

**Приложение 1**  
к ПООП по специальности  
**09.02.07 Информационные  
системы и  
программирование**

**РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

**«ПМ.01. Разработка модулей программного обеспечения для  
компьютерных систем»**

Рабочая программа профессионального модуля разработана на основе ФГОС СПО 09.02.07 Информационные системы и программирование укрупнённой группы специальностей.

Рекомендована экспертной организацией: Общество с ограниченной ответственностью «Мой регион». Зарегистрировано в государственном реестре примерных основных образовательных программ под номером: 09.02.07-170511 от 11.05.2017 г.

Организация-разработчик рабочей программы: Государственное бюджетное профессиональное образовательное учреждение «Южно-Уральский государственный колледж».

Разработчики:

Фостаковская Екатерина Викторовна, преподаватель,  
Исакова Виктория Викторовна, преподаватель,  
Замятин Андрей Павлович, преподаватель.

Рассмотрена и одобрена на заседании ПЦК «Информационных технологий» Протокол №9 от «18» апреля 2022 г.

## ***СОДЕРЖАНИЕ***

- 1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ  
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
- 2. СТРУКТУРА И СОДЕРЖАНИЕ  
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
- 3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО  
МОДУЛЯ**
- 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ  
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

## 1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

*«ПМ.01. Разработка модулей программного обеспечения для компьютерных систем»*

### 1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности Разработка модулей программного обеспечения для компьютерных систем и соответствующие ему общие компетенции и профессиональные компетенции:

#### 1.1.1. Перечень общих компетенций

<b>Код</b>	<b>Наименование общих компетенций</b>
ОК 01.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения профессиональной деятельности.
ОК 3.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 06.	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 08.	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
ОК 9.	Пользоваться профессиональной документацией на государственном и иностранном языках.

#### 1.1.2. Перечень профессиональных компетенций

<b>Код</b>	<b>Наименование видов деятельности и профессиональных компетенций</b>
ВД 1	Раздел 1. Разработка программных модулей
ПК 1.1.	ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

	Раздел 2. Поддержка и тестирование программных модулей
ПК 1.3.	ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	ПК 1.4. Выполнять тестирование программных модулей
ПК 1.5.	ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода
	Раздел 3. Технологии разработки мобильных приложений
ПК 1.2.	ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.6.	ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ
	Раздел 4. Системное программирование
ПК 1.2.	ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3.	ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	<ul style="list-style-type: none"> <li>в разработке кода программного продукта на основе готовой спецификации;</li> <li>в использовании средств на этапе отладки программного продукта;</li> <li>в проведении тестирования программного модуля по определенному сценарию;</li> <li>в использовании инструментальных средств на этапе отладки программного продукта;</li> <li>в разработке мобильных приложений</li> </ul>
уметь	<ul style="list-style-type: none"> <li>осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</li> <li>создавать программу по разработанному алгоритму как отдельный модуль;</li> <li>выполнять отладку и тестирование программы на уровне модуля;</li> <li>уметь выполнять оптимизацию и рефакторинг программного кода;</li> <li>оформлять документацию на программные средства.</li> </ul>
знать	<ul style="list-style-type: none"> <li>основные этапы разработки программного обеспечения;</li> <li>основные принципы технологии структурного и объектно-ориентированного программирования;</li> <li>способы оптимизации и приемы рефакторинга;</li> <li>основные принципы отладки и тестирования программных продуктов.</li> </ul>

## 1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов \_\_\_\_\_ 891 \_\_\_\_\_  
в том числе в форме практической подготовки \_\_\_\_\_ 812 \_\_\_\_\_

Из них на освоение МДК

МДК 01.01 \_\_\_\_\_ 327 \_\_\_\_\_

МДК 01.02 \_\_\_\_\_ 114 \_\_\_\_\_

МДК 01.03 \_\_\_\_\_ 136 \_\_\_\_\_

МДК 01.04 \_\_\_\_\_ 88 \_\_\_\_\_

в том числе самостоятельная работа \_\_\_\_\_ 44 \_\_\_\_\_

курсовая работа - 25

практики, в том числе учебная \_\_\_\_\_ 144 \_\_\_\_\_

производственная \_\_\_\_\_ 72 \_\_\_\_\_

*Промежуточная аттестация \_\_\_\_\_ проводится в форме экзамена \_\_\_\_\_.*

## 2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

### 2.1. Структура профессионального модуля

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля			Объем профессионального модуля, ак. час.							
		Суммарный объем нагрузки, час.	В т.ч. в форме практ. подготовки	Работа обучающихся во взаимодействии с преподавателем							Самостоятельная работа
				Обучение по МДК			Практики				
				Всего	В том числе						
Промежут. аттест.	Лаборат. и практ. занятий	Курсовых работ (проектов)	Учебная		Производственная	Консультации					
1	2	3	4	5	6	7 <sup>40</sup>	8	9	10	11	12
ПК 1.1. - 1.2. ОК 1-10	Раздел 1. Разработка программных модулей	327	280	280	-	115	25	72	72	17	30
ПК 1.3. - 1.5. ОК 1-10	Раздел 2. Поддержка и тестирование программных модулей	114	106	106	8	52	-	36		-	-
ПК 1.2., ПК 1.6. ОК 1-10	Раздел 3. Технологии разработки мобильных приложений	136	122	122	-	60	-	36		-	14
ПК 1.2 - 1.3	Раздел 4. Системное	88	88	88	-	44	-	-		-	-

ОК 1-10	программирован ие										
ПК ОК	Учебная практика	<b>144</b>	144	-	-	-	-	<b>144</b>		-	-
ПК ОК	Производственная практика	<b>72</b>	72						<b>72</b>	-	-
<i>ПК</i> <i>ОК</i>	Промежуточная аттестация: экзамен по ПМ	<b>10</b>	-						-	-	-
	<b><i>Всего:</i></b>	<b><i>891</i></b>	<b><i>812</i></b>	<b><i>596</i></b>	<b><i>8</i></b>	<b><i>271</i></b>	<b><i>25</i></b>	<b><i>144</i></b>	<b><i>72</i></b>	<b><i>17</i></b>	<b><i>44</i></b>



**2.2. Тематический план и содержание профессионального модуля ПМ01. Разработка модулей программного обеспечения для компьютерных систем**

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)	Объем в часах
1	2	3
<i>Раздел 1. Разработка программных модулей</i>		<b>280</b>
<i>МДК. 01.01 Разработка программных модулей</i>		<b>280</b>
<i>Тема 1.1.1 Жизненный цикл ПО</i>	<b>Содержание</b>	<b>2</b>
	1. Понятие ЖЦ ПО. Модели жизненного цикла ПО: каскадная, инкрементная, спиральная.	2
	<i>в том числе в форме практической подготовки</i>	2
	<b>Тематика практических занятий и лабораторных работ</b>	-
	<i>в том числе в форме практической подготовки</i>	-
<i>Тема 1.1.2. Структурное программирование</i>	<b>Содержание</b>	<b>8</b>
	1. Парадигмы программирования. Понятие программного модуля.	2
	2. Сравнительная характеристика структурного программирования.	2
	3. Управляющие структуры: последовательное выполнение, ветвление, цикл.	2
	4. Текст программы. Описание программы. Комментарии, отступы. Правила именования. Типы данных.	2
	<i>В том числе в форме практической подготовки</i>	8

	<b>Тематика практических занятий и лабораторных работ</b>	-
	<i>в том числе в форме практической подготовки</i>	-
<b>Тема 1.1.3 Разработка пользовательского интерфейса</b>	<b>Содержание</b>	<b>48</b>
	1. Понятия WPF, XAML. Структура проекта. Пространства имен XAML. Категории элементов управления.	2
	2. Элементы Button, Label. Окно формы, его свойства.	2
	3. Элементы TextBox, PasswordBox.	2
	4. Диалоговое окно MessageBox	2
	5. Директива Name для взаимодействия между XAML и языком C#. Правила именования объектов.	2
	6. Добавление обработчика события Click. Программирование метода, обрабатывающего событие Click.	2
	7. Элемент мультимедиа Image, свойство Source.	2
	8. Полотно InkCanvas. Настройка свойств элемента InkCanvas. Панель Canvas.	2
	9. Панель StackPanel. Свойства панели StackPanel	2
	10. Элемент RadioButton. Объединение элементов RadioButton в группы.	2
	11. Элементы RepeatButton, ToggleButton. Назначение элементов.	2
	12. Элементы TextBlock, CheckBox. Назначение элементов.	2
	13. Список ListBox. Ползунок Slider. Назначение элементов.	2

	14. Прокручиваемый список ListView. Отличия списка ListView от списка ListBox.	2
	15. Полоса прокрутки ScrollView. Свойства полосы прокрутки. Ползунок Slider.	2
	16. Раскрывающийся список ComboBox.. Элемент управления компоновкой TabControl.	2
	17. Элементы Expander, TreeView. Назначение элементов.	2
	18. Элементы DataGrid, Calendar. Назначение и отличительные особенности элементов.	2
	19. Элементы украшения окон (панелей) Menu, ToolBar, ProgressBar.	2
	20. Геометрические преобразования. Управление положением элемента с помощью применения преобразований.	2
	21. Модальное диалоговое окно. Назначение модальных диалоговых окон. Написание методов, обеспечивающих работу с диалоговым окном.	2
	22. Многооконное приложение. Дочерние формы. Описание обработки события открытия дочерних окон..	2
	23. Создание стиля с использованием размеров: абсолютных, автоматических, пропорциональных. Особенности работы с файлом App.xaml. Настройка применения стиля в выбранному элементу.	2
	24. Основы работы с трехмерной графикой. Основы анимации. Анимация в XAML.	2
	<i>в том числе в форме практической подготовки</i>	48
	<b>Тематика практических занятий и лабораторных работ</b>	<b>24</b>
	1. Разработка линейной программы с использованием графических	2

	компонентов.	
	2. Разработка приложения с использованием условных конструкций и графических компонентов.	2
	3. Отладка приложения с использованием условных конструкций и графических компонентов.	2
	4. Разработка приложения с использованием циклов и графических компонентов.	2
	5. Отладка приложения с использованием циклов и графических компонентов.	2
	6. Разработка приложения с использованием одномерных массивов и графических компонентов.	2
	7. Разработка приложения с использованием двумерных массивов и графических компонентов.	2
	8. Разработка приложения с использованием динамических массивов и графических компонентов.	2
	9. Разработка приложения с несколькими формами.	2
	10. Разработка приложения с анимацией.	2
	11. Разработка игрового приложения.	2
	12. Отладка игрового приложения.	2
	<i>в том числе в форме практической подготовки</i>	<i>24</i>
<b>Тема 1.1.4 Объектно-ориентированное программирование</b>	<b>Содержание</b>	<b>24</b>
	1. Понятие класса. Члены класса. Понятие объекта.	2

	2. Модификаторы доступа. Инкапсуляция.	2
	3. Создание нескольких ссылок на объект.	2
	4. Вложенный класс.	2
	5. Статические члены класса. Обращение к статическим свойствам и методам класса.	2
	6. Сигнатура метода. Передача параметров.	2
	7. Перегрузка методов.	2
	8. Понятие конструктора. Перегрузка конструкторов.	2
	9. Понятие абстрактного класса.	2
	10. Наследование классов	2
	11. Полиморфизм.	2
	12. Библиотека динамической компоновки (dll).	2
	<i>в том числе в форме практической подготовки</i>	24
	<b>Тематика практических занятий и лабораторных работ</b>	<b>20</b>
	1. Описание пользовательского класса.	2
	2. Обращение к свойствам и методам класса.	2
	3. Обращение к статическим членам класса.	2
	4. Работа с модификаторами доступа.	2
	5. Перегрузка метода.	2

	6. Описание конструктора класса. Перегрузка конструктора.	2
	7. Создание абстрактного класса.	2
	8. Создание наследованных классов.	2
	9. Использование полиморфизма.	2
	10. Создание библиотеки динамической компоновки (dll)	2
	<i>в том числе в форме практической подготовки</i>	20
<b>Тема 1.1.5 Изучение языков программирования для кроссплатформенной среды разработки</b>	<b>Содержание</b>	<b>38</b>
	1. Создание скрипта. Назначение скрипта объектам, дублирование скрипта на одном и на разных объектах	2
	2. Поля скрипта. Типы данных.	2
	3. Мировое пространство. Вектора. Случайные числа. Программная генерация миров.	2
	4. Объекты GameObject. Создание и импортирование.	2
	5. Компоненты и переменные. Преимущества информационно-ориентированного подхода с DOTs	2
	6. Префабы, имена и теги. Поиск объектов по имени или тегу.	2
	7. Взаимодействие объектов. Поиск дочерних и родительских объектов.	2
	8. Доступ к скриптам из разных объектов	2
	9. Добавление компонентов физики, анимации и пр. к объектам.	2
	10. Наиболее важные классы и функции.	2

	11. Методы и свойства часто используемых классов.	2
	12. Система событий. Обработчики событий мыши.	2
	13. Обработчики столкновений (коллизий).	2
	14. Условные конструкции в коде.	2
	15. Циклические конструкции в коде.	2
	16. Элементы управления (UI).	2
	17. Возможность добавления элементов управления в проект программно (GUI).	2
	18. Работа со звуком в проекте.	2
	19. Сборка программы в исполняемый файл	2
	<i>в том числе в форме практической подготовки</i>	38
	<b>Тематика практических занятий и лабораторных работ</b>	<b>30</b>
	1. Создание мирового двумерного и трехмерного пространства в приложении.	2
	2. Создание объектов из примитивных объектов в приложении.	2
	3. Импортирование объектов и изменение их свойств. Создание префабов объекта в приложении.	2
	4. Изучение элементов управления. Добавление элементов управления программно и как объект UI в приложение.	2
	5. Добавление и удаление объектов из мирового пространства программно в приложение.	2

	6. Изучение компонентов объекта в приложении.	2
	7. Создание анимации объектов в приложении.	2
	8. Создание в приложении анимации объектов с помощью кода.	2
	9. Создание приложения, в котором объекту задается управление с помощью мыши.	2
	10. Создание приложения, в котором объекту задается управление с помощью клавиатуры.	2
	11. Создание приложения, в котором задается поведение объектов при столкновении. Обработка столкновений по тегам.	2
	12. Создание приложения с перемещением управляемым объектом по движущимся платформам.	2
	13. Создание приложения с захватом объекта манипулятором (мышью).	
	14. Создание приложения с сохранением данных и получением ранее сохраненных данных в коде приложения.	2
	15. Создание приложения с добавлением фонового звука и звуковых эффектов объектам.	
	<i>в том числе в форме практической подготовки</i>	<i>30</i>
<b>Тема 1.1.6 Оптимизация и рефакторинг кода. Отладка и тестирование</b>	<b>Содержание</b>	<b>10</b>
	1. Методы оптимизации программного кода.	2
	2. Цели и методы рефакторинга.	2
	3. Обработка исключительных ситуаций.	2
	4. Метод Parse(). Метод TryParse().	2



	5. Модульное тестирование	2
	<i>в том числе в форме практической подготовки</i>	<b>10</b>
	<b><i>В том числе практических занятий и лабораторных работ</i></b>	6
	1. Оптимизация и рефакторинг кода.	2
	2. Написание тестов для проверки количества символов	2
	3. Написание теста для проверки пароля: длина пароля, строчные буквы, заглавные буквы, цифры, специальные символы.	2
	<i>в том числе в форме практической подготовки</i>	6
<b>Тема 1.1.7 Основы ADO.Net</b>	<b>Содержание</b>	<b>10</b>
	1. Работа с базами данных. Введение в ADO.Net.	2
	2. Доступ к данным. Компоненты ADO.Net. и объектная модель.	2
	3. Создание таблицы, работа с записями. Работа с данными в таблицах ADO.Net.	2
	4. Свод событий ADO.Net.	2
	5. Способы создания команд	2
	<i>в том числе в форме практической подготовки</i>	10
	<b><i>Тематика практических занятий и лабораторных работ</i></b>	<b>35+25</b> (курсовая работа)
	1. Создание диаграммы вариантов использования (UML) в векторном графическом редакторе Microsoft Visio	2
	2. Создание приложения с БД. Создание базы данных. Создание таблиц. Создание и настройка полей таблиц. Импорт данных в БД. Добавление связей между таблицами. Сохранение базы данных и создание скрипта.	2

	3. Создание каркаса приложения. Работа с глобальными стилями приложения.	2
	4. Работа с базой данных в приложении: чтение данных, реализация функции добавления данных.	2
	5. Работа с базой данных в приложении: реализация функции редактирования данных.	2
	6. Работа с базой данных в приложении: реализация функции удаления данных.	2
	7. Работа с неструктурированными данными: обработка и импорт в базу данных.	2
	8. Работа с текстовыми данными: импорт списка стран	2
	9. Работа с текстовыми данными: импорт списка отелей	2
	10. Создание запросов к БД	2
	11. Создание запросов к БД	2
	12. Создание хранимых процедур	2
	13. Разработка приложения с базой данных	2
	14. Разработка приложения с базой данных	2
	15. Разработка приложения с базой данных	2
	16. Разработка приложения с базой данных	2
	17. Разработка приложения с базой данных	2
	18. Разработка приложения с базой данных	2

	19. Работа с базами данных. Название, область применения.	2
	20. Оформление документации на программные средства	2
	21. Основание для разработки документации	2
	22. Технические требования к программному модулю	2
	23. Этапы разработки технических требований	2
	24. Оформление руководства пользователя	2
	25. Оформление руководства программиста	2
	26. Оформление приложений	2
	27. Автоматизация оформления документации	2
	28. Оформление в текстовом редакторе	2
	29. Использование электронных таблиц	2
	30. Использование графического редактора	2
	<i>в том числе в форме практической подготовки</i>	<i>60</i>
<b>Примерная тематика самостоятельной учебной работы при изучении раздела 1</b>  Разработка обучающего или игрового приложения.  Темы занятий: <ol style="list-style-type: none"> <li>1. Создание базы данных</li> <li>2. Создание базы данных</li> <li>3. Информационная закрытость Создание проекта в MS Visual Studio и подключение базы данных</li> <li>4. Связность модуля Создание проекта в MS Visual Studio и подключение базы данных</li> </ol>		<b>30</b>

5. Сцепление модулей Создание проекта в MS Visual Studio и подключение базы данных 6. Форма приложения и элементы управления 7. Форма приложения и элементы управления 8. Форма приложения и элементы управления 9. Форма приложения и элементы управления 10. Разработка формы добавления перевозки 11. Разработка формы добавления перевозки 12. Разработка формы добавления перевозки 13. Разработка формы добавления перевозки 14. Программирование события клика на кнопках 15. Программирование события клика на кнопках		
<b>Учебная практика раздела 1</b>  <b>Виды работ</b>		72
<b>Содержание учебного материала (дидактические единицы)</b>	<b>Объем часов</b>	
1. Вводный инструктаж по охране труда и противопожарной безопасности.	6	
2. Разработка алгоритма решения поставленной задачи.	6	
3. Разработка алгоритма решения поставленной задачи.	6	
4. Разработка алгоритма решения поставленной задачи.	6	
5. Разработка алгоритма решения поставленной задачи.	6	
6. Создание программы по разработанному алгоритму как отдельный модуль.	6	
7. Создание программы по разработанному алгоритму как отдельный модуль.	6	
8. Создание программы по разработанному алгоритму как отдельный модуль.	6	

9. Создание программы по разработанному алгоритму как отдельный модуль.		6	
10. Создание программы по разработанному алгоритму как отдельный модуль.		6	
11. Создание программы по разработанному алгоритму как отдельный модуль.		6	
12. Оформление документации на программные средства		6	
<b>Раздел 2 Поддержка и тестирование программных модулей</b>			<b>106</b>
<b>МДК.01.02 Поддержка и тестирование программных модулей</b>			<b>106</b>
<b>Тема 1.2.1 Отладка и тестирование программного обеспечения</b>	<b>Содержание</b>		<b>36</b>
	1. Виды ошибок. Методы отладки.		2
	2. Виды ошибок. Методы отладки.		2
	3. Виды ошибок. Методы отладки.		2
	4. Классификация тестирования по уровням.		2
	5. Классификация тестирования по уровням.		2
	6. Классификация тестирования по уровням.		2
	7. Методы тестирования.		2
	8. Методы тестирования.		2
	9. Методы тестирования.		2
	10. Регрессионное тестирование.		2
	11. Регрессионное тестирование.		2
	12. Регрессионное тестирование.		2
	13. Тестирование как часть процесса верификации программного обеспечения		2

	14. Тестирование как часть процесса верификации программного обеспечения	2
	15. Тестирование как часть процесса верификации программного обеспечения	2
	16. Тестирование производительности	2
	17. Тестирование производительности	2
	18. Тестирование производительности	2
	<i>в том числе в форме практической подготовки</i>	<i>36</i>
	<b>Тематика практических занятий и лабораторных работ</b>	<b>40</b>
	1. Интеграционное тестирование	2
	2. Интеграционное тестирование	2
	3. Интеграционное тестирование	2
	4. Интеграционное тестирование	2
	5. Интеграционное тестирование	2
	6. Модульное тестирование	2
	7. Модульное тестирование	2
	8. Модульное тестирование	2
	9. Модульное тестирование	2
	10. Модульное тестирование	2
	11. Тестирование «белым ящиком»	2
	12. Тестирование «белым ящиком»	2
	13. Тестирование «белым ящиком»	2

	14. Тестирование «белым ящиком»	2
	15. Тестирование «белым ящиком»	2
	16. Тестирование «черным ящиком»	2
	17. Тестирование «черным ящиком»	2
	18. Тестирование «черным ящиком»	2
	19. Тестирование «черным ящиком»	2
	20. Тестирование «черным ящиком»	2
	<i>в том числе в форме практической подготовки</i>	<i>40</i>
<b>Тема 1.2.2. Документирование</b>	<b>Содержание</b>	<b>18</b>
	1. Автоматизация разработки технической документации Автоматизированные средства оформления документации	2
	2. Автоматизация разработки технической документации Автоматизированные средства оформления документации	2
	3. Автоматизация разработки технической документации Автоматизированные средства оформления документации	2
	4. Документирование программного обеспечения в соответствии с Единой системой программной документации.	2
	5. Документирование программного обеспечения в соответствии с Единой системой программной документации.	2
	6. Документирование программного обеспечения в соответствии с Единой системой программной документации.	2
	7. Средства разработки технической документации. Технологии разработки документов.	2
	8. Средства разработки технической документации. Технологии разработки документов.	2
	9. Средства разработки технической документации. Технологии разработки документов.	2
	<i>в том числе в форме практической подготовки</i>	<i>18</i>

	<b>Тематика практических занятий и лабораторных работ</b>		<b>12</b>
	1. Оформление документации на программировании средств с использованием инструментальных средств		2
	2. Оформление документации на программировании средств с использованием инструментальных средств		2
	3. Оформление документации на программировании средств с использованием инструментальных средств		2
	4. Оформление документации на программировании средств с использованием инструментальных средств		2
	5. Оформление документации на программировании средств с использованием инструментальных средств		2
	6. Оформление документации на программировании средств с использованием инструментальных средств		2
	<i>в том числе в форме практической подготовки</i>		<i>12</i>
<b>Учебная практика раздела 2</b>			<b>36</b>
<b>Виды работ</b>			
<b>Содержание учебного материала (дидактические единицы)</b>		<b>Объем часов</b>	
Выполнение отладки программы на уровне модуля.		6	
Применение инструментальных средств отладки программных продуктов.		6	
Применение инструментальных средств отладки программных продуктов.		6	
Проведение тестирование программного модуля по определенному сценарию.		6	
Проведение тестирование программного модуля по определенному сценарию		6	
Использование инструментальных средств на этапе тестирования программного продукта		6	
<b>Раздел 3 Разработка мобильных приложений</b>			<b>122</b>
<b>МДК.01.03 Разработка мобильных приложений</b>			<b>122</b>
	<b>Содержание</b>		<b>28</b>
	1. Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)		2



<b>Тема 1.3.1</b> Основные платформы и языки разработки мобильных приложений	2. Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)	2
	3. Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)	2
	4. Инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)	2
	5. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения	2
	6. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения	2
	7. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения	2
	8. Основные платформы мобильных приложений, сравнительная характеристика.	2
	9. Основные платформы мобильных приложений, сравнительная характеристика.	2
	10. Основные платформы мобильных приложений, сравнительная характеристика.	2
	11. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)	2
	12. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)	2
	13. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)	2
	14. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)	2
	<i>в том числе в форме практической подготовки</i>	28
	<b>Тематика практических занятий и лабораторных работ</b>	<b>12</b>
	1. Установка инструментария и настройка среды для разработки мобильных приложений	2
	2. Установка инструментария и настройка среды для разработки мобильных приложений	2
	3. Установка среды разработки мобильных приложений с применением виртуальной машины	2
	4. Установка среды разработки мобильных приложений с применением виртуальной машины	2

	5. Установка среды разработки мобильных приложений с применением виртуальной машины	2
	6. Установка среды разработки мобильных приложений с применением виртуальной машины	2
	<i>в том числе в форме практической подготовки</i>	12
<b>Тема 1.3.2</b> Создание и тестирование модулей для мобильных приложений	<b>Содержание</b>	<b>34</b>
	1. Инструментарий среды разработки мобильных приложений	2
	2. Инструментарий среды разработки мобильных приложений	2
	3. Инструментарий среды разработки мобильных приложений	2
	4. Структура типичного мобильного приложения	2
	5. Структура типичного мобильного приложения	2
	6. Структура типичного мобильного приложения	2
	7. Элементы управления и контейнеры	2
	8. Элементы управления и контейнеры	2
	9. Элементы управления и контейнеры	2
	10. Работа со списками	2
	11. Работа со списками	2
	12. Работа со списками	2
	13. Работа со списками	2
	14. Способы хранения данных	2
	15. Способы хранения данных	2
	16. Способы хранения данных	2

	17. Способы хранения данных	2
	<i>в том числе в форме практической подготовки</i>	34
	<b>Тематика практических занятий и лабораторных работ</b>	<b>48</b>
	1. Изучение и комментирование кода	2
	2. Изучение и комментирование кода	2
	3. Изучение и комментирование кода	2
	4. Настройка режима терминала	2
	5. Настройка режима терминала	2
	6. Настройка режима терминала	2
	7. Обработка событий: переключение между экранами	2
	8. Обработка событий: переключение между экранами	2
	9. Обработка событий: подсказки	2
	10. Обработка событий: подсказки	2
	11. Обработка событий: цветовая индикация	2
	12. Обработка событий: цветовая индикация	2
	13. Передача данных между модулями	2
	14. Передача данных между модулями	2
	15. Подготовка стандартных модулей	2
	16. Подготовка стандартных модулей	2
	17. Подготовка стандартных модулей	2

	18. Создание нового проекта	2
	19. Создание нового проекта	2
	20. Создание нового проекта	2
	21. Создание нового проекта	2
	22. Создание эмуляторов и подключение устройств	2
	23. Создание эмуляторов и подключение устройств	2
	24. Создание эмуляторов и подключение устройств	2
	<i>в том числе в форме практической подготовки</i>	48
<b>Примерная тематика самостоятельной учебной работы при изучении раздела 3</b>  Разработка мобильного приложения.  Темы занятий:  1. Разработка мобильного приложения «Записная книжка» 2. Разработка мобильного приложения «Записная книжка» 3. Разработка мобильного приложения «Записная книжка» 4. Разработка мобильного приложения «Записная книжка» 5. Разработка мобильного приложения «Карманный навигатор» 6. Разработка мобильного приложения «Карманный навигатор» 7. Разработка мобильного приложения «Карманный навигатор»		14
<b>Учебная практика раздела 3</b>  <b>Виды работ</b>		36
Содержание учебного материала (дидактические единицы)	Объем часов	

1. Разработка интерфейса мобильного приложения и определение компонентов для приложения	6	
2. Разработка мобильного приложения	6	
3. Разработка кода программного модуля на современных языках программирования	6	
4. Разработка кода программного модуля на современных языках программирования	6	
5. Выполнение оптимизации и рефакторинга программного кода.	6	
6. Оформление документации на программные средства	6	
<b>Раздел модуля 4. Системное программирование</b>		<b>88</b>
<b>МДК.01.04 Системное программирование</b>		<b>88</b>
<b>Тема 1.4.1 Программирование на языке низкого уровня</b>	<b>Содержание</b>	<b>44</b>
	1. Подсистемы управления ресурсами.	2
	2. Управление процессами. Определение процесса.	2
	3. Создание процессов. Завершение процессов.	2
	4. Управление потоками. Определение потока.	2
	5. Контекст потока. Состояния потока. Параллельная обработка потоков.	2
	6. Диспетчеризация и планирование потоков. Приостановка и возобновление потоков.	2
	7. Обмен данными между процессами. Передача сообщений.	2
	8. Наследование дескрипторов. Дублирование дескрипторов.	2
	9. Анонимные и именованные каналы.	2
	10. Понятие сокета. Программирование сетевых задач.	2

	11. Межпроцессное взаимодействие.	2
	12. Идентификация машины. Серверы и клиенты.	2
	13. Тестирование программ без сети.	2
	14. Дейтаграммы. Чтение файла с сервера. Динамически подключаемые библиотеки DLL.	2
	15. Распределение и освобождение локальной памяти потока.	2
	16. Концепция сервиса. Структура сервиса. Концепция виртуальной памяти.	2
	17. Организация виртуальной памяти в Windows.	2
	18. Изменение атрибутов доступа к виртуальной странице. Создание и активация буфера экрана. экрана.	2
	19. Определение и установка параметров буфера	2
	20. Функции для работы с курсором.	2
	21. Чтение и установка атрибутов консоли.	2
	22. Прокрутка буфера экрана.	2
	<i>в том числе в форме практической подготовки</i>	<i>44</i>
	<b>Тематика практических занятий и лабораторных работ</b>	<b>44</b>
	1. Использование потоков Создание потоков. Завершение потоков.	2
	2. Динамическое изменение приоритетов потоков.	
	3. Обслуживание потоков. Синхронизация потоков и процессов.	
	4. Псевдодескрипторы процессов.	
	5. Работа с анонимными каналами в Windows. Создание анонимных каналов.	
	6. Создание клиентов с анонимным каналом.	

	7. Обмен данными. Обмен данными по анонимному каналу.		
	8. Перенаправление стандартного ввода-вывода.		
	9. Создание именованных каналов.		
	10. Соединение сервера с клиентом. Соединение клиентов с именованным каналом.		
	11. Обмен данными по именованному каналу. Копирование данных из именованного канала. Передача транзакций по именованному каналу.		
	12. Определение и изменение состояния именованного канала. Получение информации об именованном канале.		
	13. Сетевое программирование с сокетами и каналами. Сетевое программирование сокетов.		
	14. Создание DLL. Динамическая загрузка и отключение DLL		
	15. Использование DLL.		
	16. Использование файла определений.		
	17. Организация функции main, ServiceMain.		
	18. Установка сервиса.Открытие доступа к сервису. Запуск сервиса.		
	19. Управление сервисом.		
	20. Резервирование, распределение и освобождение виртуальной памяти.		
	21. Чтение и установка атрибутов консоли.		
	22. Прокрутка буфера экрана.		
	<i>в том числе в форме практической подготовки</i>	<i>44</i>	
	<b>Курсовой проект (работа)</b>		<b>25</b>
	<b>Выполнение курсового проекта (работы) по модулю является обязательным.</b>		

### Примерная тематика курсовых проектов (работ)

1. Разработка автоматизированной информационной системы “Лесси” для ветеринарной клиники
2. Разработка компьютерной игры “The universe edge”
3. Разработка игрового мобильного приложения “Walking monkey”
4. Разработка автоматизированной информационной системы для мастерской по индивидуальному пошиву одежды
5. Разработка автоматизированной информационной системы для парикмахерской
6. Разработка автоматизированной информационной системы
7. Разработка автоматизированной информационной системы “Hotel Cortez”
8. Разработка автоматизированной информационной системы “Линия” для художественного магазина
9. Разработка автоматизированной информационной системы “Учет товара”
10. Создание мобильной игры “Age of Zombies”
11. Разработка мобильной игры в жанре RPG
12. Разработка автоматизированной информационной системы “Страхование” для страховой компании
13. Разработка автоматизированной информационной системы “Инвентаризация”
14. Разработка игры “Star Wars 2d”
15. Разработка мобильного приложения “Smart” для ресурсного центра
16. Разработка приложения “КМ info” для системы автоматизации движения информационных и материальных потоков на предприятии
17. Разработка автоматизированной информационной системы
18. Разработка автоматизированной информационной системы “Ателье”
19. Разработка автоматизированной информационной системы
20. Разработка автоматизированной информационной системы “Образовательная платформа POCSscience”
21. Разработка игрового приложения “The silence of the unknown”
22. Разработка приложения “Bistro” для предприятия общественного питания
- 23. Разработка мобильного приложения “Патрульная постовая служба”**



<p><b>Обязательные аудиторные учебные занятия по курсовому проекту (работе)</b></p> <ol style="list-style-type: none"> <li>1. Работа с базами данных. Название, область применения.</li> <li>2. Оформление документации на программные средства</li> <li>3. Основание для разработки документации</li> <li>4. Технические требования к программному модулю</li> <li>5. Этапы разработки технических требований</li> <li>6. Оформление руководства пользователя</li> <li>7. Оформление руководства программиста</li> <li>8. Оформление приложений</li> <li>9. Автоматизация оформления документации</li> <li>10. Оформление в текстовом редакторе</li> <li>11. Использование электронных таблиц</li> <li>12. Использование графического редактора</li> </ol>	
<p><b>Самостоятельная учебная работа обучающегося над курсовым проектом (работой):</b></p> <ol style="list-style-type: none"> <li>1. планирование выполнения курсового проекта (работы)</li> <li>2. определение задач работы</li> <li>3. изучение литературных источников</li> <li>4. проведение пред проектного исследования</li> <li>5. написание программного кода</li> <li>6. отладка программного кода</li> </ol>	
<p><b>Производственная практика</b></p>	<p>72</p>

<b><i>Вводное занятие. Инструктаж по технике безопасности</i></b>	Вводный инструктаж по технике безопасности и противопожарной защиты. Ознакомление с организацией, отделами. Изучение правил внутреннего распорядка и трудовой дисциплины	6	
<b><i>Раздел 1. Разработка программных модулей</i></b>	Определение требований к программному продукту	6	
	Составление алгоритма написания программы	6	
	Написание кода программы	6	
	Написание кода программы	6	
<b><i>Раздел 2. Поддержка и тестирование программных модулей</i></b>	Проверка правильности работы программного продукта	6	
	Выполнение процесса оптимизации кода программы	6	
	Разработка документации к программному продукту	6	
<b><i>Раздел 3. Технологии разработки мобильных приложений</i></b>	Разработка кода программного модуля для мобильной платформы	6	
	Разработка кода программного модуля для мобильной платформы	6	
<b><i>Раздел 4. Системное программирование</i></b>	Разработка программы работы с консолью в среде Windows; Работа с файлами в MS DOS (длинные имена); работа с файловым вводом-выводом в Win32	6	
<b><i>Итоговое занятие</i></b>	Оформление результатов в соответствии с требованиями ГОСТ Подведение итогов практики. Комплексный дифференцированный зачет.	6	
<b>Всего</b>			<b>891 (с учетом консультаций и экзаменов)</b>

### **3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

**3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:**

Лаборатории Программного обеспечения и сопровождения компьютерных систем, оснащенные в соответствии с п. 6.2.1. Примерной программы по специальности:

Оснащенные базы практики, в соответствии с п. 6.2.3 Примерной программы по специальности.

#### **3.2. Информационное обеспечение реализации программы**

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендованные ФУМО, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список, может быть дополнен новыми изданиями.

##### **3.2.1. Основные печатные издания**

###### **3.2.1. Печатные издания**

1. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник. Среднее профессиональное образование, профессиональная подготовка / Г.Н. Федорова. – М.: Академия, 2020. – 384 с

###### **3.2.2. Электронные издания (электронные ресурсы)**

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование: учебное пособие для СПО / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 312 с. — ISBN 978-5-8114-9817-8. — Текст: электронный // Лань: электронно-библиотечная система.
2. Анисимов К. Г., Анисимова Е. А. Программирование и создание ИТ-продуктов по приоритетным направлениям развития цифровой экономики. — Санкт-Петербург: Лань, 2021. — 200 с. — // Лань: электронно-библиотечная система.
3. Кузнецов А.С. Системное программирование: учеб. пособие. Санкт-Петербург: Лань, 2018. — 170 с. // Лань: электронно-библиотечная система.
4. Попок, Л. Е. Разработка приложений под мобильные устройства: ОС iOS : учебное пособие / Л. Е. Попок, Д. А. Замотайлова, Д. Н. Савинская. — Краснодар : КубГАУ, 2019. — 90 с. — ISBN 978-5-907247-98-7. — Текст : электронный // Лань : электронно-библиотечная система.
5. Черников, В. Разработка мобильных приложений на C# : руководство / В. Черников. — Москва : ДМК Пресс, 2020. — 188 с. — ISBN 978-5-97060-805-0. — Текст : электронный // Лань : электронно-библиотечная система.
6. Льюис, Ш. Нативная разработка мобильных приложений : руководство / Ш. Льюис, М. Данн ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2020. — 376 с. —

ISBN 978-5-97060-845-6. — Текст : электронный // Лань : электронно-библиотечная система.

7. Федотенко, М. А. Разработка мобильных приложений. Первые шаги : руководство / М. А. Федотенко. — Москва : Лаборатория знаний, 2019. — 335 с. — ISBN 978-5-00101-640-3. — Текст : электронный // Лань : электронно-библиотечная система.
8. Локтев, Д. А. Разработка кроссплатформенного мобильного приложения для работы с картами : учебно-методическое пособие / Д. А. Локтев, Д. А. Видьманов. — Москва : МГТУ им. Н.Э. Баумана, 2019. — 32 с. — ISBN 978-5-7038-5274-3. — Текст : электронный // Лань : электронно-библиотечная система.
9. Пирская, Л. В. Разработка мобильных приложений в среде Android Studio : учебное пособие / Л. В. Пирская. — Ростов-на-Дону : ЮФУ, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст : электронный // Лань : электронно-библиотечная система.
10. Междисциплинарное электронное учебно-методическое пособие ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем по специальности 09.02.03 Программирование в компьютерных системах.

### **3.2.3. Дополнительные источники**

1. Учебники по программированию <http://programm.ws/index.php>
2. Учебное пособие. Прикладное программирование: [Электронный ресурс]. URL: [http://aco.ifmo.ru/el\\_books/applied\\_programming/](http://aco.ifmo.ru/el_books/applied_programming/)

### **3.3. Организация образовательного процесса**

Освоению данного модуля предшествует освоение профильной общепрофессиональной учебной дисциплины ОП.03 «Информационные технологии».

Обязательным условием допуска к производственной практике (по профилю специальности) в рамках профессионального модуля «*Разработка модулей программного обеспечения для компьютерных систем*» является освоение программы аудиторных занятий для формирования первичных профессиональных компетенций.

### **3.4. Кадровое обеспечение образовательного процесса**

Педагогические кадры, обеспечивающие обучение по междисциплинарным курсам в рамках данного профессионального модуля – преподаватели МДК- имеют высшее профессиональное образования в области ИТ, один из трех – имеет опыт работы на предприятиях и в организациях по профилю подготовки.

Руководство практикой осуществляют преподаватели - дипломированные специалисты в области ИТ.

Руководители практики - представители организации, на базе которой проводится практика: дипломированные специалисты с образованием, соответствующим профилю специальности «Информационные системы и программирование».

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (по разделам)

Контроль и оценка результатов освоения дисциплины / модуля осуществляется педагогическим работником в процессе проведения практических и лабораторных занятий, контрольных работ, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
<b>Раздел модуля 1. Анализ и проектирование программных решений</b>		
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<p>Оценка <b>«отлично»</b> - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма</p> <p>Оценка <b>«хорошо»</b> - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры, выполнена оценка сложности алгоритма</p> <p>Оценка <b>«удовлетворительно»</b> - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p>
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Оценка <b>«отлично»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.	Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием

	<p>Оценка <b>«хорошо»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка <b>«удовлетворительно»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<b>Раздел модуля 2. Технологии тестирования программных модулей</b>		
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	<p>Оценка <b>«отлично»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«хорошо»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.4 Выполнять тестирование программных модулей	Оценка <b>«отлично»</b> - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в	Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов

	<p>соответствии со стандартами.</p> <p>Оценка <b>«хорошо»</b> - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнено тестирование модуля и оформлены результаты тестирования.</p>	тестирования программного модуля.
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода	<p>Оценка <b>«отлично»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>Оценка <b>«хорошо»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p> <p>Оценка <b>«удовлетворительно»</b> - определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<b>Раздел модуля 3. Технологии разработки мобильных приложений</b>		
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Оценка <b>«отлично»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и полностью	Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с

	<p>соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка <b>«хорошо»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка <b>«удовлетворительно»</b> - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов</p>	<p>техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.	<p>Оценка <b>«отлично»</b> - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.</p> <p>Оценка <b>«хорошо»</b> - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие выполняемых функций спецификации с незначительными отклонениями.</p> <p>Оценка <b>«удовлетворительно»</b> - разработан модуль для заданного</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по созданию модуля для заданного мобильного устройства на основе спецификации</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>



	мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.	
<b>Раздел модуля 4. Системное программирование</b>		
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	Оценка «отлично» - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.	Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного

	<p>Оценка «<b>хорошо</b>» - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки.</p> <p>Оценка «<b>удовлетворительно</b>» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	Обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. Адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения профессиональной деятельности	Использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	Демонстрация ответственности за принятые решения; обоснованность самоанализа и коррекция результатов собственной работы	
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	Взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; обоснованность	

	анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Демонстрировать грамотность устной и письменной речи, ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	Соблюдение норм поведения ввремя учебных занятий и прохождения учебной и производственной практик	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	Эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня	Эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	

физической подготовленности		
ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.	Эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке	

**Методические материалы, определяющие процедуры оценивания результатов  
освоения модуля  
Оценочные средства для текущего контроля успеваемости ДЛЯ УСТНОГО  
ОПРОСА В АУДИТОРИИ**

***Тема 1.1.1 Жизненный цикл ПО***

Вопросы с открытым ответом:

1. Что такое жизненный цикл информационной системы?
2. Каковы этапы предпроектной стадии жизненного цикла информационной системы?
3. Каковы этапы стадии проектирования в жизненном цикле информационной системы?
4. Каковы этапы стадии внедрения в жизненном цикле информационной системы?
5. Какие процессы могут возникать на стадии функционирования в жизненном цикле информационной системы?
6. Что такое модель жизненного цикла информационной системы?
7. Каков алгоритм «модели кодирования и устранения ошибок»?
8. Что такое «каскадная стратегия»?
9. Каковы достоинства и недостатки каскадной модели?
10. Что такое инкрементная стратегия?
11. Что такое спиральная стратегия?
12. Каковы достоинства спиральной модели?
13. Каковы недостатки спиральной модели?
14. Провести сравнительный анализ моделей ЖЦ ПО.
15. Что такое RAD-разработка?
16. Что такое методология XP?
17. Каковы особенности V-модели ЖЦ ПО?
18. Что такое модель Хаоса?
19. Что такое компонентно-ориентированная модель?

Критерии оценки

Отлично - Ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности. В ответе может быть допущена 1 ошибка

Хорошо - ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности, при этом допущены две-три ошибки в ответе, исправленные самостоятельно по требованию преподавателя

Удовлетворительно - ответ полный, но при этом допущены 4-5 ошибок в ответе

Неудовлетворительно - при ответе обнаружено непонимание обучающимся основного содержания

## Оценочные средства для текущего контроля успеваемости ДЛ ПРАКТИЧЕСКОГО ЗАДАНИЯ

### Тема 1.1.2 Структурное программирование

Вариант. Выполнить сравнение алгоритмов сортировки. Результат представить в графиках. Сделать вывод.

<b>Отлично</b>	<b>Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.</b>
<b>Хорошо</b>	<b>Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.</b>
<b>Удовлетворительно</b>	<b>Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.</b>
<b>Неудовлетворительно</b>	<b>Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.</b>

## Оценочные средства для текущего контроля успеваемости ДЛ ПРАКТИЧЕСКОГО ЗАДАНИЯ

### Тема 1.1.3. Объектно-ориентированное программирование

1. Создать класс Tiles (кафель), который будет содержать поля с открытым доступом: brand, size\_h, size\_w, price и метод класса getData(). В главной функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод getData().
2. Для решения задачи описать класс Traveler, создать объект класса. Вычислить по формуле значение t для данного объекта.

Задача 2. После того, как британскими учеными было доказано, что Земля полая, отважный путешественник решил узнать, что находится в центре Земли. Для этого он проделал сквозное отверстие в Земле и, снарядившись видеокамерой, прыгнул навстречу приключениям. Известно, что радиус Земли равен 6 371 км. Ускорение свободного падения  $g = 9,8 \text{ м/с}^2$ . Программа также должна спрашивать и выводить имя путешественника, название его страны, города. Рассчитать, через сколько дней и часов путешественник достигнет своей цели. Определить, успеет ли путешественник вернуться до Нового года, если до него осталась 1 неделя.  $t=S/g$ , где t – время полётопадения, S – расстояние в метрах, g – ускорение свободного падения.

3. Определить класс Children, который содержит такие поля (члены класса): закрытые — имя ребенка, фамилию и возраст, публичные — методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.

4. Преобразовать строки двумерного массива в столбцы. Класс должен содержать два метода: один заполнит массив значениями, второй произведет замену значений строк на значения столбцов.

Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

#### **Оценочные средства для текущего контроля успеваемости ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

##### ***Тема 1.1.4. Паттерны проектирования***

Используя Fabric Method (Порождающий шаблон проектирования), распределить обязанности менеджера по найму дочерним классам.

Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

#### **Оценочные средства для текущего контроля успеваемости ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

##### ***Тема 1.1.5. Событийно-управляемое программирование***

Разработать полиформное игровое приложение с анимацией.

Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.

Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

## Оценочные средства для текущего контроля успеваемости **ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

### *Тема 1.1.6. Оптимизация и рефакторинг кода*

1. Выполнить оптимизацию кода следующей программы. Структура *vec* является вектором элементов типа *float*. Функция *combine0* вычисляет результат перемножения всех элементов вектора. Эту функцию нужно оптимизировать. Размер массива = 5000, он инициализирован случайными числами.

```
typedef struct
```

```
{
```

```
long len;
```

```
float *data;
```

```
} vec;
```

```
long vec_len(vec *v) { return v->len; }
```

```
void combine0(vec *v, float *dest)
```

```
{ long i; *dest = 1;
```

```
for (i = 0; i < vec_len(v); i++)
```

```
{ *dest *= v->data[i]; } }
```

```
#define SIZE 5000
```

```
float a[SIZE];
```

```
vec v = {SIZE, a};
```

```
int main()
```

```
{ float res;
```

```
for (i = 0; i < SIZE; i++)
```

```
// инициализация вектора случайными числами
```

```
a[i] = rand();
```

```
combine0(&v, &res);
```

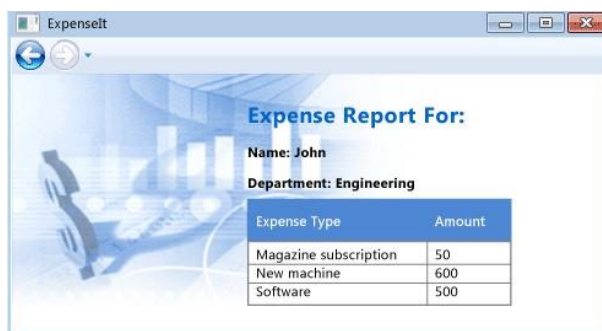


Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

### Оценочные средства для текущего контроля успеваемости **ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

#### *Тема 1.1.7. Разработка пользовательского интерфейса.*

В IDE Visual Studio на языке C# разработать следующий интерфейс пользователя.

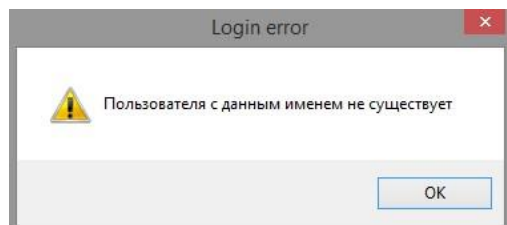


Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

### Оценочные средства для текущего контроля успеваемости **ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

### Тема 1.1.8. Основы ADO.Net

Импортировать файлы \*.csv в SQL Server. Создать приложение «Форма авторизации». Связать форму с базой данных. Обеспечить проверку пользовательского ввода в поля Логин и пароль, а также вывод соответствующих сообщений, например:



Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

### Оценочные средства для текущего контроля успеваемости **ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ** Тема 1.2.1 Отладка и тестирование программного обеспечения

1. Дана функция, которая вычисляет неотрицательную степень  $n$  числа  $x$ .

```
1 double Power(double x, int n){
2 double z=1; int i;
3 for (i=1;
4 n>=i;
5 i++)
6 {z = z*x;} /* Возврат в п.4 */
7 return z;}
```

Схематично нарисовать управляющий граф программы, отображающий поток управления программы. Определить пути и ветви.

2. Определить для следующей функции реализуемые и нереализуемые пути.

```
float H(float x,float y)
{
float H;
1 if (x*x+y*y+2<=0)
2 H = 17;
```

```

3 else H = 64;
4 return H*N+x*x;
}

```

3. Написать основной фрагмент схемы программы управления схватом робота, где интервал между моментами срабатывания схвата не определен.

4. Вставить оператор протоколирования в исходный текст метода Power:

```

double Power(double x, int n)
{
    double z=1;
    int i;
    for (i=1;n>=i;i++)
    {
        z = z*x;
    }
    return z;
}

```

5. Выполнить тестирование всех значений переменных после выполнения каждого оператора:

```

double PowerNonNeg(double x, int n)
{
    double z=1;
    int i;
    if (n>0)
    {
        for (i=1;n>=i;i++)
        {
            z = z*x;
            x,z,n,i);
        }
    }
    else printf(
        "Ошибка! Степень числа n должна быть больше 0.\n");
    return z;
}

```

Критерии оценки	Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
	Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
	Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть

		допущены несущественные неточности, устраненные с помощью преподавателя.
	Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

**Оценочные средства для текущего контроля успеваемости ДЛ** **ПРАКТИЧЕСКОГО ЗАДАНИЯ** *Тема 1.2.2. Документирование*

Оформить документацию на программный продукт с использованием инструментальных средств

1. Загрузить исходные требования Заказчика из документов Word.
2. Разработать требования в свободной текстовой форме или с помощью заданных форм (полей), например, в форме User Story, Use Cases.
3. Разработать модель UML.
4. Структурировать требования с помощью различных типов связей.
5. Распределить права доступа и ограничить видимость или возможность редактирования проектных данных.
6. Сгенерировать готовые документы по шаблону, включающему требования, модели, организованные в разделы технического задания.

Критерии оценки	Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
	Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
	Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
	Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

**Оценочные средства для текущего контроля успеваемости ДЛ** **ПРАКТИЧЕСКОГО ЗАДАНИЯ**

### **Тема 1.3.1.** Основные платформы и языки разработки мобильных приложений

1. Настроить аппаратное обеспечение для отладки приложений на устройстве.
2. Настроить операционную систему для отладки на реальном устройстве.

Критерии оценки	Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
	Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены незначительные неточности, устраненные без помощи преподавателя.
	Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены незначительные неточности, устраненные с помощью преподавателя.
	Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

### **Оценочные средства для текущего контроля успеваемости ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

### **Тема 1.3.2** Создание и тестирование модулей для мобильных приложений

1. Разработать простое мобильное приложение для мобильной операционной системы iOS. Построить пользовательский интерфейс с помощью Interface Builder. Связать графические элементы с кодом приложения. Создать обработчики событий. Протестировать готовое приложение на симуляторе iPhone 7.
2. Создать анимацию View-компонентов: изменение прозрачности, изменение размера, перемещение, поворот.

Критерии оценки	Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
	Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены незначительные неточности, устраненные без помощи преподавателя.

	Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные с помощью преподавателя.
	Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

## Оценочные средства для текущего контроля успеваемости **ДЛЯ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ**

### *Тема 1.4.1. Программирование на языке низкого уровня*

1. Выберите из предложенного списка, что может являться критерием эффективности вычислительной системы:

1. пропускная способность;
2. занятость оперативной памяти;
3. загруженность центрального процессора;
2. Системы пакетной обработки предназначены для решения задач:
  1. вычислительного характера
  2. требующих постоянного диалога с пользователем
  3. требующих решения конкретной задачи за определенный промежуток времени
3. В каких системах гарантируется выполнение задания за определенный промежуток времени:

1. пакетной обработки
2. разделения времени
3. системах реального времени
4. В системах пакетной обработки суммарное время выполнения смеси задач:
  1. равно сумме времен выполнения всех задач смеси
  2. меньше или равно суммы времен выполнения всех задач смеси
  3. больше или равно суммы времен выполнения всех задач смеси
5. В системах реального времени
  1. набор задач неизвестен заранее
  2. набор задач известен заранее
  3. известен или нет набор задач зависит от характера системы
6. Самое неэффективное использование ресурсов вычислительной системы:
  1. в системах пакетной обработки
  2. в системах разделения времени
  3. в системах реального времени
7. В многопоточных системах поток есть –
  1. заявка на ресурсы
  2. заявка на ресурс ЦП
  3. заявка на ресурс ОП
8. Потоки создаются с целью:
  1. ускорения работы процесса
  2. защиты областей памяти
  3. улучшения межпроцессного взаимодействия
9. Как с точки зрения экономии ресурсов лучше распараллелить работу:

1. создать несколько процессов
2. создать несколько потоков
3. случаи а) и б) равнозначны, можно выбирать любой из них
10. Планирование потоков игнорирует:
  1. приоритет потока
  2. время ожидания в очереди
  3. принадлежность некоторому процессу
11. В каких системах тип планирования статический
  1. реального времени
  2. разделения времени
  3. пакетной обработки
12. Состояние, которое не определено для потока в системе:
  1. выполнение
  2. синхронизация
  3. ожидание
  4. готовность
13. Каких смен состояний не существует в системе:
  1. выполнение → готовность
  2. ожидание → выполнение
  3. ожидание → готовность
  4. готовность → ожидание
14. Какой из алгоритмов планирования является централизованным:
  1. вытесняющий
  2. невытесняющий
15. При каком кванте времени в системах, использующих алгоритм квантования, время ожидания потока в очереди не зависит от длительности ее выполнения:
  1. при маленьком кванте времени
  2. при длительном кванте времени
  3. при любом кванте времени
16. Приоритет процесса не зависит от:
  1. того, является ли процесс системным или прикладным
  2. статуса пользователя
  3. требуемых процессом ресурсов
17. В каких пределах может изменяться приоритет потока в системе Windows NT:
  1. от базового приоритета процесса до нижней границы диапазона приоритета потоков реального времени
  2. от нуля до базового приоритета процесса
  3. базовый приоритет процесса  $\pm 2$
18. Каких классов прерываний нет?
  1. аппаратных
  2. асинхронных
  3. внутренних
  4. программных
19. Какие из прерываний можно считать синхронными?
  1. внешние
  2. внутренние
  3. программные
  4. динамические
20. Память с самой высокой стоимостью единицы хранения:
  1. дисковая память
  2. оперативная память

3. регистры процессора
21. Какая функция ОС по управления оперативной памятью характерна только для мультизадачных ОС:
  1. выделение памяти по запросу
  2. освобождение памяти по завершению процесса
  3. защита памяти
22. Какая стратегия управления памятью определяет, какие конкретно данные необходимо загружать в память:
  1. выборки
  2. размещения
  3. замещения
  4. загрузки
23. Виртуальные адреса являются результатом работы:
  1. пользователя
  2. транслятора
  3. компоновщика
  4. ассемблера
24. Какого типа адреса могут быть одинаковыми в разных процессах:
  1. виртуальные
  2. физические
  3. реальные
  4. сегментные
25. Недостатки распределения памяти фиксированными разделами:
  1. сложность реализации
  2. сложность защиты
  3. ограничение на число одновременно выполняющихся процессов
  4. фрагментация памяти
26. Какой процесс обязательно должен выполняться в системе памяти с перемещаемыми разделами:
  1. сжатие
  2. перемещение
  3. свопинг
27. Что из ниже перечисленного верно для свопинга:
  1. на диск выгружается неиспользуемая в настоящий момент часть процесса
  2. на диск выгружаются неиспользуемые процессом данные
  3. на диск выгружается не активный процесс
28. Таблица страниц используется для:
  1. преобразования виртуального адреса в физический
  2. для ускорения работы процесса
  3. для реализации свопинга
29. Объем страницы:
  1. выбирается по возможности максимальный
  2. выбирается минимальным
  3. для процессоров x86 стандартно равен 4 кбайта
30. Кэширование – это:
  1. способ функционирования дисковых устройств
  2. способ работы с ОП
  3. способ взаимного функционирования двух типов запоминающих устройств
31. Что может выступать в качестве кэша для ОП:
  1. дисковые устройства
  2. быстродействующая статическая память



3. виртуальная память
32. Атаки класса «отказ в обслуживании» направлены на:
  1. полный или частичный вывод ОС из строя
  2. вывод из строя аппаратуры ПК
  3. полное или частичное удаление установленного ПО
33. Какой вид многозадачности не существует?
  1. Вытесняющая многозадачность.
  2. Кооперативная (не вытесняющая) многозадачность.
  3. Симметричная многозадачность.
34. Существуют ли классификация ядер ОС по особенностям выполнения ядра в многопроцессорных системах? (учитывая, что такие системы ядром поддерживаются)
  1. Да
  2. Нет
35. Где должен располагаться код для обнаружения оборудования? (учитывая современные устройства)
  1. В ядре (или обязательных модулях, серверах для монолитных архитектур).
  2. Вне ядра, в драйверах.
36. Какое ядро современных ОС поддерживает Multiboot Specification?
  1. Windows
  2. SunOS 82
  3. MacOS
  4. Linux
  5. Все ядра BSD
37. Что означает аббревиатура PIC в контексте ОС?
  1. Programmable Interrupt Controller
  2. Past Implemented Code
  3. Position Independent Code
  4. Portable Incompatible Code
38. Какие основные преимущества микроядерной архитектуры?
  1. Упрощение переносимости
  2. Улучшение безопасности
  3. Повышенная отказоустойчивость и степень структурированности
  4. Все выше перечисленное
39. Предшественником какого современного семейства ОС была ОС Minix Эндрю Таненбаума?
  1. BSD
  2. Windows
  3. Linux
40. Нашли ли экзоядерные ОС широкое применение в современной вычислительной технике?
  1. Да
  2. Нет
41. В какой из ОС впервые был реализован стек протоколов TCP/IP?
  1. BSD
  2. Windows
  3. Linux
  4. DOS
42. Выберите не подходящее утверждение об отношении DOS к первым версиям Windows?
  1. В Windows можно было запускать приложения DOS
  2. Многие функции Windows делегировались соответствующим функциям DOS (то есть для этого производилось переключение режимов работы ЦПУ)
  3. Поддержка приложений DOS была ограниченной и неполной (при эмуляции на VDM, в рамках режима V86)

43. В какой ОС поддержка графического интерфейса пользователя (GUI) интегрирована непосредственно в ядро?
1. Windows
  2. BSD
  3. Linux
44. Укажите типы сообщений, которые могут использоваться в микроядерных ОС.
1. Синхронные и асинхронные.
  2. Только синхронные.
  3. Только асинхронные.
45. В чём главный недостаток монолитных ядер?
1. Их нельзя модифицировать во время работы
  2. Со временем они настолько разрастаются, что резко усложняется внесение каких-либо изменений
  3. Они занимают слишком много оперативной памяти
46. Укажите основное средство межпроцессного взаимодействия в микроядерных архитектурах.
1. Потоки
  2. Удалённые вызовы процедур (RPC, Remote Procedure Call)
  3. Сообщения
47. Какая нотация вызовов функций принята в системных вызовах Windows?
1. Смесь нотаций языков C и Pascal (обратный порядок аргументов, очистка стека функцией)
  2. Нотация языка Pascal (прямой порядок аргументов, очистка стека функцией)
  3. Нотация языка C (обратный порядок аргументов, очистка стека вызывающим кодом)
48. Достаточно ли установки антивирусного пакета для того, чтобы считать ОС защищенной:
1. да
  2. нет
  3. зависит от конкретных условий работы
49. Для обеспечения безопасности системы должны использоваться средства, которые при отказе переходят в состояние:
1. максимальной защиты
  2. минимальной защиты
50. При организации защиты в системе необходимо руководствоваться принципом:
1. максимальной защиты
  2. минимальной защиты
  3. баланса возможного ущерба от угрозы и затрат на ее предотвращение
51. Слабости парольной защиты:
1. трудность распознавания
  2. возможность раскрытия пароля путем подбора
  3. возможность обхода парольной защиты
52. Процесс авторизации – это процесс
1. ввода пользователем учетной информации
  2. доказательства того, что пользователь тот, за кого себя выдает
  3. выполнения действий, необходимых для того, чтобы пользователь мог начать работу в системе
53. В асимметричных системах шифрования:
1. ключ шифрования совпадает с ключом расшифрования
  2. ключ шифрования отличается от ключа расшифрования
  3. ключи генерируются случайным образом
54. Правила разграничения доступа не должны позволять:
1. присутствия ничейных объектов в системе
  2. присутствия объектов, недоступных для администраторов системы
  3. присутствия всем доступных объектов

55. Файловая система является частью:
1. дисковых систем
  2. драйверов дисков
  3. ОС
  4. пользовательских программ
56. Какую структуру образуют файлы в ФС (файловой системе) FAT?
1. древовидную
  2. сетевую
  3. реляционную
  4. плоскую
57. Определите, какое это имя файла: USER\DO\FEDYA.DOC:
1. полное
  2. простое
  3. относительное
58. Одна ФС в системах Windows занимает, как правило:
1. 1 физический диск
  2. 1 логический диск
  3. 1 раздел диска
59. В ФС FAT атрибуты файлов хранятся
1. вместе с файлом
  2. в каталогах
  3. в индексных дескрипторах
  4. в таблицах FAT
60. Диски – это память:
1. с последовательным доступом
  2. с индексно-последовательным доступом
  3. с прямым доступом
61. Какой разметки нет на диске?
1. дорожек
  2. кластеров
  3. цилиндров
  4. секторов
62. Минимальная единица, участвующая в операциях обмена с дисковым устройством:
1. байт
  2. сектор
  3. дорожка
  4. цилиндр
63. Размер логического диска:
1. меньше или равен размеру раздела
  2. равен размеру раздела
  3. больше или равен размеру раздела
64. ОС Windows поддерживают следующие типы разделов:
1. основной
  2. базовый
  3. подкачки
  4. дополнительный
65. Раздел, с которого загружается ОС при запуске компьютера называется:
1. загрузочным
  2. основным
  3. активным
66. Минимальный фактический размер файла на диске равен:

1. 1 биту
  2. 1 байту
  3. 1 сектору
  4. 1 кластеру
67. На диске не может быть кластера размером:
1. 512 байт
  2. 1024 байта
  3. 1536 байт
  4. 2048 байт
68. Числовое значение –12, 16, 32 – в ФС FAT отражает:
1. размер кластера на диске
  2. разрядность элемента в таблице FAT
  3. допустимое количество символов в имени файла
69. Максимальный размер диска, поддерживаемого FAT16:
1. практически неограничен
  2. 512 Мбайт
  3. 2 Гбайта
70. Недостатки ФС FAT:
1. сложность реализации
  2. не поддерживают разграничения доступа к файлам и каталогам
  3. не поддерживают длинных имен файлов
  4. не содержат средств поддержки отказоустойчивости
71. Какие функции выполняет операционная система?
1. обеспечение организации и хранения файлов
  2. организация диалога с пользователем, управления аппаратурой и ресурсами компьютера
  3. все выше перечисленные
72. Где находится BIOS?
1. в оперативно-запоминающем устройстве (ОЗУ)
  2. на винчестере
  3. на CD-ROM
  4. в постоянно-запоминающем устройстве (ПЗУ)
73. Папка, в которую временно попадают удалённые объекты, называется ...
1. Корзина
  2. Оперативная
  3. Портфель
  4. Блокнот
74. Текущий диск - это ...
1. диск, с которым пользователь работает в данный момент времени
  2. CD-ROM
  3. жесткий диск
  4. диск, в котором хранится операционная система
75. ОС Windows поддерживает длинные имена файлов. Длинным именем файла считается ...
1. любое имя файла без ограничения на количество символов в имени файла
  2. любое имя файла латинскими буквами, не превышающее 255 символов
  3. любое имя файла, не превышающее 255 символов
76. Внутренние команды - это ...
1. команды, предназначенные для создания файлов и каталогов
  2. команды, встроенные в DOS
  3. команды, которые имеют расширения .sys, .exe, .com
77. Загрузчик операционной системы MS DOS служит для ...
1. загрузки программ в оперативную память ЭВМ

2. обработки команд, введенных пользователем
  3. считывания в память модулей операционной системы io.sys и msdos.sys
  4. подключения устройств ввода-вывода
78. Какие команды DOS называются внешними?
1. команды, предназначенные только для работы с периферийными устройствами
  2. команды, хранящиеся на диске в виде отдельных программ и вызываемые по мере необходимости
  3. все команды, которые можно реализовать с помощью DOS
79. BIOS - это ...
1. игровая программа
  2. диалоговая оболочка
  3. базовая система ввода-вывода
  4. командный язык операционной системы
80. Операционная система сети включает в себя управляющие и обслуживающие программы. К управляющим относятся
1. Межпрограммный доступ
  2. Доступ отдельных прикладных программ к ресурсам сети
  3. Синхронизация работы прикладных программных средств
  4. Обмен информации между программами и др.
  5. Все выше перечисленные
81. Какой вид многозадачности не существует?
1. Вытесняющая многозадачность.
  2. Кооперативная (не вытесняющая) многозадачность.
  3. Симметричная многозадачность.
82. Существуют ли классификация ядер ОС по особенностям выполнения ядра в многопроцессорных системах? (учитывая, что такие системы ядром поддерживаются)
1. Да
  2. Нет
83. Где должен располагаться код для обнаружения оборудования? (учитывая современные устройства)
1. В ядре (или обязательных модулях, серверах для немонолитных архитектур).
  2. Вне ядра, в драйверах.
84. Какое ядро современных ОС поддерживает Multiboot Specification?
1. Windows
  2. SunOS 82
  3. MacOS
  4. Linux
  5. Все ядра BSD
85. Что означает аббревиатура PIC в контексте ОС?
1. Programmable Interrupt Controller
  2. Past Implemented Code
  3. Position Independent Code
  4. Portable Incompatible Code
86. Какие основные преимущества микроядерной архитектуры?
1. Упрощение переносимости
  2. Улучшение безопасности
  3. Повышенная отказоустойчивость и степень структурированности
  4. Все выше перечисленное
87. Предшественником какого современного семейства ОС была ОС Minix Эндрю Таненбаума?
1. BSD
  2. Windows

3. Linux
88. Нашли ли экзоядерные ОС широкое применение в современной вычислительной технике?
  1. Да
  2. Нет
89. В какой из ОС впервые был реализован стек протоколов TCP/IP?
  1. BSD
  2. Windows
  3. Linux
  4. DOS
90. Выберите не подходящее утверждение об отношении DOS к первым версиям Windows?
  1. В Windows можно было запускать приложения DOS
  2. Многие функции Windows делегировались соответствующим функциям DOS (то есть для этого производилось переключение режимов работы ЦПУ)
  3. Поддержка приложений DOS была ограниченной и неполной (при эмуляции на VDM, в рамках режима V86)
91. В какой ОС поддержка графического интерфейса пользователя (GUI) интегрирована непосредственно в ядро?
  1. Windows
  2. BSD
  3. Linux
92. Укажите типы сообщений, которые могут использоваться в микроядерных ОС.
  1. Синхронные и асинхронные.
  2. Только синхронные.
  3. Только асинхронные.
93. В чём главный недостаток монолитных ядер?
  1. Их нельзя модифицировать во время работы
  2. Со временем они настолько разрастаются, что резко усложняется внесение каких-либо изменений
  3. Они занимают слишком много оперативной памяти
94. Укажите основное средство межпроцессного взаимодействия в микроядерных архитектурах.
  1. Потoki
  2. Удалённые вызовы процедур (RPC, Remote Procedure Call)
  3. Сообщения
95. Какая нотация вызовов функций принята в системных вызовах Windows?
  1. Смесь нотаций языков C и Pascal (обратный порядок аргументов, очистка стека функцией)
  2. Нотация языка Pascal (прямой порядок аргументов, очистка стека функцией)
  3. Нотация языка C (обратный порядок аргументов, очистка стека вызывающим кодом)
96. Для выполнения каких операций оптимизирована серверная операционная система Novell NetWare?
  1. доступ к файлам
  2. доступ к файлам и печать
  3. почтовая служба
97. Какие из этих ОС могут использоваться для построения одноранговых сетей?
  1. NetWare
  2. Windows 95/98
  3. MS-DOS
98. Какие задачи не выполняет ОС при обмене с периферийным устройством?
  1. решает, может ли быть выполнена требуемая операция обмена
  2. передает запрос драйверу ПУ
  3. принимает информацию из сети от устройства управления ПУ
99. Сколько выделенных серверов может одновременно работать в сети?

1. нет специальных ограничений
  2. только один
  3. по числу требуемых в сети служб — для каждой сетевой службы отдельный выделенный сервер
100. Пусть сеть состоит из идентичных компьютеров, на которых установлены однотипные ОС. За одним из компьютеров административно закреплены функции по обслуживанию запросов остальных компьютеров(все пользователи сети хранят свои файлы на диске этого компьютера). К какому типу сети вы отнесете эту сеть?
1. сеть с выделенным сервером
  2. одноранговая сеть
  3. гибридная сеть

Ответы:

1. 1
2. 1
3. 3
4. 1
5. 3
6. 1
7. 3
8. 1
9. 3
10. 3
11. 3
12. 4
13. 1
14. 2
15. 3
16. 2
17. 3
18. 4
19. 2
20. 3
21. 3
22. 1
23. 2
24. 1
25. 3
26. 3
27. 3
28. 1
29. 3
30. 3
31. 1
32. 1
33. 2
34. 3
35. 1
36. 1
37. 4

38. 3  
39. 3  
40. 3  
41. 2  
42. 1  
43. 3  
44. 1  
45. 1  
46. 2  
47. 3  
48. 1  
49. 3  
50. 1  
51. 1  
52. 4  
53. 3  
54. 3  
55. 3  
56. 2  
57. 1  
58. 3  
59. 1  
60. 1  
61. 2  
62. 3  
63. 1  
64. 1  
65. 3  
66. 4  
67. 3  
68. 2  
69. 3  
70. 2  
71. 2  
72. 4  
73. 1  
74. 1  
75. 3  
76. 2  
77. 3  
78. 2  
79. 3  
80. 5  
81. 3  
82. 1  
83. 1  
84. 4  
85. 3  
86. 3  
87. 3  
88. 2



- 89. 1
- 90. 3
- 91. 1
- 92. 1
- 93. 2
- 94. 3
- 95. 1
- 96. 2
- 97. 2
- 98. 3
- 99. 2
- 100.2

**Оценочные средства для текущего контроля успеваемости ДЛЯ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ Тема 1.4.1. Программирование на языке низкого уровня**

1. Выберите из предложенного списка, что может являться критерием эффективности вычислительной системы:
  1. пропускная способность;
  2. занятость оперативной памяти;
  3. загруженность центрального процессора;
3. Системы пакетной обработки предназначены для решения задач:
  1. вычислительного характера
  2. требующих постоянного диалога с пользователем
  3. требующих решения конкретной задачи за определенный промежуток времени
4. В каких системах гарантируется выполнение задания за определенный промежуток времени:
  1. пакетной обработки
  2. разделения времени
  3. системах реального времени
5. В системах пакетной обработки суммарное время выполнения смеси задач:
  1. равно сумме времен выполнения всех задач смеси
  2. меньше или равно суммы времен выполнения всех задач смеси
  3. больше или равно суммы времен выполнения всех задач смеси
6. В системах реального времени
  1. набор задач неизвестен заранее
  2. набор задач известен заранее
  3. известен или нет набор задач зависит от характера системы
7. Самое неэффективное использование ресурсов вычислительной системы:
  1. в системах пакетной обработки
  2. в системах разделения времени
  3. в системах реального времени
8. В многопоточных системах поток есть –
  1. заявка на ресурсы
  2. заявка на ресурс ЦП
  3. заявка на ресурс ОП
9. Потоки создаются с целью:
  1. ускорения работы процесса
  2. защиты областей памяти
  3. улучшения межпроцессного взаимодействия
10. Как с точки зрения экономии ресурсов лучше распараллелить работу:
  1. создать несколько процессов

2. создать несколько потоков
3. случаи а) и б) равнозначны, можно выбирать любой из них
11. Планирование потоков игнорирует:
  1. приоритет потока
  2. время ожидания в очереди
  3. принадлежность некоторому процессу
12. В каких системах тип планирования статический
  1. реального времени
  2. разделения времени
  3. пакетной обработки
13. Состояние, которое не определено для потока в системе:
  5. выполнение
    1. синхронизация
    2. ожидание
    3. готовность
14. Каких смен состояний не существует в системе:
  1. выполнение → готовность
  2. ожидание → выполнение
  3. ожидание → готовность
  4. готовность → ожидание
15. Какой из алгоритмов планирования является централизованным:
  1. вытесняющий
  2. невытесняющий
16. При каком кванте времени в системах, использующих алгоритм квантования, время ожидания потока в очереди не зависит от длительности ее выполнения:
  1. при маленьком кванте времени
  2. при длительном кванте времени
  3. при любом кванте времени
17. Приоритет процесса не зависит от:
  1. того, является ли процесс системным или прикладным
  2. статуса пользователя
  3. требуемых процессом ресурсов
18. В каких пределах может изменяться приоритет потока в системе Windows NT:
  1. от базового приоритета процесса до нижней границы диапазона приоритета потоков реального времени
  2. от нуля до базового приоритета процесса
  3. базовый приоритет процесса  $\pm 2$
19. Каких классов прерываний нет?
  1. аппаратных
  2. асинхронных
  3. внутренних
  4. программных
20. Какие из прерываний можно считать синхронными?
  1. внешние
  2. внутренние
  3. программные
  4. динамические
21. Память с самой высокой стоимостью единицы хранения:
  1. дисковая память
  2. оперативная память
  3. регистры процессора

22. Какая функция ОС по управления оперативной памятью характерна только для мультизадачных ОС:
1. выделение памяти по запросу
  2. освобождение памяти по завершению процесса
  3. защита памяти
23. Какая стратегия управления памятью определяет, какие конкретно данные необходимо загружать в память:
1. выборки
  2. размещения
  3. замещения
  4. загрузки
24. Виртуальные адреса являются результатом работы:
1. пользователя
  2. транслятора
  3. компоновщика
  4. ассемблера
25. Какого типа адреса могут быть одинаковыми в разных процессах:
1. виртуальные
  2. физические
  3. реальные
  4. сегментные
26. Недостатки распределения памяти фиксированными разделами:
1. сложность реализации
  2. сложность защиты
  3. ограничение на число одновременно выполняющихся процессов
  4. фрагментация памяти
27. Какой процесс обязательно должен выполняться в системе памяти с перемещаемыми разделами:
1. сжатие
  2. перемещение
  3. свопинг
28. Что из ниже перечисленного верно для свопинга:
1. на диск выгружается неиспользуемая в настоящий момент часть процесса
  2. на диск выгружаются неиспользуемые процессом данные
  3. на диск выгружается не активный процесс
29. Таблица страниц используется для:
1. преобразования виртуального адреса в физический
  2. для ускорения работы процесса
  3. для реализации свопинга
30. Объем страницы:
1. выбирается по возможности максимальный
  2. выбирается минимальным
  3. для процессоров x86 стандартно равен 4 кбайта
31. Кэширование – это:
1. способ функционирования дисковых устройств
  2. способ работы с ОП
  3. способ взаимного функционирования двух типов запоминающих устройств
32. Что может выступать в качестве кэша для ОП:
1. дисковые устройства
  2. быстродействующая статическая память
  3. виртуальная память

33. Атаки класса «отказ в обслуживании» направлены на:
1. полный или частичный вывод ОС из строя
  2. вывод из строя аппаратуры ПК
  3. полное или частичное удаление установленного ПО
34. Какой вид многозадачности не существует?
1. Вытесняющая многозадачность.
  2. Кооперативная (не вытесняющая) многозадачность.
  3. Симметричная многозадачность.
35. Существуют ли классификация ядер ОС по особенностям выполнения ядра в многопроцессорных системах? (учитывая, что такие системы ядром поддерживаются)
1. Да
  2. Нет
36. Где должен располагаться код для обнаружения оборудования? (учитывая современные устройства)
1. В ядре (или обязательных модулях, серверах для монолитных архитектур).
  2. Вне ядра, в драйверах.
37. Какое ядро современных ОС поддерживает Multiboot Specification?
1. Windows
  2. SunOS 82
  3. MacOS
  4. Linux
  5. Все ядра BSD
38. Что означает аббревиатура PIC в контексте ОС?
1. Programmable Interrupt Controller
  2. Past Implemented Code
  3. Position Independent Code
  4. Portable Incompatible Code
39. Какие основные преимущества микроядерной архитектуры?
1. Упрощение переносимости
  2. Улучшение безопасности
  3. Повышенная отказоустойчивость и степень структурированности
  4. Все выше перечисленное
40. Предшественником какого современного семейства ОС была ОС Minix Эндрю Таненбаума?
1. BSD
  2. Windows
  3. Linux
41. Нашли ли экзоядерные ОС широкое применение в современной вычислительной технике?
1. Да
  2. Нет
42. В какой из ОС впервые был реализован стек протоколов TCP/IP?
1. BSD
  2. Windows
  3. Linux
  4. DOS
43. Выберите не подходящее утверждение об отношении DOS к первым версиям Windows?
1. В Windows можно было запускать приложения DOS
  2. Многие функции Windows делегировались соответствующим функциям DOS (то есть для этого производилось переключение режимов работы ЦПУ)
  3. Поддержка приложений DOS была ограниченной и неполной (при эмуляции на VDM, в рамках режима V86)

44. В какой ОС поддержка графического интерфейса пользователя (GUI) интегрирована непосредственно в ядро?
1. Windows
  2. BSD
  3. Linux
45. Укажите типы сообщений, которые могут использоваться в микроядерных ОС.
1. Синхронные и асинхронные.
  2. Только синхронные.
  3. Только асинхронные.
46. В чём главный недостаток монолитных ядер?
1. Их нельзя модифицировать во время работы
  2. Со временем они настолько разрастаются, что резко усложняется внесение каких-либо изменений
  3. Они занимают слишком много оперативной памяти
47. Укажите основное средство межпроцессного взаимодействия в микроядерных архитектурах.
1. Потоки
  2. Удалённые вызовы процедур (RPC, Remote Procedure Call)
  3. Сообщения
48. Какая нотация вызовов функций принята в системных вызовах Windows?
1. Смесь нотаций языков C и Pascal (обратный порядок аргументов, очистка стека функцией)
  2. Нотация языка Pascal (прямой порядок аргументов, очистка стека функцией)
  3. Нотация языка C (обратный порядок аргументов, очистка стека вызывающим кодом)
49. Достаточно ли установки антивирусного пакета для того, чтобы считать ОС защищенной:
1. да
  2. нет
  3. зависит от конкретных условий работы
50. Для обеспечения безопасности системы должны использоваться средства, которые при отказе переходят в состояние:
1. максимальной защиты
  2. минимальной защиты
51. При организации защиты в системе необходимо руководствоваться принципом:
1. максимальной защиты
  2. минимальной защиты
  3. баланса возможного ущерба от угрозы и затрат на ее предотвращение
52. Слабости парольной защиты:
1. трудность распознавания
  2. возможность раскрытия пароля путем подбора
  3. возможность обхода парольной защиты
53. Процесс авторизации – это процесс
1. ввода пользователем учетной информации
  2. доказательства того, что пользователь тот, за кого себя выдает
  3. выполнения действий, необходимых для того, чтобы пользователь мог начать работу в системе
54. В асимметричных системах шифрования:
1. ключ шифрования совпадает с ключом расшифрования
  2. ключ шифрования отличается от ключа расшифрования
  3. ключи генерируются случайным образом
55. Правила разграничения доступа не должны позволять:
1. присутствия ничейных объектов в системе
  2. присутствия объектов, недоступных для администраторов системы

3. присутствия всем доступных объектов
56. Файловая система является частью:
  1. дисковых систем
  2. драйверов дисков
  3. ОС
  4. пользовательских программ
57. Какую структуру образуют файлы в ФС (файловой системе) FAT?
  1. древовидную
  2. сетевую
  3. реляционную
  4. плоскую
58. Определите, какое это имя файла: USER\DO\FEDYA.DOC:
  1. полное
  2. простое
  3. относительное
59. Одна ФС в системах Windows занимает, как правило:
  1. 1 физический диск
  2. 1 логический диск
  3. 1 раздел диска
60. В ФС FAT атрибуты файлов хранятся
  1. вместе с файлом
  2. в каталогах
  3. в индексных дескрипторах
  4. в таблицах FAT
61. Диски – это память:
  1. с последовательным доступом
  2. с индексно-последовательным доступом
  3. с прямым доступом
62. Какой разметки нет на диске?
  1. дорожек
  2. кластеров
  3. цилиндров
  4. секторов
63. Минимальная единица, участвующая в операциях обмена с дисковым устройством:
  1. байт
  2. сектор
  3. дорожка
  4. цилиндр
64. Размер логического диска:
  1. меньше или равен размеру раздела
  2. равен размеру раздела
  3. больше или равен размеру раздела
65. ОС Windows поддерживают следующие типы разделов:
  1. основной
  2. базовый
  3. подкачки
  4. дополнительный
66. Раздел, с которого загружается ОС при запуске компьютера называется:
  1. загрузочным
  2. основным
  3. активным

67. Минимальный фактический размер файла на диске равен:
1. 1 биту
  2. 1 байту
  3. 1 сектору
  4. 1 кластеру
68. На диске не может быть кластера размером:
1. 512 байт
  2. 1024 байта
  3. 1536 байт
  4. 2048 байт
69. Числовое значение –12, 16, 32 – в ФС FAT отражает:
1. размер кластера на диске
  2. разрядность элемента в таблице FAT
  3. допустимое количество символов в имени файла
70. Максимальный размер диска, поддерживаемого FAT16:
1. практически неограничен
  2. 512 Мбайт
  3. 2 Гбайта
71. Недостатки ФС FAT:
1. сложность реализации
  2. не поддерживают разграничения доступа к файлам и каталогам
  3. не поддерживают длинных имен файлов
  4. не содержат средств поддержки отказоустойчивости
72. Какие функции выполняет операционная система?
1. обеспечение организации и хранения файлов
  2. организация диалога с пользователем, управления аппаратурой и ресурсами компьютера
  3. все выше перечисленные
73. Где находится BIOS?
1. в оперативно-запоминающем устройстве (ОЗУ)
  2. на винчестере
  3. на CD-ROM
  4. в постоянно-запоминающем устройстве (ПЗУ)
74. Папка, в которую временно попадают удалённые объекты, называется ...
1. Корзина
  2. Оперативная
  3. Портфель
  4. Блокнот
75. Текущий диск - это ...
1. диск, с которым пользователь работает в данный момент времени
  2. CD-ROM
  3. жесткий диск
  4. диск, в котором хранится операционная система
76. ОС Windows поддерживает длинные имена файлов. Длинным именем файла считается ...
1. любое имя файла без ограничения на количество символов в имени файла
  2. любое имя файла латинскими буквами, не превышающее 255 символов
  3. любое имя файла, не превышающее 255 символов
77. Внутренние команды - это ...
1. команды, предназначенные для создания файлов и каталогов
  2. команды, встроенные в DOS
  3. команды, которые имеют расширения .sys, .exe, .com
78. Загрузчик операционной системы MS DOS служит для ...

1. загрузки программ в оперативную память ЭВМ
2. обработки команд, введенных пользователем
3. считывания в память модулей операционной системы io.sys и msdos.sys
4. подключения устройств ввода-вывода
79. Какие команды DOS называются внешними?
  1. команды, предназначенные только для работы с периферийными устройствами
  2. команды, хранящиеся на диске в виде отдельных программ и вызываемые по мере необходимости
  3. все команды, которые можно реализовать с помощью DOS
80. BIOS - это ...
  1. игровая программа
  2. диалоговая оболочка
  3. базовая система ввода-вывода
  4. командный язык операционной системы
81. Операционная система сети включает в себя управляющие и обслуживающие программы. К управляющим относятся
  1. Межпрограммный доступ
  2. Доступ отдельных прикладных программ к ресурсам сети
  3. Синхронизация работы прикладных программных средств
  4. Обмен информации между программами и др.
  5. Все выше перечисленные
82. Какой вид многозадачности не существует?
  1. Вытесняющая многозадачность.
  2. Кооперативная (не вытесняющая) многозадачность.
  3. Симметричная многозадачность.
83. Существуют ли классификация ядер ОС по особенностям выполнения ядра в многопроцессорных системах? (учитывая, что такие системы ядром поддерживаются)
  1. Да
  2. Нет
84. Где должен располагаться код для обнаружения оборудования? (учитывая современные устройства)
  1. В ядре (или обязательных модулях, серверах для немонолитных архитектур).
  2. Вне ядра, в драйверах.
85. Какое ядро современных ОС поддерживает Multiboot Specification?
  1. Windows
  2. SunOS 82
  3. MacOS
  4. Linux
  5. Все ядра BSD
86. Что означает аббревиатура PIC в контексте ОС?
  1. Programmable Interrupt Controller
  2. Past Implemented Code
  3. Position Independent Code
  4. Portable Incompatible Code
87. Какие основные преимущества микроядерной архитектуры?
  1. Упрощение переносимости
  2. Улучшение безопасности
  3. Повышенная отказоустойчивость и степень структурированности
  4. Все выше перечисленное
88. Предшественником какого современного семейства ОС была ОС Minix Эндрю Таненбаума?



1. BSD
2. Windows
3. Linux
89. Нашли ли экзоядерные ОС широкое применение в современной вычислительной технике?
  1. Да
  2. Нет
90. В какой из ОС впервые был реализован стек протоколов TCP/IP?
  1. BSD
  2. Windows
  3. Linux
  4. DOS
91. Выберите не подходящее утверждение об отношении DOS к первым версиям Windows?
  1. В Windows можно было запускать приложения DOS
  2. Многие функции Windows делегировались соответствующим функциям DOS (то есть для этого производилось переключение режимов работы ЦПУ)
  3. Поддержка приложений DOS была ограниченной и неполной (при эмуляции на VDM, в рамках режима V86)
92. В какой ОС поддержка графического интерфейса пользователя (GUI) интегрирована непосредственно в ядро?
  1. Windows
  2. BSD
  3. Linux
93. Укажите типы сообщений, которые могут использоваться в микроядерных ОС.
  1. Синхронные и асинхронные.
  2. Только синхронные.
  3. Только асинхронные.
94. В чём главный недостаток монолитных ядер?
  1. Их нельзя модифицировать во время работы
  2. Со временем они настолько разрастаются, что резко усложняется внесение каких-либо изменений
  3. Они занимают слишком много оперативной памяти
95. Укажите основное средство межпроцессного взаимодействия в микроядерных архитектурах.
  1. Потоки
  2. Удалённые вызовы процедур (RPC, Remote Procedure Call)
  3. Сообщения
96. Какая нотация вызовов функций принята в системных вызовах Windows?
  1. Смесь нотаций языков C и Pascal (обратный порядок аргументов, очистка стека функцией)
  2. Нотация языка Pascal (прямой порядок аргументов, очистка стека функцией)
  3. Нотация языка C (обратный порядок аргументов, очистка стека вызывающим кодом)
97. Для выполнения каких операций оптимизирована серверная операционная система Novell NetWare?
  1. доступ к файлам
  2. доступ к файлам и печать
  3. почтовая служба
98. Какие из этих ОС могут использоваться для построения одноранговых сетей?
  1. NetWare
  2. Windows 95/98
  3. MS-DOS
99. Какие задачи не выполняет ОС при обмене с периферийным устройством?
  1. решает, может ли быть выполнена требуемая операция обмена

2. передает запрос драйверу ПУ
3. принимает информацию из сети от устройства управления ПУ
100. Сколько выделенных серверов может одновременно работать в сети?
  1. нет специальных ограничений
  2. только один
  3. по числу требуемых в сети служб — для каждой сетевой службы отдельный выделенный сервер
101. Пусть сеть состоит из идентичных компьютеров, на которых установлены однотипные ОС. За одним из компьютеров административно закреплены функции по обслуживанию запросов остальных компьютеров (все пользователи сети хранят свои файлы на диске этого компьютера). К какому типу сети вы отнесете эту сеть?
  1. сеть с выделенным сервером
  2. одноранговая сеть
  3. гибридная сеть

Ответ	122.1	142.1	162.3	182.1
ы:	123.2	143.3	163.1	183.1
	124.1	144.1	164.1	184.4
101.1	125.3	145.1	165.3	185.3
102.1	126.3	146.2	166.4	186.3
103.3	127.3	147.3	167.3	187.3
104.1	128.1	148.1	168.2	188.2
105.3	129.3	149.3	169.3	189.1
106.1	130.3	150.1	170.2	190.3
107.3	131.1	151.1	171.2	191.1
108.1	132.1	152.4	172.4	192.1
109.3	133.2	153.3	173.1	193.2
110.3	134.3	154.3	174.1	194.3
111.3	135.1	155.3	175.3	195.1
112.4	136.1	156.2	176.2	196.2
113.1	137.4	157.1	177.3	197.2
114.2	138.3	158.3	178.2	198.3
115.3	139.3	159.1	179.3	199.2
116.2	140.3	160.1	180.5	200.2
117.3	141.2	161.2	181.3	
118.4				
119.2				
120.3				
121.3				

Критерии оценки:

Отлично	Ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности. В тесте с выбором варианта может быть допущена 1 ошибка
Хорошо	ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности, при этом допущены две-три ошибки в тесте с выбором, исправленные самостоятельно по требованию преподавателя
Удовлетворительно	ответ полный, но при этом допущены 4-5 ошибок в тесте с выбором

Неудовлетворительно	при ответе обнаружено непонимание обучающимся основного содержания и допущены более 5 ошибок в тесте с выбором ответа
---------------------	---

**Оценочные средства для текущего контроля успеваемости ДЛЯ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ Тема 1.4.1. Программирование на языке низкого уровня**

1 Виртуальные адреса являются результатом работы:

- а) пользователя;
- б) транслятора;
- в) компоновщика;
- г) ассемблера.

2 Какого типа адреса могут быть одинаковыми в разных процессах:

- а) виртуальные;
- б) физические;
- в) реальные;
- г) сегментные.

3 Недостатки распределения памяти фиксированными разделами:

- а) сложность реализации;
- б) сложность защиты;
- в) ограничение на число одновременно выполняющихся процессов;
- г) фрагментация памяти.

4 Все файлы и каталоги в системе NTFS однозначно идентифицируются:

- а) именем;
- б) индексным дескриптором;
- в) номером записи в MFT;
- г) системным идентификатором.

5 Состояния, которые не определены для потока в системе:

- а) выполнение;
- б) синхронизация;
- в) ожидание;
- г) готовность.

6 Смены состояний в системе:

- а) выполнение → готовность;
- б) ожидание → выполнение;
- в) ожидание → готовность;
- г) готовность → ожидание.

7 Принципы подсистемы планирования потоков в ОС Windows NT:

- а) квантование;
- б) относительные приоритеты;
- в) абсолютные приоритеты;
- г) вытеснение.

8 Моменты перепланировки использования ЦП могут быть связаны с событиями:

- а) прерывания от таймера в связи с истечением кванта времени;
- б) завершение операции ввода/вывода;
- в) окончание выполнения цикла в программе;
- г) обнаружение деления на ноль в программе.

9 Способы, которыми шины выполняют прерывания:

- а) векторный;
- б) скалярный;
- в) опрашиваемый;
- г) вызываемый.

10 Синхронными прерываниями можно считать:

- а) внешние;
- б) внутренние;
- в) программные;
- г) динамические

Ответы:

- 1. б
- 2. а
- 3. в
- 4. б
- 5. г
- 6. б
- 7. а

8. в
9. г
10. б

Критерии оценки	Отлично	Ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности. В тесте с выбором варианта может быть допущена 1 ошибка
	Хорошо	ответ полный и правильный, показывающий прочные знания в области профессиональной деятельности, при этом допущены две-три ошибки в тесте с выбором, исправленные самостоятельно по требованию преподавателя
	Удовлетворительно	ответ полный, но при этом допущены 4-5 ошибок в тесте с выбором
	Неудовлетворительно	при ответе обнаружено непонимание обучающимся основного содержания и допущены более 5 ошибок в тесте с выбором ответа

#### **Оценочные средства для текущего контроля успеваемости ДЛЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ**

##### ***Тема 1.4.1. Программирование на языке низкого уровня***

1. Написать Wait-функцию, которая приостанавливает планирование некоторого потока до момента освобождения одного или нескольких синхронизирующих объектов ядра.
2. Написать функцию WaitForSingleObject, которая используется для ожидания завершения потока.
3. Написать функцию, в которой предполагается, что поток ожидает завершения одного из трех других потоков.
4. Открыть именованный канал для выполнения асинхронных операций.
5. Использовать проекцию файла для инверсии содержимого файла.

Критерии оценки	Отлично	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
	Хорошо	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные без помощи преподавателя.
	Удовлетворительно	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть

		допущены несущественные неточности, устраненные с помощью преподавателя.
	Неудовлетворительно	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

**Оценочные средства для промежуточной аттестации  
К ДИФФЕРЕНЦИРОВАННОМУ ЗАЧЕТУ ПО  
МДК 01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ  
Перечень вопросов и практических задач**

1. Назовите элементы интерфейса программы
2. Сформулируйте технологию ввода кода программы
3. Перечислите этапы алгоритма сохранения и запуска проекта
4. Сформулируйте назначение вкладок «Свойства», «События»
5. Назовите основные свойства компонентов «LABEL», «BUTTON»
6. Назначение целочисленных типов данных
7. Назначение вещественного типа данных
8. Назначение денежного типа данных
9. Назначение вариантного типа данных
10. Назначение символьного типа данных
11. Назначение интервального типа данных
12. Назначение перечисляемого типа данных
13. Основные стандартные математические функции
14. Основные свойства компоненты «EDIT»
15. Формат записи составного оператора
16. Формат записи условного оператора
17. Форма записи оператора варианта
18. Свойства компоненты «TListBox»
19. Назначение компоненты TComboBox.
20. Режимы работы компоненты «Поле со списком».
21. Назначение компоненты TCheckBox.

22. Назначение стандартных панелей сообщений.

23. Назначение компоненты TScrollBar и ее свойства

24. Назначение компоненты TPanel и ее свойства

<b>Критерии оценки</b>	Отлично	ставится обучающемуся, проявившему всесторонние и глубокие знания учебного материала, освоившему основную и дополнительную литературу, обнаружившему творческие способности в понимании, изложении и практическом использовании усвоенных знаний. Оценка «отлично» соответствует высокому уровню освоения дисциплины (или МДК).
	Хорошо	ставится обучающемуся, проявившему полное знание учебного материала, освоившему основную рекомендованную литературу, обнаружившему стабильный характер знаний и умений и способному к их самостоятельному применению, и обновлению в ходе последующего обучения и практической деятельности. Оценка «хорошо» соответствует достаточному уровню освоения дисциплины (или МДК).
	Удовлетворительно	ставится обучающемуся, проявившему знания основного учебного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, допустившему неточности при ответе, но в основном обладающему необходимыми знаниями и умениями для их устранения при корректировке со стороны преподавателя. Оценка «удовлетворительно» соответствует достаточному уровню освоения дисциплины (или МДК).
	Неудовлетворительно	ставится обучающемуся, обнаружившему существенные пробелы в знании основного учебного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине (или

		МДК). Оценка «неудовлетворительно» соответствует низкому уровню освоения дисциплины (или МДК).
--	--	--

### Оценочные средства для промежуточной аттестации

#### К ЭКЗАМЕНУ ПО

#### МДК 01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

##### Перечень вопросов и практических задач

1. Перечислите основные аспекты качества программного обеспечения. Приведите примеры характеристик, относящихся к каждому из аспектов.
2. Расскажите о методах контроля качества программного обеспечения. Объясните разницу между валидацией и верификацией.
3. Дайте определение термина «тестирование». Объясните разницу между тестированием и отладкой. Опишите общую схему тестирования.
4. Дайте определение термина «тест». Расскажите о двух стратегиях тестирования.
5. Перечислите уровни тестирования программного обеспечения. Опишите, что проверяется на этих уровнях на примере.
6. Опишите на примере проектирование тестовых наборов данных с помощью метода разбиения на классы эквивалентности.
7. Опишите на примере проектирование тестовых наборов данных с помощью метода граничных значений.
8. Опишите на примере проектирование тестовых наборов данных методом покрытия операторов.
9. Опишите на примере проектирование тестовых наборов данных методом покрытия условий.
10. Опишите на примере проектирование тестовых наборов данных методом комбинаторного покрытия условий.
11. Опишите, в чем состоит функциональное тестирование.
12. Опишите, что проверяется при тестировании взаимодействия.
13. Объясните, что такое нагрузочное и стрессовое тестирование, укажите разницу между ними.
14. Перечислите основные принципы тестирования.
15. Объясните, что такое регрессионное и повторное тестирование, укажите разницу между ними.
16. Поясните разницу между ручным и автоматизированным тестированием.
17. Опишите жизненный цикл программного дефекта.
18. Дайте определение термина «надежность программного обеспечения». Перечислите подхарактеристики надежности.
19. Охарактеризуйте понятия: ошибка, дефект, отказ. Поясните разницу между ними на примере какой-либо программы.
20. Расскажите об оценке экономической эффективности программного продукта. Приведите пример.
21. Перечислите основные виды ошибок программных средств.
22. Перечислите методы отладки и приведите примеры.
23. Опишите, что такое модульное тестирование.
24. Опишите, что такое интеграционное тестирование.
25. Назовите цели документирования программных средств.
26. Перечислите и опишите классы документов программных средств.



<b>Критерии оценки</b>	Отлично	ставится обучающемуся, проявившему всесторонние и глубокие знания учебного материала, освоившему основную и дополнительную литературу, обнаружившему творческие способности в понимании, изложении и практическом использовании усвоенных знаний. Оценка «отлично» соответствует высокому уровню освоения дисциплины (или МДК).
	Хорошо	ставится обучающемуся, проявившему полное знание учебного материала, освоившему основную рекомендованную литературу, обнаружившему стабильный характер знаний и умений и способному к их самостоятельному применению, и обновлению в ходе последующего обучения и практической деятельности. Оценка «хорошо» соответствует достаточному уровню освоения дисциплины (или МДК).
	Удовлетворительно	ставится обучающемуся, проявившему знания основного учебного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, допустившему неточности при ответе, но в основном обладающему необходимыми знаниями и умениями для их устранения при корректировке со стороны преподавателя. Оценка «удовлетворительно» соответствует достаточному уровню освоения дисциплины (или МДК).
	Неудовлетворительно	ставится обучающемуся, обнаружившему существенные пробелы в знании основного учебного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине (или МДК). Оценка «неудовлетворительно» соответствует низкому уровню освоения дисциплины (или МДК).

**Оценочные средства для промежуточной аттестации**  
**К ДИФФЕРЕНЦИРОВАННОМУ ЗАЧЕТУ ПО**  
**МДК 01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ**

Теоретические вопросы

2. Выявление функциональных требований. 2. Особенности разработки макетов интерфейсов для мобильных приложений. Концепции Human Interface Guidelines и Material Design. 3. Основные инструменты макетирования. 4. Инструменты прототипирования мобильных приложений. Понятие интерактивного прототипа. 5. Определение целей и задач разработки. Целевая аудитория проекта. Определение рамок проекта. 6. Функциональные и нефункциональные характеристики проекта. Корректное описание требований к продукту. 7. Способы схематичного отображения вариантов и сценариев использования мобильных приложений. UML-диаграммы сценариев использования. 8. Понятие базовой и альтернативной последовательности действий. 9. Инструменты тестирования и аналитики мобильных приложений. 10. Особенности жизненного цикла разработки мобильных приложений. 11. Гибкие методологии разработки программного обеспечения. Методология Agile. Специфика применения гибких методологий в разработке мобильных приложений. Понятия: Канбан-доска, Спринт и Backlog. 12. Способы описания архитектуры разработанного программного обеспечения в текстовой документации и презентации. UML-диаграммы классов. 13. Особенности описания процесса разработки программного обеспечения в текстовой документации и презентации. 14. Способы описания функциональных возможностей мобильных приложений в текстовой документации и презентации.

Перечень практических заданий:

- 1) Создание макетов интерфейсов приложения по выявленным функциональным характеристикам. Требования: а) Соответствие макетов интерфейсов концепции Human Interface Guidelines для iOS и Material Design для Android. б) Использование одного или нескольких основных инструментов макетирования. 2) Создание интерактивного прототипа по разработанным макетам.
- 2) 1) Определение целей и задач разработки. Целевая аудитория проекта. Определение рамок проекта. 2) Функциональные и нефункциональные характеристики проекта. 3) Способы схематичного отображения вариантов и сценариев использования мобильных приложений. UML-диаграммы сценариев использования. 4) Понятие базовой и альтернативной последовательности действий. Формат описания шаблонов экранов и контента. Формат описания API сервера. 5) Инструменты тестирования и аналитики мобильных приложений.
- 3) Создание дизайна интерфейсов приложения по выявленным функциональным характеристикам. 2) Создание интерактивного прототипа по разработанным макетам.
- 4) Разработка программного продукта, соответствующего выявленным функциональным и нефункциональным требованиям. Нефункциональные требования, соблюдение которых является обязательным для всех программных решений: 1) Поддержка смартфонов и планшетов (необходимые поддерживаемые устройства, версии операционных систем и разрешения экранов должны быть определены самостоятельно в зависимости от специфики конкретного продукта); 2) Соответствие дизайна мобильного программного решения концепции Human Interface Guidelines для iOS; 3) Соответствие дизайна мобильного программного решения концепции Material Design для Android; 4) Использование анимации для элементов интерфейса; 5) Использование инструмента Fabric для распространения разработанного программного продукта пользователям для тестирования.
- 5) Вопросы: 1) Постановка проблемы. Выявление целей и задач проекта. 2) Описание архитектуры разработанного программного обеспечения. UML-диаграммы классов. 3) Описание процесса разработки программного обеспечения. 4) Описание функциональных возможностей разработанного мобильного приложения. 5) Описание результата выполненной работы и планов на дальнейшее развитие продукта.

- 6) Разработка презентации по разработанному программному продукту. Требования: 1) Постановка проблемы. Обозначение целей и задач проекта. 2) Описание архитектуры разработанного программного обеспечения. 3) Описание процесса разработки программного обеспечения. 4) Описание функциональных возможностей разработанного мобильного приложения. 5) Описание результата выполненной работы и планов на дальнейшее развитие продукта.

<b>Критерии оценки</b>	Отлично	ставится обучающемуся, проявившему всесторонние и глубокие знания учебного материала, освоившему основную и дополнительную литературу, обнаружившему творческие способности в понимании, изложении и практическом использовании усвоенных знаний. Оценка «отлично» соответствует высокому уровню освоения дисциплины (или МДК).
	Хорошо	ставится обучающемуся, проявившему полное знание учебного материала, освоившему основную рекомендованную литературу, обнаружившему стабильный характер знаний и умений и способному к их самостоятельному применению, и обновлению в ходе последующего обучения и практической деятельности. Оценка «хорошо» соответствует достаточному уровню освоения дисциплины (или МДК).
	Удовлетворительно	ставится обучающемуся, проявившему знания основного учебного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, допустившему неточности при ответе, но в основном обладающему необходимыми знаниями и умениями для их устранения при корректировке со стороны преподавателя. Оценка «удовлетворительно» соответствует достаточному уровню освоения дисциплины (или МДК).
	Неудовлетворительно	ставится обучающемуся, обнаружившему существенные пробелы в знании основного учебного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической

		деятельности без дополнительной подготовки по данной дисциплине (или МДК). Оценка «неудовлетворительно» соответствует низкому уровню освоения дисциплины (или МДК).
--	--	---

### Оценочные средства для промежуточной аттестации

#### К ДИФФЕРЕНЦИРОВАННОМУ ЗАЧЕТУ ПО МДК 01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

1. Логические величины, операции, выражения.
2. Печать элементов списка
3. Дан массив А из n целых чисел. Найти сумму максимального и минимального элемента в массиве. (Поиск максимума и минимума реализовать с помощью подпрограмм-функций).
4. Подпрограмма – процедура.
5. Стеки. Объявление стека.
6. Дан файл целых чисел. Выбрать наибольшее из чисел, принадлежащее интервалу [a,b]. Концы интервала a и b вводятся с клавиатуры.
7. Подпрограмма- функция.
8. Инициализация стека. Добавление элемента в стек.
9. Дан текстовый файл F1. Переписать его содержимое в файл F2, сохраняя строчную структуру и удаляя пустые строки.
10. Рекурсия.
11. Проверка стека на пустоту. Извлечение элемента из стека.
12. Дан текстовый файл F1. Переписать его содержимое в файл F2, сохраняя строчную структуру и удаляя пустые строки.
13. Основные понятия структурного программирования.
14. Очереди. Объявление очереди.
15. Даны две символьные строки S1 и S2, содержащие только строчные латинские буквы. Построить строку S3, в которую войдут только общие символы S1 и S2 в алфавитном порядке и без повторений
16. Модуль. Структура модуля.
17. Создание и заполнение внешнего файла.
18. Даны две символьные строки S1 и S2, содержащие только строчные латинские буквы. Построить строку S3, в которую войдут только общие символы S1 и S2 в алфавитном порядке и без повторений.
19. Модуль. Структура модуля.
20. Чтение данных из внешнего файла.
21. Дан файл целых чисел. Определить, сколько раз в нем повторяется максимальное значение.
22. Списки. Объявление списка.
23. Текстовые файлы.
24. Дан файл целых чисел. Определить, сколько раз в нем повторяется максимальное значение.
25. Добавление элемента в начало списка
26. Чтение данных из внешнего файла.
27. По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка, соединяющего две точки. (длина отрезка=  $\sqrt{(\text{sgr}(x_2-x_1)+\text{sgr}(y_2-y_1))^2}$ ), где (x1,y1)- координаты одной точки, (x2,y2)-координаты второй точки отрезка).

28. Подпрограмма – процедура.
29. Создание и заполнение внешнего файла.
30. По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка, соединяющего две точки. (длина отрезка=  $\text{sgrt}(\text{sgr}(x_2-x_1)+\text{sgr}(y_2-y_1))$ ), где  $(x_1, y_1)$ - координаты одной точки,  $(x_2, y_2)$ -координаты второй точки отрезка).
31. Текстовые файлы.
32. Добавление элемента в начало списка
33. Дан текстовый файл Note.txt. Определить длину самой длинной строки этого файла.
34. Чтение данных из внешнего файла.
35. Списки. Объявление списка.
36. Дан текстовый файл Note.txt. Определить длину самой длинной строки этого файла.
37. Проверка очереди на пустоту. Извлечение элемента из очереди.
38. Создание и заполнение внешнего файла.
39. Дан текстовый файл Note.txt. Определить длину самой длинной строки этого файла.
40. Инициализация очереди. Добавление элемента в очередь.
41. Модуль. Структура модуля.
42. Дан список L, из N целых чисел. Удалить первое вхождение максимального элемента в списке.
43. Очереди. Объявление очереди.
44. Основные понятия структурного программирования.
45. Дан список L, из N целых чисел. Удалить первое вхождение максимального элемента в списке.
46. Инициализация стека. Добавление элемента в стек.
47. Логические величины, операции, выражения.
48. Дан список L, из N целых чисел. Удалить первое вхождение максимального элемента в списке.
49. Стеки. Объявление стека.
50. Подпрограмма – функция.
51. Определить среднее арифметическое чисел, хранящихся в файле Note.txt.
52. Печать элементов списка
53. Подпрограмма – процедура.
54. По заданным значениям X , Y и D вычислить
55. Вычисление MIN и MAX из двух величин оформить в виде подпрограмм – функций.
56. Добавление элемента в начало списка
57. Текстовые файлы.
58. По заданным значениям X , Y и D вычислить
59. Вычисление MIN и MAX из двух величин оформить в виде подпрограмм – функций.
60. Подпрограмма – процедура.
61. Текстовые файлы.
62. Составить рекурсивную подпрограмму вычисления N!

<b>Критерии оценки</b>	Отлично	ставится обучающемуся, проявившему всесторонние и глубокие знания учебного материала, освоившему основную и дополнительную литературу, обнаружившему творческие способности в понимании, изложении и практическом использовании усвоенных знаний. Оценка «отлично» соответствует высокому
------------------------	---------	---

		уровню освоения дисциплины (или МДК).
	Хорошо	ставится обучающемуся, проявившему полное знание учебного материала, освоившему основную рекомендованную литературу, обнаружившему стабильный характер знаний и умений и способному к их самостоятельному применению, и обновлению в ходе последующего обучения и практической деятельности. Оценка «хорошо» соответствует достаточному уровню освоения дисциплины (или МДК).
	Удовлетворительно	ставится обучающемуся, проявившему знания основного учебного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, допустившему неточности при ответе, но в основном обладающему необходимыми знаниями и умениями для их устранения при корректировке со стороны преподавателя. Оценка «удовлетворительно» соответствует достаточному уровню освоения дисциплины (или МДК).
	Неудовлетворительно	ставится обучающемуся, обнаружившему существенные пробелы в знании основного учебного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине (или МДК). Оценка «неудовлетворительно» соответствует низкому уровню освоения дисциплины (или МДК).

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

### **ПЕРЕЧЕНЬ ЭКЗАМЕНАЦИОННЫХ ВОПРОСОВ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

#### *Практические задания:*

#### *Раздел 1. Разработка программных модулей.*

1. Разработка программного продукта для предметной области «Учет клиентов компании, предоставляющей услуги мобильной связи» с применением языка программирования C#.
2. Разработка программного продукта для предметной области «Учет клиентов в

- регистратуре» с применением языка программирования C#.
3. Разработка программного продукта для предметной области «Поликлиника» с применением языка программирования C#.
  4. Разработка программного продукта для предметной области «Библиотека» с применением языка программирования C#.
  5. Разработка тестирующей программы по дисциплине «Информатика» с кодом на языке программирования C#.
  6. Разработка программного продукта для предметной области «Учет товаров в магазине» с применением языка программирования C#.
  7. Разработка тестирующей программы по дисциплине «Операционные системы» с кодом на языке программирования C#.
  8. Разработка программного продукта для предметной области «Ветеринарная клиника» с применением языка программирования C#.
  9. Разработка программного продукта для предметной области «Расчет плотности населения города с учетом численности населения районов» с применением языка программирования C#.
  10. Разработка программного продукта для предметной области «Учет продаж автомобилей на предприятии» с применением языка программирования C#.
  11. Разработка программного продукта для предметной области «Туристическое агентство» с применением языка программирования C#.
  12. Разработка программного продукта для предметной области «Учет постояльцев гостиницы» с применением языка программирования C#.
  13. Разработка программного продукта «Расчет потребляемых калорий» с применением языка программирования C#.
  14. Разработка игрового приложения «Пятнашки» с применением языка программирования C#.
  15. Разработка приложения для производственных расчетов с применением языка программирования C#.

## *Раздел 2. Поддержка и тестирование программных модулей.*

Разработать спецификации тестовых случаев, соответствующие тесты и провести тестирование для классов (по вариантам). Составить отчет в следующей форме:

Название тестового случая:

Тестирующий:

Тест пройден: Да/Нет (PASS/FAIL)

Степень важности ошибки:

Фатальная (3 уровень - crash)

Серьезная (2 уровень - расхождение в спецификации)

Незначительная (1 уровень - незначительная ошибка)

Описание проблемы:

Как воспроизвести ошибку:

Предлагаемое исправление (необязательно):

Комментарий тестирующего (необязательно):

Варианты классов:

Класс (Тип)

TBearingParam (Примитивный)

TAxleParam (Примитивный)

TCommand (Примитивный)

TLog (Примитивный)

TCommandQueue (Непримитивный)

TStore (Непримитивный)

TTerminalBearing (Непримитивный)

TTerminalAxle (Непримитивный)  
TModel (Непримитивный)  
MainForm (Непримитивный)

### *Раздел 3. Разработка мобильных приложений.*

1. Создать интерфейсы мобильного приложения для ОС iOS без использования Interface Builder.

Требования:

1. Реализация минимум 10 различных элементов интерфейса без использования Interface Builder;
  2. Соответствие реализованных элементов принципам iOS Human Interface Guidelines.
2. Создать пользовательский интерфейс мобильного приложения для ОС iOS с использованием инструмента Interface Builder.

Требования к интерфейсам:

1. Соответствие требованиям iOS Human Interface Guidelines.
  2. Использование не менее 10 различных компонентов Interface Builder.
3. Разработать мобильное приложение с функционалом перемещения между экранами мобильного приложения для ОС iOS. Сценарии использования UINavigationController и UITabBarController.

Требования к приложению:

1. Использование UINavigationController;
  2. Использование UITabBarController;
  3. Реализация сценария передачи данных с одного экрана приложения на следующий.
4. Разработать мобильное приложение с функционалом отображения списка объектов. Добавление и удаление ячеек таблицы во время выполнения программы.

Требования к приложению:

1. Использование UITableViewController или UIViewController с компонентом UITableView;
  2. Реализация сценария добавления элемента в отображаемый список;
  3. Реализация сценария удаления элемента из отображаемого списка.
5. Расширить класс NSArray. Реализация методов для: 1) подсчета количества строковых переменных в массиве, 2) объединения строковых переменных в одну строку, 3) поиска максимального элемента в массиве.
6. Создать мобильное приложение с автоматической адаптацией интерфейса для разных размеров экранов с использованием Auto Resizing Mask.

Требования к приложению:

1. Корректное отображение в вертикальном режиме;
  2. Корректное отображение в горизонтальном режиме.
7. Создать мобильное приложение с автоматической адаптацией интерфейса для разных размеров экранов с использованием Auto Layout.

Требования к приложению:

1. Корректное отображение в вертикальном режиме;
  2. Корректное отображение в горизонтальном режиме.
8. Реализовать методы расширения класса NSArray для: 1) возврата массива из элементов, которые вернул блок кода; 2) возврата объекта, который получился суммированием всех элементов массива
9. Разработать мобильное приложение с функционалом получения данных с сервера и отображения данных в виде списка объектов. Использовать решения json-server в качестве тестового сервера.

Требования к приложению:

1. Получение и отображение списка объектов;
  2. Добавление объекта в список и отправка данных на сервер;
  3. Удаление и изменение объекта списка и отправка данных на сервер
10. Разработать мобильное приложение "Тетрис". Реализация функционала игры "Тетрис" в мобильном приложении для ОС iOS.



Требования:

1. Реализация 5 типов фигур;
2. Реализация анимированного появления и исчезновения фигур;
3. Реализация управления фигурами посредством жестов.

11. Разработать мобильное приложение для ОС iOS с функционалом сохранения настроек приложения.

Требования:

1. Реализация минимум 3-х способов хранения настроек;
2. Реализация хранения минимум 5 различных типов настроек.

12. Разработать мобильное приложение для ОС iOS с функционалом подгрузки и отображения данных в разных потоках. Использовать решения json-server в качестве тестового сервера.

Требования к приложению:

1. Получение и отображение списка объектов;
2. Добавление объекта в список и отправка данных на сервер;
3. Удаление и изменение объекта списка и отправка данных на сервер.

13. Разработать мобильное приложение для ОС iOS с функционалом получения данных с сервера в фоновом режиме.

Требования к приложению:

1. Соответствие требованиям политики Apple по использованию сервисов в фоновом режиме;
2. Реализация функционала получения данных с сервера в фоновом режиме.

#### *Раздел 4. Системное программирование.*

1. Дан массив A из n целых чисел. Найти сумму максимального и минимального элемента в массиве. Поиск максимума и минимума реализовать с помощью подпрограмм функций.

2. Дан файл целых чисел. Выбрать наибольшее из чисел, принадлежащее интервалу [a,b]. Концы интервала a и b вводятся с клавиатуры.

3. Дан текстовый файл F1. Переписать его содержимое в файл F2, сохраняя строчную структуру и удаляя пустые строки.

4. Дан файл целых чисел F1. Создать два новых файла F2 и F3 из отрицательных и положительных чисел соответственно.

5. Даны две символьные строки S1 и S2, содержащие только строчные латинские буквы. Построить строку S3, в которую войдут только общие символы S1 и S2 в алфавитном порядке и без повторений.

6. Даны две символьные строки S1 и S2, содержащие только строчные латинские буквы. Построить строку S3, в которую войдут только общие символы S1 и S2 в обратном порядке и без повторений.

7. Дан файл целых чисел. Определить, сколько раз в нем повторяется максимальное значение.

8. Дан файл целых чисел. Определить, сколько раз в нем повторяется минимальное значение.

9. По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка, соединяющего две точки. (длина отрезка=  $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$ ), где (x1,y1) координаты одной точки, (x2,y2) координаты второй точки отрезка).

10. Дан файл целых чисел F1. Создать два новых файла F2 и F3 из положительных и отрицательных чисел соответственно.

11. Даны два файла целых чисел. Определить, в каком из них больше положительных значений.
12. Составить рекурсивную подпрограмму вычисления  $N!$
13. Дана вещественная матрица размера  $m \times n$ . Найти значение наибольшего по модулю элемента матрицы и указать его местоположение в матрице.
14. Определить среднее арифметическое чисел, хранящихся в файле Note.txt.
15. Дан список L из N целых чисел. Удалить первое вхождение максимального элемента в списке.
16. Дан текстовый файл Note.txt. Определить длину самой длинной строки этого файла.
17. Дан файл целых чисел. Выбрать наименьшее из чисел, принадлежащее интервалу  $[a, b]$ . Концы интервала a и b вводятся с клавиатуры.
18. Даны два файла целых чисел. Определить, в каком из них меньше положительных значений.
19. Даны два файла целых чисел. Определить, в каком из них больше отрицательных значений.
20. Даны два файла целых чисел. Определить, в каком из них больше нулевых значений.
21. Даны два файла целых чисел. Определить, в каком из них меньше отрицательных значений.
22. Даны два файла целых чисел. Определить, в каком из них меньше нулевых значений.
23. Дана вещественная матрица размера  $m \times n$ . Найти значение наименьшего по модулю элемента матрицы и указать его местоположение в матрице.
24. Определить среднее геометрическое чисел, хранящихся в файле Note.txt.
25. Дан список L из N целых чисел. Удалить первое вхождение минимального элемента в списке.
26. Дан список L из N целых чисел. Удалить последнее вхождение максимального элемента в списке.
27. Дан список L из N целых чисел. Удалить последнее вхождение минимального элемента в списке.
28. Дан текстовый файл Note.txt. Определить длину самой короткой строки этого файла.

<b>Критерии оценки</b>	<b>Отлично</b>	ставится обучающемуся, проявившему всесторонние и глубокие знания учебного материала, освоившему основную и дополнительную литературу, обнаружившему творческие способности в понимании, изложении и практическом использовании усвоенных знаний. Оценка «отлично» соответствует высокому уровню освоения дисциплины (или МДК).
------------------------	----------------	---

Хорошо	ставится обучающемуся, проявившему полное знание учебного материала, освоившему основную рекомендованную литературу, обнаружившему стабильный характер знаний и умений и способному к их самостоятельному применению, и обновлению в ходе последующего обучения и практической деятельности. Оценка «хорошо» соответствует достаточному уровню освоения дисциплины (или МДК).
Удовлетворительно	ставится обучающемуся, проявившему знания основного учебного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, допустившему неточности при ответе, но в основном обладающему необходимыми знаниями и умениями для их устранения при корректировке со стороны преподавателя. Оценка «удовлетворительно» соответствует достаточному уровню освоения дисциплины (или МДК).
Неудовлетворительно	ставится обучающемуся, обнаружившему существенные пробелы в знании основного учебного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине (или МДК). Оценка «неудовлетворительно» соответствует низкому уровню освоения дисциплины (или МДК).

УТВЕРЖДАЮ

\_\_\_\_\_  
\_\_\_\_\_/ И.О. Фамилия /

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**  
**ПРИМЕРНОЙ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

\_\_\_\_\_  
(наименование дисциплины / модуля)

по направлению подготовки / специальности / профессии

\_\_\_\_\_  
(код и наименование направления подготовки / специальности / профессии)

(год набора \_\_\_\_\_, форма обучения \_\_\_\_\_)

**на 20\_\_ / 20\_\_ учебный год**

В рабочую программу ПМ вносятся следующие изменения:

Номер изменения	Раздел рабочей программы (пункт)	Номера листов			Основание для внесения изменений
		заменен- ных	новых	аннули- рованны х	

--	--	--	--	--	--

Рассмотрен на заседании предметно-цикловой комиссии

,

\_\_\_\_\_  
 протокол от «\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_

\_\_\_\_\_  
*(должность)*

\_\_\_\_\_  
*(подпись)*

\_\_\_\_\_  
*(И.О. Фамилия)*

