

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 14.09.2024 11:11:25

Уникальный идентификатор документа: c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования «Ростовский государственный экономический университет (РИНХ)»  
Финансово-экономический колледж



УТВЕРЖДАЮ

Директор

Р. А. Сычев

2023г

**Рабочая программа МДК**  
**Инструментальные средства разработки программного обеспечения**

Специальность

09.02.07

Информационные системы и программирование

Форма обучения	очная
Часов по учебному плану	121
в том числе:	
аудиторные занятия	92
самостоятельная работа	6

Ростов-на-Дону  
2023

**Распределение часов дисциплины по семестрам**

Семестр (<Курс>.<Семестр на курсе>)	5 (3.1)		6 (3.2)		Итого	
	Неделя		17			
Вид занятий	уп	рп	уп	рп	уп	рп
Лекции	10	10	34	34	44	44
Лабораторные			4	4	4	4
Практические	16	16	28	28	44	44
Курс. проект.			20	20	20	20
Итого ауд.	26	26	86	86	112	112
Контакт. работа	26	26	86	86	112	112
Сам. работа	2	2	4	4	6	6
Промежут. аттестация			3	3	3	3
Итого	28	28	93	93	121	121

**ОСНОВАНИЕ**

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (Приказ Министерства образования и науки Российской Федерации от 9 декабря 2016 г. № 1547)

Рабочая программа составлена по образовательной программе 09.02.07 Информационные системы и программирование для набора 2022 года

программа среднего профессионального образования

Учебный план утвержден учёным советом вуза от 29.08.2023 протокол № 1

Программу составил(и): Преподаватель, Горелько Е.А.

Председатель ЦМК: Горелько Е.А.

Рассмотрено на заседании ЦМК от 30.08.2023 протокол № 1

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

- |     |  |
|-----|--|
| 1.1 | В результате изучения профессионального модуля студент должен освоить основной вид деятельности «осуществление интеграции программных модулей» и соответствующие ему профессиональные компетенции. |
|-----|--|

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ООП: МДК

- |       |  |
|-------|--|
| 2.1   | <b>Требования к предварительной подготовке обучающегося:</b>   |
| 2.1.1 | Операционные системы и среды   |
| 2.1.2 | Информационные технологии  |
| 2.1.3 | Основы алгоритмизации и программирования   |
| 2.2   | <b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b> |
| 2.2.1 | Производственная практика ПП.02  |
| 2.2.2 | Квалификационный экзамен ПМ.02   |

## 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### 3.1 Знать

#### **ПК 2.2: Выполнять интеграцию модулей в программное обеспечение.**

Модели процесса разработки программного обеспечения.

Основные подходы к интегрированию программных модулей.

Основы верификации программного обеспечения.

Современные технологии и инструменты интеграции.

Основные протоколы доступа к данным.

Методы и способы идентификации сбоев и ошибок при интеграции приложений.

Основные методы отладки.

Основные методы и виды тестирования программных продуктов.

#### **ПК 2.3: Выполнять отладку программного модуля с использованием специализированных программных средств**

Основы организации инспектирования и верификации.

Встроенные и основные специализированные инструменты анализа качества программных продуктов.

#### **ПК 2.5: Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования**

Основные принципы процесса разработки программного обеспечения.

### 3.2 Уметь

#### **ПК 2.2: Выполнять интеграцию модулей в программное обеспечение.**

Использовать выбранную систему контроля версий.

Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.

Использовать различные транспортные протоколы и стандарты форматирования сообщений.

Организовывать постобработку данных.

Создавать классы-исключения на основе базовых классов.

Выполнять ручное и автоматизированное тестирование программного модуля.

Использовать приемы работы в системах контроля версий.

#### **ПК 2.3: Выполнять отладку программного модуля с использованием специализированных программных средств**

Использовать инструментальные средства отладки программных продуктов.

Выполнять тестирование интеграции.

Выполнять отладку, используя методы и инструменты условной компиляции.

Выявлять ошибки в системных компонентах на основе спецификаций.

#### **ПК 2.5: Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования**

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

Анализировать проектную и техническую документацию.

### 3.3 Владеть

**ПК 2.2: Выполнять интеграцию модулей в программное обеспечение.**

Интегрированием модулей в программное обеспечение.

**ПК 2.3: Выполнять отладку программного модуля с использованием специализированных программных средств**

Отладкой программных модулей.

**ПК 2.5: Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования**

Инспектированием разработанных программных модулей на предмет соответствия стандартам кодирования.

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Примечание
	<b>Раздел 1. Инструментальные средства разработки программного обеспечения</b>					
1.1	Инструментальное ПО. Концепция современной интегрированной среды разработки приложений. /Лек/	5	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.2	Технология компонентной разработки COM. /Лек/	5	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.3	Технология Java. /Лек/	5	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.4	Разработка приложения на Java /Пр/	5	8	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.5	CASE-средства /Лек/	5	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.6	Работа с Case - средствами /Пр/	5	8	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.7	Самостоятельная работа Сравнительный обзор CASE-средств. Библиотека классов .NET. Обеспечение безопасности в .NET /Ср/	5	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.8	Введение в ASP.NET MVC /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.9	Первое приложение ASP.NET MVC 5 /Пр/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.10	Модели, контроллеры, представления /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.11	Работа с контроллерами /Пр/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.12	Запросы Get и Post. /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.13	Стилизация. /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.14	Основы контроллеров /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.15	Методы ASP.NET MVC /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	
1.16	Использование хелперов /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1	

1.17	Параметры ASP.NET MVC /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.18	Результаты действий /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.19	Представления в ASP.NET MVC 5 /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.20	Представления /Пр/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.21	Работа с формами /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.22	Работа с базами данных. /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.23	Работа с базами данных /Пр/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.24	Фильтрация Переопределение шаблонов /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.25	Фильтрация данных /Пр/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.26	Маршрутизация /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.27	Маршрутизация /Пр/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.28	Постраничная навигация /Пр/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.29	Валидация данных. /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.30	Привязка моделей /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.31	Привязка моделей /Пр/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.32	Работа с Ajax и jQuery. /Лек/	6	2	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.33	Работа с Ajax и jQuery /Пр/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.34	Работа над курсовым проектом. Защита курсового проекта /Курс. проект./	6	20	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.35	Тестирование и отладка /Лаб/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.36	Самостоятельная работа Сравнительный анализ WEB-технологий /Ср/	6	4	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1
1.37	Экзамен	6	3	ПК 2.2. ПК 2.3. ПК 2.5.	Л1.1Л2.1 Л2.2 Э1

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена. Перечень вопросов к экзамену:

1. Необходимые инструментальные средства разработки программ
2. Состав и виды систем программирования
3. Технология Java. Разработка приложения на Java.
4. Факторы появления Case-технологий
5. Что такое Case-технология?
6. Платформа .NET Framework
7. Массивы .NET Framework
8. Коллекции .NET Framework
9. Методы построения пользовательского интерфейса в VS 2015.
10. Основные элементы управления форм
11. Принципы построения пользовательского интерфейса на WPF
12. Клиент - серверная модель приложения
13. Технология меж процессной коммуникации – WCF
14. Обеспечение работы с удаленными объектами и Web-сервисы
15. Принципы ASPNET технологии
16. ASPNET MVC
17. Модели, контроллеры, представления
18. Запрос Get (ASPNET)
19. Запрос Post (ASPNET)
20. Стилизация
21. Основы контроллеров
22. Работа с контроллерами ASPNET MVC
23. Методы ASPNET MVC
24. Параметры ASPNET MVC
25. Представления в ASPNET MVC
26. Сетевое программирование
27. Методы отладки программ в VS
28. Развертывание приложений и контроль версий
29. Средства проектирования приложений
30. Средства тестирования программ
31. Языки программирования высокого уровня
32. Процедурно-ориентированный подход к разработке
33. Объектно-ориентированный подход к разработке
34. Структурный подход к разработке ИС
35. Объектно-ориентированный подход к разработке ИС
36. Основные элементы унифицированного языка моделирования UML
37. Диаграммы моделирования языка UML
38. Диаграмма классов, назначение и методы построения
39. Диаграммы «сущность-связь»
40. Диаграммы классов
41. Язык графического описания UML
42. Диаграмма компонентов
43. Диаграмма композитной структуры
44. Диаграмма развёртывания
45. Диаграмма объектов
46. Диаграмма пакетов
47. Диаграмма деятельности
48. Преимущества UML
49. Применение отладочных классов в проекте
50. Работа с Ajax/jQuery.

*Критерии оценивания:*

5 баллов выставляется студентам за полный и правильный ответ на все вопросы билета с логическим обоснованием аргументов, в ответе нет ошибок.

4 балла выставляется студентам, если вопросы билета раскрыты полностью, но обоснования доказательства недостаточны, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

3 балла ставится студентам за правильный ответ на вопросы билета, при этом допущено более одной ошибки по изложению фактов или более двух-трех недочетов в ответе.

2 балла ставится студентам, если допущены существенные ошибки, показавшие, что обучающийся не обладает обязательными умениями по данной теме в полной мере.

## 5.2. Фонд оценочных средств для проведения текущего контроля

Представлен в Приложении 1 к рабочей программе МДК.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 6.1. Рекомендуемая литература

<b>6.1.1. Основная литература</b>				
	Авторы,	Заглавие	Издательство, год	Количество
Л1.1	Черткова Е. А.	Программная инженерия. Визуальное моделирование программных систем: Учебник для СПО: текст электронный	Юрайт, 2022	<a href="https://urait.ru/bcode/493226/">https://urait.ru/bcode/493226/</a> неограниченный доступ зарегистрированным пользователям
<b>6.1.2. Дополнительная литература</b>				
	Авторы,	Заглавие	Издательство, год	Количество
Л2.1	Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю.	Технология разработки программного обеспечения: учебное пособие для СПО	Москва: Юрайт, 2022	<a href="https://urait.ru/bcode/492496/">https://urait.ru/bcode/492496/</a> – неограниченный доступ зарегистрированным пользователям
Л2.2	Стружкин Н. П., Годин В. В.	Базы данных: проектирование: учебник для СПО	Москва: Юрайт, 2022	<a href="https://urait.ru/bcode/495973/">https://urait.ru/bcode/495973/</a> неограниченный доступ зарегистрированным пользователям
<b>6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»</b>				
Э1	Ежемесячный компьютерный журнал КомпьютерПресс – <a href="https://www.compress.ru/">https://www.compress.ru/</a>			
<b>6.3. Перечень программного обеспечения</b>				
6.3.1	Офисный пакет - LibreOffice			
6.3.2	Интернет-браузер - Chromium			
<b>6.4 Перечень информационных справочных систем</b>				
6.4.1	ИСС «КонсультантПлюс»			
6.4.2	ИСС «Гарант»			

<b>7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b>	
7.1	Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения.

<b>8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)</b>	
Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе МДК.	

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### МДК.02.02 Инструментальные средства разработки программного обеспечения

#### 1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

##### 1.1 Показатели и критерии оценивания компетенций:

УУД, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
<b>ПК 2.2: Выполнять интеграцию модулей в программное обеспечение.</b>			
<p><b>Знать:</b>            Модели процесса разработки программного обеспечения.            Основные подходы к интегрированию программных модулей.            Основы верификации программного обеспечения.            Современные технологии и инструменты интеграции.            Основные протоколы доступа к данным.            Методы и способы идентификации сбоев и ошибок при интеграции приложений.            Основные методы отладки.            Основные методы и виды тестирования программных продуктов.</p>	<p><b>Получение систематических знаний</b>            в области современных технологий и инструментов интеграции</p>	<p><b>Уровень знаний –</b>            понимать модели и принципы организации интеграции</p>	<p><b>Т (1-20), ПЗ (1-25), ЛР</b></p>
<p><b>Уметь:</b>            Использовать выбранную систему контроля версий.            Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.            Использовать различные транспортные протоколы и стандарты форматирования сообщений.            Организовывать постобработку данных.            Создавать классы-исключения на основе базовых классов.            Выполнять ручное и автоматизированное тестирование программного модуля.            Использовать приемы работы в системах контроля версий.</p>	<p><b>Сформировать систематическое умение</b>            в организации заданной интеграции модулей</p>	<p><b>Уровень умения –</b>            организовать требуемую интеграцию</p>	<p><b>Т (1-20), ПЗ (1-25), ЛР</b></p>
<p><b>Владеть:</b>            Интегрированием модулей в программное обеспечение.</p>	<p><b>Сформировать систематическое владение в</b></p>	<p><b>Уровень владения –</b>            интегрировать и отлаживать программные</p>	<p><b>ПЗ (1-25), ЛР, КП(1-35)</b></p>



	интегрировании и отладке программных модулей	модули в программное обеспечение в соответствии с требованиями.	
<b>ПК 2.3: Выполнять отладку программного модуля с использованием специализированных программных средств</b>			
<b>Знать:</b> Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов.	<b>Получение систематических знаний</b> о методах и видах разработки, интегрирования и отладки программного обеспечения	<b>Уровень знаний</b> – знать методы разработки, интегрирования и отладки программного обеспечения	<b>Т (1-20), ПЗ (1-25), ЛР</b>
<b>Уметь:</b> Использовать инструментальные средства отладки программных продуктов. Выполнять тестирование интеграции. Выполнять отладку, используя методы и инструменты условной компиляции. Выявлять ошибки в системных компонентах на основе спецификаций.	<b>Сформировать систематическое умение</b> по тестированию и отладке с использованием системы контроля версий	<b>Уровень умения</b> использовать систему контроля версий, выполнять тестирование и отладку.	<b>Т (1-20), ПЗ (1-25), ЛР</b>
<b>Владеть:</b> Отладкой программных модулей.	<b>Сформировать систематическое владение</b> средствами отладки с учетом соответствия стандартам.	<b>Уровень владения</b> отлаживать программные модули, инспектировать их на предмет соответствия стандартам кодирования.	<b>ПЗ (1-25), ЛР, КП(1-35)</b>
<b>ПК 2.5: Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования</b>			
<b>Знать:</b> Основные принципы процесса разработки программного обеспечения.	<b>Получение систематических знаний</b> по разработке и анализу программного обеспечения	<b>Уровень знаний</b> основных принципов процесса разработки	<b>Т (1-20), ПЗ (1-25), ЛР</b>
<b>Уметь:</b> Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию.	<b>Сформировать систематическое умение</b> анализировать документацию, разработанный код и выявлять ошибки	<b>Уровень умения</b> - выявлять ошибки в системных компонентах на основе спецификаций.	<b>Т (1-20), ПЗ (1-25), ЛР</b>
<b>Владеть:</b> Инспектированием разработанных программных модулей на предмет соответствия стандартам кодирования.	<b>Сформировать систематическое владение</b> методами инспектирования разработанных модулей	<b>Уровень владения</b> - инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.	<b>ПЗ (1-25), ЛР, КП(1-35)</b>

*Т – тестовые задания, ПЗ - практические задания, ЛР – лабораторная работа, КП- курсовой проект.*

**2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

## Тестовые задания:

### 1 семестр

Выберите один или несколько вариантов, составляющих ответ на поставленный вопрос:

#### 1. Программная инженерия:

- а) software engineering
- б) Инструменты создания программного обеспечения
- в) Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
- г) Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения
- д) Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
- е) Инженерная индустрия применения прикладного программного обеспечения
- ж) Совокупность инженерных методов и средств создания программного обеспечения
- з) Прикладное программное обеспечение для решения офисных задач

#### 2. Построение SADT-модели включает в себя выполнение следующих действий:

- а) Написание программного обеспечения для разрабатываемой системы по требованиям заказчика
- б) Сбор информации об объекте, определение его границ
- в) Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм
- г) Представление исследуемой системы в графическом виде
- д) Представление исследуемого объекта средствами системного моделирования
- е) Критическая оценка, рецензирование и комментирование
- ж) Разработка, отладка и тестирование программного обеспечения
- з) Использование графических пакетов для представления системы в виде модели

#### 3. Моделирование основывается на принципах:

- а) Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение
- б) Декомпозиции системы на отдельные подзадачи
- в) Инкапсуляции и полиморфизма
- г) Децентрализации управления системой
- д) Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности
- е) Открытой трансформируемой системы
- ж) Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга
- з) Анализа и синтеза проектирования систем

#### 4. В бизнес-процессах выделяют классы процессов:

- а) Решающие бизнес-процессы
- б) Регламентирующие бизнес-процессы
- в) Основные бизнес-процессы
- г) Бизнес-процессы поведения системы
- д) Программируемые бизнес-процессы
- е) Экономические бизнес-процессы
- ж) Обеспечивающие бизнес-процессы
- з) Бизнес-процессы управления

#### 5. CASE-средства классифицируются по следующим признакам:

- а) По применяемым методологиям и моделям систем и БД
- б) По используемому программному обеспечению
- в) По этапам жизненного цикла программного обеспечения
- г) По степени интегрированности с СУБД

- д) По уровням детализации и декомпозиции проектируемой системы
- е) По доступным платформам
- ж) По используемым языкам программирования
- з) По степени сложности моделируемой системы

6. К малым интегрированным средствам моделирования относятся:

- а) ARIS Toolset
- б) Design/IDEF
- в) ERwin
- г) BPwin
- д) Designer/2000
- е) Paradigm Plus
- ж) Model Mart
- з) Rational Rose

7. К средним интегрированным средствам моделирования относятся:

- а) Rational Rose
- б) Design/IDEF
- в) BPwin
- г) Designer/2000
- д) ARIS Toolset
- е) Model Mart
- ж) Paradigm Plus
- з) ERwin

## 2 семестр

8. Объектно-ориентированная методология (ООМ) включает в себя составные части:

- а) Объектно-ориентированный анализ
- б) Объектно-ориентированный подкласс
- в) Объектно-ориентированное проектирование
- г) Объектно-ориентированная парадигма
- д) Объектно-ориентированная экспозиция
- е) Объектно-ориентированное моделирование
- ж) Объектно-ориентированное программирование
- з) Объектно-ориентированная декомпозиция

9. К основным понятиям объектно-ориентированного подхода относятся:

- а) Обобщение
- б) Полиморфизм
- в) Инкапсуляция
- г) Реализация
- д) Агрегирование
- е) Наследование
- ж) Ассоциация
- з) Композиция

10. Главные принципы объектного подхода:

- а) Абстрагирование
- б) Наследование
- в) Ограничение доступа или инкапсуляция
- г) Безграничный доступ или инкапсуляция
- д) Модульность и иерархия
- е) Агрегирование
- ж) Композиция
- з) Обобщение и специализация

11. Дополнительные принципы объектного подхода:

- а) Реализация
- б) Типизация
- в) Параллелизм
- г) Внедрение
- д) Перпендикулярность
- е) Сохраняемость или устойчивость
- ж) Несохранимость или неустойчивость
- з) Динамичность

12. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:

- а) Rational Rose
- б) Model Mart
- в) MS Visio
- г) ARIS
- д) IDEF1X
- е) Erwin
- ж) BPwin
- з) JAM

13. К инструментальным средствам представления функциональных моделей относятся:

- а) JAM
- б) Model Mart
- в) MS Visio
- г) ARIS
- д) IDEF0
- е) Erwin
- ж) BPwin
- з) Rational Rose

14. Методологии, поддерживаемые в BPwin:

- а) IDEF1X
- б) IDEF0
- в) IDEF1
- г) IDEF3
- д) IDEFX
- е) IDEF5
- ж) DFD
- з) DFD1X

15. Диаграмма IDEF0 может содержать следующие типы диаграмм:

- а) Диаграмму классов
- б) Контекстную диаграмму, диаграмму декомпозиции
- в) Диаграмму компонентов
- г) Диаграмму дерева узлов
- д) Диаграмму взаимодействий
- е) Диаграмму только для экспозиции (FEO)
- ж) Диаграмму последовательности, диаграмму кооперации
- з) Диаграмму узлов

16. Уровни логической модели:

- а) Диаграмма сущность
- б) Диаграмма связь
- в) Диаграмма пакетов
- г) Диаграмма сущность-связь

- д) Модель данных, основанная на классах
- е) Модель данных, основанная на ключах
- ж) Полная операционная модель
- з) Полная атрибутивная модель

17. Внутренние стрелки не входящие в состав диаграммы IDEF0:

- а) mechanism- output
- б) output-input
- в) mechanism- input
- г) output-control
- д) output-input feedback
- е) output-control feedback
- ж) output-mechanism
- з) control feedback- mechanism

18. Типы стрелок не входящие в состав диаграммы IDEF0:

- а) Input
- б) Editor
- в) Control
- г) Properties
- д) Output
- е) Mechanism
- ж) Call
- з) Dictionary

19. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:

- а) Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
- б) Report Header. Печатается единожды в начале отчета
- в) Columnar. Простой табличный отчет
- г) Page Header. Печатается в верхней части каждой страницы
- д) Vertical. Простой вертикальный отчет
- е) Group Header. Печатается в начале каждой группы
- ж) Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные
- з) Detail. Печатается для каждой строчки набора данных

20. BPwin допускает следующие переходы с одной нотации на другую:

- а) IDEF3 → DFD
- б) DFD → IDEF0
- в) IDEF0 → DFD
- г) DFD → DFD
- д) IDEF3 → IDEF0
- е) IDEF0 → IDEF3
- ж) IDEF3 → IDEF3
- з) DFD → IDEF3

### **Критерии оценивания:**

- 5 баллов выставляется, если правильные ответы даны на 85-100% тестовых заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% тестовых заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% тестовых заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% тестовых заданий

## Практические задания:

### 1 семестр

1. Составить программу для вычисления площади треугольника по формуле Герона
2. Дана последовательность действительных чисел. Выяснить, будет ли она возрастающей.
3. Составить программу для вычисления площади треугольника по через синус угла и двум прилежащим к нему сторонам.
4. Найти произведение положительных элементов одномерного массива A размера N.
5. Определить время года по номеру месяца. Номер месяца вводить с клавиатуры
6. Составить программу для анализа, введенного пользователем числа (целое или нет; положительное, отрицательное или нуль; четное или нечетное).
7. В переменную последовательно вводятся N вещественных чисел. Вычислить максимальное значение.
8. В массив A[N] занесены натуральные числа. Найти сумму тех элементов, которые кратны данному K.
9. Составить программу вычисления факториала введенного с клавиатуры числа. // результат вывести в таком виде: fact=1\*2\*3=6 ( при n =3)
10. Определить максимальный элемент массива A[10] и его порядковый номер.

### 2 семестр

11. Составить программу для вычисления суммы всех натуральных чисел, кратных числу b и меньших 100.
12. В переменную последовательно вводят числа, отличные от нуля. Окончание ввода - ноль. Определить среднее арифметическое отрицательных чисел.
13. С клавиатуры вводятся числа. Суммировать числа до тех пор, пока сумма не станет больше 100. Вывести сумму и количество просуммированных чисел.
14. Составить программу, которая запрашивает дату (число, месяц, год) и проверяет корректность введенным пользователем данных.
15. Найти сумму положительных элементов одномерного массива A размера N.
16. В массиве целых чисел есть нулевые элементы. Создать массив из номеров этих элементов.
17. Написать программу вывода на экран четных чисел из интервала от 0 до 100.
18. Составить программу, которая по номеру дня недели выводит на экран расписание уроков в вашей группе в соответствующий день.
19. Написать программу для подсчета суммы чисел, кратных 3 в диапазоне от 30 до 60.
20. Составить программу для нахождения минимального значения среди элементов, стоящих до первого четного элемента.
21. Составить программу для поиска произведения положительных элементов массива.
22. Написать программу для нахождения в массиве из N элементов количества нулевых элементов.
23. Найти количество положительных элементов одномерного массива A размера N.
24. Написать программу для определения максимального элемента массива A[20].

25. Определить минимальный элемент массива A[15] и его порядковый номер.

**Критерии оценивания:**

- 5 баллов выставляется, если правильные ответы даны на 85-100% практических заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% практических заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% практических заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% практических заданий.

**Лабораторная работа:**

**Тема «Тестирование и отладка»**

**Цель:**

- 1) освоить процесс тестирования интерфейса пользователя средствами инструментальной среды разработки;
- 2) освоить процесс отладки проекта.

**Теоретические сведения**

- 1) Функциональное тестирование пользовательского интерфейса состоит из пяти фаз:
  - анализ требований к пользовательскому интерфейсу;
  - разработка тест-требований и тест-планов для проверки пользовательского интерфейса;
  - выполнение тестовых примеров и сбор информации о выполнении тестов;
  - определение полноты покрытия пользовательского интерфейса требованиями.
  - составление отчетов о проблемах в случае несовпадения поведения системы и требований, либо в случае отсутствия требований на отдельные интерфейсные элементы.

Все эти фазы точно такие же, как и в случае тестирования любого другого компонента программной системы. Отличия заключаются в трактовке некоторых терминов в применении к пользовательскому интерфейсу и в особенностях автоматизированного сбора информации на каждой фазе.

Так, тест-планы для проверки пользовательского интерфейса как правило представляют собой сценарии, описывающие действия пользователя при работе с системой. Сценарии могут быть записаны либо на естественном языке, либо на формальном языке какой-либо системы автоматизации пользовательского интерфейса. Выполнение тестов при этом производится либо оператором в ручном режиме, либо системой, которая эмулирует поведение оператора.

При сборе информации о выполнении тестовых примеров как правило применяются технологии анализа выводимых на экран форм и их элементов (в случае графического интерфейса) или выводимого на экран текста (в случае текстового), а не проверка значений тех или иных переменных, устанавливаемых программной системой.

Под полнотой покрытия пользовательского интерфейса понимается то, что в результате выполнения всех тестовых примеров каждый интерфейсный элемент был использован хотя бы один раз во всех доступных режимах.

Отчеты о проблемах в пользовательском интерфейсе могут включать в себя как описания несоответствий требований и реального поведения системы, так и описания проблем в требованиях к пользовательскому интерфейсу. Основной источник проблем в этих требованиях – их тестонепригодность, вызванная расплывчатостью формулировок и неконкретностью.

2) Отладка (debug, debugging) — этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки. Чтобы понять, где возникла ошибка, приходится: узнавать текущие значения переменных; выяснять, по какому пути выполнялась программа.

отладки начинается с попытки воспроизвести проблему, что может оказаться не простой задачей при программировании параллельных процессов или при некоторых необычных ошибках, известных как гейзенбаги.

Технологии отладки.

1) Использование отладчиков — программ, которые включают в себя пользовательский интерфейс для пошагового выполнения программы: оператор за оператором, функция за функцией, с остановками на некоторых строках исходного кода или при достижении определённого условия.

2) Вывод текущего состояния программы с помощью расположенных в критических точках программы операторов вывода — на экран, принтер, громкоговоритель или в файл. Вывод отладочных сведений в файл называется журналированием.

Инструменты отладки.

1. Отладчик – программный инструмент, позволяющий программисту наблюдать за выполнением исследуемой программы, останавливать и перезапускать её, прогонять в замедленном темпе, изменять значения в памяти и даже, в некоторых случаях, возвращать назад по времени.

2. Профилировщики – позволяют определить сколько времени выполняется тот или иной участок кода, а анализ покрытия позволит выявить неисполняемые участки кода.

3. API логгеры – позволяют программисту отследить взаимодействие программы и активность журнала в терминале

4. Дизассемблеры позволят программисту посмотреть ассемблерный код исполняемого файла

5. Сниферы помогут программисту проследить сетевой трафик генерируемой программой

6. Сниферы аппаратных интерфейсов позволят увидеть данные, которыми обменивается система и устройство.

7. Логи системы.

Использование языков программирования высокого уровня, таких как Java, обычно упрощает отладку, поскольку содержат такие средства как обработка исключений, сильно облегчающие поиск источника проблемы. В некоторых низкоуровневых языках, таких как Ассемблер, ошибки могут приводить к незаметным проблемам — например, повреждениям памяти или утечкам памяти, и бывает довольно трудно определить, что стало первоначальной причиной ошибки. В этих случаях, могут потребоваться изощрённые приёмы и средства отладки.

Отладка = Тестирование + Поиск ошибок + Редактирование

Виды отладки ПО, включая тестирование (в нашей стране).

1.1. Автономная отладка. Последовательное раздельное тестирование различных частей программ, входящих в ПО, с поиском и исправлением в них фиксируемых при тестировании ошибок. Она фактически включает отладку каждого программного модуля и отладку сопряжения модулей.



1.2. Комплексная отладка. Тестирование ПО в целом с поиском и исправлением фиксируемых при тестировании ошибок во всех документах (включая тексты программ ПО), относящихся к ПО в целом. К таким документам относятся определение требований к ПО, спецификация качества ПО, функциональная спецификация ПО, описание архитектуры П.О. и тексты программ ПО.

2.1. Синтаксическая отладка. Синтаксические ошибки выявляет компилятор, поэтому исправлять их достаточно легко.

2.2. Семантическая (смысловая) отладка. Ее время наступает тогда, когда синтаксических ошибок не осталось, но результаты программа выдает неверные. Здесь компилятор сам ничего выявить не сможет, хотя в среде программирования обычно существуют вспомогательные средства отладки, о которых мы еще поговорим.

Взаимосвязь процессов тестирования и отладки через алгоритм отладки.

После того как написан рабочий код производятся тестовые запуски программы на различных наборах тестовых данных.

При этом тестер или программист заранее должны получить контрольный результат, с которым будет идти сверка работы проверяемого кода.

В случае обнаружения расхождений между контрольным и фактическим результатами, начинается поиск проблемного участка кода и выявление ошибок вышеуказанными способами.

Средства автоматического тестирования исходного кода программ.

Основной прием здесь это создание тестов исходного текста, которые будут применены к тестируемому участку кода, а система тестирования сообщит об их результатах.

Примерами таких систем могут быть: встроенный модуль doctest в Python и мультиязыковая библиотека тестирования xUnit, распространяемая на условиях GNU/GPL и LGPL. Основа применения всех этих средств и техник — это разбиение одной большой задачи на ряд четких и более маленьких задач.

### **Задание.**

- 1) Проведите тестирование интерфейса пользователя средствами инструментальной среды разработки;
- 2) Проведите отладку в проекте.

### **Критерии оценивания:**

**5 баллов** выставляется, если студент:

- полно раскрыл содержание материала в объеме, предусмотренном программой и учебником;
- изложил материал грамотным языком в определенной логической последовательности, точно используя терминологию;
- продемонстрировал знание основ, сформированность и устойчивость используемых при ответе умений и навыков.

**4 балла** выставляется, если он удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков:

- в изложении допущены небольшие пробелы, не исказившие содержание ответа;
- допущены один - два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя.

**3 балла** ставится в следующих случаях:

- неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения программного материала;

- имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, исправленные после нескольких наводящих вопросов преподавателя.

**2 балла** ставится в следующих случаях:

- не раскрыто основное содержание учебного материала;
- обнаружено незнание или непонимание учеником большей или наиболее важной части учебного материала;
- допущены ошибки в определении понятий, при использовании терминологии, которые не исправлены после нескольких наводящих вопросов преподавателя.

### **Темы курсовых проектов:**

1. Разработать программное обеспечение для предметной области «Приемная комиссия»
2. Разработать программное обеспечение для предметной области «Абоненты ГТС (городской телефонной станции)»
3. Разработать программное обеспечение для предметной области «Автобусный парк»
4. Разработать программное обеспечение для предметной области «Автомагазин»
5. Разработать программное обеспечение для предметной области «Автохозяинство»
6. Разработать программное обеспечение для предметной области «Аптека»
7. Разработать программное обеспечение для предметной области «Библиотека»
8. Разработать программное обеспечение для предметной области «ГИБДД»
9. Разработать программное обеспечение для предметной области «Городское агентство недвижимости»
10. Разработать программное обеспечение для предметной области «Городское бюро по найму»
11. Разработать программное обеспечение для предметной области «Гостиница»
12. Разработать программное обеспечение для предметной области «Деканат»
13. Разработать программное обеспечение для предметной области «Журнал успеваемости»
14. Разработать программное обеспечение для предметной области «Журналистский архив»
15. Разработать программное обеспечение для предметной области «ЖЭУ»
16. Разработать программное обеспечение для предметной области «Канцелярия»
17. Разработать программное обеспечение для предметной области «Морское пароходство»
18. Разработать программное обеспечение для предметной области «Оранжерея»
19. Разработать программное обеспечение для предметной области «Отдел кадров»
20. Разработать программное обеспечение для предметной области «Сеть магазинов»
21. Разработать программное обеспечение для предметной области «Питомник»
22. Разработать программное обеспечение для предметной области «Поликлиника»
23. Разработать программное обеспечение для предметной области «Редакционное издательство»
24. Разработать программное обеспечение для предметной области «Салон красоты»
25. Разработать программное обеспечение для предметной области «Станция технического обслуживания»
26. Разработать программное обеспечение для предметной области «Стол заказов»
27. Разработать программное обеспечение для предметной области «Телеателье»
28. Разработать программное обеспечение для предметной области «Торгово-посредническое предприятие»
29. Разработать программное обеспечение для предметной области «Транспортно-экспедиционное предприятие»
30. Разработать программное обеспечение для предметной области «Турагентство»
31. Разработать программное обеспечение для предметной области «Универмаг»
32. Разработать программное обеспечение для предметной области «Химчистка/прачечная»
33. Разработать программное обеспечение для предметной области «Строительная компания»
34. Разработать программное обеспечение для предметной области «Почтовое отделение»
35. Разработать программное обеспечение для предметной области «Ремонт компьютеров».

### **Критерии оценивания:**

**5 баллов** выставляется за работу, написанную на актуальную тему и имеющую практическую значимость. При этом студенту необходимо максимально полно раскрыть тему курсовой работы, проанализировать литературные источники, использовать современные научные методики исследования, оформить с учётом требований ГОСТа.

**4 балла** выставляется за работу, которая в целом соответствующая вышеприведенным требованиям к оценке «5 баллов», но в которой допущены следующие недостатки:

- а) при раскрытии темы упущены некоторые существенные вопросы, или
- б) не нашли отражения современные научные данные, содержащиеся в литературе, или
- в) обнаружилось недостаточное использование современной нормативной базы, или
- г) допущено пять ошибок в оформлении.

Оценка снижается также за неточные ответы на вопросы.

**3 балла** ставится за работу, написанную на актуальную тему, правильно оформленную, но при защите которой студент показал поверхностные теоретические и практические знания, отсутствие умений четко ориентироваться в защищаемой теме. Оценка снижается также при наличии совокупности двух и более замечаний, указанных для оценки «4 балла», если в ходе защиты студент не смог убедительно ответить на претензии к своей работе.

В **2 балла** оценивается работа, выполненная на низком теоретическом уровне, не имеющая практической значимости, при защите которой студент не смог ответить на поставленные вопросы. По результатам оценивания курсовой работы студент вправе подать письменную апелляцию о несогласии с оценкой. Апелляция подается лично обучающимся в апелляционную комиссию не позднее следующего рабочего дня после проставления оценки в экзаменационную ведомость или экзаменационный лист.

### **3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций состоит из текущего контроля.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации и учитываются при оценивании знаний, умений, навыков и (или) опыта деятельности.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

### МДК.02.02 Инструментальные средства разработки программного обеспечения

Методические указания для студентов по освоению МДК являются частью рабочей программы МДК (РПД) (приложением к рабочей программе).

РПД – рабочая программа, утвержденная директором колледжа для изучения МДК. Она определяет цели и задачи МДК, формируемые в ходе ее изучения компетенции и их компоненты, содержание изучаемого материала, виды занятий и объем выделяемого учебного времени, а также порядок изучения и преподавания МДК.

Для самостоятельной учебной работы студента важное значение имеют разделы «Структура и содержание дисциплины (модуля)» и «Учебно-методическое и информационное обеспечение дисциплины (модуля)». В первом указываются разделы и темы изучаемой МДК, а также виды занятий и планируемый объем (в академических часах), во втором – рекомендуемая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Для подготовки к текущему контролю студенты могут воспользоваться оценочными средствами, представленными в Приложении 1 к рабочей программе МДК.

#### 1. Описание последовательности действий студента

Приступая к изучению МДК необходимо в первую очередь ознакомиться с содержанием РПД, где в разделе «Структура и содержание дисциплины (модуля)» приведено общее распределение часов аудиторных занятий и самостоятельной работы по темам МДК.

Залогом успешного освоения МДК является регулярное посещение занятий и выполнение предусмотренных программой заданий. Пропуск одного, а тем более нескольких занятий может осложнить освоение разделов курса.

Лекции имеют целью дать систематизированные основы научных знаний по содержанию МДК. При изучении и проработке теоретического материала необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы подготовить конспект, используя рекомендованные в РПД литературные источники и электронные образовательные ресурсы.

Практические занятия и лабораторные работы проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы с учебной литературой.

В процессе практического занятия, лабораторной работы, как вида учебных занятий, обучающиеся выполняют одно или несколько практических заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Выполнение обучающимися практических работ направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплины;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

При подготовке к практическому занятию необходимо изучить или повторить лекционный материал по соответствующей теме.

## **2. Самостоятельная работа студента**

Самостоятельная работа студента – самостоятельная учебная деятельность студента, организуемая колледжем и осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- воспитание самостоятельности, как личностного качества будущего специалиста.

Самостоятельная работа студента по МДК выполняется:

- самостоятельно вне расписания учебных занятий;
- с использованием современных образовательных технологий;
- работа со специальной литературой для подготовки к тестовым, практическим и лабораторным заданиям.

## **3. Рекомендации по работе с литературой и источниками**

Работу с литературой следует начинать с анализа РПД, содержащей список основной и дополнительной литературы, а также знакомства с учебно-методическими разработками.

В случае возникновения затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным.

Работа с литературой не только полезна как средство более глубокого изучения МДК, но и является неотъемлемой частью профессиональной деятельности будущего выпускника.