

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Федорова Марина Владимировна  
Должность: Директор филиала  
Дата подписания: 29.09.2023 10:21:08  
Уникальный программный ключ:  
e766def0e2eb455f02135d659e45051ac23041da

Приложение №9.4.43  
к ППСЗ по специальности 09.02.03  
Программирование в компьютерных  
системах

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ  
ПМ.03 УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ  
для специальности  
09.02.03 Программирование в компьютерных системах  
Уровень подготовки - базовый  
Год начала подготовки 2020**

## СОДЕРЖАНИЕ

1.	ОБЩИЕ ПОЛОЖЕНИЯ.....	4
2.	РЕЗУЛЬТАТЫ ОСВОЕНИЯ МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ.....	5
3.	ОЦЕНКА ОСВОЕНИЯ ТЕОРЕТИЧЕСКОГО КУРСА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	8
4.	ОЦЕНКА ПО УЧЕБНОЙ И ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ..... 68	
5.	ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО).....	69
6.	ОЦЕНОЧНАЯ ВЕДОМОСТЬ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ.....	70
7.	ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	74

### Общие положения

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности. Участие в интеграции программных модулей.

Формой аттестации по профессиональному модулю является экзамен (квалификационный). Итогом этого экзамена является однозначное решение: «Вид профессиональной деятельности освоен / не освоен».

### Формы контроля и оценивания элементов профессионального модуля

Таблица 1

Элемент модуля	Форма контроля и оценивания	
	Промежуточная аттестация	Текущий контроль
МДК.03.01 Технология разработки программного обеспечения	Дифференцированный зачёт	- защита лабораторных и практических работ, внеаудиторных работ; - контрольные работы по темам МДК
МДК.03.02 Инструментальные средства разработки программного обеспечения	Дифференцированный зачёт	- защита лабораторных и практических работ, внеаудиторных работ ; - контрольные работы по темам МДК
МДК.03.03 Документирование и сертификация	Дифференцированный зачёт	- защита лабораторных и внеаудиторных работ;
УП.03.01.	Дифференцированный зачёт	Защита по каждому из разделов МДК.
ПП.03.01.	Дифференцированный зачёт	наблюдение и экспертная оценка выполнения работ Решение зачетной работы Защита отчетов по производственной практике
ПМ.03	Экзамен (квалификационный)	

В рамках программы учебной дисциплины реализуется программа воспитания, направленная на формирование следующих личностных результатов (дескрипторов):

ЛР 5. Демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России.

ЛР 7. Осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности.

ЛР 13. Демонстрирующий готовность обучающегося соответствовать ожиданиям работодателей: ответственный сотрудник, дисциплинированный, трудолюбивый, нацеленный на достижение поставленных задач, эффективно взаимодействующий с членами команды, сотрудничающий с другими людьми, проектно мыслящий.

ЛР 17. Ценностное отношение обучающихся к своему Отечеству, к своей малой и большой Родине, уважительного отношения к ее истории и ответственного отношения к ее современности.

ЛР 18. Ценностное отношение обучающихся к людям иной национальности, веры,

культуры; уважительного отношения к их взглядам.

ЛР 19. Уважительное отношения обучающихся к результатам собственного и чужого труда.

ЛР 22 Приобретение навыков общения и самоуправления.

ЛР 23. Получение обучающимися возможности самораскрытия и самореализация личности.

## Результаты освоения модуля, подлежащие проверке

### 2.1. Профессиональные и общие компетенции:

Таблица 2

Профессиональные компетенции	Показатели оценки результата
ПК 1. Анализировать проектную и техническую документацию на уровне взаимодействия компонентов программного обеспечения.	Анализ требований к программному обеспечению. Определение характера взаимодействия компонентов программного обеспечения. Анализ проектной и технической документации на уровне взаимодействия компонент программного обеспечения. Точность и грамотность оформления технологической документации.
ПК 2. Выполнять интеграцию модулей в программную систему.	Определение этапов разработки программного обеспечения. Демонстрация построения концептуальной, логической и физической моделей программного обеспечения и отдельных модулей. Выбор технологии разработки исходного модуля исходя из его назначения. Выбор методов разработки программных модулей. Выбор средств разработки программных модулей. Демонстрация навыков модификации программных модулей.
ПК 3. Выполнять отладку программного продукта с использованием специализированных программных средств.	Выявление ошибок в программных модулях. Определение возможности увеличения быстродействия программного продукта. Определение способов и принципов оптимизации. Выбор методов отладки программных модулей и программного продукта. Выбор специализированных средств для отладки программного продукта. Демонстрация навыков использования программных средств для отладки программного продукта
ПК 4. Осуществлять разработку тестовых наборов и тестовых сценариев.	Разработка тестовых наборов и тестовых сценариев. Демонстрация устранения ошибок в программных модулях. Демонстрация использования методов тестирования программного обеспечения. Демонстрация навыков внесения изменения в программные модули для обеспечения качества программного обеспечения. Демонстрация навыков правильного использования инструментальных средств тестирования программных модулей.
ПК 5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования	Выбор методов обеспечения качества и надежности в процессе разработки сложных программных средств. Изложение основных принципов тестирования. Способен производить инспектирование компонент программного продукта на предмет соответствия стандартам

	кодирования.
ПК 6. Разрабатывать технологическую документацию	Правильность выбора методов средств разработки программной документации. Точность и грамотность оформления технологической документации.
<b>Общие компетенции</b>	<b>Показатели оценки результата</b>
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	Демонстрация интереса к будущей профессии
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	Выбор и применение методов и способов решения профессиональных задач в области разработки и администрирования баз данных. Оценка эффективности и качества выполнения.
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Решение стандартных и нестандартных профессиональных задач в области разработки и администрирования баз данных
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	Эффективный поиск необходимой информации. Использование различных источников, включая электронные.
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.	Разрабатывать, программировать и администрировать базы данных.
ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	Взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения.
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	Самоанализ и коррекция результатов собственной работы.
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	Организация самостоятельных занятий при изучении профессионального модуля.
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Анализ инноваций в области разработки и администрирования баз данных.

2.2. В результате изучения профессионального модуля обучающийся должен:

**иметь практический опыт:**

ПО 1. Участия в выработке требований к программному обеспечению;

ПО 2. Участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

**уметь:**

У 1. Владеть основными методологиями процессов разработки программного обеспечения;

У 2. Использовать методы для получения кода с заданной функциональностью и степенью качества;

У 3. Оформлять документацию на программные средства.

У 4. Использовать инструментальные средства для автоматизации оформления документации.

**знать:**

З 1. модели процесса разработки программного обеспечения;

З 2. основные принципы процесса разработки программного обеспечения;

З 3. основные подходы к интегрированию программных модулей;

З 4. основные методы и средства эффективной разработки;

З 5. основы верификации и аттестации программного обеспечения;

З 6. концепции и реализации программных процессов;

З 7. принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;

З 8. методы организации работы в коллективах разработчиков программного обеспечения;

З 9. основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;

З 10. стандарты качества программного обеспечения;

З 11. методы и средства разработки программной документации.

### 2.3 Критерии оценки:

Академическая оценка	Критерии оценки
5 «отлично»	Проект разработан полностью с соблюдением всех этапов разработки и использованием интегрированной среды программирования, программный код составлен грамотно, рационально, в соответствии со стандартом кодировки, интерфейс приложения удовлетворяет условиям поставленной задачи и соответствует всем требованиям к пользовательскому интерфейсу, программа работает корректно, проходит всю систему тестовых заданий, обрабатывает исключительные ситуации и выдает ожидаемый результат. Имеется необходимая документация на программный продукт.
4 «хорошо»	Допущено не более двух недочетов на каком-нибудь этапе разработки, но приложение работает и выдает результат.
3 «удовлетворительно»	Имеются недостатки в программном коде, либо не обрабатываются исключения, либо выбран не совсем удачный алгоритм решения проблемы, программный код работает, хотя и не совсем корректно. Сопровождающая документация имеется.
2 «неудовлетворительно»	Приложение разработано с нарушениями, программный код написан с ошибками, не проходит тестирования, не имеется соответствующей документации на программный продукт.

## **Оценка освоения теоретического курса профессионального модуля ПМ.03 Участие в интеграции программных модулей**

Контроль и оценка осуществляются с использованием следующих форм и методов: для проведения текущего и рубежного контроля – тест, контрольная работа, самостоятельная работа, фронтальный опрос; для промежуточной аттестации – дифференцированный зачет, квалификационный экзамен.

### **Текущий контроль**

#### **Практические занятия по МДК.03.01 Технология разработки программного обеспечения**

#### **Практическое занятие № 1**

**Тема:** Анализ предметной области ПО.

**Цель:** изучение различных форм представления знаний об объектах и их практическое использование, формирование умения выбирать форму представления знаний об объектах, ориентированную на определённый язык программирования

#### **Задание и порядок проведения лабораторной работы**

1. Составить перечень заинтересованных лиц.
2. Провести интервью и/или анкетирование с каждым заинтересованным лицом.
3. Проанализировать полученную информацию и сформулировать актуальность проблемы и потребности заинтересованных лиц.

#### **Контрольные вопросы**

1. Что является исходными данными для анализа проблемы (предметной области)?
2. Что является результатом этапа системного анализа предметной области?
3. Как определить заинтересованных лиц?
4. Какой метод сбора информации наиболее эффективен?
5. Для чего проводятся интервьюирование и анкетирование?

Перечень тем для проектирование информационной системы

1. Магазин детских вещей
2. Магазин по продажам автозапчастей
3. Детская библиотека
4. Туристическая фирма
5. Склад
6. Грузовые перевозки
7. Детский сад
8. Танцевальная группа
9. Хобби
10. Спортивная секция
11. Общественный фонд
12. Издательство научно-технической литературы
13. Химчистка
14. Архив

#### **Практическое занятие №2**

**Тема:** Оформление спецификаций требований ПО

**Цель :** Формирование навыков разработки спецификаций на программный продукт

**Оборудование:** 1) персональный компьютер; 2) среда Microsoft Office.

**Контрольные вопросы**



- Для чего разрабатываются спецификации на программный продукт?
- Что должны включать спецификации на программный продукт?
- Что должна содержать спецификация процессов
- Что такое словарь терминов и для чего он используется?
- Что такое диаграмма переходов состояний и для чего ее используют?
- Что такое диаграмма потоков и для чего ее используют?

#### **Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1 и техническим заданием выполнить разработку спецификаций на программный продукт, которые должны включать:

- спецификации процессов;
- словарь терминов;
- диаграммы переходов состояний;
- диаграммы потоков с детализацией.

2. Оформить отчет.

#### **Содержание отчета**

- тема практического занятия;
- цель практического занятия;
- ответы на контрольные вопросы;
- задание на практическое занятие;
- разработанные спецификации процессов;
- словарь терминов;
- диаграммы переходов состояний;
- диаграммы потоков с детализацией;
- выводы по проделанной работе.

#### **Практическое занятие №3-4**

**Тема:** Проектирование модулей.

**Цель :** Формирование навыков проектирования приложений, с использованием методов структурного и объектного подхода к разработке.

**Оборудование:** 1) персональный компьютер; 2) среда Microsoft Office.

#### **Контрольные вопросы**

- В чем суть структурного подхода к разработке приложений?
- Охарактеризуйте объектный подход к разработке приложений.
- Какие виды диаграмм используются при объектном подходе к программированию?
- Для каких целей используется диаграмма вариантов использования? Что они показывают?
- Для чего используют диаграмму последовательности системы?
- Для чего используют диаграмму последовательности действий? Что она показывает?
- Диаграмма кооперации.
- Каким образом разрабатываются классы в приложении и какие диаграммы при этом используются?

#### **Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1:

- построить диаграмму вариантов использования;
- составить диаграмму последовательности системы по диаграмме последовательностей действий (либо диаграмму кооперации, используя концептуальную модель предметной области);
- разработать структуру классов и уточнить отношения между классами;
- используя диаграмму потоков и пункты 2, 3, и 4 разработать структуру приложения (схему программы, схему данных, схему работы системы);

2. Оформить отчет.

#### **Содержание отчета**

- тема практического занятия;
- цель практического занятия;
- ответы на контрольные вопросы;
- задание на практическое занятие;
- построенные диаграммы;
- разработанные схемы программ ;
- выводы по проделанной работе.

#### **Практическое занятие №5**

**Тема:** Разработка модулей.

**Цель :** Формирование навыков разработки графических пользовательских интерфейсов с использованием средств Visual Studio.NET.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

#### **Контрольные вопросы**

- Что такое интерфейс?
- Какие типы пользовательских интерфейсов существуют?
- Перечислите этапы разработки пользовательских интерфейсов?
- К какому типу интерфейсов будет относиться интерфейс, разработанный в данной лабораторной работе?
- Какие модели интерфейсов существуют?
- Какая модель интерфейса будет использована в данной работе?
- Что такое диалог?
- Какие типы диалогов существуют?
- Какие формы диалога Вы знаете?
- Какой тип диалога и какая форма диалога будет использована в данной работе?

#### **Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1 и спроектированным приложением в практическом занятии №3 разработать пользовательский интерфейс в среде визуального проектирования Visual Studio.NET..
2. Используя схемы и диаграммы, разработанные в практическом занятии №3, выполнить проектирование сценариев диалога и построить граф диалога.
3. Выполнить кодирование обработчиков событий в среде Visual Studio.NET..
4. Оформить отчет.

#### **Содержание отчета**

- тема практического занятия;
- цель практического занятия;
- ответы на контрольные вопросы;
- задание на практическое занятие;
- окна с разработанным пользовательским интерфейсом;
- таблицу компонентов, свойств, значений, используемых в приложении;
- сценарии диалогов и графы диалогов;
- листинг приложения;
- вывод о проделанной работе.

#### **Практическое занятие №6**

**Тема:** Оценивание характеристик качества ПО

**Цель :** тестирование и оценивание качественных показателей ПС.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

### **Контрольные вопросы**

Как определяется понятие «качество» государственным и международным стандартами?

Какие способы получения информации о ПС приведены в стандарте ГОСТ 28195-89?

Какой стандарт необходимо применить для оценки качества ПС, если необходимо оценить уровень автоматизации ПС?

Дайте определение фактора надёжность ПС и перечислите его критерии качества согласно ГОСТ 28195-89.

Чем обусловлена объективная необходимость повышения качества продукции и программных систем, в том числе, в современных условиях?

## **Практическое занятие № 7-8**

**Тема: Визуальное моделирование**

**Цель:** получить практические навыки построения модели разрабатываемой информационной системы

**Задание для самостоятельного выполнения**

По заданию своего варианта построить диаграмму классов, объектов и последовательностей.

## **Практическое занятие № 9**

**Тема: Разработка тестов. Автоматическая генерация тестов на основе формального описания**

**Цель:** Формирование навыков разработки тестов в среде Visual Studio.NET.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

### **Контрольные вопросы**

1. Что понимают под тестированием программных продуктов.
2. Какие виды тестирования программных продуктов Вы знаете?
3. Что называют графом программы?
4. Какими способами можно описать граф программы?
5. Как построить тестовые пути?
6. Что понимают под комплексным тестированием?
7. Какие ошибки в программах существуют?
8. Что понимают под локализацией ошибок?
9. Что понимают под отладкой программы?
10. Чем отладка отличается от тестирования?
11. Какие методы отладки существуют?

### **Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1 и спроектированным приложением в практическом занятии №4, разработанными пользовательским интерфейсом и структурой приложения построить граф программы, упакованную в матрицу смежности, области тестовых данных и тестовые пути.

2. Используя пункт 1 данного практического занятия подобрать тестовые данные для выборочного тестирования разрабатываемого приложения.

4. Оформить отчет.

### **Содержание отчета**

- граф программы, упакованную матрицу смежности, области тестовых данных и тестовые пути;
- окна с результатом тестирования и анализом ошибок;
- окна с результатами отладки;

- листинг приложения;
- вывод о проделанной работе.

### Задание 1

Для тестового случая №1 необходимо составить полный *список* всех возможных альтернативных путей (см. подраздел "*Список* альтернативных путей") и разработать соответствующие тесты.

Кроме того, необходимо:

- выбрать **случай использования** на основании дерева решений;
- составить пошаговое описание выбранного случая использования;
- учесть все альтернативные пути;
- составить спецификации тестовых случаев;
- разработать соответствующие **тестовые случаи (тесты)** ;
- составить **тестовые отчеты**.

### Задание 2

Для случая автоматического тестирования на основе MSC-диаграмм нужно повторить те же задания, что и для ручного тестирования.

### Практическое занятие № 10

**Тема:** Средства автоматизации тестирования

**Цель:** Научиться использовать средства автоматизации тестирования приложения.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

#### Контрольные вопросы

1. Какие основные подходы к автоматизации тестирования существуют?
2. Что такое тестовый сценарий?
3. Какова одна из главных проблем автоматизированного тестирования?
4. Что такое инструментарий автоматизированного тестирования?

#### Методические указания к выполнению практического занятия

1. В соответствии с вариантом задания на практическое занятие №1 и спроектированным приложением в практическом занятии №4, разработанными пользовательским интерфейсом и структурой приложения провести автоматизированное тестирование приложения.

2. Оформить отчет.

#### Содержание отчета

- окна с результатом тестирования и анализом ошибок;
- вывод о проделанной работе.

#### Задание

Выполнить автоматическое тестирование своего приложения в среде разработки Microsoft Visual Studio.

### Практическое занятие № 11

**Тема:** Тестирование и отладка программы

**Цель:** Формирование навыков тестирования и отладки приложений в среде Visual Studio.NET.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

#### Контрольные вопросы

1. Что понимают под тестированием программных продуктов.
2. Какие виды тестирования программных продуктов Вы знаете?
3. Что называют графом программы?
4. Какими способами можно описать граф программы?
5. Как построить тестовые пути?
6. Что понимают под комплексным тестированием?

7. Какие ошибки в программах существуют?
8. Что понимают под локализацией ошибок?
9. Что понимают под отладкой программы?
10. Чем отладка отличается от тестирования?
11. Какие методы отладки существуют?

**(Для текущего контроля можно использовать следующий тест)**

В каждом тестовом задании может быть несколько вариантов ответа. После проведения теста, студенты могут попробовать обосновать свои неверные ответы.

1. Тестовое окружение может использоваться для:
  1. запуска и выполнения тестируемого модуля
  2. передачи входных данных
  3. сбора ожидаемых выходных данных
  4. сравнения реальных выходных данных с ожидаемыми
  5. поддержки отчуждения отдельных модулей системы от всей системы

**Ответ: 1, 2, 4, 5**

2. Тестовое окружение для программного кода на структурных языках программирования состоит из:
  1. драйвера
  2. тестов
  3. заглушек
  4. исходного кода

**Ответ: 1, 3**

3. Модульное тестирование проводится для того, чтобы:
  1. удостовериться в корректной работе системы в целом
  2. удостовериться в корректной работе набора модулей
  3. удостовериться в корректной работе отдельного модуля

**Ответ: 3**

4. Модуль – это (с точки зрения наших семинарских занятий):
  1. часть программного кода, выполняющая одну функцию с точки зрения функциональных требований
  2. программный модуль, т.е. минимальный компилируемый элемент программной системы
  3. задача в списке задач проекта
  4. участок кода, который может уместиться на одном экране или одном листе бумаги
  5. один класс или их множество с единым интерфейсом.

**Ответ: 2**

5. Какие основные задачи решаются в ходе модульного тестирования?
  1. Поиск и документирование несоответствий требованиям
  2. Поддержка разработки и рефакторинга низкоуровневой архитектуры системы и межмодульного взаимодействия
  3. Рефакторинг модулей
  4. Поддержка рефакторинга модулей
  5. Отладка
  6. Поддержка устранения дефектов и отладки

**Ответ: 1, 2, 4, 6**

**Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1 и спроектированным приложением в практическом занятии №4 разработанными

пользовательским интерфейсом и структурой приложения построить граф программы, упакованную в матрицу смежности, области тестовых данных и тестовые пути.

2. Используя пункт 1 данного практического занятия подобрать тестовые данные для выборочного тестирования разрабатываемого приложения.

3. Выполнить отладку приложения.

4. Охарактеризовать типы ошибок, которые возникали в процессе отладки.

4. Оформить отчет.

#### **Содержание отчета**

- граф программы, упакованную матрицу смежности, области тестовых данных и тестовые пути;
- окна с результатом тестирования и анализом ошибок;
- окна с результатами отладки;
- листинг приложения;
- вывод о проделанной работе.

#### **Практическое занятие № 12**

**Тема: Оформление документации, сопровождающей процесс верификации и тестирования**

**Цель:** Изучить порядок оформления документации, сопровождающей процесс верификации и тестирования.

**Оборудование:** 1). Персональный компьютер;2). Среда Microsoft Office;3). Среда Visual Studio.NET.

#### **Контрольные вопросы**

1. Что такое верификация?
2. В чем основная цель верификации?
3. Чем отличается верификация от валидации?
4. Перечислите документы, создаваемые во время верификации ПО.

#### **Методические указания к выполнению практического занятия**

1. В соответствии с вариантом задания на практическое занятие №1 и спроектированным приложением в практическом занятии №4 разработанными пользовательским интерфейсом и структурой приложения разработать план тестирования, тест-требования и отчет о проблемах.

2. Оформить отчет.

#### **Содержание отчета**

- План тестирования;
- Тест-требования;
- Отчет о проблемах.

#### **Практическое занятие № 13**

**Тема: Тестирование приложения**

**Цель:** формирование навыков тестирования программного обеспечения.

**Оборудование:** персональный компьютер, ресурсы сети интернет.

#### **Перечень контрольных вопросов:**

1. Что такое юзабилити-тестирование?
2. Какие цели у юзабилити-тестирования?
3. В чем состоит роль эксперта при таком тестировании?

#### **Практическое занятие № 14-15**

**Тема: Подходы к проектированию тестов**

**Цель:** изучить подходы к проектированию тестов, получить практические навыки разработки системы тестов и тестовых сценариев.

**Оборудование:** персональный компьютер, ресурсы сети интернет.

**Задание на занятие.**

Разработать систему тестов по методу «белого ящика» и протестировать приложение по своему варианту.

Разработать систему тестов по методу «черного ящика» и протестировать приложение по своему варианту.

### **Контрольные вопросы**

1. Перечислите основные подходы к проектированию тестов.
2. В чем заключаются методы «белого ящика» и «черного ящика»?
3. Какие еще подходы к проектированию тестов вы знаете?

## **Практическое занятие № 16-17**

### **Тема: Разработка тестов ПО**

**Цель:** получение практических навыков в работе по тестированию программных средств

**Оборудование:** персональный компьютер, ресурсы сети интернет.

### **Порядок выполнения работы**

- 1) Разработать тестовые наборы для функционального тестирования (Приложение 1).
- 2) Провести тестирование программы и представить результаты в виде таблицы. (Приложение 1).
- 3) Выработать рекомендации для корректировки тестируемой программы.
- 4) Представить отчет по практическому занятию для защиты.

### **Защита отчета по практическому занятию**

Отчет по практическому занятию должен быть оформлен согласно требований филиала и состоять из следующих структурных элементов:

1. титульный лист;
2. текстовая часть;

Текстовая часть отчета должна включать пункты:

- условие задачи;
- порядок выполнения
- полученные результаты.

Защита отчета по практическому занятию заключается в предъявлении преподавателю полученных результатов, демонстрации полученных навыков в ответах на вопросы преподавателя.

### **Контрольные вопросы**

1. Что такое тестирование ПС?
2. Чем тестирование отличается от отладки ПС?
3. Для чего проводится функциональное тестирование?
4. Что такое комплексное тестирование?
5. Каковы правила тестирования программы «как черного ящика»?
6. Как проводится тестирования программы по принципу «белого ящика»?
7. Что такое модульное тестирование?
8. Как осуществляется сборка программы при модульно тестировании?

## **Практическое занятие № 18**

**Тема: Выполнение отладки с помощью инструментария средств разработки**

**Цель:** получение практического навыка отладки приложения средствами интегрированной среды разработки Visual Studio.NET

**Задание на занятие:**

1. Изучить теоретический материал по теме занятия.
2. Выполнить отладку приложения средствами интегрированной среды разработки Visual Studio.NET в соответствии с методическими рекомендациями.
3. Оформить отчет

### **Практическое занятие № 19-21**

**Тема: Интеграция системы**

**Цель:** получение практического навыка интеграции компонентов приложения в единое целое

**Задание на занятие:**

1. Изучить теоретический материал по теме занятия.
2. Выполнить интеграцию компонентов приложения средствами интегрированной среды разработки Visual Studio.NET в соответствии с методическими рекомендациями.
3. Оформить отчет

### **Практическое занятие № 22-24**

**Тема: Использование инструментальных средств интегрального тестирования**

**Цель:** Получить практические навыки тестирования программного обеспечения инструментальными средствами

**Задание на занятие:**

1. Изучить теоретический материал по теме занятия.
2. Выполнить интегральное тестирование приложения средствами интегрированной среды разработки Visual Studio.NET в соответствии с методическими рекомендациями.
3. Оформить отчет

### **Практическое занятие № 25-27**

**Тема: Использование инструментальных средств системного тестирования**

**Цель:** Получение практического навыка системного тестирования программного обеспечения

**Задание на занятие:**

4. Изучить теоретический материал по теме занятия.
5. Выполнить системное тестирование приложения средствами интегрированной среды разработки Visual Studio.NET в соответствии с методическими рекомендациями.
6. Оформить отчет

### **Практическое занятие № 28-29**

**Тема: Технические командные роли**

**Цель:** Получить практические навыки распределения командных ролей в группе.

**Задание на занятие:**

1. Создать группу для разработки проекта.
2. Распределить роли.
3. Описать деятельность каждого члена группы согласно его роли.

### **Практическое занятие № 30-31**

**Тема: Типы совместной деятельности**

**Цель:** изучить типы совместной деятельности в группе, получить практические навыки работы в группе

**Задание на занятие:**

1. Определить тип совместной деятельности, подходящий для выполнения задачи.
2. Описать представление о проекте каждому члену команды.
3. Обсудить представление о проекте и выработать тактику совместной деятельности.



4. Составить документ "Концепция проекта", который:
- описывает не только то, что делает продукт, но и то, чего он не делает;
  - конкретизирует продукт (например, позволяет включать и исключать определенные функциональные возможности из данной версии);
  - содержит описание путей реализации проекта, благодаря чему проектная группа и заказчик могут начать работу.

### **Практическое занятие № 32-33**

**Тема: Средства поддержки коллективной разработки**

**Цель:** получить практический навык создания группового проекта

**Задание на занятие:**

1. Изучить теоретический материал.
2. По заданию преподавателя создать командный проект по разработке программного приложения.

### **3.1.4 Практические работы по МДК.03.03 Документирование и сертификация**

#### **Практическое занятие №1.**

**Тема:** Порядок проведения сертификации информационно-программных средств.

**Цель:** Изучить правила проведения сертификации информационно-программных средств.

**Задание:**

1. Изучить и описать порядок проведения сертификации информационно-программных средств.
2. Изучить и заполнить документы, необходимые для проведения сертификации информационно-программных средств. (Комплект документов для проведения сертификации программных средств располагается в папке Документы для сертификации, которая находится в папке Мои документы).

**Ход занятия:**

- a. Изучить теоретический материал по заданной теме.
- b. Заполнить шаблоны документов на проведение сертификации информационно-программных средств.
- c. Оформить отчет по практическому занятию. В отчет включить заполненные документы и ответы на контрольные вопросы.

#### **Общие сведения**

#### **Порядок проведения сертификации программного обеспечения**

Процедуры и вся технология проведения работ по сертификации определяются схемой сертификации, которая устанавливает четкую совокупность действий, по результатам которых принимается решение о соответствии или несоответствии продукции заданным требованиям. Согласно идеологии Международной организации по стандартизации (ИСО) общепризнанными являются восемь основных схем сертификации. Они используются и в комплекте основополагающих документов системы сертификации ГОСТ Р. При этом число схем сертификации, принятых Госстандартом России, в два раза больше, чем принято в зарубежной и международной практике. Схемы сертификации, принятые в системе сертификации ГОСТ Р, приведены в приложении 1. Для каждой схемы сертификации продукции приводятся условия ее применения с учетом степени опасности продукции. При проведении сертификации программного обеспечения наиболее удобно применение схемы 10а. Госстандартом России предусматривается ее

использование в качестве доказательства соответствия (несоответствия) продукции (программного обеспечения) установленным требованиям декларации, о соответствии прилагаемым к ней документам, подтверждающим соответствие продукции установленным требованиям.

Порядок проведения сертификации программного обеспечения средств измерений, информационно-измерительных систем и аппаратно-программных комплексов определен такими методиками как МИ 2891-2004 "ГСИ. Общие требования к программному обеспечению средств измерений" и МИ 2955-2005 "Типовая методика аттестации программного обеспечения средств измерений и порядок ее проведения".

Кроме того, в настоящее время в связи с принятием 11 июня 2008 г. новой редакции Закона РФ "Об обеспечении единства измерений", где в статье 9, п. 1 говорится о том, что "в состав обязательных требований к средствам измерений ... в необходимых случаях включаются также требования к ... программному обеспечению", ФГУП ВНИИМС приступил к разработке национального стандарта ГОСТ Р "ГСИ. Требования к программному обеспечению средств измерений и информационно - измерительных систем".

Порядок проведения сертификации программного обеспечения включает:

- подачу заявки на сертификацию;
- принятие решения по заявке на сертификацию, в том числе назначение экспертов на проведение основных работ по сертификации из числа экспертов органа по сертификации;
- оформление договора на проведение работ по сертификации;
- проведение сертификационной проверки ПО, в том числе при необходимости проведение испытаний/контроля ПО по согласованным с заказчиком методикам;
- принятие решения о выдаче Сертификата соответствия и разрешения использования знака соответствия либо об отказе в выдаче Сертификата соответствия;
- выдача Сертификата соответствия и разрешения использования знака соответствия;
- занесение заявителя/изготовителя ПО и перечня сертифицированных ПО в Реестр СДС ПО;
- проведение инспекционного контроля сертифицированных ПО.

Результатом сертификации является возможность приобрести программный продукт в Российской Федерации с соответствующей поддержкой от производителя или его официального представителя.

В результате проведенной сертификации производитель ПО получает:

- Экспертное заключение;
- Свидетельство о сертификации;
- Право использовать логотип «Проверено IT Expert».

Сертификация выгодна и для покупателей соответствующего программного обеспечения.

Покупатель получит:

- Предметную оценку функционала программного обеспечения;
- Возможность сравнения продуктов между собой;
- Возможность самостоятельной оценки продуктов по своим критериям.

### **Перечень информации предоставляемой заявителем для прохождения процедуры сертификации**

- описание структуры сертифицируемого программного обеспечения, выполняемых функций, в том числе последовательность обработки данных;
- описание функций сертифицируемого ПО и параметров программного обеспечения, существенных для их работы;

- описание реализованных в сертифицируемом программном обеспечении алгоритмов функционирования, в том числе вычислительных алгоритмов, а также их блок-схемы;
  - описание модулей программного обеспечения;
  - перечень интерфейсов и перечень команд для каждого интерфейса, включая заявление об их полноте;
  - список, значение и действие всех команд, получаемых от устройств ввода (клавиатуры, мыши, сенсорных устройств и т.п.);
  - описание реализованных методов идентификации сертифицируемого программного обеспечения;
  - описание реализованных методов защиты сертифицируемого программного обеспечения и данных от влияющих факторов;
  - описание интерфейсов пользователя, всех меню и диалогов;
  - описание хранимых или передаваемых наборов данных;
  - руководство пользователя на сертифицируемое программное обеспечение;
  - характеристики необходимых системных и аппаратных средств, если эта информация не приведена в руководстве пользователя.
- Перечень документов, сопровождающих программное обеспечение, может корректироваться соглашением между исполнителем и заказчиком сертификации ПО.

Приложение 1. Бланк заявки на проведение сертификации

Приложение 2. Бланк договора на проведение сертификации

### **Контрольные вопросы**

1. Что такое сертификация программной продукции?
2. Что означает термин "программная продукция" и почему говорится о сертификации программной продукции, а не программных средств или программ?
3. Является ли сертификация программной продукции обязательной?
4. Что относится к нормативным документам, на соответствие которым проводится сертификация?
5. Какие материалы нужно представить на сертификацию и кто это может сделать?
6. Что в заявке на сертификацию означает фраза "Схема сертификации №3"?
7. Что включает в себя процесс сертификации?
8. Как проводится проверка соответствия ПС разделам и пунктам нормативных документов?
9. Срок действия сертификата соответствия?
10. В каких случаях сертификат приостанавливается или отменяется?
11. Какие виды программных средств могут быть сертифицированы в ОС ПС?
12. Сколько продолжается процесс сертификации?
13. Чем сертификат отличается от лицензии?
14. Какая копия сертификата считается действительной?
15. Что означает «Сертифицируемый объем ПС», упомянутый в заявке?
16. Может ли быть сертифицировано зарубежное программное средство?

### **Практическое занятие №2.**

**Тема:** Разработка технического задания на создание программного средства.

**Цель:** Ознакомиться с правилами написания технического задания.

#### **Ход занятия:**

1. Изучить теоретический материал по заданной теме.
2. Заполнить шаблоны технического задания на разработку приложения и на разработку сайта.
3. Оформить отчет по практическому занятию. В отчет включить заполненные документы и ответы на контрольные вопросы.

## Общие сведения

1. Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата А4 и А3 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляют в верхней части листа над текстом.
2. Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78. Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.
3. Для внесения изменений и дополнений в техническое задание на последующих стадиях разработки программы или программного изделия выпускают дополнение к нему. Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.
4. Техническое задание должно содержать следующие разделы:
  - название программы и область применения;
  - основание для разработки;
  - назначение разработки;
  - технические требования к программе или программному изделию;
  - технико-экономические показатели;
  - стадии и этапы разработки;
  - порядок контроля и приемки;
  - приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.
5. Содержание разделов
  - 5.1. В разделе «Наименование и область применения» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.
  - 5.2. В разделе «Основание для разработки» должны быть указаны:
    - документ (документы), на основании которых ведется разработка;
    - организация, утвердившая этот документ, и дата его утверждения;
    - наименование и (или) условное обозначение темы разработки.
  - 5.3. В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.
  - 5.4. Раздел «Технические требования к программе или программному изделию» должен содержать следующие подразделы:
    - требования к функциональным характеристикам;
    - требования к надежности;
    - условия эксплуатации;
    - требования к составу и параметрам технических средств;
    - требования к информационной и программной совместимости;
    - требования к маркировке и упаковке;
    - требования к транспортированию и хранению;
    - специальные требования.
  - 5.5. В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.
  - 5.6. В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т. п.).

5.7. В подразделе «Условия эксплуатации» должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т. п. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.

5.8. В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их технических характеристик.

5.9. В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования. При необходимости должна обеспечиваться защита информации и программ.

5.10. В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.

5.11. В подразделе «Требования к транспортированию и хранению» должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.

5.12. В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

5.13. В разделе «Стадии и этапы разработки\*» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

5.14. В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

5.15. В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

#### **Пример разработки технического задания на разработку приложения.**

##### **Введение.**

Работа выполняется в рамках проекта «Автоматизированная система оперативно-диспетчерского управления электро-, теплоснабжением корпусов института».

##### **Основание для разработки**

1. Основанием для данной работы служит договор № 1234 от 10 января 2015 г.

2. Наименование работы:

«Модуль автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов института».

Исполнители: ОАО «Лаборатория создания программного обеспечения».

Соисполнители: нет.

##### **Назначение разработки**

Создание модуля для контроля и оперативной корректировки состояния основных параметров теплообеспечения корпусов Московского института.

##### **Технические требования**

1. Требования к функциональным характеристикам.

1.1. Состав выполняемых функций. Разрабатываемое ПО должно обеспечивать:

- сбор и анализ информации о расходовании тепла, горячей и холодной воды по данным теплосчетчиков SA-94 на всех тепловых выходах.;
- сбор и анализ информации с устройств управления системами воздушного отопления и кондиционирования типа РТ1 и РТ2 (разработки кафедры СММЭ и ТЦ);

- предварительный анализ информации на предмет нахождения параметров в допустимых пределах и сигнализирование при выходе параметров за пределы допуска;
- выдачу рекомендаций по дальнейшей работе;
- отображение текущего состояния по набору параметров - циклически постоянно (режим работы круглосуточный), при сохранении периодичности контроля прочих параметров;
- визуализацию информации по расходу теплоносителя:
  - текущую, аналогично показаниям счетчиков;
  - с накоплением за прошедшие сутки, неделю, месяц - в виде почасового графика для информации за сутки и неделю;
  - суточный расход - для информации за месяц.

Для устройств управления приточной вентиляцией текущая информация должна содержать номер приточной системы и все параметры, выдаваемые на собственный индикатор.

По отдельному запросу осуществляются внутренние настройки.

В конце отчетного периода система должна архивировать данные.

## 1.2. Организация входных и выходных данных.

Исходные данные в систему поступают в виде значений с датчиков, установленных в помещениях института. Эти значения отображаются на компьютере диспетчера. После анализа поступившей информации оператор диспетчерского пункта устанавливает необходимые параметры для устройств, регулирующих отопление и вентиляцию в помещениях. Возможна также автоматическая установка некоторых параметров для устройств регулирования.

Основной режим использования системы - ежедневная работа.

## 2. Требования к надежности.

Для обеспечения надежности необходимо проверять корректность получаемых данных с датчиков.

## 3. Условия эксплуатации и требования к составу и параметрам технических средств.

Для работы системы должен быть выделен ответственный оператор.

Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

## 4. Требования к информационной и программной совместимости.

Программа должна работать на платформах Windows XP/ NT/2000.

## 5. Требования к транспортировке и хранению. Программа поставляется на лазерном носителе информации.

Программная документация поставляется в электронном и печатном виде.

## 6. Специальные требования.

Программное обеспечение должно иметь дружелюбный интерфейс, рассчитанный на пользователя (в плане компьютерной грамотности) средней квалификации.

Ввиду объемности проекта задачи предполагается решать поэтапно, при этом модули ПО, созданные в разное время, должны предполагать возможность наращивания системы и быть совместимы друг с другом, поэтому документация на принятое эксплуатационное ПО должна содержать полную информацию, необходимую для работы программистов с ним.

Язык программирования – по выбору исполнителя, должен обеспечивать возможность интеграции программного обеспечения с некоторыми видами периферийного оборудования (например, счетчик SA-94 и т. П.).

### Требования к программной документации

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой Системы Программной Документации (ЕСПД); руководство пользователя, руководство администратора, описание применения.

Технико-экономические показатели

Эффективность системы определяется удобством использования системы для контроля и управления основными параметрами теплообеспечения помещений института, а также экономической выгодой, полученной от внедрения аппаратно-программного комплекса.

#### Порядок контроля и приемки

После передачи Исполнителем отдельного функционального модуля программы Заказчику, последний имеет право тестировать модуль в течение 7 дней. После тестирования Заказчик должен принять работу по данному этапу или в письменном виде изложить причину отказа от принятия. В случае обоснованного отказа Исполнитель обязуется доработать модуль.

#### Календарный план работ.

№ этапа	Название этапа	Сроки этапа	Чем закрывается этап
1.	Изучение предметной области. Проектирование системы. Разработка предложений по реализации системы.	01.02.2001 - 28.02.2001	Предложения по работе системы. Акт сдачи-приёмки.
2.	Разработка программного модуля по сбору и анализу информации со счётчиков и устройств управления. Внедрение системы для одного из корпусов.	01.03.2001 - 31.08.2001	Программный комплекс.
3.	Тестирование и отладка модуля. Внедрение системы во всех корпусах.	01.09.2001 - 30.12.2001	Готовая система контроля теплоснабжения, установленная в диспетчерском пункте. Программная документация. Акт сдачи-приёма работ.

Руководитель работ

Сидоров А.В.

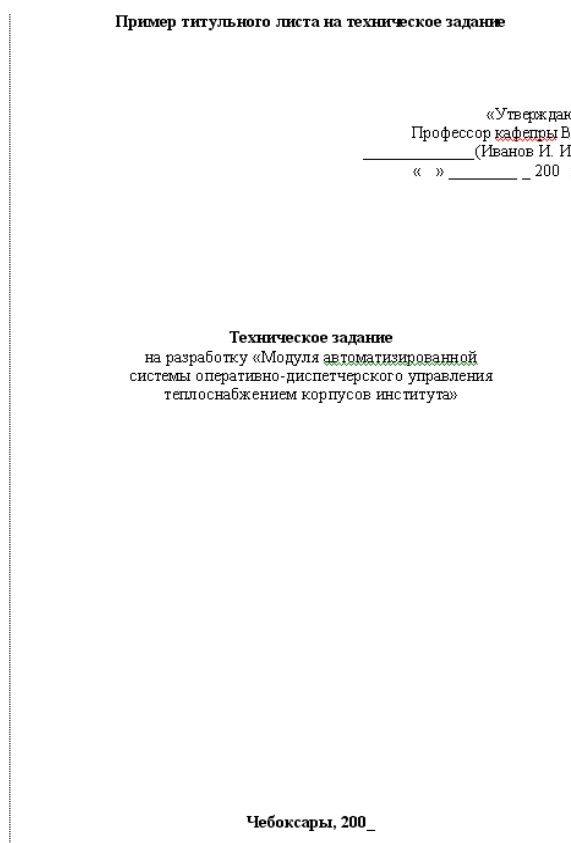


Рис. 1.1. Пример титульного листа на техническое задание

### **Индивидуальные задания:**

Разработайте техническое задание на создание программного продукта по всем требованиям в соответствии со своим вариантом:

1. Разработка программного комплекса «Автотранспорт».
2. Разработка программного комплекса «Деканат института».
3. Разработка программного комплекса «Обслуживание банкомата».
4. Разработка программного комплекса «Управление гостиницей».
5. Разработка программного комплекса «Выдача кредитов в банке».
6. Разработка программного комплекса «Строительная фирма».
7. Разработка программного комплекса «Управление библиотечным фондом».
8. Разработка программного комплекса «АРМ работника склада»
9. Разработка программного комплекса «АРМ администратора ателье по ремонту оргтехники»
10. Разработка программного комплекса «АРМ администратора автосалона».
11. Разработка программного комплекса «АРМ администратора ресторана».
12. Разработка программного комплекса «АРМ сотрудника ЖЭСа».
13. Разработка программного комплекса «АРМ администратора аэропорта».
14. Разработка программного комплекса «АРМ работника отдела кадров».
15. Разработка программного комплекса «АРМ администратора спорткомплекса».

### **Практическое занятие №3.**

**Тема:** Разработка технологической документации на программное средство.

**Цель:** Изучить правила оформления технологической документации на программное средство.

#### **Задание:**

1. Изучить документ ГОСТ 19.101-77 Виды программ и программных документов.



2. Изучить документ ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению
3. Изучить правила оформления технологической документации на программное средство.
  4. Оформить пояснительную записку для программного средства согласно варианту:
    - 1) Разработка программного комплекса «Автотранспорт».
    - 2) Разработка программного комплекса «Деканат института».
    - 3) Разработка программного комплекса «Обслуживание банкомата».
    - 4) Разработка программного комплекса «Управление гостиницей».
    - 5) Разработка программного комплекса «Выдача кредитов в банке».
    - 6) Разработка программного комплекса «Строительная фирма».
    - 7) Разработка программного комплекса «Управление библиотечным фондом».
    - 8) Разработка программного комплекса «АРМ работника склада»
    - 9) Разработка программного комплекса «АРМ администратора ателье по ремонту оргтехники»
    - 10) Разработка программного комплекса «АРМ администратора автосалона».
    - 11) Разработка программного комплекса «АРМ администратора ресторана».
    - 12) Разработка программного комплекса «АРМ сотрудника ЖЭСа».
    - 13) Разработка программного комплекса «АРМ администратора аэропорта».
    - 14) Разработка программного комплекса «АРМ работника отдела кадров».
    - 15) Разработка программного комплекса «АРМ администратора спорткомплекса».

### **Общие сведения**

#### **Документация, создаваемая и используемая в процессе разработки программных средств**

При разработке ПС создается и используется большой объем разнообразной документации. Она необходима как средство передачи информации между разработчиками ПС, как средство управления разработкой ПС и как средство передачи пользователям информации, необходимой для применения и сопровождения ПС. На создание этой документации приходится большая доля стоимости ПС.

Эту документацию можно разбить на две группы:

- Документы управления разработкой ПС.
- Документы, входящие в состав ПС.

*Документы управления разработкой ПС (software process documentation)* управляют и протоколируют процессы разработки и сопровождения ПС, обеспечивая связи внутри коллектива разработчиков ПС и между коллективом разработчиков и *менеджерами ПС (software managers)* - лицами, управляющими разработкой ПС. Эти документы могут быть следующих типов:

- *Планы, оценки, расписания.* Эти документы создаются менеджерами для прогнозирования и управления процессами разработки и сопровождения ПС.
- *Отчеты об использовании ресурсов в процессе разработки.* Создаются менеджерами.
- *Стандарты.* Эти документы предписывают разработчикам, каким принципам, правилам, соглашениям они должны следовать в процессе разработки ПС. Эти стандарты могут быть как международными или национальными, так и специально созданными для организации, в которой ведется разработка ПС.
- *Рабочие документы.* Это основные технические документы, обеспечивающие связь между разработчиками. Они содержат фиксацию идей и проблем, возникающих в процессе разработки, описание используемых стратегий и подходов, а также рабочие (временные) версии документов, которые должны войти в ПС.
- *Заметки и переписка.* Эти документы фиксируют различные детали взаимодействия между менеджерами и разработчиками.

*Документы, входящие в состав ПС (software product documentation)*, описывают программы ПС как с точки зрения их применения пользователями, так и с точки зрения их разработчиков и сопровождаемых (в соответствии с назначением ПС). Здесь следует отметить, что эти документы будут использоваться не только на стадии эксплуатации ПС (в ее фазах применения и сопровождения), но и на стадии разработки для управления процессом разработки (вместе с рабочими документами) - во всяком случае, они должны быть проверены (протестированы) на соответствие программам ПС. Эти документы образуют два комплекта с разным назначением:

- Пользовательская документация ПС (П-документация).
- Документация по сопровождению ПС (С-документация).

### **Пользовательская документация программных средств**

*Пользовательская документация ПС (user documentation)* объясняет пользователям, как они должны действовать, чтобы применить разрабатываемое ПС. Она необходима, если ПС предполагает какое-либо взаимодействие с пользователями. К такой документации относятся документы, которыми должен руководствоваться пользователь при *инсталляции* ПС (при установке ПС с соответствующей настройкой на среду применения ПС), при применении ПС для решения своих задач и при управлении ПС (например, когда разрабатываемое ПС будет взаимодействовать с другими системами). Эти документы частично затрагивают вопросы сопровождения ПС, но не касаются вопросов, связанных с модификацией программ.

В связи с этим следует различать две категории пользователей ПС: ординарных пользователей ПС и администраторов ПС.

*Ординарный пользователь ПС (end-user)* использует ПС для решения своих задач (в своей предметной области). Это может быть инженер, проектирующий техническое устройство, или кассир, продающий железнодорожные билеты с помощью ПС. Он может и не знать многих деталей работы компьютера или принципов программирования.

*Администратор ПС (system administrator)* управляет использованием ПС ординарными пользователями и осуществляет сопровождение ПС, не связанное с модификацией программ. Например, он может регулировать права доступа к ПС между ординарными пользователями, поддерживать связь с поставщиками ПС или выполнять определенные действия, чтобы поддерживать ПС в рабочем состоянии, если оно включено как часть в другую систему.

Состав пользовательской документации зависит от аудиторий пользователей, на которые ориентировано разрабатываемое ПС, и от режима использования документов. Под *аудиторией* здесь понимается контингент пользователей ПС, у которого есть необходимость в определенной пользовательской документации ПС. Удачный пользовательский документ существенно зависит от точного определения аудитории, для которой он предназначен. Пользовательская документация должна содержать информацию, необходимую для каждой аудитории. Под *режимом использования* документа понимается способ, определяющий, каким образом используется этот документ. Обычно пользователю достаточно больших программных систем требуются либо документы для изучения ПС (использование в виде *инструкции*), либо для уточнения некоторой информации (использование в виде *справочника*).

Можно считать типичным следующий состав пользовательской документации для достаточно больших ПС:

- *Общее функциональное описание ПС.* Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС.

- *Руководство по установке ПС.* Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.
- *Инструкция по применению ПС.* Предназначена для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для ее изучения.
- *Справочник по применению ПС.* Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
- *Руководство по управлению ПС.* Предназначено для администраторов ПС. Оно должно описывать сообщения, генерируемые, когда ПС взаимодействует с другими системами, и как должен реагировать администратор на эти сообщения. Кроме того, если ПС использует системную аппаратуру, этот документ может объяснять, как сопровождать эту аппаратуру.

Разработка пользовательской документации начинается сразу после создания внешнего описания. Качество этой документации может существенно определять успех ПС. Она должна быть достаточно проста и удобна для пользователя (в противном случае это ПС, вообще, не стоило создавать). Поэтому, хотя черновые варианты (наброски) пользовательских документов создаются основными разработчиками ПС, к созданию их окончательных вариантов часто привлекаются профессиональные технические писатели. Кроме того, для обеспечения качества пользовательской документации разработан ряд стандартов, в которых предписывается порядок разработки этой документации, формулируются требования к каждому виду пользовательских документов и определяются их структура и содержание.

### **Документация по сопровождению программных средств**

*Документация по сопровождению ПС (system documentation)* описывает ПС с точки зрения ее разработки. Эта документация необходима, если ПС предполагает изучение того, как оно устроена (сконструирована), и модернизацию его программ. Как уже отмечалось, сопровождение - это продолжающаяся разработка. Поэтому в случае необходимости модернизации ПС к этой работе привлекается специальная команда разработчиков-сопроводителей. Этой команде придется иметь дело с такой же документацией, которая определяла деятельность команды первоначальных (основных) разработчиков ПС, - с той лишь разницей, что эта документация для команды разработчиков-сопроводителей будет, как правило, чужой (она создавалась другой командой). Чтобы понять строение и процесс разработки модернизируемого ПС, команда разработчиков-сопроводителей должна изучить эту документацию, а затем внести в нее необходимые изменения, повторяя в значительной степени технологические процессы, с помощью которых создавалось первоначальное ПС.

Документация по сопровождению ПС можно разбить на две группы:

1. документация, определяющая строение программ и структур данных ПС и технологию их разработки;
2. документацию, помогающую вносить изменения в ПС.

Документация первой группы содержит итоговые документы каждого технологического этапа разработки ПС. Она включает следующие документы:

- Внешнее описание ПС (Requirements document).
- Описание архитектуры ПС (description of the system architecture), включая внешнюю спецификацию каждой ее программы (подсистемы).
- Для каждой программы ПС - описание ее модульной структуры, включая внешнюю спецификацию каждого включенного в нее модуля.
- Для каждого модуля - его спецификация и описание его строения (design description).
- Тексты модулей на выбранном языке программирования (program source code listings).
- Документы установления достоверности ПС (validation documents), описывающие, как устанавливалась достоверность каждой программы ПС и как информация об установлении достоверности связывалась с требованиями к ПС.

Документы установления достоверности ПС включают, прежде всего, документацию по тестированию (схема тестирования и описание комплекта тестов), но могут включать и результаты других видов проверки ПС, например, доказательства свойств программ. Для обеспечения приемлемого качества этой документации полезно следовать общепринятым рекомендациям и стандартам.

Документация второй группы содержит

- *Руководство по сопровождению ПС (system maintenance guide)*, которое описывает особенности реализации ПС (в частности, трудности, которые пришлось преодолевать) и как учтены возможности развития ПС в его строении (конструкции). В нем также фиксируются, какие части ПС являются аппаратно- и программно-зависимыми.

Общая проблема сопровождения ПС - обеспечить, чтобы все его представления шли в ногу (оставались согласованными), когда ПС изменяется. Чтобы этому помочь, связи и зависимости между документами и их частями должны быть отражены в руководстве по сопровождению, и зафиксированы в базе данных управления конфигурацией.

#### **Контрольные вопросы**

1. Что такое менеджер программного средства?
2. Что такое ординарный пользователь программного средства?
3. Что такое администратор программного средства?
4. Что такое руководство по инсталляции программного средства?
5. Что такое руководство по управлению программным средством?
6. Что такое руководство по сопровождению программного средства?

#### **Практическое занятие №4.**

**Тема:** Разработка эксплуатационной документации на программное средство.

**Цель:** Изучить правила оформления эксплуатационной документации на программное средство.

#### **Задание:**

1. Изучить документ ГОСТ 19.502-78 Описание применения. Требования к содержанию и оформлению
2. Изучить документ ГОСТ 19.503-79 Руководство системного программиста. Требования к содержанию и оформлению
3. Изучить документ ГОСТ 19.504-79 Руководство программиста. Требования к содержанию и оформлению

4. Изучить документ ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению
5. Изучить правила оформления эксплуатационной документации на программное средство.
6. Разработать руководство программиста и руководство пользователя для программного средства, разработанного в рамках курсовой работы по дисциплине «Технология разработки программного обеспечения».

### 3.1.5 Внеаудиторные самостоятельные работы (пример)

#### Внеаудиторная самостоятельная работа № 1

##### Текст задания

Подготовка реферата по одной из тем:

1. Государственные стандарты РФ (ГОСТ Р)
2. Международный стандарт ISO/IEC 12207: 1995-08-

Время выполнения: 40 минут

##### Проверяемые результаты обучения

Наименование объектов контроля и оценки	Основные показатели оценки результата	Оценка (кол-во баллов)
– У3 Оформлять документацию на программные средства.	Разработка руководств пользователя, программиста, системного администратора	2 балла
– У4 Использовать инструментальные средства для автоматизации оформления документации	Разработка документации с использованием шаблонов стандартных документов на ПО	

За выполнение задания выставляется – 2 балла.

За невыполнение задания – 0 баллов.

### 3.1.6 Задания для получения допуска к сдаче дифференцированного зачёта по МДК .03.03

Задание 1. Постановка задачи

Вариант 1

Разработать техническое задание на программный продукт «Текстовый редактор»

Вариант 2

Разработать техническое задание на программный продукт «Программа проверки знаний (тестирование по заданной теме)»

Вариант 3

Разработать техническое задание на web-сайт «Магазин автозапчастей»

Задание 2. Разработать документацию к оконному приложению.

Критерии оценки программного продукта согласно показателям качества:

1. Показатели надежности программного продукта:
  - устойчивость функционирования;
  - работоспособность.
2. Показатели сопровождения:
  - структурность.
3. Простота конструкции.
4. Наглядность.
5. Повторяемость.

6. Показатели удобства применения.

7. Легкость освоения.

Критерии оценки:

«5»	«4»	«3»	«2»
Созданный программный продукт разработан в полном соответствии с показателями качества.	Созданный программный продукт имеет несоответствие по одному из показателей: простота конструкции.	Созданный программный продукт имеет несоответствие по двум показателям: простота конструкции, показатели удобства применения.	Созданный программный продукт не соответствует более 2 показателям.

### 3.2. Задания для оценки освоения МДК

#### 3.2.1. Задания для оценки освоения МДК.03.01. Технология разработки программного обеспечения.

##### Типовой тест к дифференцированному зачету

##### Раздел 1. «Общие принципы разработки программных продуктов»

1. Программный продукт – это
  - 1) программа для удовлетворения нужд разработчиков, предназначенная для продажи
  - 2) комплекс взаимосвязанных программ для решения определенной проблемы массового спроса, подготовленный к реализации как любой вид промышленной продукции
  - 3) программная реализация решения задачи на компьютере
  - 4) результат разработки какого-либо технического задания
2. Отличительной особенностью программных продуктов является
  - 1) системность
  - 2) простота
  - 3) универсальность
  - 4) надежность
3. Сопровождение программного продукта – это
  - 1) снабжение программного продукта необходимой документацией
  - 2) обнаружение и исправление ошибок
  - 3) поддержка работоспособности программного продукта, переход на его новые версии, внесение изменений, исправление обнаруженных ошибок и т.д.
  - 4) проверка работоспособности каждой разработанной функции, процедуры, модуля
4. Мобильность программных продуктов – это
  - 1) независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.
  - 2) точность выполнения предписанных функций обработки
  - 3) способность к внесению изменений

- 4) обеспечение дружелюбного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства
5. В условиях существования рынка программных продуктов важными его характеристиками являются:
  - 1) количество продаж, наличие программ-конкурентов, длительность продаж
  - 2) стоимость, количество продаж, время нахождения на рынке, известность фирмы-разработчика и программы
  - 3) внешний интерфейс программы, количество продаж, наличие программ-конкурентов
  - 4) модифицируемость, надежность, универсальность, известность фирмы - разработчика
6. Утилитарные программы выполняют роль...
  - 1) сервиса
  - 2) клиента
  - 3) сервера
  - 4) программного средства разработки приложений
7. Основными показателями качества программных продуктов является:
  - 1) алгоритмическая сложность, полнота и системность функций обработки, объем файлов программы
  - 2) стоимость, количество продаж, наличие программных продуктов аналогичного назначения
  - 3) мобильность, надежность, эффективность, модифицируемость, коммуникативность, учет человеческого фактора
  - 4) модифицируемость, надежность, наличие программных продуктов аналогичного назначения
8. Функциональные задачи – это
  - 1) задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей
  - 2) основа для разработки сервисных средств ПО (утилиты, библиотеки)
  - 3) совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей
  - 4) задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере
9. Алгоритм – это
  - 1) комплекс математических вычислений для решения задачи
  - 2) последовательность команд, предназначенных для решения задач
  - 3) программная реализация на компьютере решения определенной задачи
  - 4) результат интеллектуального труда, для которого характерно творчество
10. При индивидуальной разработке фирма-разработчик создает программный продукт для...
  - 1) конкретного заказчика
  - 2) массового использования
  - 3) внедрения в специальные организации
  - 4) для удовлетворения собственных нужд
11. Модифицируемость программных продуктов означает...
  - 1) независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.
  - 2) точность выполнения предписанных функций обработки

- 3) способность к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п.
  - 4) обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства
12. Жизненный цикл программы – это
- 1) временной интервал, начиная с момента замысла программы и кончая прекращением всех видов его пользований
  - 2) временной интервал, начиная с момента введения программы в эксплуатацию
  - 3) промежуток времени, который определяет наиболее эффективное использование создаваемой программы
  - 4) временная характеристика разработки программного продукта
13. Программы малого Жизненного Цикла – это программы
1. когда время разработки программы значительно меньше времени эксплуатации программы
  2. когда время разработки программы значительно больше времени использования программы
  3. когда время разработки программы равно времени эксплуатации программы
  4. нет правильного ответа
14. Выбрать правильный ответ
- a. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, необходимость обеспечения безопасности системы и данных
  - b. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, функции, которые должен выполнять программный продукт
  - c. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, сроки написания программы
  - d. На этапе сбора и анализа требований заказчик должен собрать литературу по разрабатываемому программному продукту
15. Самая распространенная модель Жизненного цикла программного продукта это
- 1) итерационная
  - 2) V - образная
  - 2) спиральная
  - 3) каскадная
16. Классическая модель ЖЦПО характеризуется следующими основными особенностями
1. последовательным выполнением входящих в ее состав этапов
  2. наличием обратных связей между этапами
  3. отсутствием временного перекрытия этапов
  4. отсутствием (или определенным ограничением) возврата к предыдущим этапам
  5. наличием результата после каждого этапа разработки
17. Выберите правильную последовательность этапов спиральной модели жизненного цикла программного продукта:
- 1) техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка
  - 2) кодирование, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка
  - 3) кодирование, техническое проектирование, уточнение функциональных характеристик, тестирование и отладка



- 4) определение требований, анализ, реализация и тестирование, внедрение
18. V – образная модель ЖЦ разработки ПО предполагает:
1. отсутствие временного перекрытия этапов
  2. наличие обратной связи
  3. возможность сокращения времени разработки ПО
  4. возможность увеличения жизненного цикла программного продукта
19. На втором этапе каскадной модели ЖЦ разработки ПО (Требования ПО) осуществляется...
1. составление концептуальной структуры системы
  2. определение функциональности программного компонента
  3. составление детальной спецификации архитектуры системы
  4. составление набора тестовых данных
20. Происходит ли интеграция отдельных компонент системы при разработке ПП по каскадной модели?
1. да
  2. нет
21. Проверка корректности требований при использовании V – образной модели ЖЦ разработки ПО осуществляется...
1. после каждого этапа разработки
  2. после разработки всей системы
  3. после разработки черновой версии системы
  4. после разработки набора тестовых данных
22. Выберите правильную последовательность этапов жизненного цикла программного продукта:
- 1) техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка
  - 2) сбор и анализ требований, проектирование системы, кодирование, создание программной документации, сопровождение
  - 3) кодирование, сбор и анализ требований заказчика, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка
  - 4) сбор и анализ требований заказчика, уточнение функциональных характеристик, техническое проектирование, кодирование, тестирование и отладка, сопровождение ПП
23. Во вспомогательные процессы ЖЦ программного продукта входит:
- 1) документирование, верификация, аттестация, обеспечение качества, совместная оценка, разрешение проблем, аудит
  - 2) управление, создание инфраструктуры, усовершенствование, обучение
  - 3) разработка, приобретение, поставка, эксплуатация, сопровождение
  - 4) кодирование, тестирование, сопровождение
24. Метод получения информации при проектировании программного продукта путем анализа материала подразумевает:
- 1) изучение материала, с которым будет осуществляться работа с использованием данного ПП
  - 2) изучение работы одного из исполнителей с учетом того, что другие исполнители будут выполнять те же действия и операции
  - 3) накопление опыта разработки программного продукта
  - 4) накопление информации в том случае, если были получены противоречивые сведения
25. Одним из достоинств классического жизненного цикла программного продукта является

- 1) дает план и временной график по всем этапам проекта
  - 2) в конце всей работы заказчику будут доступны результаты проекта
  - 3) системный анализ каждого элемента программы
  - 4) отсутствие временного перекрытия этапов разработки программного продукта
26. Итерационная модель ЖЦПО характеризуется следующими основными особенностями:
1. последовательным выполнением входящих в ее состав этапов
  2. наличием обратных связей между этапами
  3. отсутствием временного перекрытия этапов
  4. отсутствием (или определенным ограничением) возврата к предыдущим этапам
  5. возможность проведение корректировки после каждого этапа
27. В конце каждого витка спирали спиральной модели ЖЦ разработки ПО получаем...
1. готовый программный продукт
  2. одну версию программного продукта
  3. версию программного продукта с набором тестовых данных
  4. черновую модель программного продукта
28. Спиральная модель ЖЦ разработки ПО предполагает:
1. отсутствие временного перекрытия этапов
  2. наличие обратной связи
  3. возможность сокращения времени разработки ПО
29. На втором этапе каскадной модели ЖЦ разработки ПО (Требования к ПО) осуществляется...
1. определение функциональности программного компонента
  2. составление детальной спецификации архитектуры системы
  3. составление концептуальной структуры системы
  4. написание программного кода
30. Происходит ли интеграция отдельных компонент системы при разработке ПП по экстремальной модели ЖЦ?
1. да
  2. нет
31. Какую модель жизненного цикла разработки ПО целесообразнее использовать, если нет четко определенных требований к будущей системе?
- 1) каскадную
  - 2) спиральную
  - 3) V – образную
  - 4) итерационную
32. В каких годах разработана основная часть документов единой системы программной документации?
- 1) 60-70-х гг
  - 2) 70-80-х гг
  - 3) 80-90-х гг
  - 4) 60-90-х гг
33. На сколько частей можно разделить ЕСПД?
- 1) 6
  - 2) 8
  - 3) 5
  - 4) 10
34. Группа «0» ЕСПД указывает на ...
- 1) резервные группы
  - 2) общее положение
  - 3) основополагающие стандарты

- 4) прочие стандарты
35. Обозначение стандарта ЕСПД должно состоять из трех частей, где первое число обозначает...
- 1) присвоение к классу стандартов ЕСПД
  - 2) код классификационной группы стандарта
  - 3) год регистрации стандарта
  - 4) номер документа в реестре
36. Техническое задание – это
- 1) набор правил, по которым строится ПП
  - 2) задание, которое необходимо выполнить на ПК по техническим характеристикам
  - 3) набор правил эксплуатации программного продукта
  - 4) совокупность требований к программным средствам, которые могут использоваться как критерий проверки и приемки разработанного ПП
37. ГОСТ 19.102-77 называется:
- 1) «Правила внесения изменений в программные документы, выполняемые непечатным способом»
  - 2) «Стадии разработки»
  - 3) «Техническое задание. Требования к содержанию и оформлению»
  - 4) «Основные надписи»
38. Техническое задание, как одно из стадий разработки, состоит из трех частей. Каких?
- 1) научно-исследовательская работа, разработка эскизного проекта, разработка технического проекта
  - 2) разработка программной документации, утверждение эскизного и технического проектов, подготовка и передача программы
  - 3) обоснование необходимой разработки программы, научно-исследовательская работа, разработка и утверждение технического задания
  - 4) разработка программной документации, утверждение выбранных методов разработки, утверждение технического проекта, кодирование
39. Одной из составных частей рабочего проекта является
- 1) разработка программ и программной документации
  - 2) разработка технического задания;
  - 3) выбор языка программирования
  - 4) разработка технического проекта
40. Системное программное обеспечение – это
- 1) комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области
  - 2) совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ
  - 3) совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов
41. Предметная (прикладная) область – это
- 1) проблема, подлежащая решению
  - 2) совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей
  - 3) программная реализация решения задачи
  - 4) точная формулировка решения задачи на компьютере с описанием входной и выходной информации
42. Задача – это
- 1) проблема, подлежащая решению

- 2) совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей
  - 3) программная реализация решения задачи
  - 4) точная формулировка решения задачи на компьютере с описанием входной и выходной информации
43. Программное средство - это
- 1) программа для удовлетворения нужд разработчиков, предназначенная для продажи
  - 2) программа, предназначенная для многократного применения на различных объектах и разработанная любым способом
  - 3) программная реализация решения задачи на компьютере
  - 4) результат разработки какого-либо технического задания
44. Технологические задачи – это
- 1) задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей
  - 2) основа для разработки сервисных средств ПО (утилиты, библиотеки)
  - 3) совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей
  - 4) задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере
45. Дайте расшифровку аббревиатуре ЕСПД.
- 1) Единая схема проектирования документов
  - 2) Единая система программной документации
  - 3) Единая схема программных документов
  - 4) Единственная система программной документации

## **Раздел 2. «Методология проектирования программных продуктов»**

1. На современном этапе выделяют 2 основных подхода к проектированию ПП. Какие?
  - 1) структурный и процедурный
  - 2) объектно-ориентированный и структурный
  - 3) метод проектирования Джексона и объектно-ориентированный
  - 4) иерархический и сетевой
2. Методами структурного проектирования являются
  - 1) модульное программирование, нисходящее проектирование, кодирование и тестирование, структурное проектирование;
  - 2) интегрированное и модульное проектирование;
  - 3) функционально – ориентированное и объектно-ориентированное проектирование
  - 4) структурное программирование, модульное проектирование, тестирование и кодирование
3. Что не использует структурный подход проектирования программного продукта?
  - 1) диаграммы декомпозиции
  - 2) интегрированную структуру данных предметной области
  - 3) структурные схемы
  - 4) анализ предметной области
4. Объектно-ориентированный подход проектирования программного продукта основан на:
  - 1) проектировании
  - 2) кодировании и тестировании
  - 3) создании иерархии классов, наследовании свойств объектов и методов их обработки
  - 4) выделении классов объектов

5. Проектирование – это
- 1) итерационный процесс, при помощи которого требования к программным средствам транслируются в инженерное представление
  - 2) процесс построения модели будущего программного средства
  - 3) инженерное представление программного продукта на каком-либо языке программирования
  - 4) представление программного продукта совокупностью объектов и их свойств
6. Дополните фразу: предварительное проектирование программного продукта формирует...
- 1) уточнение абстракций и добавляет подробности алгоритмического уровня
  - 2) абстракцию архитектурного уровня
  - 3) идентификацию подсистемы и определение основных принципов управления подсистемами
  - 4) набор тестовых данных
7. Какие модели можно использовать при структурировании системы?
- 1) модель абстракционной машины, трехуровневую модель, модель хранилища данных, модель клиент-сервер
  - 2) модель событийного управления, модель хранилища данных, модель потока данных, трехуровневую модель
  - 3) модель объекта, модель централизованного управления, модель хранилища данных, модель абстракционной машины
  - 4) модель объекта, модель централизованного управления, модель абстрактной машины
8. Назовите виды моделей управления.
- 1) модель потока данных и модель хранилища данных
  - 2) модель клиент-сервер и модель управления прерываниями
  - 3) модель централизованного и событийного управления
  - 4) модель централизованного и периферийного управления
9. При разбиении программного средства на отдельные модули можно выделить 2 модели:
- 1) модель потока данных и модель событий
  - 2) модель потока данных и модель объекта
  - 3) модель объекта и модель управления
  - 4) модель управления и модель событий
10. В основе модели потока данных лежит –
- 1) сцепление компонентов
  - 2) разделение данных
  - 3) разбиение по функциям
  - 4) выделение отдельных компонентов и их свойств
11. К классическим методам проектирования ПС ориентированных на процедурную реализацию относят:
- 1) структурный метод проектирования и метод проектирования Джексона
  - 2) метод проектирования Джексона и объектно-ориентированный метод проектирования
  - 3) объектно-ориентированный метод проектирования и структурный метод проектирования
  - 4) иерархический и структурный методы
12. Структурный метод проектирования ПП основан на:
- 1) разбиении единой системы на автономные объекты реального мира
  - 2) выделении объектов и их методов
  - 3) разбиении всей программы на отдельные логические части
  - 4) последовательной декомпозиции всей системы на отдельные компоненты

13. Исходными данными для структурного метода проектирования ПП являются
- 1) логические компоненты, составляющие программное средство
  - 2) отдельные компоненты модели анализа программных средств
  - 3) специальные компоненты, имеющие в основе собственный набор данных
  - 4) объекты системы, их свойства и методы
16. Проектирование для потоков типа преобразования состоит из
- 1) 3 уровней
  - 2) 5 уровней
  - 3) 4 уровней
  - 4) 7 уровней
17. Модуль — это...
- 1) самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки автономно от других программ
  - 2) упорядоченный набор команд, обеспечивающий выполнение определенных функций
  - 3) алгоритм построения программного продукта
18. Головной модуль –
- 1) обеспечивает вызов других модулей на обработку
  - 2) управляет запуском программного продукта
  - 3) выполняет функции обработки
  - 4) осуществляет обслуживающие функции
19. Синтаксическая спецификация модуля программного продукта позволяет
- 1) построить на используемом языке программирования синтаксически правильное обращение к модулю
  - 2) описать семантику функций, выполняемых этим модулем по каждому из его входов
  - 3) описать древовидную структуру модуля
20. Метод восходящей разработки программного продукта заключается в
- 1) первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с нижнего уровня
  - 2) первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с головного уровня
  - 3) проектировании всех модулей одновременно и последующая их сборка в единую систему
21. Конструктивный подход в разработке ПП основывается на
- 1) восходящей разработке ПП
  - 2) архитектурной разработке ПП
  - 3) нисходящей разработке ПП
  - 4) является комбинированным способом проектирования
22. Спецификация программного модуля содержит
- 1) общие требования к ПП
  - 2) функциональную и синтаксическую спецификацию
  - 3) требования к каждому модулю ПП
23. Для контроля структуры программы можно использовать
- 1) смежный контроль
  - 2) тестовый контроль
  - 3) итоговый контроль
  - 4) сквозной контроль
24. В основе структуры программного продукта лежит
- 1) функциональная организация продукта и его функций
  - 2) модульная организация продукта и его функций
  - 3) пошаговое программирование всех функций продукта
25. Рабочий модуль

- 1) обеспечивает вызов других модулей на обработку
  - 2) управляет запуском программного продукта
  - 3) выполняют функции обработки
  - a) осуществляют обслуживающие функции
26. Модульную структуру ПП можно представить
- 1) в виде сетевой структуры
  - 2) в виде древовидной структуры
  - 3) в реляционной структуры
27. При нисходящем тестировании первым тестируется
- 1) управляющий модуль
  - 2) головной модуль
  - 3) рабочий модуль
  - 4) исполняемый модуль
28. При конструктивном подходе к разработке ПП головной модуль программируется исходя из
- 1) спецификаций модуля
  - 2) спецификаций программы в целом
  - 3) общих требований к программному продукту
29. В классическом методе нисходящей разработки программного продукта рекомендуется
- 1) сначала запрограммировать весь программный продукт, а затем начинать нисходящее их тестирование
  - 2) сначала все модули разрабатываемой программы запрограммировать, а затем начинать нисходящее их тестирование
  - 3) сначала запрограммировать весь программный продукт, а затем начинать детально прорабатывать каждый модуль
30. Статический контроль состоит в
- 1) мысленном прокручивание структуры программы при выполнении заранее разработанных тестов
  - 2) контроле со стороны разработчиков архитектуры и внешнего описания ПС
  - 3) оценке структуры программы сточки зрения хорошо ли программа разбита на модули
31. Цели структуризации программного продукта.
- 1) контролировать трудозатраты и стоимость проектных работ
  - 2) разрозненное выполнение отдельных функций программного продукта
  - 3) распределить работы по исполнителям, обеспечив приемлемую их загрузку и требуемые сроки разработки программных продуктов
32. Управляющий модуль -
- 1) обеспечивает вызов других модулей на обработку
  - 2) управляет запуском программного продукта
  - 3) выполняют функции обработки
  - 4) осуществляют обслуживающие функции
33. Функциональная спецификация модуля позволяет
- 1) построить на используемом языке программирования синтаксически правильное обращение к модулю
  - 2) описать семантику функций, выполняемых этим модулем по каждому из его входов
  - 3) описать древовидную структуру модуля
34. Метод нисходящей разработки программного продукта заключается в
- 1) проектировании всех модулей одновременно и последующая их сборка в единую систему
  - 2) первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с нижнего уровня

- 3) первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с головного уровня
35. Архитектурный подход в разработке ПП основывается на
- 1) восходящей разработке ПП
  - 2) нисходящей разработке ПП
  - 3) является комбинированным способом проектирования
36. В рамках конструктивного подхода сначала реализуются
- 1) простейшие модули
  - 2) более сложные модули
  - 3) разрабатывается программа целиком
37. Для контроля структуры программы можно использовать
- 1) смежный контроль
  - 2) тестовый контроль
  - 3) модульный контроль
  - 4) сквозной контроль
38. Модули бывают
- 1) главные, вспомогательные, сервисные, управляющие
  - 2) головные, управляющие, рабочие, сервисные
  - 3) главные, побочные, рабочие
  - 4) первичные, вторичные, локальные, глобальные
39. Сервисный модуль
- 1) обеспечивает вызов других модулей на обработку
  - 2) управляет запуском программного продукта
  - 3) выполняет функции обработки
  - 4) осуществляет обслуживающие функции
40. Назовите методы разработки структуры программного продукта.
- 1) нисходящий, восходящий, конструктивный и архитектурный метод разработки
  - 2) разработка сверху вниз и восходящая разработка
  - 3) нисходящая, восходящая, детальная разработка
  - 4) конструктивна, восходящая, нисходящая разработка
41. При восходящем тестировании программного продукта для каждого модуля необходимо
- 1) создавать отдельный набор тестирующих данных и проверять взаимодействие модуля с остальными модулями и головной программой
  - 2) создавать ведущую программу, которая должна подготовить для тестируемого модуля необходимое состояние информационной среды и произвести требуемое обращение к нему
  - 3) создавать дополнительный модуль, который тестирует исходный модуль
42. Важным при архитектурном подходе проектирования программного продукта является
- 1) разработка конкретной программы
  - 2) разработка конкретной функции программы
  - 3) повышение уровня используемого языка программирования
  - 4) возможность разработки архитектуры программного продукта после проектирования основных модулей
43. Достоинством целенаправленной конструктивной реализации является то,
- 1) что на ранней стадии создается протестированный программный продукт
  - 2) что уже на ранней стадии создается работающий вариант разрабатываемой программы
  - 3) что ПП создается поэтапно
44. Сквозной контроль – это



- 1) мысленное прокручивание структуры программы при выполнении заранее разработанных тестов
  - 2) контроль со стороны разработчиков архитектуры и внешнего описания ПС
  - 3) проверка каждого действия разработчика
  - 4) проверка структуры программы, набора разработанных тестов и последующее применение тестов к программе
45. Какую связь устанавливает пользовательский интерфейс?
- 1) серверную
  - 2) доменную
  - 3) клиент-серверную
  - 4) клиентскую
46. Выберите правильную характеристику системного программиста.
- 1) системные программисты знают тонкости построения ПС и могут ее модифицировать
  - 2) системные программисты умеют строить алгоритмы и на основе их реализовывать программу
  - 3) системные программисты способны разрабатывать базовые методы и средства оснащения ПО
  - 4) системные программисты не имеют опыт работы с ПК
47. Каким базовым требованиям должен отвечать интерфейс, чтобы он был удобным?
- 1) понятность, эстетичность, устойчивость к неполадкам
  - 2) простота, надежность, предсказуемость, адаптивность, стандартность
  - 3) легкая восприимчивость пользователями, несхожесть с другими интерфейсами
  - 4) предсказуемость, модифицируемость, адаптивность, эффективность
48. Интерактивный режим работы с программами заключается в том, что...
- 1) действия пользователей ограничивает задание
  - 2) пользователь вводит команды и получает результат
  - 3) оба варианта правильны
49. Одной из особенностей диалогового режима работы программы является:
- 1) предназначен для непрограммистов
  - 2) широко применяется в современных ИС
  - 3) программа занимает мало ресурса, в том числе и памяти
  - 4) применяется в MSDOS
50. Пакетный режим работы программ предназначен для...
- 1) работы с самой программой
  - 2) коммерческих целей
  - 3) работы системных программистов и администраторов сети

### **Раздел 3. «Разработка программных продуктов»**

1. Язык программирования – это...
  - 1) некоторая структура, в которой указаны правила синтаксиса и семантики
  - 2) множество текстов некоторого алфавита, удовлетворяющих правилам синтаксиса и семантики
  - 3) шаблон построения программных продуктов, с заранее указанными правилами синтаксиса и семантики
  - 4) совокупность правил описания алгоритма
2. Алфавит языка программирования включает в себя
  - 1) буквы и специальные знаки
  - 2) зарезервированные слова
  - 3) оба варианта правильны
3. Семантика языка программирования – это...

- 1) правила, определяющие какие операции, и в какой последовательности должна выполнять ЭВМ
  - 2) правила, определяющие какие операции должна выполнять ЭВМ
  - 3) набор правил, которым должна удовлетворять любая программа
  - 4) описание допустимых элементов, используемых языком программирования
4. Абстракция объектов –
- 1) представление данных, позволяющее использовать их, не зная подробности их создания и организации
  - 2) происходит в виде процедур, функций, когда фрагмент программы может быть использован другими пользователями, знающими данный фрагмент
  - 3) правильного варианта нет
5. Одной из сравнительных характеристик языка программирования является уровень языка, который характеризуется
- 1) разнообразием создаваемых программ
  - 2) сложностью задач с использованием данного языка программирования
  - 3) свойствами совокупности понятий, служащих для описания данного языка программирования
  - 4) сложностью тестирования задач, разработанных на данном языке программирования
6. Гибкость языка программирования обеспечивает –
- 1) описание задачи с использованием определенной предметной области
  - 2) легкость выражения данным языке программирования, необходимое для решения задачи
  - 3) независимость языка от других программных и аппаратных средств
  - 4) возможность тестирования программы на другом языке программирования
7. Полнота языка программирования обеспечивает –
- 1) легкость восприятия пользователем программы
  - 2) легкость понимания семантики языков конструкций
  - 3) простоту разработки программного продукта
  - 4) описание задачи с использованием определенной предметной области
8. Что такое консольное приложение?
- 1) это монитор и клавиатура, рассматриваемые как единое устройство
  - 2) устройством вывода
  - 3) программа, предназначенная для работы в операционной системе MS-DOS
  - 4) обычное приложение
9. Модульная программа – это программа, в которой...
- 1) модульную часть логической структуры можно изменить, не внося изменений в основную часть программы
  - 2) модульную часть программы можно оставить без изменений, а внести изменения в саму программу
  - 3) модульная часть функционирует независимо от программы
10. Модуль – это
- 1) отдельная единица, которая представляет собой «шаблон» для построения программных продуктов
  - 2) отдельно компилируемая часть программы
  - 3) одна из составляющих программного средства, которая функционирует при подключении к программе
  - 4) именованная часть программы, которая создается с использованием языка программирования и его настроек
11. Основным свойством модуля является:
- 1) надежность
  - 2) независимость

- 3) взаимодействие с программными средствами
  - 4) наличие внешних связей
12. Логика модуля означает
- 1) что делает модуль
  - 2) как модуль связан с другими модулями программы
  - 3) наличие внутренних и внешних потоков информации
  - 4) как реализован модуль
13. Внутренняя связь модуля характеризуется...
- 1) понятием прочности модуля
  - 2) понятием надежности модуля
  - 3) понятием взаимозависимости модуля
  - 4) логикой модуля
14. К уровням прочности модуля можно отнести:
- 1) прочность по совпадению, процедурно прочные модули, функционально прочные модули
  - 2) прочность по классу, информационные модули
  - 3) оба варианта правильны
  - 4) прочность по объектам и используемым локальным и глобальным переменным
15. Внешние связи характеризуются степенью...
- 1) независимости модуля по отношению к данным
  - 2) зависимости модуля по отношению к данным
  - 3) прочностью модуля по отношению к данным
16. Загрузочный модуль – это...
- 1) отдельная программа, независима от других выполняемых программ
  - 2) выполняемый \*.exe модуль, фактически отдельная программа
  - 3) обычный текстовый файл с нужным расширением
  - 4) средство языка программирования служащее для увеличения уровня языка программирования
17. Подпрограмма – это...
- 1) средство языка программирования, служащее для увеличения уровня языка программирования
  - 2) набор откомпилированных программ, собранных в специально форматированный файл
  - 3) практическая часть текста программы, которая подставляется при каждом вызове, увеличивая ее размер
  - 4) обычный текстовый файл с нужным расширением
18. Объектный модуль – это...
- 1) отдельная программа, независима от других выполняемых программ
  - 2) модуль специальной структуры, созданный при компиляции программы
  - 3) обычный текстовый файл с нужным расширением
  - 4) средство языка программирования служащее для увеличения уровня языка программирования
19. Загрузочный модуль – это...
- 1) отдельная программа, независима от других выполняемых программ
  - 2) выполняемый \*.exe модуль, фактически отдельная программа
  - 3) обычный текстовый файл с нужным расширением
  - 4) средство языка программирования служащее для увеличения уровня языка программирования
20. Модуль, выполняющий (реализующий) одну какую-либо определенную функцию называется...
- 1) информационно прочным
  - 2) объектным

- 3) независимым
  - 4) функционально прочным
21. Сцепление модуля - это
- 1) мера его зависимости по данным от других модулей
  - 2) набор откомпилированных программ, собранных в специально форматированный файл
  - 3) практическая часть текста программы, которая подставляется при каждом вызове, увеличивая ее размер
  - 4) степень взаимосвязи локальных переменным модуля и глобальных переменных основной программы
22. Сцепление модулей по общей области означает что...
- 1) несколько модулей используют одну и ту же область памяти
  - 2) один из них имеет прямые ссылки на состав другого модуля
  - 3) один модуль имеет ссылку на переменные другого модуля
  - 4) несколько модулей имеют ссылку на переменные другого модуля
23. Сцепление модулей по содержимому означает что...
- 1) несколько модулей используют одну и ту же область памяти
  - 2) несколько модулей имеют ссылку на переменные другого модуля
  - 3) один из них имеет прямые ссылки на состав другого модуля
  - 4) один модуль имеет ссылку на переменные другого модуля
24. Объектно-ориентированная декомпозиция обеспечивает...
- 1) разбиение единой системы на автономные объекты
  - 2) разбиение единой системы на объекты реального мира
  - 3) оба варианта правильны
25. Объектно-ориентированное программирование строится на 3-х принципах:
- 1) на абстрагировании, инкапсуляции, модульности
  - 2) на структурированности, абстрагировании, кодировании
  - 3) на инкапсуляции, логике, модульности
26. Абстракция концентрирует внимание ...
- 1) на внутренние представления объекта
  - 2) на внешние представления объекта
  - 3) на определении способности подвергать декомпозиции
27. Инструментами иерархической структуры ООП программы являются:
- 1) структуры из классов и объектов
  - 2) структуры из подпрограмм и классов
  - 3) структуры из наследования классов
28. Выбрать правильный ответ.
- 1) объект – структурная единица, составляющая в целом программу
  - 2) объект – это именованная часть любой программы
  - 3) объект – это конкретное представление отдельной абстракции
29. Состояние объекта –
- 1) это свойство, отличающее один объект от других
  - 2) характеризуется перечнем функций и свойств данного объекта
  - 3) характеризует то, как данный объект «общается» с другими объектами
  - 4) описывает связь одного объекта программы с другими объектами
30. Оптимизация-это
- 1) улучшение программы
  - 2) увеличение объема программы
  - 3) увеличение количества выполняемых функций
  - 4) уменьшение объема программы и увеличение времени ее выполнения
31. Глобальная оптимизация – это

- 1) оптимизация, при которой экономия времени достигается путём добавления новых циклов
  - 2) оптимизация, при которой экономия ресурсов достигается путём изменения алгоритма всей программы или большей ее части
  - 3) оптимизация, при которой экономия ресурсов достигается путём изменения участка программы
  - 4) способ улучшения программ за счет уменьшения объема памяти отводимой под информационные замены
32. Чистка программы
- 1) в цикле имеются элементы, которые не изменяются при цикле и их выносят за пределы участка цикла
  - 2) способ улучшения программ за счет уменьшения объема памяти отводимой под информационные замены
  - 3) способ повышения качества за счет удаления ненужных объектов и конструкций
  - 4) способ повышения качества программы за счет выполнения определённых вычислений на этапе трансляции
33. Время работы подпрограммы подразумевает
- 1) программу, подлежащую оптимизации следует разделить на подпрограммы и выявить процент общего времени, используемого каждой подпрограммой, количество человеко-часов для оптимизации и % отношение оптимизации подпрограммы
  - 2) определение фактического времени работы каждой подпрограммы
  - 3) вынесение неизменяющихся участков цикла за его пределы
34. Необходимо ли делать проверки внутри цикла
- 1) да
  - 2) нет
35. Оптимизация бывает
- 1) локальная и глобальная
  - 2) локальная и периодическая
  - 3) глобальная и периодическая
  - 4) сетевая и иерархическая
36. Локальная оптимизация – это
- 1) оптимизация, при которой экономия времени достигается путём добавления новых циклов
  - 2) оптимизация, при которой экономия ресурсов достигается путём изменения участка программы
  - 3) оптимизация, при которой экономия ресурсов достигается путём изменения алгоритма всей программы или большей ее части
  - 4) способ улучшения программ за счет уменьшения объема памяти отводимой под информационные замены
37. Реализация действий
- 1) в цикле имеются элементы, которые не изменяются при цикле и их выносят за пределы участка цикла
  - 2) способ улучшения программ за счет уменьшения объема памяти отводимой под информационные замены
  - 3) способ улучшения программы за счет сокращения ее размеров: чистка, запроцедурирование
  - 4) способ повышения качества программы за счет выполнения определённых вычислений на этапе трансляции
38. Сегментация программы подразумевает
- 1) программу, подлежащую оптимизации следует разделить на подпрограммы и выявить процент общего времени, используемого каждой подпрограммой,

количество человеко-часов для оптимизации и % отношение оптимизации подпрограммы

- 2) необходимость установить фактическое время работы каждой подпрограммы
- 3) вынесение неизменяющихся участков цикла за его пределы
- 4) улучшение программы за счет уменьшения объема памяти отводимой под информационные замены

39. Необходимо ли выполнять проверку вводимых пользователем данных

- 1) да
- 2) нет

40. Качество ПП - это

- 1) совокупность свойств этого продукта, которые удовлетворяют определенным потребностям пользователей в соответствии с его назначением;
- 2) те свойства данного продукта, благодаря которым программный продукт может функционировать в любой программной среде;
- 3) совокупность свойств программного продукта, которые удовлетворяют требованиям ЕСПД и базовым международным стандартам.

41. Изучаемость ПП включает в себя:

- 1) удобочитаемость, тестируемость, информативность;
- 2) внедряемость, понятность, удобочитаемость;
- 3) документированность, понятность, удобочитаемость.

42. Функциональная пригодность программного продукта включает в себя:

- 1) точность, защищенность, надежность;
- 2) эффективность и внедряемость;
- 3) понятность, стабильность, надежность.

43. Понятность ПП заключается в ...

- 1) наличию в составе программы информации необходимой и достаточной для понимания назначения программы, существующих ограничений, входных и выходных данных и результатов обработки;
- 2) степени, которой пользователь может изучить назначение ПП, результат ее работы и текст этой программы;
- 3) быстрой модификации с целью приспособления к изменяющимся условиям функционирования.

44. Программа является надежной, если...

- 1) выдаваемый результат работы имеет допустимые значения отклонений от аналогичных отклонений;
- 2) она продолжает свою работу при возникновении сбоев;
- 3) она при всех одинаково вводимых данных обеспечивает полную повторяемость результата.

45. Программа является эффективной, если...

- 1) она правильно работает при любых допустимых вариантах исходных данных;
- 2) объем требуемых ресурсов для ее выполнения не превышает допустимой границы;
- 3) она работает должным образом не только автономно, но и как часть информационной системы.

46. Программа является совместимой, если...

- 1) она работает должным образом не только автономно, но и как часть информационной системы;
- 2) ее качества могут быть продемонстрированы на практике;
- 3) она допускает быструю модификацию с целью приспособления к изменяющимся условиям функционирования.

47. Одним из источников ошибок в программе может быть

- 1) перевод программы;

- 2) взаимопонимание;
  - 3) оба варианта правильны.
48. Сложность программы может заключаться в...
- 1) сложность в построении неформальной модели предметной области;
  - 2) реализации программы, сложность в самой предметной области;
  - 3) сложность в создании загрузочного модуля.
49. Что такое цикл?
- 1) Оператор выбора
  - 2) Оператор условия
  - 3) Оператор повторений
  - 4) Оператор перехода
50. Какой из приведенных ниже циклов, является циклом с параметром?
- 1) WHILE
  - 2) FOR
  - 3) DO
  - 4) REPEAT
51. Какой из приведенных ниже циклов, является циклом с предусловием?
- 1) CASE
  - 2) FOR
  - 3) REPEAT
  - 4) WHILE
52. Какой из приведенных ниже циклов, является циклом с постусловием?
- 1) REPEAT
  - 2) FOR
  - 3) WHILE
  - 4) VAR
53. Консольное приложение запускается клавишами...
- 1) F9
  - 2) CTRL+F9
  - 3) ALT+F9
  - 4) F10
54. В общем виде инструкция обращения к функции выглядит так...
- 1) Переменная := Функция (Параметры)
  - 2) Функция (Параметры):= Переменная
  - 3) Переменная := Функция
  - 4) Функция:= Функция (Параметры)
55. Какое зарезервированное слово обозначает, что переменная именно символьного типа?
- 1) CHR
  - 2) CHAR
  - 3) BOOLEAN
  - 4) SUCC;
56. Какую структуру имеет подпрограмма?
- 1) Только имя подпрограммы
  - 2) Имя и тело подпрограммы
  - 3) Тело подпрограммы
  - 4) Тело подпрограммы и вызов себя самой
57. При составлении блок-схемы для программы, блок-схема подпрограммы оформляется ...
- 1) Отдельно
  - 2) Внутри блок-схемы программы
  - 3) Не учитывается в основной
  - 4) не оформляется блок-схема вообще

58. Сколько подпрограмм можно сделать в программе?
- 1) Только одну
  - 2) Две
  - 3) Сколь необходимо
  - 4) Не больше трех
59. Для вызова подпрограммы необходимо написать...
- 1) Только ее имя
  - 2) Ее имя и фактические параметры
  - 3) Параметры подпрограммы
  - 4) Ее имя и формальные параметры
60. В программе процедуры и функции описываются ...
- 1) вне основной программы
  - 2) сразу после основной программы
  - 3) в разделе описания подпрограмм
  - 4) после слова begin основной программы
61. Что такое алгоритм?
- 1) Последовательность команд исполнителю, выполнение которых приводит к решению поставленной задачи.
  - 2) Последовательность действий для ЭВМ.
  - 3) Последовательность команд для ЭВМ.
  - 4) Решение задачи
62. Способы задания алгоритма?
- 1) Линейный, разветвляющийся
  - 2) Табличный, словесный, блок-схема
  - 3) Табличный, блок-схема
  - 4) Линейный, циклический, разветвляющийся
63. Типы алгоритма?
- 1) Табличный, блок-схема, словесный
  - 2) Линейный, циклический, разветвляющийся
  - 3) Табличный, блок-схема
  - 4) Линейный, циклический,
64. Что такое программа?
- 1) Последовательность действий для решения задачи
  - 2) Последовательность алгоритмов для решения задачи
  - 3) Упорядоченная последовательность действий для ЭВМ, реализующая алгоритм решения кокой - либо задачи
  - 4) Последовательность команд исполнителю, выполнение которых приводит к решению поставленной задачи.
65. Что такое вспомогательный алгоритм?
- 1) Набор формальных и фактических параметров
  - 2) Процедура и функция
  - 3) Алгоритм, помогающий при решении задачи
  - 4) Алгоритм решения некоторой подзадачи из основной решаемой задачи
66. Как выглядят операторские скобки в языке Си?
- 1) ( )
  - 2) { }
  - 3) (\* \*)
  - 4) Begin ...End
67. Общий вид оператора IF?
- 1) If (условие) оператор1 Else оператор2
  - 2) If (оператор1) оператор2 Else условие
  - 3) If ( условие) оператор2 Else оператор1



- 4) If (параметры) условие Else оператор
68. Как выглядят комментарии в программе?
- 1) ( )
  - 2) { }
  - 3) //
69. Что такое комментарий?
- 1) оператор языка программирования
  - 2) пояснения к программе и для программы
  - 3) операторные скобки
  - 4) пояснения к программе для программиста
70. Какого типа данных не существует?
- 1) целочисленного
  - 2) малочисленного
  - 3) литерного
  - 4) булевского
71. Сопоставьте циклы

1.For	a. цикл с последующим условием
2.WHILE	b. цикл с предшествующим условием
3.DO WHILE	c. цикл с параметром

72. Индивидуальность объекта – это...
- 1) перечень всех свойств объекта и текущие значения каждого из этих свойств
  - 2) модель воздействия одного объекта на другие объекты в терминах изменений его состояния и передачи сообщений
  - 3) методы и свободные подпрограммы, ассоциированные с конкретным объектом
  - 4) характеристика объекта, которая отличает его от всех других объектов
73. Поведение объекта – это...
- 1) перечень всех свойств объекта и текущие значения каждого из этих свойств
  - 2) модель воздействия одного объекта на другие объекты в терминах изменений его состояния и передачи сообщений
  - 3) методы и свободные подпрограммы, ассоциированные с конкретным объектом
  - 4) характеристика объекта, которая отличает его от всех других объектов

#### **Раздел 4. «Отладка, тестирование и сопровождение программных продуктов»**

1. Под ошибкой подразумевается
  - 1) место в программе, где искажение проявляется или становится очевидным
  - 2) неправильность, погрешность или неумышленное искажение объекта или процесса
  - 3) место в программе, где возникают условия для появления искажений
  - 4) исправление выявленных искажений в процессе тестирования программы
2. Источником ошибок в программе может быть
  - 1) недостаточная квалификация специалиста
  - 2) сложность программы
  - 3) большой объем программы
  - 4) недостаточное знание заказчиком предметной области
3. Структурный подход к разработке программы является методом борьбы с...
  - 1) переводом программы
  - 2) неквалифицированностью специалиста
  - 3) взаимопониманием
  - 4) сложностью программы
4. Одним из признаков классификации ошибок является
  - 1) уровень сложности и устойчивости
  - 2) степень заикливания

- 3) правильность описания программы
- 4) возможность описания программы
5. Процесс отладки включает следующие подпроцессы:
  - 1) выявление ошибок, диагностика и локализация ошибок, исправление ошибок
  - 2) выявление ошибок и их локализация
  - 3) диагностика ошибок, исправление ошибок и повторное тестирование программы
  - 4) выявление ошибки, исправление ошибки
6. Отладка начинается с того момента как
  - 1) не выдается сообщение об ошибках
  - 2) не выдаются сообщения о синтаксических ошибках
  - 3) программа полностью описана
  - 4) прописаны отдельные модули программы
7. Точка обнаружения – это...
  - 1) место в программе, где ошибка себя проявляет или становится очевидной
  - 2) неправильность, погрешность или неумышленное искажение объекта или процесса
  - 3) место в программе, где ошибку можно локализовать
  - 4) место в программе, где возникают условия для появления ошибки
8. Что может являться источником ошибки в программе?
  - 1) перевод программы
  - 2) недостаточная квалификация специалиста
  - 3) модульное программирование
  - 4) объектно-ориентированное программирование
9. Контроль (проверка, испытания) программы является методом борьбы с...
  - 1) переводом программы
  - 2) взаимопониманием
  - 3) сложностью программы
  - 4) описанием программы
10. Выделяют следующие виды ошибок программ:
  - 1) синтаксические, семантические, первичные
  - 2) ошибки анализа, общего и физического характера
  - 3) ошибки анализа, первичные и вторичные
  - 4) ошибки описания, определения функций и кодирования
11. Под отладкой понимается процесс
  - 1) нахождения и исправления ошибок
  - 2) позволяющий получить программу, которая функционирует с требуемыми характеристиками
  - 3) оптимизации программы
  - 4) тиражирования программы
12. Для тестирования программы используют
  - 1) простые тестовые данные
  - 2) просчитанные данные
  - 3) сложные данные
  - 4) произвольные данные
13. Точка происхождения – это...
  - 1) место в программе, где ошибка себя проявляет или становится очевидной
  - 2) неправильность, погрешность или неумышленное искажение объекта или процесса
  - 3) место в программе, где возникают условия для появления ошибки
  - 4) место в программе, где ошибку можно локализовать
14. Выберите возможные источники ошибки в программе.

- 1) модульное программирование
  - 2) трудность во взаимопонимании между заказчиком и разработчиком
  - 3) сложность понимания языка программирования
  - 4) объектное описание программы
15. Переход на формальные стороны взаимодействия является методом борьбы с...
- 1) переводом программы
  - 2) взаимопониманием
  - 3) сложностью программы
  - 4) пониманием языка программирования
16. Одним из признаков классификации ошибок является
- 1) синтаксис и семантика
  - 2) степень заикливания
  - 3) первичные и побочные ошибки
  - 4) первостепенные и второстепенные ошибки
17. Отладка бывает:
- 1) ручная и семантическая
  - 2) ручная и автоматизированная
  - 3) разрушающая и неразрушающая
  - 4) разрушающая, семантическая, оптимизирующая
18. Тестирование – это...
- 1) оптимизация программ
  - 2) действие, направленное на выявление ошибок
  - 3) регистрация программы
  - 4) исправление выявленных ошибок
19. Тестирование – это...
- 1) процесс создания загрузочного файла программы
  - 2) запуск программы на выполнение
  - 3) процесс многократного выполнения программы с целью обнаружения максимального количества ошибок
  - 4) процесс нахождения и исправления ошибок
20. Тестовый набор данных должен включать
- 1) входные, промежуточные и выходные данные
  - 2) входные и выходные данные
  - 3) все промежуточные результаты проверки тестов и конечный результат выполнения каждой функции
  - 4) входные, выходные данные и результаты проверки каждого условия
21. Тестирование бывает
- 1) нисходящее, восходящее, промежуточное, завершенное
  - 2) структурное, функциональное, промежуточное, полное
  - 3) нисходящее, восходящее, структурное, полное
  - 4) нисходящее, восходящее, структурное, функциональное
22. Что известно при тестировании «черного ящика»?
- 1) функции программы
  - 2) внутренняя структура программы
  - 3) работа каждой функции на всей области определения
  - 4) внутренние элементы программы и связи между ними
23. При тестировании «белого ящика» исследуется...
- 1) функции программы
  - 2) внутренняя структура программы
  - 3) работа каждой функции на всей области определения
  - 4) внутренние элементы программы и связи между ними
24. К методам «белого ящика» относятся...

- 1) метод покрытия решений, метод граничных решений, метод функциональных диаграмм, метод покрытия условий
  - 2) метод эквивалентных разбиений, метод функциональных диаграмм, анализ граничных решений
  - 3) метод покрытия условий, метод покрытия операторов, метод покрытия решений, анализ граничных решений
  - 4) метод покрытия условий, метод покрытия операторов, метод покрытия решений, метод покрытия решений и условий
25. Метод эквивалентных разбиений основан на...
- 1) разработке такого числа эквивалентных тестов, достаточного для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз
  - 2) разбиении входной области программы на классы по определенным признакам
  - 3) разработке достаточного количества тестов, чтобы каждое решение на этих тестах выполнялось по крайней мере один раз
  - 4) выполнении каждого оператора хотя бы один раз
26. Метод покрытия условий основан на...
- 1) разработке такого числа эквивалентных тестов, достаточного для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз
  - 2) разбиении входной области программы на классы по определенным признакам
  - 3) разработке достаточного количества тестов, чтобы возможные результаты каждого условия в решении выполнялось по крайней мере один раз
  - 4) выполнении каждого оператора хотя бы один раз
27. Тестирование «черного ящика» выполняется
- 1) на ранних этапах разработки программы
  - 2) когда разработан весь программный продукт и протестированы отдельные его модули
  - 3) на поздних стадиях тестирования программы
  - 4) на ранних стадиях тестирования программы
28. Техника «черного ящика» ориентирована на...
- 1) выявление класса ошибок
  - 2) выявление отдельных ошибок
  - 3) сокращение количества тестовых вариантов
  - 4) увеличение количества тестовых наборов
29. Тестирование включает в себя ...
- 1) создание текстового, загрузочного файла и их проверка
  - 2) разработка тестов и непосредственное тестирование по ним
  - 3) проверка разработанного набора тестов на исполняемом файле
  - 4) составление алгоритма решения задачи, текста программы, набора тестовых данных и их проверка
30. Чему равна вероятность наличия необнаруженных ошибок в какой-то части программы?
- 1) обратно пропорциональна числу ошибок обнаруженных в программе
  - 2) количеству обнаруженных в программе ошибок
  - 3) пропорциональна числу ошибок обнаруженных в программе
  - 4)  $1/3$  числу обнаруженных ошибок
31. Что известно при тестировании «белого ящика»?
- 1) функции программы
  - 2) внутренняя структура программы

- 3) работа каждой функции на всей области определения
  - 4) внутренние элементы программы и связи между ними
32. При тестировании «черного ящика» исследуется...
- 1) функции программы
  - 2) внутренняя структура программы
  - 3) работа каждой функции на всей области определения
  - 4) внутренние элементы программы и связи между ними
33. К методам «черного ящика» относятся...
- 1) метод покрытия решений, метод граничных решений, метод функциональных диаграмм, метод покрытия условий
  - 2) метод эквивалентных разбиений, метод функциональных диаграмм, анализ граничных решений
  - 3) метод покрытия условий, метод покрытия операторов, метод покрытия решений, анализ граничных решений
  - 4) метод покрытия условий, метод покрытия операторов, метод покрытия решений, метод покрытия решений и условий
34. Метод покрытия операторов при тестировании программ основан на...
- 1) разработке такого числа эквивалентных тестов, достаточного для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз
  - 2) разбиении входной области программы на классы по определенным признакам
  - 3) разработке достаточного количества тестов, чтобы каждое решение на этих тестах выполнялось по крайней мере один раз
  - 4) выполнении каждого оператора хотя бы один раз
35. Граничные условия – это
- 1) условия, ситуация, возникающая непосредственно на границе выше или ниже границ входных или выходных элементов класса эквивалентности
  - 2) ситуация, возникающая непосредственно на промежуточных элементах класса эквивалентности
  - 3) условия, ситуация, возникающие внутри программы, когда выполнены все тестовые наборы
36. При тестировании программ методами «черного ящика» необходимо разрабатывать набор тестов, который...
- 1) показывает нормальное функционирование программы
  - 2) выявляет все ошибки программы и по ним позволяет оптимизировать программу
  - 3) показывает нормальное и аномальное функционирование программы
37. Тестирование «белого ящика» выполняется
- 1) на ранних этапах разработки программы
  - 2) когда разработан весь программный продукт и протестированы отдельные его модули
  - 3) на поздних стадиях тестирования программы
  - 4) на ранних стадиях тестирования программы
38. Тестирование «черного ящика» обеспечивает поиск следующих категорий ошибок:
- 1) ошибок во внутренних структурах данных
  - 2) ошибок интерфейса
  - 3) ошибок во внешних структурах данных
  - 4) ошибок в циклах и ветвлениях
  - 5) ошибок характеристик
39. К программным средствам защиты программного продукта не относятся....
- 1) криптографическую защиту
  - 2) ограничение доступа к программному продукту

- 3) патентную защиту
  - 4) нестандартное форматирование диска, на котором находится программный продукт
40. Лицензирование программного продукта относится к...
- 1) правовой защите ПП
  - 2) программной защите ПП
  - 3) технической защите ПП
  - 4) физической защите ПП
41. Каким знаком обозначается авторское право на программный продукт?
- 1) ©
  - 2) тм
  - 3) ®
42. Каким знаком обозначается регистрация права на программный продукт?
- 1) ©
  - 2) тм
  - 3) ®
43. Этап Эволюции при сопровождении программного продукта предполагает...
- 1) выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
  - 2) внесение изменения в программу в ответ на изменившиеся условия
  - 3) использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
  - 4) проектирование программного продукта, тестирование, тиражирование и утилизацию
44. Этап Сохранение при сопровождении программного продукта предполагает...
- 1) выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
  - 2) внесение изменения в программу в ответ на изменившиеся условия
  - 3) использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
  - 4) проектирование программного продукта, тестирование, тиражирование и утилизацию
45. Этап Чистое сопровождение при сопровождении программного продукта предполагает...
- 1) выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
  - 2) внесение изменения в программу в ответ на изменившиеся условия
  - 3) использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
  - 4) проектирование программного продукта, тестирование, тиражирование и утилизацию

### **Раздел 5. «Коллективная разработка программных средств»**

1. Существует две основные модели организации коллектива при разработке ПО:
  - 1) иерархическая модель и модель группы
  - 2) структурная и объектная модель
  - 3) иерархическая и объектная модель
  - 4) модель группы и сетевая модель
2. Какая модель коллективной разработки программного продукта определяет структуру коллектива с точки зрения отдела кадров?
  - 1) модель группы
  - 2) иерархическая модель
  - 3) структурная модель

- 4) сетевая модель
3. Какая модель коллективной разработки программного продукта не определяет структуру коллектива с точки зрения отдела кадров?
  - 1) модель группы
  - 2) иерархическая модель
  - 3) структурная модель
  - 4) сетевая модель
4. Основными недостатками иерархической модели коллективной разработки программных продуктов является:
  - 1) несогласованное представление о разных сторонах проекта
  - 2) нехватка информации
  - 3) разрозненная связь с внешними источниками информации
  - 4) сложность расстановки приоритетов
5. Основными недостатками групповой модели коллективной разработки программных продуктов является:
  - 1) несогласованное представление о разных сторонах проекта
  - 2) нехватка информации
  - 3) разрозненная связь с внешними источниками информации
  - 4) сложность расстановки приоритетов
6. Для скрытия недостатков иерархической модели коллективной разработки программных продуктов предусматривают
  - 1) сплочение коллектива путем приобретения большого числа заказов
  - 2) распределение обязанностей руководителя между членами коллектива
  - 3) объединение обязанностей руководителя и отдельных членов коллектива
  - 4) определение целей проекта и распределение обязанностей в соответствии с целями
7. В чем заключается задача модели проектной группы при коллективной разработке программного продукта?
  - 1) сплочение коллектива путем приобретения большого числа заказов
  - 2) распределение обязанностей руководителя между членами коллектива
  - 3) объединение обязанностей руководителя и отдельных членов коллектива
  - 4) определение целей проекта и распределение обязанностей между членами группы
8. Основной целью менеджера продукта при коллективной разработке программного продукта является:
  - 1) удовлетворение требований заказчика
  - 2) соблюдение ограничений проекта
  - 3) соответствие спецификациям
  - 4) выпуск программного продукта только после выявления и устранения проблем
9. Основной целью менеджера программы при коллективной разработке программного продукта является:
  - 1) удовлетворение требований заказчика
  - 2) соблюдение ограничений проекта
  - 3) соответствие спецификациям
  - 4) выпуск только после выявления и устранения проблем
10. Основной целью разработчика при коллективной разработке программного продукта является:
  - 1) удовлетворение требований заказчика
  - 2) соблюдение ограничений проекта
  - 3) соответствие спецификациям
  - 4) выпуск программного продукта только после выявления и устранения проблем
11. Основной целью тестера при коллективной разработке программного продукта является:
  - 1) удовлетворение требований заказчика

- 2) соблюдение ограничений проекта
  - 3) соответствие спецификациям
  - 4) выпуск программного продукта только после выявления и устранения проблем
12. Основной целью инструктора при коллективной разработке программного продукта является:
- 1) удовлетворение требований заказчика
  - 2) соблюдение ограничений проекта
  - 3) повышение эффективности труда пользователя
  - 4) выпуск программного продукта только после выявления и устранения проблем
13. Основной целью инструктора при коллективной разработке программного продукта является:
- 1) удовлетворение требований заказчика
  - 2) простота развертывания и постоянное сопровождение программного продукта
  - 3) повышение эффективности труда пользователя
  - 4) выпуск программного продукта только после выявления и устранения проблем
14. Главная задача менеджера продукта при коллективной разработке программного продукта заключается...
- 1) в формировании общего представления о поставленной задаче и о том, как ее решать
  - 2) в ведении процесса разработки с учетом всех ограничений
  - 3) в испытании продукта в реальных условиях
  - 4) в повышении эффективности труда пользователей
14. Главная задача тестера при коллективной разработке программного продукта заключается...
- 1) в формировании общего представления о поставленной задаче и о том, как ее решать
  - 2) в ведении процесса разработки с учетом всех ограничений
  - 3) в испытании продукта в реальных условиях
  - 4) в повышении эффективности труда пользователей
15. Главная задача менеджера продукта при коллективной разработке программного продукта заключается...
- 1) в формировании общего представления о поставленной задаче и о том, как ее решать
  - 2) в ведении процесса разработки с учетом всех ограничений
  - 3) в испытании продукта в реальных условиях
  - 4) в повышении эффективности труда пользователей
16. Главная задача инструктора при коллективной разработке программного продукта состоит...
- 1) в формировании общего представления о поставленной задаче и о том, как ее решать
  - 2) в ведении процесса разработки с учетом всех ограничений
  - 3) в испытании продукта в реальных условиях
  - 4) в повышении эффективности труда пользователей
17. Главная задача логистика при коллективной разработке программного продукта состоит...
- 1) в формировании общего представления о поставленной задаче и о том, как ее решать
  - 2) в проверке, чтобы все серверы развертывания и рабочие станции пользователей удовлетворяли указанным требованиям
  - 3) в испытании продукта в реальных условиях
  - 4) в повышении эффективности труда пользователей



18. Кто из членов группы при коллективной разработке программных продуктов составляет график работ?
- 1) менеджер продукта
  - 2) менеджер программы
  - 3) логистик
  - 4) инструктор
19. Кто из членов группы при коллективной разработке программных продуктов выполняет проектирование архитектуры программного продукта?
- 1) менеджер продукта
  - 2) разработчик
  - 3) логистик
  - 4) инструктор
20. Кто из членов группы при коллективной разработке программных продуктов разрабатывает стратегию, планы, графики и сценарии тестирования?
- 1) менеджер продукта
  - 2) разработчик
  - 3) логистик
  - 4) тестер
21. Кто из членов группы при коллективной разработке программных продуктов составляет документацию, определяет требования к резервному копированию данных и разрабатывает план восстановления на случай отказа систем?
- 1) менеджер продукта
  - 2) разработчик
  - 3) логистик
  - 4) тестер
22. Кто из членов группы при коллективной разработке программных продуктов участвует в создании пользовательского интерфейса, сокращая тем самым затраты на сопровождение продукта и поддержку пользователей?
- 1) менеджер продукта
  - 2) разработчик
  - 3) инструктор
  - 4) тестер
23. К недостаткам иерархической модели коллективной разработки программных продуктов можно отнести:
- 1) несогласованное представление о разных сторонах проекта
  - 2) невозможностью учесть все особенности проекта
  - 3) разрозненная связь с внешними источниками информации
  - 4) отсутствием полноценной связи между всеми участниками проекта, так как вся информация идет в одном направлении — вверх по иерархии, к главному менеджеру
24. К недостаткам иерархической модели коллективной разработки программных продуктов можно отнести:
- 1) несогласованность личных планов членов группы
  - 2) невозможностью учесть все особенности проекта
  - 3) отсутствие опыта, снижающее эффективность коллективной работы
  - 4) отсутствием полноценной связи между всеми участниками проекта, так как вся информация идет в одном направлении — вверх по иерархии, к главному менеджеру
25. Какие задачи необходимо решить, чтобы проект считался удачным?
- 1) удовлетворить требования заказчика
  - 2) соблюсти ограничения
  - 3) спроектировать систему по объектно-ориентированному методу

- 4) выполнить спецификации, основанные на требованиях пользователей
- 5) выпустить продукт только после выявления и устранения всех проблем
- 6) выполнить программный продукт с учетом ситуации на рынке программ
- 7) гарантировать простоту развертывания и управления

### **3.1.2. Задания для оценки освоения МДК.03.02. Инструментальные средства разработки программного обеспечения.**

Типовой тест к дифференцированному зачету

1. Программное средство, предназначенное для поддержки разработки других программ, называется -...
  - 1) аппаратным инструментом
  - 2) программным инструментом
  - 3) программной средой
  - 4) инструментарий технологии программирования
2. Анализаторы обеспечивают...
  - 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
  - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
  - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
3. Преобразователи обеспечивают...
  - 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
  - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
  - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
4. Сколько классов инструментальных средств выделяют в инструментальной среде разработки и сопровождения программ?
  - 1) 2
  - 2) 4
  - 3) 3
  - 4) 5
5. Среда программирования предназначена для...
  - 1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
  - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
  - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
6. Инструментальные среды программирования бывают
  - 1) языково-ориентированные среды и среды общего назначения
  - 2) объектно-ориентированные и языково-ориентированные среды

- 3) среды общего назначения и прикладные среды
  - 4) среды общего назначения, прикладные среды, логические и математические среды
7. Для поддержки разработки программного продукта на каком-либо одном языке программирования используют...

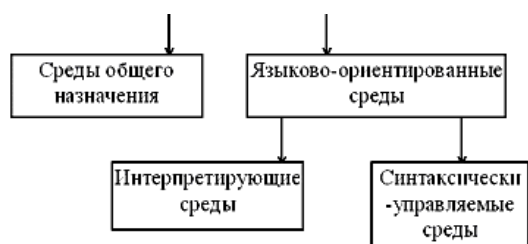
- 1) среду программирования общего назначения
  - 2) языково-ориентированную среду программирования
  - 3) интерпретирующую среду программирования
  - 4) прикладную среду программирования
8. Синтаксически-управляемая инструментальная среда программирования базируется на знании

- 1) семантики языка программирования
  - 2) синтаксиса языка программирования
  - 3) синтаксиса и семантики языка программирования
  - 4) основных управляющих структур языка программирования
9. На рисунке представлена классификация



- 1) инструментальной системы технологии программирования
  - 2) инструментальной среды разработки и сопровождения программ
  - 3) рабочего места компьютерной технологии
  - 4) языков программирования
10. Инструментальная система технологии программирования – это...
- 1) программное средство, предназначенное для поддержки разработки других программ
  - 2) устройство компьютера, специально предназначенное для поддержки разработки программного средства
  - 3) интегрированная совокупность программных и аппаратных инструментов, поддерживающая все процессы разработки и сопровождения больших программных продуктов
  - 4) логически связанная совокупность программных и аппаратных инструментов поддерживающих разработку ПП
11. Устройство компьютера, специально предназначенное для поддержки разработки программного средства, называется -...
- 1) аппаратным инструментом
  - 2) программным инструментом
  - 3) программной средой
  - 4) инструментарий технологии программирования
12. Редакторы обеспечивают...
- 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
  - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
  - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
13. Инструменты, поддерживающие процесс выполнения программ обеспечивают...

- 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматический привод документов к другой форме представления или перевод документа одного вида к документу другого вида
  - 3) возможность выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
  - 4) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
14. Инструментальная система технологии программирования предназначена для...
- 1) поддержки всех процессов разработки и сопровождения в течение всего жизненного цикла ПС и ориентирована на коллективную разработку больших программных систем с длительным жизненным циклом
  - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
  - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
  - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
15. Рабочее место компьютерной технологии предназначено для...
- 1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла
  - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
  - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
  - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
16. Инструментальные среды программирования содержат
- 1) редактор, анализатор и компилятор
  - 2) редактор, интерпретатор и компилятор
  - 3) интерпретатор, компилятор, преобразователь
  - 4) редактор и интерпретатор
17. Для поддержки разработки программного продукта на разных языках программирования (например, текстовый редактор, редактор связей или интерпретатор языка целевого компьютера) используют...
- 1) среду программирования общего назначения
  - 2) языково-ориентированную среду программирования
  - 3) интерпретирующую среду программирования
  - 4) прикладную среду программирования
18. На рисунке представлена классификация



- 1) инструментальной системы технологии программирования
- 2) инструментальной среды программирования
- 3) рабочего места компьютерной технологии

- 4) языков программирования
19. При использовании компьютерных технологий для разработки ПП жизненный цикл ПП представлен следующей цепочкой:
  - 1) прототипирование – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
  - 2) прототипирование – разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
  - 3) разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
  - 4) прототипирование – разработка спецификаций – кодогенерация – аттестация, применение, сопровождение
20. Основными чертами инструментальной системы технологии программирования являются...
  - 1) массовость, дискретность, результативность, определенность, понятность
  - 2) комплексность, ориентированность на коллективную разработку, технологическая определенность, интегрированность
  - 3) актуальность, непротиворечивость, полнота
  - 4) комплексность, актуальность, интегрированность, массовость, понятность
21. Современные крупные проекты информационных систем характеризуются следующими особенностями:
  - 1) сложность описания, требующая тщательного моделирования и анализа данных и процессов
  - 2) наличие совокупности тесно взаимодействующих компонентов
  - 3) наличие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений
  - 4) невозможность интеграции существующих и вновь разрабатываемых приложений;
22. Под CASE-средства понимаются программные средства, поддерживающие...
  - 1) процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО
  - 2) процессы тиражирования программного продукта
  - 3) процессы создания и эксплуатации программного продукта
  - 4) процессы компиляции и интерпретации программных продуктов
23. Репозиторий Case – средства – это...
  - 1) специализированная база данных проекта, предназначенная для отображения состояния проектируемой системы в каждый момент времени
  - 2) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
  - 3) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
  - 4) компонент, позволяющий получать информацию о проектах в виде отчетов
  - 5) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
  - 6) набор системных утилит по обслуживанию репозитория
24. Графический редактор Case – средства – это...
  - 1) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
  - 2) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
  - 3) компонент, позволяющий получать информацию о проектах в виде отчетов

- 4) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
25. Верификатор Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
  - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
  - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
  - 4) набор системных утилит по обслуживанию репозитория
26. Документатор проекта Case – средства – это...
- 1) компонент, позволяющий получать информацию о проектах в виде отчетов
  - 2) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
  - 3) набор системных утилит по обслуживанию репозитория
  - 4) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
27. Сервис Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
  - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
  - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
  - 4) набор системных утилит по обслуживанию репозитория
28. Администратор проекта Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
  - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
  - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
  - 4) набор системных утилит по обслуживанию репозитория
29. Какие методологии проектирования используют Case – средства?
- 1) структурного и модульного проектирования
  - 2) структурного и объектно-ориентированного проектирования
  - 3) объектно-ориентированного и нисходящего проектирования
  - 4) нисходящего и восходящего проектирования
30. Структурное проектирование системы основано на...
- 1) объектно-ориентированной декомпозиции
  - 2) алгоритмической декомпозиции
  - 3) модульной декомпозиции
  - 4) функциональной декомпозиции
31. Объектно-ориентированное проектирование системы основано на...
- 1) объектно-ориентированной декомпозиции
  - 2) алгоритмической декомпозиции
  - 3) модульной декомпозиции
  - 4) функциональной декомпозиции
32. Case – средства представляют собой...
- 1) набор инструментальных средств для проектирования программного продукта
  - 2) набор программных средств для сопровождения программного продукта
  - 3) набор программных и инструментальных средств, поддерживающие все процессы жизненного цикла программного продукта
  - 4) набор аппаратных средств, поддерживающих все процессы жизненного цикла программного продукта

33. Компания-разработчик приобрела новое Case – средство. Сразу ли компания получит ожидаемый результат от применения новой технологии?
- 1) Да 2) нет
34. Сколько классов Case – средств выделяют?
- 1) 5 2) 3 3) 7 4) 2
35. Case – средства анализа и проектирования, предназначенные для
- 1) моделирования данных и генерации схем баз данных
  - 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
  - 3) обеспечения комплексной поддержки требований к создаваемой системе
  - 4) поддержки всего жизненного цикла программного продукта
36. Case – средства управления требованиями предназначены для
- 1) моделирования данных и генерации схем баз данных
  - 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
  - 3) обеспечения комплексной поддержки требований к создаваемой системе
  - 4) поддержки всего жизненного цикла программного продукта
37. Case – средства проектирования баз данных предназначены для
- 1) моделирования данных и генерации схем баз данных
  - 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
  - 3) обеспечения комплексной поддержки требований к создаваемой системе
  - 4) поддержки всего жизненного цикла программного продукта
38. На каких стандартах базируется технология освоения и внедрения Case – средств?
- 1) ГОСТ 2107-99
  - 2) IEEE (IEEE Std 1348-1995 и IEEE Std 1209-1992)
  - 3) AES
  - 4) ISO
39. Из каких этапов состоит процесс освоения и внедрения Case – средств?
- 1) определение потребностей в CASE-средствах, оценка и выбор CASE-средств, практическое внедрение CASE-средств
  - 2) определение потребностей в CASE-средствах, оценка и выбор CASE-средств, выполнение пилотного проекта, практическое внедрение CASE-средств
  - 3) определение потребностей в CASE-средствах, проектирования CASE-средств, практическое применение CASE-средств
  - 4) проектирование CASE-средств, оценка и внедрение CASE-средств, практическое применение CASE-средств
40. Критериями для выбора CASE-средств могут являться
- 1) открытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, обеспечение целостности проекта, независимость от программно-аппаратной платформы и СУБД
  - 2) модифицируемость, простота, эффективность, учет человеческого фактора, многоплатформенность
  - 3) закрытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, простота, эффективность
  - 4) максимальная зависимость от программных и аппаратных средств системы и характеристик самой системы, жесткая привязка к конкретным информационным процессам, прочность внутренней связи отдельных компонентов системы
41. Комплексность компьютерной поддержки разработки ПП с использованием инструментальной системы технологии программирования означает
- 1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

- 2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов
  - 3) что все инструменты объединены единым пользовательским интерфейсом
  - 4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга
42. Ориентированность инструментальной системы технологии программирования на коллективную разработку означает
- 1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана
  - 2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов
  - 3) что все инструменты объединены единым пользовательским интерфейсом
  - 4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга
43. Технологическая определенность инструментальной системы технологии программирования означает
- 1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана
  - 2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов
  - 3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования
  - 4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга
44. Интегрированность инструментальной системы технологии программирования по данным означает
- 1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана
  - 2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов
  - 3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования
  - 4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга
45. Интегрированность инструментальной системы технологии программирования по пользовательскому интерфейсу означает
- 1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана
  - 2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов
  - 3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования
  - 4) что все инструменты объединены единым пользовательским интерфейсом



### **3.1.3. Задания для оценки освоения МДК.03.03. Документирование и сертификация.**

Типовой тест к дифференцированному зачету

1. Зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать:
  - 1) Сообщение;
  - 2) Документ;
  - 3) Текст.
2. Что не относится к свойствам, которыми должно обладать сообщение, чтобы стать документом?
  - 1) Легитимность;
  - 2) Доступность;
  - 3) Качество.
3. Что не относится к внутренним признакам документа?
  - 1) Стиль;
  - 2) Управляющая сила;
  - 3) Размер.
4. К каким признакам документа относятся: авторство, наименование вида документа, датирование, удостоверение, форма и размер, способ документирования и используемые носители?
  - 1) Внешние;
  - 2) Документообразующие;
  - 3) Внутренние.
5. По какому признаку классифицированы следующие документы: протоколы, приказы, отчеты?
  - 1) Вид документа;
  - 2) Качество документа;
  - 3) Разновидность документа.
6. По какому признаку классифицирована информация на общедоступную и конфиденциальную?
  - 1) Порядок представления;
  - 2) Порядок распространения;
  - 3) Категория доступа.
7. Что не относится к юридически значимым элементам оформления документа?
  - 1) Телефон организации, являющейся автором документа;
  - 2) Подпись документа;
  - 3) Регистрационный номер документа.
8. Частное понятие, которое детализирует, уточняет характер деятельности, документируемой тем или иным видом:
  - 1) Вид документа;
  - 2) Разновидность документа;
  - 3) Признак документа.
9. Совокупность присущих документу существенных признаков и особенностей, позволяющих выделить его из среды других предметов:
  - 1) Качество документа;
  - 2) Признак документа;

- 3) Вид документа.
- 10. К чему относится изменение во времени яркости элементов изображения в телевизионных системах?
  - 1) Документ;
  - 2) Информация;
  - 3) Сообщение.
- 11. Информация, содержащая сведения об источниках, где могут находиться необходимые данные (факты):
  - 1) Постоянная;
  - 2) Фактографическая;
  - 3) Документальная.
- 12. Задача документоведения:
  - 1) Практическое обоснование процессов документационного обеспечения аппарата управления обществом;
  - 2) Теоретическое обоснование процессов документационного обеспечения аппарата управления обществом;
  - 3) 1 и 2.

### 3.1.4 Материалы для дифференцированного зачета по МДК.03.01 Технология разработки программного обеспечения

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА  
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ  
СООБЩЕНИЯ» в г.Алатыре**

<p><b>ОДОБРЕНО</b> на заседании ЦК математических и общих естественнонаучных дисциплин Протокол № _____ Председатель <u>Р.В.Пасюнина</u> «__» _____ 2020г.</p>	<p align="center"><u>Дифференцированный</u> <u>зачет по</u> <u>МДК.03.01 Технология</u> <u>разработки программного</u> <u>обеспечения</u> <small>Дисциплина</small> Группа <u>ПО-18-1</u> Семестр <u>8</u> Билет №<u>1</u></p>	<p><b>УТВЕРЖДАЮ:</b> Заместитель директора по УР <u>Т.Ю.Базилевич</u> «__» _____ 2020г.</p>
--	--	---

**Коды проверяемых и общих компетенций ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9.**

**Место проведения экзамена – кабинет №301**

Инструкция:

Внимательно прочитайте задание.

При выполнении задания вы можете воспользоваться методическими пособиями по программе из папки «К экзамену»

Время выполнения задания – 30 минут.

Задания

1. Основные понятия программного обеспечения (программное обеспечение, программа, приложение, задача, предметная область).
2. Разработка требований к программному продукту.
3. Задача. Создать приложение для подсчета суммы цифр введенного целого числа.

Критерии оценки

Оценка «5»:	- Выполнены полностью часть 1, 2, 3
Оценка «4»	- Выполнены полностью часть 1,2 задание из части 3 выполнено не полностью
Оценка «3»	- Выполнена только часть 1 и 2, или 1 и 3, или 2 и 3
Оценка «2»	- Работа не выполнена

Преподаватель \_\_\_\_\_ Скворцова И.И.

#### **4. Оценка по учебной и производственной практикам**

Оценка по учебной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и требованиями организации, в которой проходила практика.

Итогом производственной практики (по профилю специальности) является оценка, которая выставляется руководителем практики от образовательного учреждения на основании собеседования со студентом с учетом его характеристики от производства, отношения к работе, качества выполненных отчетных документов и полученной квалификации. Оценка по практике приравнивается к оценке теоретического обучения и учитывается при подведении итогов успеваемости.

После прохождения практики проводится квалификационный экзамен

5. Контрольно-оценочные материалы для экзамена (квалификационного)

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА  
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ  
СООБЩЕНИЯ» в г.Алатыре**

<b>ОДОБРЕНО</b> на заседании ЦК математических и общих естественнонаучных дисциплин Протокол № __ от __» 2020г. Председатель ____ Р.В.Пасю ина	Экзаменационный билет № 1 <u>ПМ.03 Участие в</u> <u>интеграции программных</u> <u>модулей</u> ПМ, МДК Группа <u>ПО-17-1</u> Семестр <u>8</u>	<b>УТВЕРЖДАЮ</b> Заместитель директора по УР ____ Т.Ю. Базилевич «__» _____ 2020г .
--	--	--

**Коды проверяемых результатов обучения: ПК 3.1, ПК 3.2, ПК 3.3,  
ПК 3.4, ПК 3.5, ПК 3.6, ОК1, ОК2, ОК3, ОК4, ОК6, ОК7, ОК8, ОК9.**

**Место проведения экзамена – кабинет №301**

**Инструкция**

1. Внимательно прочитайте задание.
2. Вы можете воспользоваться: справочной литературой
3. Максимальное время выполнения задания – 30 мин.

**Задание.**

**Задание 1.** Ответьте на вопрос: Структурное программирование.

**Задание 2.** Ответьте на вопрос: Защита ПО от несанкционированного доступа.

**Задание 3.** Выполните практическое задание:

Загрузите проект с удаленного репозитория Git :student@gitexam:/git/black. Сделайте слияние ветви alpha с основной. Загрузите результат на удаленный репозиторий.

**Критерии оценки:**

- оценка «отлично» выставляется студенту, если даны полные правильные ответы на все вопросы и решена задача;
- оценка «хорошо» выставляется, если даны ответы на вопросы, но допущены неточности и решены задачи.
- оценка «удовлетворительно» выставляется, если даны неполные ответы на вопросы и задачи решены с ошибками;
- оценка «неудовлетворительно» выставляется, если нет правильного ответа ни на один вопрос и не решены задачи.

Преподаватели: \_\_\_\_\_/И.И.Скворцова/  
\_\_\_\_\_ /К.А.Новикова/

**6. ОЦЕНОЧНАЯ ВЕДОМОСТЬ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**  
**СВОДНАЯ ВЕДОМОСТЬ ДОПУСКА**  
 к экзамену (квалификационному) по ПМ.03 Участие в интеграции программных модулей  
код и название модуля

Специальность 09.02.03 «Программирование в компьютерных системах»

Группа ПО-17-1

Курс 4

	ФИО студента	Результаты аттестации			УП.03.01 Учебная практика	ПП.03.01 Производственная
		МДК .03.01 Технология обеспечения разработки программного	МДК.03.02 Инструментальные средства разработки программного	МДК.03.03 Документирование и сертификация		
1.	Авксентьева Анжелика Дмитриевна					
2.	Бакун Данила Витальевич					
3.	Днепровская Ксения Виленовна					
4.	Кормишин Егор Александрович					
5.	Кудашкин Илья Александрович					
6.	Лавренко Дмитрий Владимирович					
7.	Ларин Игорь Сергеевич					
8.	Назаров Андрей Вячеславович					
9.	Никишин Даниил Валерьевич					
10.	Плюснина Мария Сергеевна					
11.	Прохоров Андрей Евгеньевич					
12.	Сегов Данила Алексеевич					
13.	Сторожилова Алена Андреевна					
14.	Уральский Владимир Викторович					
15.	Царев Никита Сергеевич					
16.	Чаткин Александр Александрович					
17.	Чижов Дмитрий Александрович					

Заведующий отделением \_\_\_\_\_ Н.А.Мальков  
подпись

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

**ПРОТОКОЛ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО)**  
 по профессиональному модулю ПМ.03 «Участие в интеграции программных модулей»  
наименование профессионального модуля

образовательной программы специальности 09.02.03 Программирование в компьютерных системах наименование специальности, программы профессиональной подготовки, переподготовки, повышения квалификации

«\_\_» \_\_\_\_\_ 20\_\_ г.

Профессиональный модуль осваивался в объеме \_\_\_ час. с «\_\_» \_\_\_\_\_ 20\_\_ г. по «\_\_» \_\_\_\_\_ 20\_\_ г.

Место проведения квалификационного экзамена филиал СамГУПС в г.Алатыре.

Состав аттестационной комиссии на основании приказа по техникуму № \_\_\_\_\_ от «\_\_» \_\_\_\_\_ 20\_\_ г.:

Состав комиссии	Должность, место работы	ФИО
Председатель комиссии	Начальник отдела автоматизации и программирования АО «Завод Электроприбор»	Федоров Иван Иванович
Члены комиссии		
Заместитель председателя	Заместитель директора по УР АТЖТ филиала СамГУПС в г.Алатыре	Базилевич Татьяна Юрьевна
	Председатель предметной (цикловой) комиссии математических и общих естественнонаучных дисциплин	Пасюнина Раиса Викторовна
	Заведующий отделением	Мальков Николай Анатольевич
	Преподаватель	Скворцова Ирина Ивановна
	Преподаватель	Новикова Ксения Андреевна

## ОЦЕНОЧНАЯ ВЕДОМОСТЬ

*ФИО кандидата*

по профессиональному модулю Участие в интеграции программных модулей

*наименование профессионального модуля*

образовательной программы специальности 09.02.03 Программирование в компьютерных системах

*наименование специальности, программы профессиональной подготовки, переподготовки, повышения квалификации*

Профессиональный модуль осваивался в объеме \_\_\_ час. с «\_\_» \_\_\_ 20\_\_ г. по «\_\_» \_\_\_ 20\_\_ г.

Итоги квалификационного экзамена по профессиональному модулю:

Профессиональные компетенции	Оценка («освоена / не освоена»)	
ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонентов программного обеспечения.		
ПК 3.2. Выполнять интеграцию модулей в программную систему.		
ПК 3.3. Выполнять отладку программного продукта с использованием специализированных программных средств.		
ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев.		
ПК 3.5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования		
ПК 3.6. Разрабатывать технологическую документацию		
Итоговый результат по профессиональному модулю: Вид профессиональной деятельности «Участие в интеграции программных модулей» _____ освоен / не освоен. Оценка _____ <i>наименование вида профессиональной деятельности</i>		
Состав комиссии	Подпись	ФИО
Председатель аттестационной комиссии		Федоров Иван Иванович
Члены комиссии:		Базилевич Татьяна Юрьевна
		Пасюнина Раиса Викторовна
		Мальков Николай Анатольевич
		Скворцова Ирина Ивановна
		Новикова Ксения Андреевна

Дата проведения квалификационного экзамена \_\_\_\_\_, протокол № \_\_\_\_\_.

С оценочной ведомостью ознакомлен(а) \_\_\_\_\_.

*подпись кандидата, дата*

МП



### Сведения о курсовом проектировании

1. ФИО студента \_\_\_\_\_

2. Группа \_\_\_\_\_

3. Специальность 09.02.03 Программирование в компьютерных системах

<b>№ п/п</b>	<b>Индекс МДК</b>	<b>Тема курсового проекта</b>	<b>Оценка</b>	<b>Ф.И.О. руководителя курсового проекта</b>	<b>Подпись руководителя курсового проекта</b>
1.	МДК.03.01				

## 7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

### Основные источники:

1. Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие / Т. М. Зубкова. — Санкт-Петербург : Лань, 2019. — 324 с. — ISBN 978-5-8114-3842-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/122176>
2. Иванова, Г.С. Технология программирования : учебник / Иванова Г.С. — Москва : КноРус, 2018. — 333 с. — ISBN 978-5-406-06109-1. — URL: <https://book.ru/book/926372>

### Дополнительные источники:

1. Долженко, А.И. Технологии командной разработки программного обеспечения информационных систем : курс лекций / Долженко А.И. — Москва : Интуит НОУ, 2016. — 300 с. — URL: <https://book.ru/book/918260>