качества программного продукта.
5. Оценка качества программ с помощью характеристики Боэма.
6. Группы показателей качества. Эргономические показатели.
7. Группы показателей качества. Технологические показатели.
8. Группы показателей качества. Конструктивные показатели
9. Надежность. Основные понятия теории надежности.
10. Проблемы надежности ПО. Отказ, сбой, восстановление при исполнении программ.
11. Факторы, определяющие надежность функционирования сложных комплексов программ.
12. Показатели надежности.
13. Сложность программ. Оценка программ по метрикам размера.
14. Сложность программ. Оценка программ по метрикам сложности потока управления.
15. Сложность программ. Оценка программ по метрикам сложности потока данных.
16. Оценка программ по метрикам стилистики.
17. Корректность программ
18. Методы измерений и проверки корректности
19. Методы измерения качества ПО.
20. Инструментальные, программные и аппаратные средства измерений и количественной оценки качества программного обеспечения.

## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. *Что такое технический барьер?*
2. *В каких областях осуществляется техническое регулирование?*
3. *Перечислите принципы технического регулирования.*
4. *Какими обстоятельствами вызвана реформа технического регулирования?*
5. *В чем проявляется защитная функция технического регулирования?*
6. *Укажите объекты ОТР.*
7. *Укажите объекты СТР.*

8. Перечислите вопросы, которыми руководствуются при принятии решения о разработке ТР.
9. В чем заключается принцип идентификации проблемы при принятии решения о разработке ТР?
10. Что понимается под объектом технического регулирования?
11. Какие требования предъявляются к такому структурному элементу, как «требования безопасности»?
12. Приведите примеры гармонизации с международными стандартами планируемых к разработке ТР.
13. Приведите примеры органов исполнительной власти, которые несут ответственность за реализацию ТР.
14. Какие требования предъявляются к порядку разработки ТР?
15. В каких формах может быть принят ТР? Какая из них основная?
16. Какие подзаконные акты разрабатываются для реализации утвержденного ТР?
17. Какова необходимость включения в содержание ТР такого раздела (главы, статьи), как «переходный период»?
18. Каковы права органов, осуществляющих госконтроль (надзор) за соблюдением требований ТР? 20. На какой стадии жизненного цикла продукции осуществляется ГКиН?
19. Что вы знаете о разработке ТР в рамках переходного периода, предусмотренного ФЗ о техническом регулировании?
20. Что вы знаете о развитии технического законодательства за рубежом?
21. Какие документы охватывает понятие «нормативный документ»?
22. Какие из перечисленных нормативных документов содержат обязательные требования: государственные стандарты, кодексы установившейся практики, правила, технические регламенты, отраслевые стандарты, общероссийские классификаторы, стандарты общественных объединений?
23. При реализации каких целей выполняются следующие функции: а) охранная; б) ресурсосберегающая; в) коммуникативная; г) цивилизующая?
24. При разработке каких нормативных документов используется метод систематизации объектов?
25. За счет чего удастся повысить качество готовой продукции при осуществлении комплексной стандартизации?
26. Почему опережающая стандартизация позволяет повысить конкурентоспособность продукции?
27. Какие обязательные требования к продукции установил ФЗ о техническом регулировании?
28. Как расшифровать аббревиатуру ГОСТ?
29. Прерогативой каких документов является установление обязательных требований?
30. Чем отличаются правила по стандартизации от рекомендаций по стандартизации? Приведите пример того и другого документа.
31. Что такое вид стандарта? Перечислите основные виды стандартов.
32. Что такое основополагающий стандарт? Приведите примеры организационно-методических и общетехнических стандартов.
33. Какие требования предъявляются к стандартам на методы контроля?

34. *Что такое знак соответствия национальному стандарту?*
35. *Какие требования предъявляются к применению знака соответствия?*
36. *Какие ранее действовавшие категории стандартов заменяет стандарт организации?*
37. *В чем преимущества стандарта организации перед национальным стандартом?*
38. *Назовите объекты стандартов организаций.*
39. *Назовите субъекты стандартов организаций.*
40. *Приведите примеры государственных стандартов, используемых в организациях розничной торговли.*
41. *В каком источнике содержится информация о действующих государственных стандартах РФ?*
42. *Какой вариант применения международного стандарта в Российской Федерации реализован в стандарте ГОСТ Р ИСО 9000—2001 (судя по обозначению)?*
43. *Какой вариант применения международного стандарта в Российской Федерации реализован в стандарте ГОСТ Р 50231—92 (ИСО 7173—89) (судя по обозначению)?*
44. *Назовите основные правила, предусмотренные Соглашением по техническим барьерам в торговле.*
45. *Требования каких международных профессиональных объединений следует учитывать при продвижении товара на внешний рынок?*
46. *Приведите примеры технических барьеров из области стандартизации.*
47. *Каким основным документом в странах ЕС представлено техническое законодательство?*
48. *Какие комплексы стандартов особенно широко используются для целей сертификации?*
49. *Какой основной документ является главным результатом работ по Единой системе классификации и кодирования технико-экономической информации?*
50. *Назовите источник первичных данных, положенных в основу каталогизации продукции.*
51. *В каких случаях технические условия выполняют роль технических документов и нормативных документов?*
52. *Назовите объекты технических условий.*
53. *Назовите специфические виды стандартов, используемые при стандартизации услуг.*
54. *Укажите приоритетные направления технического регулирования в области стандартизации.*
55. *Какие условия в сфере стандартизации должна выполнить Россия для вступления в ВТО?*
56. *Какие нормативные документы в области стандартизации за рубежом носят добровольный характер?*
57. *Какие федеральные законы составляют техническое законодательство России?*
58. *Каковы два условия обеспечения единства измерений?*
59. *Что такое размер измеряемой величины?*
60. *По каким признакам подразделяют СИ?*
61. *Какую функцию выполняют стандартные образцы?*
62. *В чем различие в назначении рабочих СИ и эталонов?*

65. Назовите метрологические характеристики, определяющие: — область применения СИ; — качество измерения.
66. Какая характеристика определяет точность измерения СИ?
67. В чем различие понятий «сходимость результатов измерений» и «воспроизводимость результатов измерений»?
68. При передаче размера единицы от какого СИ получают размер «рабочий эталон 0-го разряда»?
69. Как расшифровывается аббревиатура ГСИ?
70. Что представляет организационная подсистема ГСИ?
71. Назовите сферы государственного метрологического контроля и надзора.
72. Кто проводит государственный метрологический контроль и надзор?
73. В каких случаях необходимо осуществлять процедуру «утверждение типа» СИ?
74. Что такое поверка СИ?
75. Что является объектом поверки СИ?
76. Как подтверждаются положительные результаты поверки?
77. В каком порядке проводится надзор за количеством товаров?
78. Как установить правомерность отклонения в массе нетто при надзоре за количеством фасованных товаров?
79. Сравните поверку и калибровку СИ.
80. Укажите, какие из перечисленных СИ подлежат государственному метрологическому контролю: метры в магазине «Ткани»; метры для работ на садовом участке; термометры в больнице; термометр в жилище; весы на сельскохозяйственном рынке; весы для взвешивания в домашних условиях; весы в банке; весы в санатории; весы в продовольственном магазине; весы, используемые на учебных занятиях.
81. Укажите, какой из перечисленных метрологических процедур подлежат весы, используемые продовольственным магазином: поверке, утверждению типа, калибровке?
82. Какой из метрологических процедур подлежат СИ, впервые ввезенные в страну в порядке импортных поставок: поверке, утверждению типа, калибровке?
83. В рамках какого вида государственного метрологического надзора осуществляются в магазине контрольные закупки?
84. По каким причинам единство измерений становится объектом технических регламентов?
85. Для чего нужна сертификация СИ?
86. Перечислите направления совершенствования метрологической деятельности.
87. В чем заключается административная ответственность за нарушение метрологических правил?
88. В каких случаях лицо, виновное в нарушении метрологических правил, может быть привлечено к уголовной ответственности?
89. Каково принципиальное обличье проекта нового ФЗ от действующего Закона РФ «Об обеспечении единства измерений»?
90. Какие лица или органы участвуют в подтверждении соответствия?
91. Какая сторона подтверждает соответствие: а) при сертификации соответствия; б) при декларировании соответствия?
92. Какую сторону представляет продавец: а) как получатель товара; б)

- при реализации товара покупателю?*
93. Назовите законодательные акты, предусматривающие обязательную сертификацию.
  94. Кем утверждаются перечни продукции, подлежащие сертификации соответствия и декларированию соответствия?
  95. Укажите нормативные документы, требования которых проверяются при обязательной сертификации.
  96. В чем заключается специфическая цель обязательной сертификации?
  97. В чем состоят общие цели обязательной и добровольной сертификации?
  98. В чем заключается специфическая цель добровольной сертификации?
  99. Какая форма подтверждения соответствия преобладает в России, какая — за рубежом?
  100. В чем сходство в процедурах обязательной сертификации и декларирования соответствия?
  101. В чем различие в процедурах обязательной сертификации и декларирования соответствия?
  102. Какие из перечисленных товаров являются объектом обязательной сертификации и декларирования соответствия: продукты питания для детей, хлебобулочные изделия, алкогольные напитки, одеяла, электроприборы, фотообъективы?
  103. Кем заверяется копия сертификата соответствия?
  104. Какие федеральные органы исполнительной власти создают системы сертификации?
  105. Какая система сертификации (с точки зрения принадлежности к федеральному органу исполнительной власти, сформировавшему систему) охватывает товары народного потребления и услуги населению?
  106. В чем заключаются функции органа сертификации?
  107. В чем различие понятий «схема сертификации» и «порядок сертификации»?
  108. Какая схема сертификации продукции является самой жесткой?
  109. Какова цель инспекционного контроля в работах по сертификации?
  110. Какие дополнительные документы может запросить у заявителя орган по сертификации?
  111. Какие иностранные сертификаты признаются в России?
  112. В чем отличие схем сертификации продукции от схем сертификации услуг?
  113. Какие специфические виды государственных стандартов используются при сертификации услуг?
  114. Какие нормативные документы используются при сертификации систем качества?
  115. Какова цель сертификации систем качества?
  116. Какие предписания выдаются при госнадзоре организациям, которые нарушают обязательные требования государственных стандартов?
  117. Как решается проблема «снятия избыточности» обязательной сертификации?
-

Тема 2.2. Документирование основных процессов ЖЦ ПО

Тема 2.3. Документирование вспомогательных процессов ЖЦ ПО

### **Оценочные средства для проведения промежуточной аттестации**

*Вопросы к дифференцированному зачету*

1. *Взаимосвязь дискретной математики с другими дисциплинами. Практические проблемы, изучаемые методами дискретной математики*
2. *Составные высказывания. Простейшие связки. Логические отношения.*
3. *Варианты импликации.*
4. *Основные законы, определяющие свойства логических операций.*
5. *Булевы функции.*
6. *Свойства элементарных булевых функций.*
7. *Дизъюнктивные и конъюнктивные нормальные формы алгебры высказываний.*
8. *Совершенная дизъюнктивная и совершенная конъюнктивная нормальные формы.*
9. *Многочлены Жегалкина.*
10. *Специальные классы булевых функций: функции, сохраняющие единицу, функции, сохраняющие нуль, самодвойственные функции, линейные функции, монотонные функции. Теорема Поста о функциональной полноте.*
11. *Понятие множества. Способы задания множества. Подмножества. Операции над множествами.*
12. *Соотношения между множествами и составными высказываниями.*
13. *Соотношение между высказываниями и соответствующими им множествами истинности.*
14. *Абстрактные законы операций над множествами.*
15. *Кортежи и декартово произведение множеств. Степень множества.*
16. *Бинарные отношения в множестве и их свойства.*
17. *Отношения строгого и нестрогого порядка.*
18. *Отображение множеств. Функции.*
19. *Определенность и неопределенность функций. Композиция отображений.*
20. *Метод математической индукции. База индукции. Индукционный переход. Полная и неполная индукция.*
21. *Основные правила комбинаторики. Методы алгоритмического перечисления (генерации) основных комбинаторных объектов: перестановка, сочетание, размещение.*
22. *Комбинация элементов с повторениями. Бином Ньютона.*
23. *Предикаты. Применение предикатов в алгебре.*
24. *Булева алгебра предикатов.*
25. *Кванторы. Формулы логики предикатов.*
26. *Равносильные формулы логики предикатов. Приведенные и нормальные формы в логике предикатов.*
27. *Исчисления предикатов.*

28. Основные понятия теории графов. Степень вершины. Маршрут, цепи, циклы. Связность графа.
29. Ориентированные графы.
30. Изоморфизм графов.
31. Плоские графы. Операции над графами.
32. Способы задания графов. Некоторые типы графов.
33. Сети. Сетевые модели представления информации. Применение графов и сетей.
34. Вычислимые функции и алгоритмы.
35. Рекурсивные функции.
36. Нормальный алгоритм Маркова.
37. Машины Тьюринга.
38. Понятия конечного автомата. Определения и способы задания конечного автомата.
39. Примеры конечных автоматов.
40. Канонические уравнения автомата.

### **Практические задания к зачёту**

1 Создайте концептуальную модель некоторого университета, которая бы позволяла получить следующую информацию:

- Список студентов-юношей старше 20 лет;
- Список студентов-юношей, приписанных к определенному [военкомату](#).

2 Создайте концептуальную модель некоторого университета, которая бы позволяла получить следующую информацию:

- Список студентов определенной группы, поступивших занятию по болезни за период;
- Количество студентов, пропустивших занятия без уважительной причины по группам.

3 Создайте концептуальную модель некоторой библиотеки, которая бы позволяла получить следующую информацию:

- Список книг определенного автора, имеющих в библиотеке;
- Список книг, находящихся у читателей.

4 Создайте концептуальную модель некоторой библиотеки, которая бы позволяла получить следующую информацию:

- Список книг, находящихся у определенного читателя;
- Количество читателей-должников на определенную дату.

5 Создайте концептуальную модель некоторой библиотеки, которая бы позволяла получить следующую информацию:

- Список книг, имеющих в наличии на определенную дату;

6 Создайте концептуальную модель некоторой библиотеки, которая бы позволяла получить следующую информацию:

- Список читателей-должников на определенную дату;
- Список книг, находящихся у определенного читателя больше года.

7 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Список сотрудников, принятых на работу за период;
- Количество сотрудников старше 50 лет.



8 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Список сотрудников, уволенных за период;
- Количество сотрудников в возрасте от 25 до 50 лет.

9 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Количество сотрудников, принятых на работу за период;
- Список сотрудников, имеющих [высшее образование](#).

10 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Количество сотрудников, уволенных за период;

11 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Список сотрудников-юбиляров в текущем году;
- Количество сотрудников со стажем работы в нашей организации более 5 лет.

12 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Список сотрудников указанного отдела;
- Количество сотрудников, занимающих определенную должность.

13 Создайте концептуальную модель некоторого отдела кадров, которая бы позволяла получить следующую информацию:

- Количество сотрудников в указанном отделе;
- Список сотрудников со стажем работы в организации до 5 лет.

14 Создайте концептуальную модель некоторого отдела банка, которая бы позволяла получить следующую информацию:

- Список клиентов, имеющих вклады больше 50000 руб.;
- Количество выданных [ипотечных кредитов](#).

15 Создайте концептуальную модель некоторого банка, которая бы позволяла получить следующую информацию:

- Список клиентов, имеющих больше двух вкладов;
- Количество выданных потребительских кредитов.

16 Создайте концептуальную модель некоторого банка, которая бы позволяла получить следующую информацию:

- Кредиты, оформленные в текущем месяце;
- Сведения о клиентах банка.

17 Создайте концептуальную модель некоторого университета, которая бы позволяла получить следующую информацию:

- Список юношей, обучающихся на экономическом факультете;
- Количество студентов, имеющих по дисциплине «Базы данных» 2 балла.

18 Создайте концептуальную модель некоторого университета, которая бы позволяла получить следующую информацию:

- Количество юношей в учебном заведении;
- Оценки определенного студента по всем дисциплинам.

19 Создайте концептуальную модель некоторой [строительной фирмы](#), которая бы позволяла получить следующую информацию:

- Список всех договоров, заключенных с определенным поставщиком;
- Количество поставщиков материала «кирпич».

20 Создайте концептуальную модель некоторой строительной фирмы, которая бы позволяла получить следующую информацию:

- Количество поставок, выполненных по определенному договору с

поставщиком;

- Поставки за месяц.

21 Создайте концептуальную модель некоторой строительной фирмы, которая бы позволяла получить следующую информацию:

- Наличие оплаты по договору;

- Поставщик с максимальным числом договоров.

22 Создайте концептуальную модель некоторой строительной фирмы, которая бы позволяла получить следующую информацию:

- Количество действующих договоров с поставщиками;

### Примерные тестовые задания:

### Тест

#### Вариант 1

- Докажите тождественную истинность формулы  $X \rightarrow (X \rightarrow Y)$ .
- С помощью таблиц истинности проверить, являются ли эквивалентными высказывания:  $f_1 = XA (Y \rightarrow Z)$  и  $f_2 = (X \wedge Y) \vee (X \wedge Z)$ .
- Определите для каждого из следующих высказываний, будет ли оно логически истинным, противоречивым: ни тем, ни другим.  
а)  $X \sim X$ , \_\_\_\_\_ б)  $X \wedge X$ , в)  $(X \vee Y) \wedge (X \wedge Y)$ , г)  $(X \wedge Y) \wedge (Y \wedge X)$ ,  
д)  $(X \wedge Y) \wedge (Y \wedge Z) \wedge (X \rightarrow Z)$ , е)  $(X \wedge Y) \wedge X$ , ж)  $((X \wedge Y) \wedge X$
- Доказать закон отрицания конъюнкции  $(A \wedge B) \rightarrow \neg(A \wedge \neg B)$
- Найти значение  $A \vee B \rightarrow A \vee B$  и убедиться, что при всех значениях  $A$  и  $B$  - это истинное значение.  
6. С помощью основных равносильностей доказать закон обобщенного склеивания  $x \vee y \wedge z = x \vee yz$ .
- Составьте таблицу истинности булевой функции трех переменных  $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \vee x_1 \wedge x_2 \wedge x_3$  и найдите ее двоичный набор.

#### Вариант 2

- Докажите эквивалентность функции:  $f(x, y, z) = xA(x \vee z)A(y \vee z)$  и  $f(x, y, z) = (xAy)Y(xAz)$ .
- Найдите СДПФ и СКНФ функции  $f(x, y, z)$ , заданной следующей таблицей истинности:

$X_1$	$X_2$	$X_3$	$f(X_1, X_2, X_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- Опрос 100 студентов выявил следующие данные о числе студентов, изучающих различные иностранные языки: английский — 28; немецкий - 30; французский — 42, английский и немецкий - 8; английский и французский — 10; немецкий и французский - 5;

все три языка - 3.

- 1) Сколько студентов не изучают ни одного иностранного языка?
- 2) Сколько студентов изучают один французский язык?
- 3) Сколько студентов изучают немецкий язык в том и только в том случае, если они изучают французский язык?

Решение. Нарисовать диаграмму Эйлера-Венна в виде трех кругов, обозначающих множество студентов, изучающих соответственно французский, немецкий и английский языки. В каждую из 8-ми областей вписать данные, используя приведенные цифры. Начинать с конца списка и двигаться к началу.

Ответ: 1) 20; 2) 30; 3) 38.

- 4) Изобразите с помощью диаграмм Эйлера-Венна множества:
  - 1)  $A' = 3$  и  $B \cap C$ ; 2)  $A \subset 3$ ;  $3 \cap A \setminus B = 0$ ; 3)  $A \cap B = 3$ ;  $3 \cap C = C$  и  $C = A \cup 3$ ;
  - 4)  $A' = B$ ;  $3' = C$  и  $A \cap B \cap C = 0$ ; 5)  $(A \setminus B) \cap (B \setminus A) = 3$ ;
- 5) Воспользовавшись диаграммой Эйлера-Венна, определите, какие из следующих высказываний истинны: а)  $X \cap X \subset A$ ; б)  $X \cap A \subset A$ ; в)  $X \cup (A \cap B) \subset A \cup B$ ; г)  $X \rightarrow (Y \cap X)$ ; д)  $X \cap D \subset (Y \cap A)$ .
- 6) Пусть  $A = \{1, 2\}$ . Выписать все элементы декартова произведения  $A \times A$ .
- 7) Рассмотрим два множества  $A = \{a, b, c, d, e, f, g, h\}$  и  $B = \{1, 2, 3, 4, 5, 6, 7, 8\}$ .

### Вариант 3

1. Проверьте на линейность функцию  $f(x_1, x_2, x_3)$ , если ее двоичный набор  $F = 11100001$ .
2. Найдите правую и левую области отношения  $R = \{<1, 5>; <1, 6>; <1, 7>\}$ .
3. Если  $L = \{2, 3, 4, 5, 6, 7, 8\}$ , запишите бинарное отношение  $R = \{<x, y> : x, y \in L, x \text{ делит } y, \text{ и } x \neq y\}$ .
4. Являются ли следующие отношения функциями:
  - 1)  $\{<1, 2>; <2, 3>; <3, 2>\}$ ; 2)  $\{<1, 2>; <1, 3>; <2, 3>\}$ ;
  - 3)  $\{x, x^2 - 2x - 3 : x \in A\}$ ?
5. Футбольный мяч сшит из 32 лоскутов: белых шестиугольников и черных пятиугольников. Каждый черный лоскут граничит только с белыми, а каждый белый — с тремя черными и тремя белыми. Сколько лоскутов белого цвета?
6. Из 4 первокурсников, 5 второкурсников и 6 третьекурсников надо выбрать 3 студента на конференцию. Сколькими способами можно осуществить этот выбор, если среди выбранных должны быть студенты разных курсов?
7. Сколько можно составить четырехзначных чисел так, чтобы любые две соседние цифры были различны?

Форма аттестации	Знания	Умения	Практический опыт (владеть)	Личные качества обучающегося	Примеры оценочных средств
зачет	Контроль	Оценка умения	Оценка навыков	Оценка способности	Вопросы:

знания базовых положений в области документирования и сертификации	понимать специальную терминологию	логического сопоставления и характеристики объектов	грамотно и четко излагать материал	прилагаются
	Оценка умения решать типовые задачи в области профессиональной деятельности	Оценка навыков логического мышления при решении задач в области профессиональной деятельности	Оценка способности грамотно и четко излагать ход решения задач в области проф. деятельности и аргументировать результаты	Задачи прилагаются

#### 7.4.2. Примерные задачи для проведения промежуточной аттестации

### Итоговый тест

#### Список вопросов теста

##### Вопрос 1

**Множество, не содержащее ни одного элемента, называется ...**

##### Варианты ответов

- Пустым
- Конечным
- Нулевым

##### Вопрос 2

**Математический символ  $\emptyset$  обозначает ...**

##### Варианты ответов

- Нулевое множество
- Бесконечное множество
- Пустое множество

##### Вопрос 3

**Существует ли множество без элементов?**

##### Варианты ответов

- В любом множестве должно быть не менее 1 элемента
- Да
- Нет

##### Вопрос 4

**При пересечении двух множеств получаем третье множество, которое ...**

##### Варианты ответов

- Может состоять из одного элемента
- Всегда состоит из одного элемента
- Всегда не содержит элементов
- Иногда не содержит элементов

##### Вопрос 5

**Как можно изобразить множество графически?**

##### Варианты ответов

- Интервалом на числовой оси
- Частью координатной плоскости
- Кругами Эйлера (диаграммами Эйлера-Венна)
- Этого нельзя сделать

Вопрос 6

**При обозначении множеств используют ...**

Варианты ответов

- Только фигурные скобки
- Только круглые скобки
- Иногда круглые, иногда фигурные, иногда одновременно оба вида скобок

Вопрос 7

**Если можно найти разность двух множеств, то можно найти и их ...**

Варианты ответов

- Объединение
- Сумму
- Произведение
- Частное

Вопрос 8

**Если все элементы множества  $A$  входят в множество  $B$ , то можно сказать, что ...**

Варианты ответов

- $A$  – образ множества  $B$
- $A$  – подмножество  $B$
- $B$  – прообраз множества  $A$
- $B$  – подмножество  $A$

Вопрос 9

**Множества обозначаются ...**

Варианты ответов

- Кириллицей
- Большими латинскими буквами
- Малыми латинскими буквами

Вопрос 10

**Какой операции над множествами соответствует выражение?**

**“Элемент, принадлежащий полученному множеству, принадлежит множеству  $A$  ИЛИ множеству  $B$ ”.**

Варианты ответов

- Объединение множеств
- Перечисление множеств
- Пересечение множеств

Вопрос 11

**Способ выражения мыслей или форма, по которой они строятся - это ...**

Варианты ответов

- Форма мышления
- Форма умозаключения
- Форма доказательства

Вопрос 12

**Согласны ли Вы с тем, что вопросительные и повелительные предложения не являются логическими высказываниями?**

Варианты ответов

- Да
- Нет

Вопрос 13

**Какой приоритет выполнения логических операций в логических выражениях?**

*Варианты ответов*

- Отрицание
- Конъюнкция
- Дизъюнкция; исключая дизъюнкция
- Следование; эквивалентность

*Вопрос 14*

**Является логическое отрицание (инверсия) унарной операцией?**

*Варианты ответов*

- Да
- Нет

*Вопрос 15*

**Сложные высказывания представляют собой объединение простых высказываний с помощью ...**

*Варианты ответов*

- Логических запятых
- Логических скобок
- Логических связок

*Вопрос 16*

**Логическое выражение, принимающее значение истина при любых значениях входящих в него переменных - это ...**

*Варианты ответов*

- Выполнимое выражение
- Тождественно–ложное выражение
- Тождественно–истинное выражение

*Вопрос 17*

**Закон тождества говорит о том, что каждое высказывание тождественно самому себе и имеет математическую форму записи ...**

*Варианты ответов*

- $A \equiv A$
- $A < = A$
- $A > = A$
- $A = A$

*Вопрос 18*

**Какие две формы интерпретации ФАЛ существуют?**

*Варианты ответов*

- Серверная интерпретация
- Теоретико-множественная интерпретация
- Игровая интерпретация
- Выказательная интерпретация

*Вопрос 19*

**Какое высказывание является ложным?**

*Варианты ответов*

- Знаком  $\&$  обозначается логическая операция И
- Знаком  $\&$  обозначается логическая операция дизъюнкция
- Логическую операцию И также называют логическим умножением

*Вопрос 20*

**Соединение двух простых высказываний А и В в одно составное с помощью союза И – это ...**

*Варианты ответов*

- Конъюнкция
- Импликация
- Дизъюнкция

*Вопрос 21*

**В теоретико-множественной интерпретации дизъюнкция соответствует ...**

*Варианты ответов*

- Объединению множеств
- Пересечению множеств
- Разности множеств

*Вопрос 22*

**Определите правильный порядок действий при составлении таблицы истинности.**

*Варианты ответов*

- Определить количество строк в таблице.
- Определить количество столбцов, которое будет равно количеству переменных + количество логических операций.
- Построить таблицу, заполнить шапку и внести возможные наборы значений исходных логических переменных.
- Заполнить таблицу истинности по столбцам, выполняя базовые логические операции в необходимой последовательности.

*Вопрос 23*

**Функция алгебры логики  $f(x_1, x_2, \dots, x_n)$  называется линейной если она может быть представлена ...**

*Варианты ответов*

- Полиномом Жегалкина
- Полиномом Вебба
- Полиномом Шеффера

*Вопрос 24*

**Два одинаково разрядных двоичных набора называются противоположными, если они различаются ...**

*Варианты ответов*

- Во всех разрядах
- Хоть в одном разряде
- В двух разрядах

*Вопрос 25*

**Из определения двойственности функций следует, что для любой ФАЛ существует двойственная функция, причём единственная.**

**Согласны ли Вы с этим утверждением?**

*Варианты ответов*

- Да
- Нет

## 8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Основные понятия и определения. Классификация

ТЕОРИЯ ГРАФОВ - это область дискретной математики, особенностью которой является геометрический подход к изучению объектов. Теория графов находится сейчас в самом расцвете. Обычно её относят к топологии (потому что во многих случаях рассматриваются лишь топологические свойства графов), однако она пересекается со многими разделами теории множеств, комбинаторной математики, алгебры, геометрии, теории матриц, теории игр, математической логики и многих других математических дисциплин. Основной объект теории графов-граф и его обобщения.

1. Граф - Пара объектов  $G = (X, \Gamma)$ , где  $X$  - конечное множество, а  $\Gamma$  -конечное подмножество прямого произведения  $X \times X$ . При этом  $X$  называется множеством вершин, а  $\Gamma$  - множеством дуг графа  $G$ .
2. Любое конечное множество точек (вершин), некоторые из которых попарно соединены стрелками, (в теории графов эти стрелки называются дугами), можно рассматривать как граф.
3. Если в множестве  $\Gamma$  все пары упорядочены, то такой граф называют ориентированным.
  1. Дуга- ребро ориентированного графа.
  2. Граф называется вырожденным, если у него нет рёбер.
  3. Вершина  $X$  называется инцидентной ребру  $G$ , если ребро соединяет эту вершину с какой-либо другой вершиной.
4. Подграфом  $G(V_1, E_1)$  графа  $G(V, E)$  называется граф с множеством вершин  $V_1 \subset V$  и множеством ребер (дуг)  $E_1 \subset E$ , - такими, что каждое ребро (дуга) из  $E_1$  инцидентно (инцидентна) только вершинам из  $V_1$ . Иначе говоря, подграф содержит некоторые вершины исходного графа и некоторые рёбра (только те, оба конца которых входят в подграф).
5. Подграфом, порождённым множеством вершин  $U$  называется подграф, которого -  $U$ , содержащий те и только те рёбра, оба конца которых входят в  $U$ .
6. Подграф называется остовным подграфом, если множество его вершин совпадает с множеством вершин самого графа.
7. Вершины называются смежными, если существует ребро, их соединяющее.
8. Два ребра  $G_1$  и  $G_2$  называются смежными, если существует вершина, инцидентная одновременно  $G_1$  и  $G_2$ .
9. Каждый граф можно представить в пространстве множеством точек, соответствующих вершинам, которые соединены линиями, соответствующими ребрам (или дугам - в последнем случае направление обычно указывается стрелочками). - такое представление называется укладкой графа.
10. Доказано, что в 3-мерном пространстве любой граф можно представить в виде



укладки таким образом, что линии, соответствующие ребрам (дугам) не будут пересекаться во внутренних точках.

Для 2-мерного пространства это, вообще говоря, неверно. Допускающие представление в виде укладки в 2- мерном пространстве графы называют плоскими (планарным).

Другими словами, планарным называется граф, который может быть изображен на плоскости так, что его рёбра не будут пересекаться.

11. Гранью графа, изображенного на некоторой поверхности, называется часть поверхности, ограниченная рёбрами графа.

Данное понятие грани, по существу, совпадает с понятием грани многогранника. В качестве поверхности в этом случае выступает поверхность многогранника. Если многогранник выпуклый, его можно изобразить на плоскости, сохранив все грани. Это можно наглядно представить следующим образом: одну из граней многогранника растягиваем, а сам многогранник «расплющиваем» так, чтобы он весь поместился внутри этой грани. В результате получим плоский граф. Грань, которую мы растягивали «исчезнет», но ей будет соответствовать грань, состоящая из части плоскости, ограничивающей граф.

Таким образом, можно говорить о вершинах, рёбрах и гранях многогранника, а оперировать соответствующими понятиями для плоского графа.

12. Пустым называется граф без рёбер. Полным называется граф, в котором каждые две вершины смежные.

13. Конечная последовательность необязательно различных рёбер  $E_1, E_2, \dots, E_n$  называется маршрутом длины  $n$ , если существует последовательность  $V_1, V_2, \dots, V_n$  необязательно различных вершин, таких, что  $E_i = (V_{i-1}, V_i)$ .

14. Если совпадают, то маршрут замкнутый.

15. Маршрут, в котором все рёбра попарно различны, называется цепью.

16. Замкнутый маршрут, все рёбра которого различны, называется циклом. Если все вершины

цепи или цикла различны, то такая цепь или цикл называются простыми.

17. Маршрут, в котором все вершины попарно различны, называется простой цепью.

18. Цикл, в котором все вершины, кроме первой и последней, попарно различны, называется

простым циклом.

19. Граф называется связным, если для любых двух вершин существует путь, соединяющий эти

вершины.

20. Любой максимальный связный подграф (то есть, не содержащийся в других связных подграфах) графа  $G$  называется компонентой связности. Несвязный граф имеет, по крайней мере, две компоненты связности.

21. Граф называется  $k$  - связным ( $k$  - реберно - связным), если удаление не менее  $k$  вершин (ребер) приводит к потере свойства связности.

22. Маршрут, содержащий все вершины или ребра графа и обладающий

определенными свойствами, называется обходом графа.

23. Длина маршрута (цепи, простой цепи) равна количеству ребер а порядке их прохождения.

Длина кратчайшей простой цепи, соединяющей вершины  $v_i$  и  $v_j$  в графе  $G$ , называется расстоянием  $d(v_i, v_j)$  между  $v_i$  и  $v_j$ .

24. Степень вершины - число ребер, которым инцидентна вершина  $V$ , обозначается  $D(V)$ .

С помощью различных операций можно строить графы из более простых, переходить от графа к более простому, разбивать графы на более простые и т.д.

Среди одноместных операций наиболее употребительны: удаление и добавление ребра или вершины, стягивание ребра (отождествление пары смежных вершин), подразбиение ребра (т.е. замена ребра  $(u, v)$  на пару  $(u, w)$ ,  $(w, v)$ , где  $w$  - новая вершина) и др.

Известны двуместные операции: соединение, сложение, различные виды умножений графов и др. Такие операции используются для анализа и синтеза графов с заданными свойствами.

25. Два графа  $G_1=(V_1;E_1)$ ,  $G_2=(V_2;E_2)$ , называются изоморфными, если существует взаимнооднозначное соответствие между множествами вершин  $V_1$  и  $V_2$  и между множествами ребер  $E_1$  и  $E_2$ , такое, чтобы сохранялось отношение инцидентности.

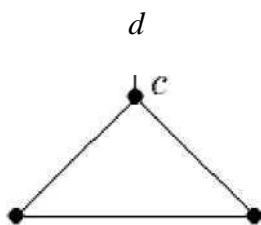
Понятие изоморфизма для графов имеет наглядное толкование. Представим ребра графов эластичными нитями, связывающими узлы - вершины. Тогда, изоморфизм можно представить как перемещение узлов и растяжение нитей.

26. Связный граф без циклов называется деревом.

Деревья особенно часто возникают на практике при изображении различных иерархий. Например, популярные генеалогические деревья.

### Способы задания графов:

1. Геометрический:



( П Б

2. Матрица смежности:

	a	B	c	d
A	0	1	1	0
B	1	0	1	0
C	1	1	0	1
D	0	0	1	0

Матрица смежности - квадратная матрица, размерности, равной количеству вершин. При этом  $a[i, j]$  - целое число, равное количеству ребер, связывающих  $i$ -ю,  $j$ -ю вершину. Если в графе нет петель, то диагональные элементы равны 0.

Если рёбра не повторяются, то все элементы 0 или 1. Если граф неориентированный, то матрица симметрична.

3. Матрица инцидентности:

	a	B	c	d
A	1	1	0	0
B	0	1	1	0
C	1	0	1	0
D	0	0	1	1

4. Явное задание графа как алгебраической системы:

$$\langle \{a,b,c,d\}, \{u,v,w,x\}; \{(u,a),(u,b),(v,b),(v,c),(w,c),(w,a),(x,c), (x,d)\} \rangle.$$

Так как мы рассматриваем только простые графы, граф нам проще определять как модель, носителем которой является множество вершин, а отношение - бинарное отношение смежности вершин. Тогда данный граф запишется как  $\langle \{a,b,c,d\}; \{(a,b), (b,a), (b,c), (c,b), (a,c), (c,a), (c,d), (d,c)\} \rangle$ . В таком представлении ребру соответствуют две пары вершин  $(v_1, v_2)$  и  $(v_2, v_1)$ , инцидентных данному ребру. Чтобы задать такое представление, достаточно для каждого ребра указать двухэлементное множество вершин - его мы и будем отождествлять с ребром. Для данного графа рёбра задаются множеством  $\{\{a,b\}, \{b,c\}, \{a,c\}, \{c,d\}\}$  и граф мы будем записывать как пару  $(V, E)$ , где  $V$  - множество вершин, а  $E$  - множество рёбер.

5. Наконец, граф можно задать посредством **списков**.

Например:

*вариант 1:* списком пар вершин, соединенных ребрами (или дугами);

*вариант 2:* списком списков для каждой вершины множества смежных с ней вершин.

### Примеры применения в моделировании

Развитие теории графов в основном обязано большому числу всевозможных приложений. По-видимому, из всех математических объектов графы занимают одно из первых мест в качестве формальных моделей реальных систем.

Графы нашли применение практически во всех отраслях научных знаний: физике, биологии, химии, математике, истории, лингвистике, социальных науках, технике и т.п. Наибольшей популярностью теоретико-графовые модели используются при исследовании коммуникационных сетей, систем информатики, химических и генетических структур, электрических цепей и других систем сетевой структуры.

Далее перечислим некоторые типовые задачи теории графов и их приложения:

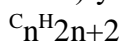
- Задача о кратчайшей цепи

- \* замена оборудования
- \* составление расписания движения транспортных средств
- \* размещение пунктов скорой помощи
- \* размещение телефонных станций
- Задача о максимальном потоке
  - \* анализ пропускной способности коммуникационной сети
- \* организация движения в динамической сети
- \* оптимальный подбор интенсивностей выполнения работ
- \* синтез двухполюсной сети с заданной структурной надежностью
- \* задача о распределении работ
- Задача об упаковках и покрытиях

- \* оптимизация структуры ПЗУ
- \* размещение диспетчерских пунктов городской транспортной сети
  - Раскраска в графах
- \* распределение памяти в ЭВМ
- \* проектирование сетей телевизионного вещания
  - Связность графов и сетей
- \* проектирование кратчайшей коммуникационной сети
- \* синтез структурно-надежной сети циркуляционной связи
- \* анализ надежности стохастических сетей связи
  - Изоморфизм графов и сетей
- \* структурный синтез линейных избирательных цепей
- \* автоматизация контроля при проектировании БИС
  - Изоморфное вхождение и пересечение графов
- \* локализация неисправности с помощью алгоритмов поиска МИПГ
- \* покрытие схемы заданным набором типовых подсхем
  - Автоморфизм графов
- \* конструктивное перечисление структурных изомеров для производных органических соединений
- \* синтез тестов цифровых устройств

### **Графы и химия**

Еще А. Кэли рассмотрел задачу о возможных структурах насыщенных (или предельных) углеводородов, молекулы которых задаются формулой:



Молекула каждого предельного углеводорода представляет собой дерево. Если удалить все атомы водорода, то оставшиеся атомы углеводорода также будут образовывать дерево, каждая вершина которого имеет степень не выше 4. Следовательно, число возможных структур предельных углеводородов, т. е. число гомологов данного вещества, равно числу деревьев с вершинами степени не больше четырех.

Таким образом, подсчет числа гомологов предельных углеводородов также приводит к задаче о перечислении деревьев определенного типа. Эту задачу и ее обобщения рассмотрел Д. Пойа.

### **Графы и биология**

Деревья играют большую роль в биологической теории ветвящихся процессов. Для простоты мы рассмотрим только одну разновидность ветвящихся процессов - размножение бактерий. Предположим, что через определенный промежуток времени каждая бактерия либо делится на две новые, либо погибает. Тогда для потомства одной бактерии мы получим двоичное дерево.

Нас будет интересовать лишь один вопрос: в скольких случаях  $n$ -е поколение одной бактерии насчитывает ровно  $k$  потомков? Рекуррентное соотношение, обозначающее число необходимых случаев, известно в биологии под названием процесса Гальтона-Ватсона. Его можно рассматривать как частный случай многих общих формул.

