

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Коротков Сергей Леонидович  
Должность: Директор филиала СамГУПС в г. Ижевске  
Дата подписания: 03.05.2024 15:21:57  
Уникальный программный ключ:  
d3cff7ec2252b3b19e5caaa8cefa396a11af1dc5

Приложение к ППССЗ по специальности  
09.02.07 Информационные системы  
и программирование

## **ФОНД**

### **КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

#### **ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

#### **ДЛЯ СПЕЦИАЛЬНОСТИ**

#### **09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ**

**Базовый уровень подготовки**

**Год начала подготовки - 2022**

## СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ	5
3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	7
4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ	9
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	20
ПРИЛОЖЕНИЕ	25

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (ФОС) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки специалистов среднего звена по профессиональному модулю ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.

ФОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

ФОС текущего контроля используется для оперативного и регулярного управления учебной деятельностью студентов.

ФОС промежуточной аттестации студентов по профессиональному модулю предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения междисциплинарных курсов профессионального модуля, экзамена (квалификационного) по завершению изучения профессионального модуля в целом.

ФОС разработан на основании:

- программы подготовки специалистов среднего звена по специальности СПО 09.02.07 Информационные системы и программирование;
- рабочей программы профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем;
- учебного плана по специальности СПО 09.02.07 Информационные системы и программирование.

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом в рамках освоения профессионального модуля *ПМ.01 Разработка модулей программного обеспечения для компьютерных систем* является овладение студентами вида профессиональной деятельности *Разработка модулей программного обеспечения для компьютерных систем*, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач 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 4	Эффективно взаимодействовать и работать в коллективе и команде;
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 6	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 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 8	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.»;

В результате освоения профессионального модуля студент должен:

**иметь практический опыт:**

- ПО<sub>1</sub> в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- ПО<sub>2</sub> в использовании инструментальных средств на этапе отладки программного продукта;
- ПО<sub>3</sub> в проведении тестирования программного модуля по определенному сценарию;
- ПО<sub>4</sub> в разработке мобильных приложений.

**уметь:**

- У<sub>1</sub> осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

- У<sub>2</sub> создавать программу по разработанному алгоритму как отдельный модуль;
- У<sub>3</sub> выполнять отладку и тестирование программы на уровне модуля;
- У<sub>4</sub> осуществлять разработку кода программного модуля на современных языках программирования;
- У<sub>5</sub> уметь выполнять оптимизацию и рефакторинг программного кода;
- У<sub>6</sub> оформлять документацию на программные средства.

**знать:**

- З<sub>1</sub> основные этапы разработки программного обеспечения;
- З<sub>2</sub> основные принципы технологии структурного и объектно-ориентированного программирования;
- З<sub>3</sub> способы оптимизации и приемы рефакторинга;
- З<sub>4</sub> основные принципы отладки и тестирования программных продуктов.

### 3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Оценивание уровней сформированности профессиональных и общих компетенций проводится в рамках текущего и промежуточного контроля.

В результате освоения профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем студенты демонстрируют три уровня сформированности профессиональных компетенций: пороговый, базовый и повышенный.

Для каждого конкретного этапа формирования компетенции определены категории «знать», «уметь», «практический опыт», в которые вкладывается следующий смысл:

«приобрести практический опыт» – решать усложненные задачи на основе приобретенных умений и навыков, с их применением в профессиональных деятельности;

«уметь» – решать типичные задачи на основе воспроизведения стандартных алгоритмов решения;

«знать» - воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты.

Наименование Разделов, МДК, тем	Результаты обучения - коды ПК, ОК	Приобретенный практический опыт, освоенные умения и знания	Уровень освоения ПК, ОК	Формы и методы контроля и оценки
1	2	3	4	5
<i>Раздел 1. Разработка программных модулей</i>				
МДК.01.01 Разработка программных модулей	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> У <sub>5</sub> У <sub>6</sub> З <sub>1</sub> З <sub>2</sub> З <sub>3</sub>	2,3	– оценка на практических занятиях; – оценка защиты лабораторных работ; – оценка выполнения индивидуального задания; – экзамен по МДК 01.01
Тема 1.1.1 Жизненный цикл ПО	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>6</sub> З <sub>1</sub>	2,3	
Тема 1.1.2 Структурное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.3 Объектно-ориентированное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.4 Паттерны проектирования	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.5 Событийно-управляемое программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.6 Оптимизация и рефакторинг кода	ПК 1.4 ПК 1.5 ОК 1-ОК 9	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>4</sub> У <sub>5</sub> З <sub>3</sub>	2,3	
1.1.7 Разработка пользовательского интерфейса	ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>2</sub>	2,3	
Тема 1.1.8 Основы ADO.Net	ПК 1.1 ПК 1.2	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>2</sub>	2,3	

	ОК 1-ОК 9			
<i>Раздел 2. Поддержка и тестирование программных модулей</i>				
МДК.01.02 Поддержка и тестирование программных модулей	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>3</sub> У <sub>6</sub> З <sub>1</sub> З <sub>4</sub>	2,3	<ul style="list-style-type: none"> <li>– оценка на практических занятиях;</li> <li>– оценка защиты лабораторных работ;</li> <li>– оценка выполнения индивидуального задания;</li> <li>– экзамен по МДК 01.02</li> </ul>
Тема 1.2.1 Отладка и тестирование программного обеспечения	ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>3</sub> З <sub>1</sub> З <sub>4</sub>	2,3	
1.2.2 Документирование	ПК 1.2 ОК 1-ОК 9	У <sub>6</sub> З <sub>1</sub> З <sub>4</sub>	2,3	
<i>Раздел 3. Разработка мобильных приложений</i>				
МДК.01.03 Разработка мобильных приложений	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> ПО <sub>4</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	<ul style="list-style-type: none"> <li>– оценка на практических занятиях;</li> <li>– оценка защиты лабораторных работ;</li> <li>– оценка выполнения индивидуального задания;</li> <li>– экзамен по МДК 01.03</li> </ul>
Тема 1.3.1 Основные платформы и языки разработки мобильных приложений	ПК 1.6 ОК 1-ОК 9	ПО <sub>4</sub> У <sub>4</sub> З <sub>2</sub>	2,3	
Тема 1.3.2 Создание и тестирование модулей для мобильных приложений	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> ПО <sub>4</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	
<i>Раздел 4. Системное программирование</i>				
МДК.01.04 Системное программирование	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	<ul style="list-style-type: none"> <li>– оценка на практических занятиях;</li> <li>– оценка защиты лабораторных работ;</li> <li>– оценка выполнения индивидуального задания;</li> <li>– экзамен по МДК 01.04</li> </ul>
Тема 1.4.1 Программирование на языке низкого уровня	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ  
ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ  
КОМПЬЮТЕРНЫХ СИСТЕМ  
(С ПРАВИЛЬНЫМИ ОТВЕТАМИ)**

№ пп	Содержание вопроса	Правильный ответ	Проверяемые компетенции
1.	Система объектно-ориентированного визуального программирования – это ...	Delphi	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
2.	Среда быстрой разработки приложений – это ...	RAD	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
3.	Язык программирования Delphi – это ...	Object Pascal	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
4.	Интегрированная среда разработки – это ...	IDE	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
5.	Совокупность объектов и способов их взаимодействия называют объектно-ориентированной	программой	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
6.	Основным окном разрабатываемого приложения является ...	форма	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
7.	Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...	Windows	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
8.	Заготовка разрабатываемого приложения – ЭТО Окно проектировщика ...	форм	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
9.	Набор всех файлов,	проектом	ПК 1.2



	необходимых для создания приложения, называется...		ПК 1.3 ПК 1.4 ОК 1-ОК 9
10.	При сборке приложения описание из DFM-файла помещается в область ...	ресурсов	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9
11.	Класс формы объявляется в разделе...	Type	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9
12.	Примерами комбоэлементов не являются ...	комбополе	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9
13.	Любой элемент управления, ограничивающий набор значений, доступных для ввода пользователем, является ограничивающим элементом ...	ввода	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
14.	Каждый приемник широковещательных сообщений является наследником класса ...	BroadcastReceive	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
15.	Операционная система: система программ, которая обеспечивает совместную работу всех устройств компьютера по обработке ...	информации	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
16.	Программное обеспечение (ПО) – это: совокупность программ, позволяющих организовать решение задач на ...	компьютере	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
17.	Операционные системы бывают DOS, Windows, ...	Unix	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
18.	Загрузочный модуль программы – результат работы ...	Транслятора	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9

19.	Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется ...	Компилятор	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
20.	Результатом компиляции программы на языке высокого уровня является ... файл	Объектный	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9

**ТЕСТ**  
(с правильными ответами)

№ пп	Содержание вопроса	Правильный ответ	Проверяемые компетенции
1.	IDE <u>не</u> включает в себя: (несколько вариантов ответа) 1. Редактор кода 2. Высокопроизводительный компилятор в машинный код 3. Объектно-ориентированную модель компонент 4. Эффекты и переходы 5. Визуальное построение приложений 6. Сопровождение ПП 7. Средство для построения баз данных	4, 6	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
2.	Названия процедур на определенное событие: 1. Компоненты 2. Обработчики событий 3. Редактор кода	2	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
3.	Назначение главного меню: 1. Осуществление функций управления при разработке программ 2. Сопровождение программных продуктов 3. Автоматизирует процесс тестирования программ	1	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
4.	Назначение панели инструментов: 1. Обработка событий 2. Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто употребляемым командам Главного меню 3. Построение баз данных	2	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
5.	Назначение палитры компонентов: 1. Содержит пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение 2. Подделка подписей 3. Управление Paint	1	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9
6.	Главное окно управляет окнами ... (несколько	1, 3, 4	ПК 1.1

	<p>вариантов ответа)</p> <ol style="list-style-type: none"> <li>1. Инспектор объектов</li> <li>2. Язык ассемблера</li> <li>3. Редактора кода</li> <li>4. Проектировщика форм</li> <li>5. Трансляции программы</li> </ol>		<p>ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9</p>
7.	<p>Обнуление строки ввода:</p> <ol style="list-style-type: none"> <li>1. Edit1.Text :='';</li> <li>2. Edit1.Text := TMemo;</li> <li>3. Edit1.Text :='Закругляемся'</li> </ol>	1	<p>ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9</p>
8.	<p>Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:</p> <ol style="list-style-type: none"> <li>1. Окно компилятора</li> <li>2. Окно редактора кода</li> <li>3. Окно проектировщика</li> <li>4. Главное окно</li> </ol>	2	<p>ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9</p>
9.	<p>Задание метки текста из редактора ввода:</p> <ol style="list-style-type: none"> <li>1. Memo1.Caption := Edit1.Text;</li> <li>2. Label1. Caption := Edit1.Text;</li> <li>3. Form1. Caption := Edit1.Text;</li> </ol>	2	<p>ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9</p>
10.	<p>Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:</p> <ol style="list-style-type: none"> <li>1) Android SDK</li> <li>2) JDK</li> <li>3) плагин ADT</li> <li>4) Android NDK</li> </ol>		<p>ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9</p>
11.	<p>Библиотеки, реализованные на базе PacketVideo OpenCORE:</p> <ol style="list-style-type: none"> <li>1) Media Framework</li> <li>2) SQLite</li> <li>3) FreeType</li> <li>4) 3D библиотеки</li> </ol>		<p>ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9</p>
12.	<p>Intel XDK поддерживает разработку под:</p> <ol style="list-style-type: none"> <li>1) JavaFX Mobile</li> <li>2) Apple iOS, BlackBerry OS</li> <li>3) MtkOS, Symbian OS, Microsoft Windows 8</li> <li>4) Android, Apple iOS, Microsoft Windows 8, Tizen</li> </ol>		<p>ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9</p>
13.	<p>Какой язык разметки используется для описания иерархии компонентов графическо- го пользовательского интерфейса Androidприложения?</p> <ol style="list-style-type: none"> <li>1) html</li> <li>2) xml</li> <li>3) gml</li> <li>4) xhtml</li> </ol>		<p>ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9</p>
	<p>Системные оболочки – это:</p> <ol style="list-style-type: none"> <li>1) специальная кассета для удобного размещения дискет с операционной системой</li> <li>2) специальная программа, упрощающая диалог</li> </ol>	2	<p>ПК 1.2 ПК 1.3 ПК 1.4</p>

	пользователь – компьютер, выполняет команды операционной системы 3) система приемов и способов работы конкретной программы при загрузке программ и завершении работы		ОК 1-ОК 9
<b>14.</b>	Использование одного имени для задания общих для класса действий, что означает способность объектов выбирать внутренний метод, исходя из типа данных, определяет свойство ООП 1) Полиморфизм 2) Управление событиями 3) Инкапсуляция 4) Наследование	<b>3</b>	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9
<b>15.</b>	Понятие класса в ООП включает в себя 1) Поля и методы класса 2) Процедуры и функции обработки 3) Поля и функции обработки 4) Поля и процедуры обработки	<b>1</b>	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9
<b>16.</b>	Назначение конструктора объекта 1) Только выделяет память под объект 2) Выделяет память и задает начальное значение полям 3) Задает начальное значение полям 4) Выделяет память, задает начальное значение полям, выполняет любые проверки, заданные программистом	<b>1</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>17.</b>	Как описывается конструктор объекта 1) procedure create; 2) constructor create; 3) function create; 4) function constructor	<b>4</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>18.</b>	Моделями типа «черный ящик» являются 1) Модели мышления 2) Модели, описывающие зависимость параметров состояния объекта от входных параметров 3) Модели, описывающие входные и выходные параметры объекта без учета внутренней структуры объекта 4) Модели «аварийного» ящика на самолетах	<b>3</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>19.</b>	Программа «драйвер» служит для 1) запуска программы на выполнение 2) имитации запуска программы на выполнение 3) проверки правильности работы программы 4) передачи параметров в процедуры и функции	<b>3</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>20.</b>	Какие методы сборки программы существуют 1) монолитная 2) пошаговая 3) одновременная 4) постепенная	<b>2, 3</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6

			ОК 1-ОК9
<b>21.</b>	Какой метод тестирования программы учитывает закон распределения входных данных 1) детерминированное тестирование 2) функциональное тестирование 3) стохастическое тестирование 4) логическое тестирование	<b>4</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>22.</b>	Результатом компиляции программы на языке высокого уровня является 1) Командный файл 2) Объектный файл 3) Исходный текст программы на языке высокого уровня 4) Дисплейный файл	<b>2</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>23.</b>	Алгоритм-это: 1) Указание на выполнение действий, 2) Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи, 3) Процесс выполнения вычислений, приводящих к решению задачи	<b>2</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>24.</b>	Свойство алгоритма – дискретность, выражает, что: 1) Команды должны следовать последовательно друг за другом, 2) Каждая команда должна быть описана в расчете на конкретного исполнителя, 3) Разбиение алгоритма на конечное число команд	<b>3</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>25.</b>	Самое важное свойство алгоритма: 1) Визуальность, 2) Массовость, 3) Дискретность, 4) Аудиальность	<b>2</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК9
<b>26.</b>	Псевдокоманда DW означает 1) определить байт; 2) определить слово (2 байта); 3) определить двойное слово (4 байта); 4) определить 6 байт 5) определить учетверенное слово (8 байт);	<b>2</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
<b>27.</b>	Псевдокоманда DD означает 1) определить байт; 2) определить слово (2 байта); 3) определить двойное слово (4 байта); 4) определить 6 байт 5) определить учетверенное слово (8 байт);	<b>3</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
<b>28.</b>	Запись регистра в память осуществляется при помощи команды 1) MOV 2) XCHG 3) XLAT	<b>1</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
<b>29.</b>	Порядковый номер элемента в массиве, который	<b>1</b>	ПК 1.1

	всегда начинается с нуля 1) Индекс 2) Команда 3) Псевдокоманда 4) Макрокоманда		ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9
<b>30.</b>	Позволяет сохранять длину исходного текста программы 1) Индекс 2) Команда 3) Псевдокоманда 4) Макрокоманда	<b>4</b>	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК9

#### **4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

##### **4.1. Типовые задания для оценки освоения МДК.01.01 Разработка программных модулей**

###### **Вопросы для подготовки к экзамену/зачёту**

1. Трёхслойная клиент-серверная архитектура
2. Основные элементы HTML
3. Основные теги HTML
4. Форматирование текста в HTML. Символьные элементы
5. Гиперссылки в HTML
6. Фреймы в HTML
7. Таблицы в HTML. Атрибуты таблиц и ячеек
8. Таблицы в HTML. Объединение ячеек таблиц
9. Списки в HTML. Упорядоченные и неупорядоченные списки
10. Формы в HTML. Основные элементы форм. Атрибуты форм
11. Графические возможности HTML
12. Карты изображений в HTML
13. Каскадные таблицы стилей и их подключение к HTML-документу
14. Структура документа с PHP-скриптом. Включение PHP в HTML
15. Переменные и константы в PHP
16. Типы данных PHP. Преобразование типов
17. Операторы PHP. Приоритет операторов
18. Вывод контента в PHP.
19. Функции даты и времени в PHP
20. Строки в PHP. Функции работы со строками
21. Условный оператор в PHP
22. Оператор выбора в PHP
23. Циклы с заданным числом повторений в PHP
24. Циклы с предусловием и постусловием в PHP
25. Функции в PHP. Повторное использование кода
26. Массивы в PHP. Массивы с числовыми индексами и ассоциативные массивы
27. Массивы в PHP. Функции для работы с массивами
28. Обработка форм в PHP
29. Основные понятия JavaScript, структура программы, поддержка браузерами
30. Вывод результатов работы программы и ввод данных в JavaScript.
31. Переменные и типы данных JavaScript

32. Операторы JavaScript. Приоритет выполнения операторов
33. Типы данных JavaScript, преобразование типов данных
34. Массивы в JavaScript. Назначение массивов, объявление и инициализация массивов
35. Пользовательские функции JavaScript. Расположение функций внутри HTML-кода
36. Оператор ветвления в JavaScript
37. Оператор выбора JavaScript
38. Многократное выполнение блоков кода. Цикл с заданным числом повторений в JavaScript
39. Циклы с предусловием и постусловием в JavaScript
40. Встроенный класс Global в JavaScript. Основные функции и примеры использования
41. Встроенный класс Number в JavaScript. Работа с числами
42. Встроенный класс String в JavaScript. Обработка строк
43. Встроенный класс Array в JavaScript. Работа с массивами
44. Встроенный класс Math в JavaScript. Использование математических функций
45. Встроенный класс Date в JavaScript. Работа с датой и временем
46. События в JavaScript. Определение событий, обработчики событий. События мыши, пример использования
47. События в JavaScript. Определение событий, обработчики событий. События документа, пример использования
48. События в JavaScript. Определение событий, обработчики событий. События формы, пример использования.
49. События в JavaScript. Написание обработчиков событий, примеры
50. События в JavaScript. Объект event
51. Объектная модель документа DOM. Функции для работы с объектной моделью документа
52. Доступ к значениям элементов форм в JavaScript
53. Описание среды программирования на C++
54. Структура программы на языке C++
55. Понятие алгоритма. Свойства алгоритмов
56. Базовые структуры алгоритмов. Способы описания алгоритмов
57. Переменные и константы в языке C++
58. Типы данных в языке C++. Явное и неявное преобразование типов
59. Операции в языке C++
60. Математические функции в языке C++. Примеры использования математических функций
61. Условный оператор в языке C++. Назначение. Формат записи, блок-схема. Примеры
62. Оператор выбора в языке C++. Назначение. Формат записи, блок-схема. Примеры
63. Циклы с условием в языке C++. Назначение. Формат записи, блок-схема. Примеры
64. Циклы с заданным числом повторений в языке C++. Назначение. Формат записи, блок-схема. Примеры
65. Операторы передачи управления в языке C++
66. Алгоритм нахождения суммы ряда с заданной точностью. Блок-схема алгоритма и реализация на C++
67. Нахождение корней уравнения методом половинного деления. Блок-схема алгоритма и реализация на C++
68. Нахождение корней уравнения методом хорд. Блок-схема алгоритма и реализация на C++
69. Вычисление определенного интеграла методом прямоугольников. Блок-схема алгоритма и реализация на C++
70. Вычисление определенного интеграла методом трапеций. Блок-схема алгоритма и реализация на C++
71. Одномерные массивы. Объявление и инициализация одномерных массивов

72. Случайные и псевдослучайные числа. Генерация псевдослучайных чисел в C++
73. Получение и изменение значений элементов массива. Перебор элементов массива
74. Матрицы. Объявление и инициализация матриц
75. Получение и изменение значений элементов матрицы. Перебор элементов матрицы
76. Нахождение минимального и максимального элемента массива. Блок-схема алгоритма и реализация в C++
77. Нахождение суммы элементов массива. Блок-схема алгоритма и реализация на C++
78. Сортировка массивов методом нахождения максимального элемента
79. Сортировка массивов методом «пузырька»
80. Указатели и операции над указателями. Выражения и арифметические действия с указателями
81. Взаимосвязь между указателями и массивами
82. Динамические массивы
83. Объявление и инициализация строк. Доступ к символам строки. Перебор символов строки
84. Основные функции для работы со строками
85. Пользовательские функции в языке C++. Объявление, определение и вызов функций
86. Пользовательские функции в языке C++. Расположение объявления и определения функций.  
Заголовочные файлы
87. Пользовательские функции в языке C++. Способы передачи параметров в функцию
88. Пользовательские функции в языке C++. Передача массивов в функции
89. Пользовательские функции в языке C++. Статические переменные
90. Пользовательские функции в языке C++. Рекурсия
91. Работа с файлами в языке C++. Открытие и закрытие файла
92. Работа с файлами в языке C++. Запись в файл и чтение из файла
93. Структуры. Объявление и инициализация структур
94. Структуры. Доступ к элементам структур
95. Структуры. Массивы структур
96. Распределение оперативной памяти для программ на C++. Динамическая память
97. Функции для работы с динамической памятью
98. Динамические структуры данных. Классификация динамических структур
99. Динамические структуры данных. Объявление динамических структур
100. Динамические структуры данных. Доступ к данным в динамических структурах
101. Динамические структуры данных. Работа с памятью при использовании динамических структур
102. Однонаправленные списки. Определение, основные операции с однонаправленным списком
103. Однонаправленные списки. Создание и просмотр однонаправленного списка
104. Однонаправленные списки. Вставка элемента в однонаправленный список
105. Однонаправленный список. Удаление элемента из однонаправленного списка
106. Однонаправленный список. Поиск элемента в однонаправленном списке
107. Стек. Определение, основные операции со стеком
108. Очередь. Определение, основные операции с очередью
109. Основные понятия объектно-ориентированного программирования
110. Классы и объекты. Объявление классов в языке C++
111. Классы и объекты. Объявление и определение атрибутов и методов
112. Классы и объекты. Конструкторы и деструкторы
113. Классы и объекты. Статические атрибуты и методы
114. Динамическое создание объектов. Указатели на объекты
115. Наследование. Множественное наследование
116. Виртуальные методы



117. Абстрактные методы и классы
118. Перегрузка операций
119. Платформа Microsoft .NET. Структура платформы
120. Платформа Microsoft .NET. Выполнение программы в .NET
121. Основные принципы объектно-ориентированного программирования
122. Среда Visual Studio .NET. Типы создаваемых проектов. Основные окна программы
123. Язык программирования C#. Типы данных, классификация типов. Преобразование типов
124. Язык программирования C#. Переменные. Объявление и инициализация переменных, область действия переменных
125. Язык программирования C#. Операции и выражения
126. Язык программирования C#. Ввод и вывод информации
127. Язык программирования C#. Математические функции, класс Math
128. Язык программирования C#. Операторы ветвления. Условный оператор. Логические операции
129. Язык программирования C#. Операторы ветвления. Оператор выбора
130. Язык программирования C#. Операторы цикла. Цикл с условием
131. Язык программирования C#. Операторы цикла. Цикл с постусловием
132. Язык программирования C#. Операторы цикла. Цикл с заданным числом повторений
133. Язык программирования C#. Операторы цикла. Цикл перебора
134. Язык программирования C#. Операторы передачи управления
135. Язык программирования C#. Обработка исключительных ситуаций
136. Язык программирования C#. Массивы. Объявление и инициализация одномерных массивов
137. Язык программирования C#. Массивы. Объявление и инициализация прямоугольных массивов
138. Язык программирования C#. Символы и массивы символов
139. Язык программирования C#. Строки. Функции для работы со строками
140. Язык программирования C#. Классы и объекты. Основные понятия
141. Язык программирования C#. Классы. Поля и методы
142. Язык программирования C#. Классы. Конструкторы и деструкторы
143. Язык программирования C#. Классы. Свойства
144. Язык программирования C#. Классы. Наследование
145. Язык программирования C#. Классы. Виртуальные методы
146. Язык программирования C#. Классы. Абстрактные классы
147. Событийно-управляемое программирование
148. Этапы разработки программ под Windows
149. Основы программирования под Windows. Класс Control
150. Основы программирования под Windows. Элементы управления Label, Button, TextBox
151. Основы программирования под Windows. Меню MainMenu и ContextMenu
152. Основы программирования под Windows. Флажок CheckBox, переключатель RadioButton, панель GroupBox, список ListBox
153. Основы программирования под Windows. Класс Form
154. Основы программирования под Windows. Диалоговые окна
155. Основы программирования под Windows. Работа с графикой

### **Примеры практических заданий**

1. Создать форму для ввода массива и кнопкой «Перемешать». При нажатии на кнопку элементы массива перемешиваются

2. Палиндромом называют последовательность символов, которая читается как слева направо, так и справа налево. Создать web-форму для ввода строки и кнопкой, при нажатии на которую определяется, является ли введенная строка палиндромом
3. Создать web-форму с двумя полями для ввода чисел и выбором операции (+, -, \*, /). Вычислить соответствующее выражение и вывести результат в браузер
4. Найти наибольшее из трех чисел, введенных пользователем в форму
5. Написать программу нахождения синуса и косинуса угла, выбор действия осуществляется с помощью переключателя
6. Средствами PHP вывести время и дату на странице
7. Создать web-форму для ввода строки и кнопкой, при нажатии на которую строка переворачивается и выводится в браузер
8. Написать программу, которая для введенной строки подсчитывает сумму всех содержащихся в ней цифр
9. Создать класс Point, разработав следующие элементы класса: поля (x, y); конструкторы, позволяющие создать экземпляр класса с нулевыми координатами или с заданными координатами; методы, позволяющие вывести координаты точки на экран, рассчитать расстояние от начала координат до точки, переместить точку на плоскости на вектор (a, b); свойства для получения-установки координаты точки (доступное для чтения и записи), для умножения координаты точки на скаляр (доступное только для записи)
10. Создать класс Triangle, разработав следующие элементы класса: поля (a, b, c); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон треугольника на экран, рассчитать периметр треугольника, рассчитать площадь треугольника; свойства для получения-установки длин сторон треугольника (доступное для чтения и записи), для определения, существует ли треугольник с данными длинами сторон (доступное только для чтения)
11. Создать класс Rectangle, разработав следующие элементы класса: поля (a, b); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон прямоугольника на экран, рассчитать периметр прямоугольника, рассчитать площадь прямоугольника; свойства для получения-установки длин сторон прямоугольника (доступное для чтения и записи), для определения, является ли данный прямоугольник квадратом (доступное только для чтения)
12. Создать класс Round, разработав следующие элементы класса: поля (r); конструктор, позволяющий создать экземпляр класса с заданным радиусом; методы, позволяющие рассчитать длину окружности, рассчитать площадь круга; свойства для получения-установки радиуса круга (доступное для чтения и записи)
13. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создать производные классы: Triangle (треугольник), Circle (круг) со своими методами вычисления площади и периметра. Создать массив n фигур и вывести полную информацию о фигурах на экран
14. Посчитать сумму элементов побочной диагонали квадратной матрицы
15. Найти среднее арифметическое положительных элементов одномерного массива. Количество элементов массива вводится пользователем, массив заполняется случайными числами
16. Посчитать в строке количество цифр. Строка вводится пользователем с клавиатуры
17. Создать форму Windows. Программа должна предоставлять возможность менять цвет фона формы
18. Создать Windows-приложение, демонстрирующее работу с модальными и немодальными окнами
19. Написать программу на языке C++, которая считает сумму элементов одномерного массива, не используя операцию индексирования [].
20. Написать программу на языке C++, демонстрирующую способы передачи параметров в функции

21. Написать рекурсивную функцию для нахождения факториала числа
22. Написать программу на языке C++ для подсчета суммы элементов одномерного массива. Размерность массива вводится пользователем с клавиатуры
23. Написать программу, демонстрирующую возможность создания односвязного списка и добавления элемента в список
24. Написать программу, демонстрирующую возможность создания стека и добавления элемента в любое место стека
25. Создать класс «Прямоугольник», который имеет два конструктора. Первый конструктор принимает в качестве параметров длину и ширину. Второй конструктор создает квадрат с заданной стороной
26. Написать программу, демонстрирующую работу с абстрактными классами
27. Создать функцию на языке C++, подсчитывающую сумму элементов массива. Массив передается как параметр в функцию
28. Написать программу для нахождения максимального элемента матрицы.
29. Написать программу для сортировки массива методом пузырька.
30. Написать программу для нахождения корней квадратного уравнения  $ax^2+bx+c=0$ . Коэффициенты  $a$ ,  $b$ ,  $c$  вводятся пользователем

#### **4.2. Типовые задания для оценки освоения МДК.01.02 Поддержка и тестирование программных модулей**

**Вопросы для подготовки к экзамену/зачёту**

**Валидация(аттестация) и верификация**

**Общие вопросы верификации ПО.**

**Цели и задачи верификации ПО**

1. Статические и динамические методы верификации ПО
2. Факторы и атрибуты внешнего и внутреннего качества ПО
3. Виды и методы тестирования. Смоук-тестирование, регрессионное тестирование, тестирование белого и черного ящиков
4. Тестирование нефункциональных требований
5. Взаимосвязь разработки и тестирования. V-модель разработки ПО.
6. Уровни тестирования. Модульное (unit), интеграционное (integration),
7. системное (system), приемочное (acceptance) тестирование.
8. Техники тест дизайна. Разбиение на классы эквивалентности и тестирование граничных значений.
9. Понятие дефекта. Основные определения и классификация дефектов.
10. Описание дефектов
11. Атрибуты дефектов. Приоритет(priority) и серьезность(severity) дефектов
12. Определение серьезности дефекта по его описанию — практическое задание
13. Версионирование ПО на разных стадиях разработки.
14. Инструментальные средства поддержки тестирования. Системы отслеживания ошибок (Bug Tracking Systems)
15. Артефакты разработки ПО, относящиеся к тестированию. Тест-кейсы (test cases)
16. Артефакты разработки ПО, относящиеся к тестированию. План тестирования (test plan).

#### **4.3. Типовые задания для оценки освоения МДК.01.03 Разработка мобильных приложений**

**Вопросы для подготовки к экзамену/зачёту**

1. Понятие защищенной информационной системы.
2. Свойства защищенной ОС.
3. Безопасность информационных систем в нормативных документах.
4. Классификация защищенности ОС по международным стандартам.
5. Политика безопасности, формальное представление политик.
6. Классификация изъянов защиты.
7. Категории изъянов защиты в ОС.
8. Мобильное программирование, платформы для разработки.
9. Основы работы с сенсорным вводом. Обработка нескольких касаний. Использование изображений.
10. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
11. Особенности использования pivot и panorama.
12. Краткая история ОС Android.
13. Intel для Android: партнерство и инструментарий разработчика.
14. Архитектура приложений для Android. Ресурсы приложения. Пользовательский интерфейс. Инструментарий разработки приложений для Android.
15. Обзор шагов разработки типичного приложения под Android. Особенности разработки с использованием эмулятора. Отладка кода в эмуляторе и на реальных приложениях. Пример простейших программ Android-приложения. Запуск приложения на эмуляторе.
16. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS).
17. Планирование покадровой анимации, анимирование, анимация шаблонов, видов, использование класса Camera.
18. Проверка безопасности, работа со службами, основанными на местоположении, использование HTTP-служб, службы AIDL.
19. Подготовка AndroidManifest.xml для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK-файла для загрузки, работа пользователя с Android Market.
20. Работа с инструментами Intel для оптимизации отладки Android-приложений

### **Примеры практических заданий**

1. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск заметок. Два варианта хранения заметок: А) в базе SQLite. Б) С использованием файловой системы.
2. Создать приложение с графическим пользовательским интерфейсом с функциями: Определение местоположения пользователя на карте Google Map; Определение скорости и направления движения пользователя; Масштабирование карты. Программа должна быть конфигурируемой. Настройки: Режим определения местоположения (через GPS либо по сотам); Включение/отключение режима поиска.
3. Разработать приложение-таймер с использованием датчика ориентации в виде песочных часов. Каждый раз для того чтобы активировать таймер, необходимо перевернуть экран мобильного устройства вверх ногами. Используйте анимацию для показа «перетекающего песка» и переворота песочных часов. Для задания времени перетекания песка требуется разработать push-notification сервер. Через форму ввода на сервере можно отправлять на клиент (приложение-таймер) указанное время (числовой ввод).
4. Разработка программы для обмена мгновенными сообщениями. Требуется разработать приложение для обмена мгновенными сообщениями через Wi-Fi/Bluetooth. Поддерживаемые режимы: 1. Активный режим. Приложение занимает весь экран, содержит поля для отправки сообщений и список принятых сообщений. 2. Режим уведомлений. Приложение через уведомления показывает принятые сообщения.

5. Разработка мобильного сайта с адаптивным дизайном. Требуется разработать сайт, пригодный для просмотра на смартфонах и планшетах, с использованием принципов адаптивного дизайна: 1) Гибкая сетка. 2) Пропорциональные шрифты. 3) Масштабируемые изображения. 4) Медиа-запросы

#### **4.4. Типовые задания для оценки освоения МДК.01.04 Системное программирование**

##### **Вопросы для подготовки к экзамену/зачёту**

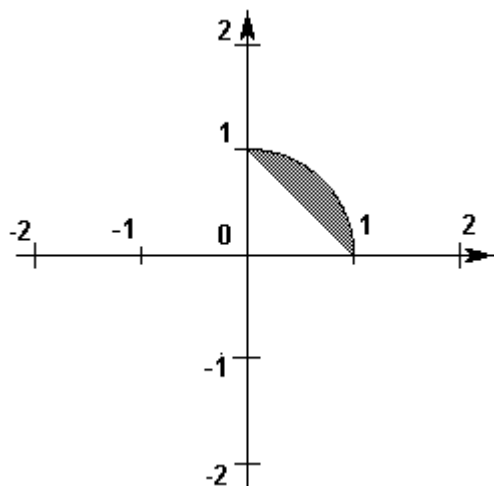
1. Что такое системное программирование?
2. Инструментальные средства для создания и запуска программ на языке C
3. Структура программы на языке C
4. Интерпретаторы и компиляторы
5. Программирование на языке C в среде Visual Studio. Отладка программы
6. Алфавит языка C
7. Идентификаторы
8. Классификация данных в языке C
9. Типы данных языка C
10. Объявление и инициализация переменных в языке C
11. Ввод данных с клавиатуры и вывод данных на экран в языке C
12. Работа с файлами в языке C. Чтение данных из файла и запись в файл
13. Классификация операций языка C. Основные операции
14. Логические переменные в языке C. Операции отношения, сравнения, логические операции, поразрядные логические операции, операции сдвига, условная операция
15. Математические функции языка C
16. Видимость переменных в языке C, локальные и глобальные переменные, статические переменные
17. Условный оператор
18. Оператор выбора
19. Циклические программы. Классификация циклов
20. Цикл с предусловием
21. Цикл с постусловием
22. Цикл с заданным числом повторений
23. Алгоритм вычисления суммы бесконечного ряда с заданной точностью
24. Алгоритм вычисления определенного интеграла с заданной точностью
25. Алгоритм решения уравнений методом дихотомии
26. Функции в языке C. Описание, вызов функции, прототипы функций
27. Передача параметров в функции по имени и по адресу
28. Рекурсивные функции
29. Создание многомодульных программных проектов
30. Массивы. Описание массивов и их инициализация в языке C. Доступ к элементам массива
31. Методы сортировки одномерных массивов
32. Алгоритмы нахождения минимального (максимального) элемента массива и подсчёта суммы элементов массива

Символы и строки. Объявление и инициализация строк. Работа со строками

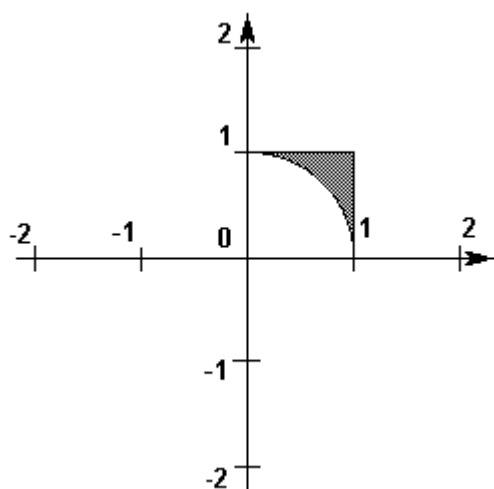
##### **Примеры практических заданий**

1. Написать программу, которая удаляет из введенной с клавиатуры строки первые пробелы (если их нет, то оставить как есть)
2. Подсчитать количество слов в строке
3. Выделить первое слово из строки
4. Найти максимальное из трех введенных целых чисел
5. Найти сумму цифр трехзначного числа
6. Решить квадратное уравнение  $ax^2+bx+c=0$ , коэффициенты уравнения вводятся с клавиатуры

7. Определить, кратна ли сумма цифр трехзначного числа. Число вводится с клавиатуры
8. Найти площадь прямоугольного треугольника по известным катету и гипотенузе. Исходные данные вводятся с клавиатуры
9. Переписать все символы строки в обратном порядке
10. Написать программу, которая вводит координаты точки (x, y) и определяет, попадает ли точка в заштрихованную область на рисунке. Попадание на границу области считать попаданием в область.



11. Построить программу, которая вводит координаты точки (x, y) и определяет, попадает ли точка в заштрихованную область на рисунке. Попадание на границу области считать попаданием в область.



12. Написать программу для сортировки одномерных массивов
13. Вывести на экран все трехзначные числа, в которых хотя бы две цифры одинаковые
14. Вывести на экран все трехзначные числа, которые начинаются и заканчиваются на одну и ту же цифру
15. Вывести на экран таблицу значений функции  $y=x^3$  при  $x=-4\dots4$ . Шаг изменения  $x$  вводится с клавиатуры
16. Найти сумму ряда с точностью 0.0001

$$a_n = (-1)^n \frac{n+1}{n^3+2}$$

17. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 найти сумму положительных элементов

18. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 количество отрицательных элементов
19. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 найти среднее арифметическое четных элементов
20. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 заменить все отрицательные элементы нулями
21. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 переставить местами минимальный и максимальный элементы
22. Определить среднее арифметическое элементов главной диагонали квадратной матрицы
23. Определить максимальный элемент побочной диагонали квадратной матрицы
24. Найти сумму нечетных элементов прямоугольного массива, состоящего из пяти строк и семи столбцов
25. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями
26. Определить, является ли сумма элементов матрицы 3x2 нечетной
27. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями
28. Определить, является ли сумма элементов матрицы 3x2 нечетной
29. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями
30. Определить, является ли сумма элементов матрицы 3x2 нечетной

#### **4.5. Типовые задания для экзамена квалификационного по ПМ.01**

##### **Задание 1**

Разработать приложение с пользовательским графическим интерфейсом по предложенному заданию

##### **Задание 2**

Провести отладку и тестирования разработанного приложения

##### **Задание 3**

Создать мобильное приложение, реализующее часть функционала созданного приложения

### **Вопросы для устного опроса по разделам**

#### **по МДК 01.01. Разработка программных модулей**

1. Дайте понятие ЖЦ ПО. Перечислите этапы ЖЦ ПО?
2. Объясните технологию структурного программирования.
3. Какие существуют инструментальные средства оформления и документирования алгоритмов программ?
4. Расскажите об оценке сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи.
5. Перечислите и поясните основные принципы объектно-ориентированного программирования. В чем его отличие от остальных видов программирования. Что такое классы?
6. Перегрузка методов. Операции класса. Иерархия классов.
7. Синтаксис интерфейсов.
8. Интерфейсы и наследование. Структуры. Делегаты.

9. Регулярные выражения. Коллекции. Параметризованные классы. Указатели.
10. Операции со списками. Назначение и виды паттернов.
11. Основные шаблоны. Порождающие шаблоны. Структурные шаблоны. Поведенческие шаблоны.
12. Событийно-управляемое программирование.
13. Элементы управления. Диалоговые окна. Обработчики событий.
14. Введение в графику. Методы оптимизации программного кода.
15. Цели и методы рефакторинга.
16. Правила разработки интерфейсов пользователя.
17. Работа с базами данных. Доступ к данным.
18. Создание таблицы, работа с записями. Способы создания команд.

#### по МДК 01.02. Поддержка и тестирование программных модулей

1. Расскажите о тестировании как о части процесса верификации программного обеспечения.
2. Какие существуют виды ошибок и методы отладки?
3. Перечислите методы тестирования.
4. Как классифицируются тестирования по уровням?
5. Тестирование производительности: в чем суть?
6. Что такое регрессионное тестирование?
7. Перечислите средства разработки технической документации. Расскажите о технологии разработки документов.
8. Документирование программного обеспечения в соответствии с Единой системой программной документации: основное.
9. Автоматизация разработки технической документации.
10. Автоматизированные средства оформления документации.

#### по МДК 01.03. Разработка мобильных приложений

1. Какие существуют основные платформы мобильных приложений, представьте их сравнительную характеристику?
2. Что такое нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, каковы их области применения?
3. Перечислите основные языки для разработки мобильных приложений (Java, Objective-C и др.)
4. Какие существуют инструменты разработки мобильных приложений (JDK/ AndroidStudio/ WebView/ Phonegap и др.)
5. Инструментарий среды разработки мобильных приложений.
6. Расскажите о структуре типичного мобильного приложения.
7. Элементы управления и контейнеры.
8. Работа со списками.
9. Способы хранения данных.

#### по МДК 01.04. Системное программирование



1. Подсистемы управления ресурсами.
2. Управление процессами.
3. Управление потоками.
4. Параллельная обработка потоков.
5. Создание процессов и потоков.
6. Обмен данными между процессами. Передача сообщений.
7. Анонимные и именованные каналы.
8. Сетевое программирование сокетов.
9. Динамически подключаемые библиотеки DLL
10. Сервисы.
11. Виртуальная память. Выделение памяти процессам.
12. Работа с буфером экрана.

### ***Критерии оценки:***

**Отметка «5»:** ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, литературным языком. Ответ самостоятельный.

**Отметка «4»:** ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

**Отметка «3»:** ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный.

## **Фонд тестовых заданий**

### по МДК 01.01. Разработка программных модулей

1. Delphi – это...
  - 1) Язык программирования низкого уровня
- 2) Система объектно-ориентированного визуального программирования**
  - 3) Программа обработки видео
  - 4) Программа для перевода кода программы на язык машинных кодов.
2. RAD – это...
  - 1) Программа записи видео
  - 2) Средство модульного программирования
- 3) Среда быстрой разработки приложений**
  3. Язык программирования Delphi
    - 1) Object Pascal**
    - 2) Java
    - 3) C#
  4. IDE – это...

- 1) История развития Delphi
- 2) Приложение, разрабатываемое программистом

### 3) Интегрированная среда разработки

5. IDE не включает в себя: (несколько вариантов ответа)

- 1) Редактор кода
- 2) Высокопроизводительный компилятор в машинный код
- 3) Объектно-ориентированную модель компонент

### 4) Эффекты и переходы

- 5) Визуальное построение приложений

### 6) Сопровождение ПП

- 7) Средство для построения баз данных
  6. Объектно-ориентированная программа - ...
- 1) Программирование, основанное на объектах

### 2) Совокупность объектов и способов их взаимодействия

- 3) Структура среды программирования
  7. Установить соответствие

1) Объект	а) Атрибуты (основные характеристики), которые описывают особенности объекта (цвет, ширина, положение и т.д.)
2) Событие	б) Совокупность данных (компонентов) и методов работы с ними
3) Свойство	с) Отклик на внешнее воздействие
<b>1 – б, 2 – с, 3 - а</b>	

8. Дополнить предложение. Основным окном разрабатываемого приложения является ...

- 1) Код
- 2) Форма**
- 3) Библиотека
- 4) Объект

9. Дополнить предложение. Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...

- 1) Компилятора
- 2) C++

### 3) Windows

10. Названия процедур на определенное событие:

- 1) Компоненты

### 2) Обработчики событий

- 3) Редактор кода

11. Назначение главного меню:

### 1) Осуществление функций управления при разработке программ

- 2) Сопровождение программных продуктов
- 3) Автоматизирует процесс тестирования программ

12. Назначение панели инструментов:

- 1) Обработка событий

2) **Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто используемым командам Главного меню**

3) Построение баз данных

13. Назначение палитры компонентов:

1) **содержит пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение**

2) Подделка подписей

3) Управление Paint

14. Дополнить предложение. Главное окно управляет окнами ... *(несколько вариантов ответа)*

1) **Инспектор объектов**

2) Язык ассемблера

3) **Редактора кода**

4) **Проектировщика форм**

5) Трансляции программы

15. Заготовка разрабатываемого приложения - ...

1) Компилятор

2) Свойство

3) **Окно проектировщика форм**

4) Главное окно

16. Действия которые нельзя выполнить с помощью проектировщика форм

1) Добавить компоненты в форму

2) **Сменить язык программирования**

3) Модифицировать форму и её компоненты

4) Связать обработчики событий компонента с программой на Object Pascal, содержащейся в редакторе кода

5) **Документирование программы**

17. Дополнить предложение. Инспектор объектов позволяет ... *(несколько вариантов ответа)*

1) **Устанавливать свойства объектов**

2) Изменять структуру программного кода

3) Изменять наследование классов объектов

4) **Назначать методы обработки событий**

18. Дополнить предложение. Инспектор объектов состоит из ... *(несколько вариантов ответа)*

1) Unit – программного модуля

2) **Properties – списка свойств,**

3) **Events – списка событий.**

4) Begin...end.

19. Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:

1) Окно компилятора

2) **Окно редактора кода**

3) Окно проектировщика

4) Главное окно

20. Задание метки текста из редактора ввода:

- 1) Memo1.Caption := Edit1.Text;
- 2) Label1. Caption := Edit1.Text;**
- 3) Form1. Caption := Edit1.Text;

21. Обнуление строки ввода:

- 1) Edit1.Text :='';**
- 2) Edit1.Text := TМемо;
- 3) Edit1.Text :='Закругляемся'

22. Передача фокуса ввода на редактор ввода

- 1) Label1.SetFocus;
- 2) Edit1.Add;
- 3) Edit1.SetFocus;**

23. Открыть файл проекта Project1 можно, нажав:

- 1) CTRL+F4
- 2) CTRL+ALT+F10
- 3) CTRL+F12**
- 4) ALT+F12

24. Для обозначения комментария не используются:

- 1) (\*комментарий\*)
- 2) \комментарий/**
- 3) {комментарий}
- 4) //комментарий

25. Зарезервированное слово, объявляющее блок подключаемых к проекту модулей:

- 1) Program
- 2) Begin..end
- 3) Uses**
- 4) Forms

26. Дополнить предложение. Объект Application определяет ...

- 1) Свойства и методы программы-приложения для Windows**
- 2) Отображение размещения формы
- 3) Перемещение по программному коду

27. Метод Application.Run:

- 1) Создает окно формы, регистрирует входящие в него компоненты
- 2) Подключает автоматический цикл обработки сообщений Windows к приложению**
- 3) Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия

28. Контейнер, содержащий все элементы программы:

- 1) Unit1
- 2) Form1**
- 3) Edit1
- 4) Memo1

29. Дополнить предложение. Свойство **Position** определяет ...

- 1) Размещение и размер формы**
- 2) Подключение модулей

3) Цвет шрифта

### 30. poDesigned:

- 1) Форма выводится в центр экрана
- 2) Windows автоматически определяет начальную позицию и размеры формы

#### 3) Форма отображается в той позиции и с теми размерами, которые были установлены при её конструировании

31. Определяет способ выравнивания компонента внутри контейнера:

- 1) Свойство Position

#### 2) Свойство Align

- 3) Свойство Memo
- 4) Свойство ScrollBars

### 32. alClient:

- 1) Компонент занимает всю поверхность контейнера
- 2) Выравнивание не используется
- 3) Компонент помещается в нижнюю часть контейнера

### 33. alCustom:

- 1) Компонент помещается в верхнюю часть контейнера
- 2) Компонент помещается в нижнюю часть контейнера

#### 3) Размеры и положение компонента устанавливаются разработчиком

34. Объект Memo1 - ...

- 1) Однострочный редактор

#### 2) Многострочный редактор, содержащий несколько строк текста

- 3) Текстовая надпись

35. Свойство ScrollBars:

#### 1) Задаёт в поле редактирования полосы прокрутки

- 2) Определяет способ выравнивания компонента внутри контейнера
- 3) Задаёт имя объекта

36. Функциональная кнопка с изображением устанавливается из:

- 1) Палитра компонентов -> Standart -> Button
- 2) Палитра компонентов -> Standart -> Label

#### 3) Палитра компонентов -> Additional -> BitBtn

37. Дополнить предложение. За стиль оформления внешней и внутренней рамок отвечают свойства ... (несколько вариантов ответа)

- 1) Align

#### 2) BevelOuter

#### 3) BevelInner

- 4) ssBoth

- 5) Caption

38. Установить соответствие

1) bvLowered	a) Компонент помещается в нижнюю часть контейнера
2) poScreenCenter	b) Есть обе полосы прокрутки
3) bvNone	c) Рамка вдавлена
4) alBottom	d) Форма выводится в центре экрана, её высота и ширина не изменяются
5) ssBoth	e) Рамка отсутствует

1 –с, 2- d, 3 – е, 4 – а, 5 -b

39. VCL – это...

- 1) Приложение Delphi
- 2) Алгоритмизация решения задачи
- 3) Библиотека визуальных компонентов**
- 4) Характеристики объектов

40. Дополнить предложение. Компоненты, которые видны на форме, как во время создания приложения, так и во время работы приложения называются ...

**1) Визуальными**

- 2) Реальными
- 3) Невизуальными
- 4) Основными

41. К невидимым компонентам относят:

- 1) Кнопки, метки, списки блоков
- 2) Таймеры, компоненты для работы с базами данных, списки изображений**
- 3) Библиотеки, звуки, коды

42. Установить соответствие:

1) Standart	a) Обеспечивает доступ к 32-битным элементам Windows
2) Additional	b) Компоненты, реализующие интерфейс с пользователем и процесс управления данными для БД
3) Win32	c) Включает стандартные компоненты, обеспечивающие некоторые функции интерфейса пользователя
4) System	d) Специализированные компоненты, организующие доступ к БД
5) DataAccess	e) Набор компонентов для доступа к системным ресурсам (OLE, DDE)
6) DataControl	f) Дополнительные интерфейсные компоненты для красочного оформления приложения
<b>1) – с, 2) – f, 3) – а, 4) – е, 5) – d, 6) – b;</b>	

43. Дополнить предложение. Имя компонента состоит из... (несколько вариантов ответа)

- 1) Названия компонента**
- 2) Заготовки компонента
- 3) Порядкового номера компонента**
- 4) Произвольного значения ItemIndex

44. Дополнить предложение. Свойства Height и Width показывают...

- 1) Какое действие должен выполнить программист
- 2) Размеры компонента**
- 3) Исполняемый блок программы

45. Дополнить предложение. Проект Delphi представляет собой...

- 1) Набор программных единиц – модулей**
- 2) Основной интерфейсный элемент в Delphi
- 3) Атрибуты (основные характеристики), которые описывают особенности объекта

46. Модуль – это...

- 1) Средство для создания, изменения исходных файлов, которые содержат написанную на языке программирования программу

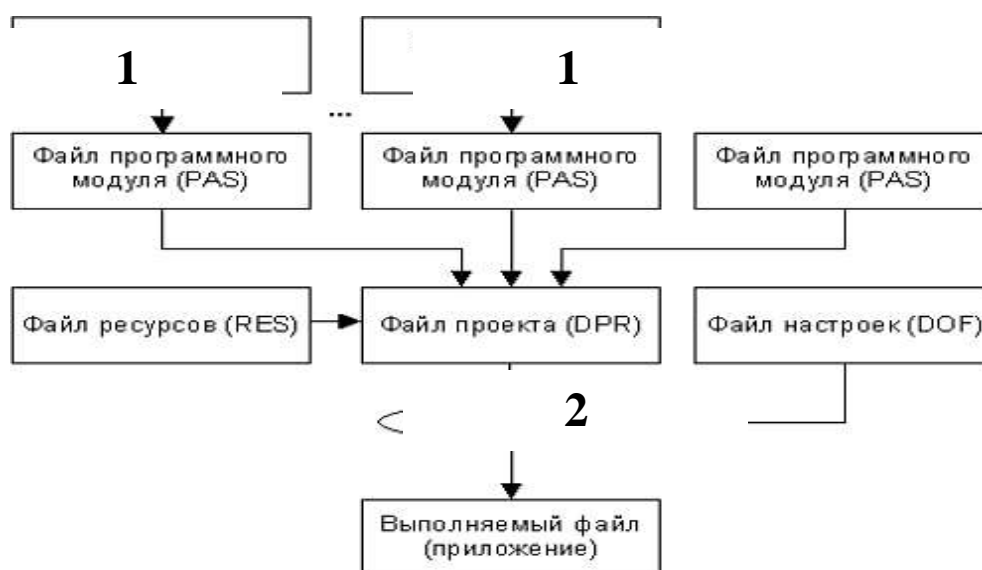
2) Автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры и функции) и, возможно некоторые исполняемые операторы иницирующей части

3) Набор инструментов, которые используются для преобразования символов

47. Установить соответствие

1) Файл проекта	a) (.pas) соответствующий файл модуль для хранения кода
2) Файл модуля	b) (.res) содержит пиктограмму и прочие ресурсы
3) Файл формы	c) (.dpr) текстовый файл используется для хранения информации о формах и модулях, содержит операторы инициализации и запуска программы на выполнение
4) Файл опций проекта	d) (.dfm) двоичный файл, который создается для хранения информации о ваших формах и фреймах
5) Файл ресурсов	e) (.cfg) хранит установки проекта
6) Файл конфигурации проекта	f) (.dof) хранит установки опций проекта
<b>1) – с, 2) – а, 3) – d, 4) – f, 5) – b, 6) - e</b>	

48. Заполнить схему:



1) 1 – файл формы (DFM); 2 – файл проекта (DPR)

2) 1 – файл проекта (DPR); 2 – файл формы (DFM)

3) 1 – файл формы (DPR); 2 – файл проекта (DFM)

4) 1 – файл проекта (DFM); 2 – файл формы (DPR)

49. Дополнить предложение. Набор всех файлов, необходимых для создания приложения, называется...

1) Модулем

**2) Проектом**

3) Задачей

4) Формой

50. Файлы описания форм – это...

1) Отклики на внешнее воздействие

2) Файлы подключающие все используемые программные модули

**3) Текстовые файлы с расширением DFM, в котором сохраняются значения свойств формы и ее компонентов**

51. Содержимое dfm-файла вызывается:

**1) Контекстное меню формы -> View as Text**

2) Контекстное редактора кода -> View as Form

3) File -> Save as...

52. Дополнить предложение. При сборке приложения описание из DFM-файла помещается в...

1) Файл модуля

**2) Область ресурсов**

3) Область установок проекта

53. Программный модуль (Unit) содержит:

**1) Все относящиеся к форме объявления и методы обработки событий, написанные на языке Object Pascal**

2) Заготовку, макет одного из окон разрабатываемого приложения

3) Пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение

54. Дополнить предложение. Имя модуля должно совпадать с...

1) Приложением Windows

2) Спецификацией задачи

3) Объектом в Delphi

**4) Именем файла при сохранении File->Save as...**

55. Интерфейсная секция модуля (interface) содержит: (несколько вариантов ответа)

**1) Список стандартных модулей библиотеки VCL**

**2) Раздел описания типов**

3) Компилятор

**4) Раздел описания переменных (объявление собственно объекта формы)**

5) Инспектор объектов и проектировщик форм

56. Дополнить предложение. Класс формы объявляется в разделе...

**1) Type**

2) Uses

3) Project

4) Var

57. Дополнить предложение. В описании класса форм помещенные на форму компоненты представлены...

1) Границами формы

**2) Полями формы**

3) Индексами формы

58. Published – это...

**1) Атрибут видимости, позволяющий работать с компонентами и методами обработки событий на визуальном уровне**

2) Директива препроцессора

3) Текст программы на языке Object Pascal

59. Пустые секции, в которых можно размещать любые вспомогательные поля, методы и свой-



ства: *(несколько вариантов ответа)*

**1) Private**

2) Uses

3) Var

**4) Public**

5) Program

60. Объявление объекта формы:

1) unit Unit1;

2) type TForm1 = class(TForm)

**3) var Form1: TForm1;**

61. implementation – это...

1) Тип данных Delphi

2) Племя в Южной Африке

**3) Раздел реализации объявлений интерфейса**

62. Подключение файла описания формы:

**1) {\$R \*.dfm}**

2) {\$R \*.pas}

3) {R \*.dpr}

63. Дополнить предложение. После подключения файла описания формы следует...

1) Раздел описания типов

2) Атрибут видимости

3) Описание размеров используемых компонентов

**4) Реализация методов обработки событий**

64. Установить метод обработки события Button1Click: *(несколько вариантов ответа)*

**1) Двойной щелчок по компоненту Button1 на форме**

**2) Выделить Button1 ->Events -> OnClick**

3) Выделить BitBtn1 ->Events -> OnClick

65. Программный файл проекта – это...

1) Программист

**2) Главная часть приложения, с которой начинается выполнение программы и которая обеспечивает инициализацию других модулей**

3) Пустая секция

66. К комментариям не относятся: *(несколько вариантов ответа)*

**1) Program**

2) {Form1}

**3) {\$R \*.res}**

4) //выполнение команды

67. Дополнить предложение. Сочетание end со следующей за ней точкой называется...

1) Терминалом программной единицы

**2) Терминатором программной единицы**

3) Андроидом программной единицы

68. Содержимое DPR-файла вызывается: *(несколько вариантов ответа)*

1) Ctrl+alt

2) File -> New->Application

**3) Project -> View Source**

4) Ctrl+F12

69. Дополнить предложение. Зарезервированное слово **uses** сообщает компилятору о...

**1) Модулях, которые должны рассматриваться как необъемлемые части программы и которые содержатся в других файлах**

2) Начале работы приложения

3) Ошибках, которые могут возникнуть при компилировании

70. Модуль Forms – это...

1) Дополнительный модуль, содержащий неиспользуемые компоненты

2) Модуль для построения более красочного приложения

**3) Обязательный для всех программ модуль, который содержит определение объекта Application**

71. Запись `{SR *.res}` означает:

1) Комментарий, указывающий программисту где искать ресурсы

**2) Директиву, подключающую к результирующему выполняемому файлу ресурсы**

3) Директиву, подключающую к результирующему выполняемому файлу форму

72. Установить соответствие:

1) Application.Initialize	a) загружает и инициализирует форму
2) Application.CreateForm	b) активизирует форму и начинает выполнение приложения
3) Application.Run	c) подготавливает приложение к работе
<b>1 – c; 2 – a; 3 – b</b>	

73. Включение множественного выбора элементов компонента ListBox1

1) `ListBox2.MultiSelect:=false;`

**2) `ListBox1.MultiSelect:=true;`**

3) `ListBox1.MultiSelect:=false;`

4) `ListBox1.Checked:=true;`

74. Запись `ListBox1.Selected[3]:=true` означает что:

1) Необходимо установить три дополнительных элемента

2) Необходимо удалить три лишних элемента

**3) Выделен четвертый элемент в компоненте**

4) Выделен третий элемент в компоненте

75. Простейшая функция вывода на экран текстовых сообщений:

1) `ListBox1.Items.Add();`

**2) `ShowMessage();`**

3) `Case...of;`

4) `CheckBox1.Checked:=true;`

76. Полная очистка списка:

**1) `ListBox1.Clear;`**

2) `ListBox1.Items.Add();`

3) `RadioGroup1.ItemIndex:=0;`

77. Снятие «флажка» компонента CheckBox1

- 1) ListBox1.Checked:=true;
- 2) CheckBox1.Checked:=false;**
- 3) CheckBox1.Checked:=true;
- 4) CheckBox1.Caption :='Снять флажок';

78. Свойство Align=AllBottom:

- 1) Компонент перемещается в центр формы
- 2) Компонент перемещается в верхнюю часть контейнера
- 3) Компонент перемещается в нижнюю часть контейнера и его ширина становится равной ширине контейнера**

79. Объект ComboBox - ...

- 1) Многострочный редактор
- 2) Комбинированный список**
- 3) Функциональная кнопка

80. Установка в ComboBox1 первой строки массива Item:

- 1) ItemIndex=0**
- 2) ItemIndex=-1
- 3) ItemIndex=1
- 4) ItemIndex=356

81. Объект RadioGroup - ...

- 1) Команда «Авторадио»
- 2) Панель группы радиокнопок**
- 3) Метка

82. Дополнить предложение. Компонент-флажок, который используется для обозначения включения или отключения какой-либо опции, называется...

- 1) RadioGroup
- 2) Функциональная кнопка
- 3) CheckBox**
- 4) ListBox

### по МДК 01.03. Разработка мобильных приложений

1) Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

- а) Android SDK
- б) JDK
- в) плагин ADT
- г) Android NDK

2) С какой целью был создан Open Handset Alliance?

- А) писать историю развития ОС Android
- б) продавать смартфоны под управлением Android
- в) рекламировать смартфоны под управлением Android
- г) разрабатывать открытые стандарты для мобильных устройств

3) Библиотеки, реализованные на базе PacketVideo OpenCORE:

- А) Media Framework
- Б) SQLite В) FreeType
- Г) 3D библиотеки

4) Какой движок баз данных используется в ОС Android?

- A) InnoDB
- Б) DBM
- В) MyISAM
- Г) SQLite

5) С какой целью инструмент Intel\* Integrated Performance Primitives (Intel\* IPP) используется в среде разработки Intel\* Beacon Mountain?

- A) для оптимизированной обработки данных и изображений
- Б) позволить разработчикам оптимизировать загрузку системы при использовании процедур OpenGL
- В) для ускорения работы эмулятора в среде разработки
- Г) позволить разработчикам эффективно распараллелить C++ мобильные приложения

6) Intel XDK поддерживает разработку под:

- A) JavaFX Mobile
- Б) Apple iOS, BlackBerry OS
- В) MtkOS, Symbian OS, Microsoft Windows 8
- Г) Android, Apple iOS, Microsoft Windows 8, Tizen

7) Каждый приемник широковещательных сообщений является наследником класса ...

- A) ViewReceiver
- Б) IntentReceiver
- В) ContentProvider
- Г) BroadcastReceiver

8) Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?

- A) GUI
- Б) View
- В) UIComponent
- Г) Widget

9) Какой слушатель используется для отслеживания события касания экрана устройства?

- A) OnPressListener
- Б) onTouchListener
- В) OnClickListener
- Г) OnInputListener

10) В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?

- A) res/value
- Б) res/items
- В) res/layout
- Г) res/menu

11) Фоновые приложения ...

- A) после настройки не предполагают взаимодействия с пользователем, большую часть времени на ходятся и работают в скрытом состоянии
- Б) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями
- В) небольшие приложения, отображаемые в виде графического объекта на рабочем столе
- Г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

- 12) Полный иерархический список обязательных файлов и папок проекта можно увидеть на вкладке ...
- А) Package Explorer
  - Б) Internet Explorer
  - В) Navigator
  - Г) Project Explorer
- 13) Какой компонент управляет распределенным множеством данных приложения? А) сервис (Service)  
Б) активность (Activity)  
В) приемник широкоэмитательных сообщений (Broadcast Receiver)  
Г) контентпровайдер (Content Provider)
- 14) Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?  
А) html  
Б) xml  
В) gml  
Г) xhtml
- 15) Выберите верную последовательность действий, необходимых для создания в приложении контентпровайдера.
- А) Создание класса наследника от класса ContentProvider; Определение способа организации данных; Заполнение контентпровайдера данными
  - Б) Проектирование способа хранения данных; Определение способа организации данных;
  - В) Создание класса наследника от класса ContentProvider; Заполнение контентпровайдера данными; Определение способа работы с данными
  - Г) Проектирование способа хранения данных; Создание класса наследника от класса ContentProvider; Определение строки авторизации провайдера, URI для его строк и имен столбцов
- 16) Выберите верные утверждения относительно объекта намерения (Intent).
- А) представляет собой структуру данных, содержащую описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса
  - Б) используется для передачи сообщений пользователю
  - В) используется для получения инструкций от пользователя
  - Г) используются для передачи сообщений между основными компонентами приложений
- 17) Расположение элементов мобильного приложения: А) полезно для передачи иерархии Б) влияет на удобство использования  
В) полезно для создания пространственных отношений между объектами на экране и объектами реального мира  
Г) все варианты ответа верны
- 18) Какие элементы управления применяются для действий по настройке?  
А) командные элементы управления Б) элементы выбора В) элементы ввода Г) элементы отображения
- 19) Примерами комбоэлементов не являются:  
А) комбосписок  
Б) все вышеперечисленное  
В) комбокнопка  
Г) комбополе

- 20) Дизайн или проектирование интерфейса для графических дизайнеров: А) все варианты ответа верны  
Б) прозрачность и понятность информации  
В) тон, стиль, композиция, которые являются атрибутами бренда  
Г) передача информации о поведении посредством ожидаемого назначения
- 21) Более крупные элементы:  
А) привлекают больше внимания  
Б) все варианты ответа верны  
В) размер не влияет на уровень внимания  
Г) привлекают меньше внимания
- 22) К традиционным типографическим инструментам не относят  
А) масштаб  
Б) цвет  
В) разреженность  
Г) выравнивание по сетке
- 23) К элементам ввода относят:  
А) ограничивающие элементы ввода  
Б) ползунки  
В) счетчики  
Г) все вышеперечисленное
- 24) Выделяют следующие категории плотности экрана для Android-устройств:  
А) LDPI, MDPI, HDPI, XHDPI, и XXXHDPI  
Б) правильный вариант ответа отсутствует  
В) LDPI, MDPI, HDPI, XHDPI, и XXXHDPI  
Г) LDPI, MDPI, HDPI
- 25) Следующие утверждения не верны:  
А) не используйте интерфейсные элементы  
Б) картинки работают быстрее, чем слова  
В) на любом шаге должна быть возможность вернуться назад  
Г) если объекты похожи, они должны выполнять сходные действия
- 26) Следующие утверждения верны:  
А) текстура бесполезна для передачи различий или привлечения внимания  
Б) восприятие направления затруднено при больших размерах объектов  
В) все варианты ответа верны  
Г) люди легко воспринимают контрастность
- 27) Основные вкладки (FixedTabs) удобны при отображении  
А) от четырех вкладок  
Б) двух вкладок  
В) трех и более вкладок  
Г) трех и менее вкладок
- 28) Диалоговое окно, содержащее линейку процесса выполнения какого-то действия — это А) DatePickerDialog  
Б) AlertDialog  
В) ProgressDialog  
Г) DialogFragment
- 29) Уведомления стоит использовать, когда

- А) сообщение не требует ответа пользователя, но важно для продолжения его работы
- Б) сообщение является важным и требует немедленного прочтения и ответа
- В) сообщение является важным, однако требует немедленного прочтения, но не ответа
- Г) сообщение является важным, однако не требует немедленного прочтения и ответа

30) Какой метод запускает новую активность?

- А) startActivity()
- Б) beginActivity()
- В) intentActivity()
- Г) newActivity()

31) ProgressDialog это:

- А) контейнер для создания собственных диалоговых окон
- Б) диалоговое окно с предопределенным интерфейсом, позволяющее выбрать дату или время
- В) диалоговое окно, содержащее линейку процесса выполнения какого-то действия
- Г) диалоговое окно, которое может содержать заголовок, до трех кнопок, список выбираемых значений или настраиваемое содержимое

32) AlertDialog это:

- А) контейнер для создания собственных диалоговых окон
- Б) диалоговое окно, содержащее линейку процесса выполнения какого-то действия
- В) диалоговое окно, которое может содержать заголовок, до трех кнопок, список выбираемых значений или настраиваемое содержимое
- Г) диалоговое окно с предопределенным интерфейсом, позволяющее выбрать дату или время

33) Что необходимо сделать при добавлении в проект новой активности? А) скачать и установить специальный инструмент

- MultiActivity SDK
- Б) прописать в манифесте информацию о новой активности
- В) создать новый проект
- Г) запустить эмулятор

34) Системы позиционирования смартфона могут включать

- А) все перечисленное
- Б) систему GPS
- В) систему ГЛОНАСС
- Г) сигналы WiFi и Bluetooth

35) Какая константа не определена в классе MotionEvent, для обозначения сенсорных событий А) ACTION\_DOWN

- Б) ACTION\_UP
- В) ACTION\_CLICK
- Г) ACTION\_MOVE

36) С какой целью используется метод release() в классах MediaPlayer и MediaRecorder?

- А) конец жизненного цикла объекта и освобождение ресурсов
- Б) перевод объекта в ожидающее состояние
- В) обновление объекта и запуск его работы
- Г) создание объекта и запуск его работы

37) К датчикам окружающей среды, встроенным в мобильное устройство относят

- А) датчики вектора вращения
- Б) датчики освещенности

- В) акселерометры
- Г) гироскопы

38) Библиотека Universal Image Loader for Android позволяет:

- А) парсить HTML-страницы
- Б) строить графики и диаграммы
- В) загружать, кешировать и отображать изображения
- Г) использовать анимацию, доступную только с версии 3.x, на более ранних вариантах платформы Android

39) Facebook SDK for Android — это библиотека, позволяющая:

- А) получать доступ к информации любого пользователя
- Б) отправлять рекламные сообщения от имени пользователя
- В) писать сообщения на стену, читать и менять статусы, смотреть ленту друзей
- Г) парсить страницы пользователей

40) Что из перечисленного не относится к правилам безопасности при подключении библиотек?

- А) с осторожностью использовать библиотеки из сомнительных источников
- Б) ознакомиться с форумами и сайтами, где могут обсуждаться библиотеки
- В) лично познакомиться с разработчиками библиотеки
- Г) использовать скомпрометированные библиотеки

41) Библиотека MapNavigator предназначена для:

- А) работы с любыми картами
- Б) работы с Яндекс.Картами
- В) морской навигации
- Г) работы с картами Google Maps

42) Библиотека jsoup не позволяет:

- А) находить и извлекать данные, используя DOM и селекторы CSS
- Б) манипулировать HTML-элементами, атрибутами и текстом
- В) писать сообщения на стену, читать и менять статусы, смотреть ленту друзей
- Г) принимать в качестве параметра URL, файл или строку

43) При настройке обратной совместимости необходимо добавить в файл манифеста следующую информацию:

- А) только минимальную версию Android SDK
- Б) минимальную и основную (целевую) версии Android SDK
- В) информацию о подключенной библиотеке
- Г) только основную (целевую) версию Android SDK

44) Какая библиотека предназначена для упрощения загрузки изображений? А) Yandex.Metrica for Apps;

- Б) Universal Image Loader for Android
- В) ActionBarSherlock
- Г) NineOldAndroids

45) Библиотеки совместимости предназначены для:

- А) сбора статистики
- Б) рисования графиков
- В) использования возможностей, появившиеся в какой-то версии ОС Android, на более ранних версиях платформы
- Г) подключения нестандартных элементов управления



- 46) Какая библиотека предназначена для использования анимации? А) Universal Image Loader for Android  
Б) NineOldAndroids  
В) Yandex.Metrica for Apps  
Г) ActionBarSherlock
- 47) Для чего служит папка res/anim/ проекта?  
А) в этой папке находятся файлы, содержащие набор картинок, предназначенных для кадровой анимации  
Б) в этой папке находятся файлы, содержащие анимированные ролики для воспроизведения в приложении  
В) в этой папке находятся XML файлы, задающие реализацию анимации свойств  
Г) в этой папке находятся XML файлы, задающие последовательность инструкций анимации преобразований
- 48) В какой файл обязательно добавляется информация при создании нового Activity в приложении?  
А) AndroidManifest.xml  
Б) main.java  
В) layout.xml  
Г) activity.xml
- 49) Какой метод жизненного цикла активности вызывается системой непосредственно перед появлением активности на экране?  
А) onVisible()  
Б) onOpen()  
В) onResume()  
Г) onCreate()
- 50) С какой целью используется метод SurfaceHolder.lockCanvas()?  
А) блокировка Canvas для перерисовки  
Б) игнорирование дальнейшего взаимодействия с Canvas  
В) сокрытие Canvas  
Г) блокировка Canvas от сворачивания.

По МДК 01.04 Системное программирование

1. Операционная система:
  - а) система программ, которая обеспечивает совместную работу всех устройств компьютера по обработке информации**
  - б) система математических операций для решения отдельных задач
  - в) система планового ремонта и технического обслуживания компьютерной техники
2. Программное обеспечение (ПО) – это:
  - а) совокупность программ, позволяющих организовать решение задач на компьютере**
  - б) возможность обновления программ за счет бюджетных средств
  - в) список имеющихся в кабинете программ, заверен администрацией школы
3. Загрузка операционной системы – это:
  - а) запуск специальной программы, содержащей математические операции над числами
  - б) загрузка комплекса программ, которые управляют работой компьютера и организуют диалог пользователя с компьютером**
  - в) вложение дискеты в дисковод
4. Система программирования – это:
  - а) комплекс любимых программ программиста
  - б) комплекс программ, облегчающий работу программиста**

- c) комплекс программ, обучающих начальным шагам программиста
- 5. Прикладное программное обеспечение – это:
  - a) справочное приложение к программам
  - b) текстовый и графический редакторы, обучающие и тестирующие программы, игры**
  - c) набор игровых программ
- 6. Прикладное программное обеспечение:
  - a) программы для обеспечения работы других программ
  - b) программы для решения конкретных задач обработки информации**
  - c) программы, обеспечивающие качество работы печатающих устройств
- 7. Операционные системы:
  - a) DOS, Windows, Unix**
  - b) Word, Excel, Power Point
  - c) (состав отделения больницы): зав. отделением, 2 хирурга, 4 мед. Сестры
- 8. Системное программное обеспечение:
  - a) программы для организации совместной работы устройств компьютера как единой системы**
  - b) программы для организации удобной системы размещения программ на диске
  - c) набор программ для работы устройства системного блока компьютера
- 9. Сервисные (обслуживающие) программы:
  - a) программы сервисных организаций по бухгалтерскому учету
  - b) программы обслуживающих организаций по ведению делопроизводства
  - c) системные оболочки, утилиты, драйвера устройств, антивирусные и сетевые программы**
- 10. Системные оболочки – это:
  - a) специальная кассета для удобного размещения дискет с операционной системой
  - b) специальная программа, упрощающая диалог пользователь – компьютер, выполняет команды операционной системы**
  - c) система приемов и способов работы конкретной программы при загрузке программ и завершении работы
- 11. Использование одного имени для задания общих для класса действий, что означает способность объектов выбирать внутренний метод, исходя из типа данных, определяет свойство ООП
  - a) Полиморфизм
  - b) Управление событиями
  - c) Инкапсуляция**
  - d) Наследование
- 12. Понятие «инкапсуляция» относится к
  - a) Технологии модульного программирования
  - b) Технологии объектно – ориентированного программирования**
  - c) Технологии императивного программирования
  - d) Технологии модульного программирования
- 13. Свойство ООП, которое может быть смоделировано с помощью таксономической классификационной схемы (иерархии) называется
  - a) Инкапсуляция
  - b) Управление событиями
  - c) Полиморфизм
  - d) Наследование**
- 14. Понятие класса в ООП включает в себя
  - a) Поля и методы класса**
  - b) Процедуры и функции обработки
  - c) Поля и функции обработки
  - d) Поля и процедуры обработки

15. Назначение конструктора объекта
- a) **Только выделяет память под объект**
  - b) Выделяет память и задает начальное значение полям
  - c) Задает начальное значение полям
  - d) Выделяет память, задает начальное значение полям, выполняет любые проверки, заданные программистом
16. Как описывается конструктор объекта
- a) procedure create;
  - b) constructor create;
  - c) function create;
  - d) **function constructor;**
17. Как описывается деструктор объекта
- a) procedure free;
  - b) **destructor free;**
  - c) free;
  - d) function free;
18. Понятия объекта в ООП - это
- a) представитель класса
  - b) **конкретные данные, заданные в классе.**
  - c) компонент панели инструментов
  - d) встроенный объект Delphi
19. Моделями типа «черный ящик» являются
- a) Модели мышления
  - b) Модели, описывающие зависимость параметров состояния объекта от входных параметров
  - c) **Модели, описывающие входные и выходные параметры объекта без учета внутренней структуры объекта**
  - d) Модели «аварийного» ящика на самолетах
20. Моделями типа «белый ящик» являются
- a) Модели мышления
  - b) **Модели, описывающие зависимость параметров состояния объекта от входных параметров**
  - c) Модели, описывающие входные и выходные параметры объекта с учетом внутренней структуры объекта
  - d) Модели, описывающие выходные данные в программе
21. Программа «драйвер» служит для
- a) запуска программы на выполнение
  - b) имитации запуска программы на выполнение
  - c) **проверки правильности работы программы**
  - d) передачи параметров в процедуры и функции
22. Программа «заглушка» служит для
- a) запуска программы на выполнение
  - b) **имитации запуска другой программы на выполнение**
  - c) проверки правильности работы программы
  - d) имитации передачи параметров в другой модуль
23. Какие методы сборки программы существуют
- a) монолитная
  - b) **пошаговая**
  - c) **одновременная**
  - d) постепенная

24. Какой метод тестирования программы учитывает закон распределения входных данных
- a) детерминированное тестирование
  - b) функциональное тестирование
  - c) стохастическое тестирование
  - d) логическое тестирование**
25. Программирование сверху вниз – это
- a) Процесс, при котором от начального предположения осуществляется движение по направлению к лучшим решениям
  - b) Процесс пошагового разбиения алгоритма на все более мелкие части с целью получения таких элементов, для которых можно написать конкретные команды**
  - c) Метод сведения трудной задачи к последовательности более простых
  - d) Исследование древовидной модели пространства решений и ориентация на поиск оптимального решения
26. Загрузочный модуль программы – результат работы
- a) Грамматики
  - b) Транслятора**
  - c) Интерпретатора
  - d) Редактора связей (компоновщика)
27. Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется
- a) строителем кода

- b) компилятор**
- c) переводчиком
- d) преобразователем

28. Результатом компиляции программы на языке высокого уровня является

- a) Командный файл
- b) Объектный файл**
- c) Исходный текст программы на языке высокого уровня
- d) Дисплейный файл

### Основы Ассемблера

1. Установите порядок, в котором осуществляется перевод произвольного двоичного числа в системе счисления с основанием  $q=2^n$

Укажите порядок следования всех 3 вариантов ответа: 1. рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием  $q = 2^n$ ; 2. двоичное число разбить слева и справа (целую и дробную части) на группы по цифр в каждой; 3. если в последних правой и левой группах окажется меньше разрядов, то их надо дополнить справа и слева нулями до нужного числа разрядов

- a) 3 1 2**
- b) 2 1 3
- c) 1 2 3
- d) 3 2 1

2. По сколько цифр в группе нужно разбивать двоичное число при переводе его в четверичную систему счисления?

- a) 6

**b) 2**

- c) 3
- d) 4

3. Переведите двоичное число 100011111011 в шестнадцатеричную систему счисления:

- a) 5fa
- b) 8fb**
- c) 8fc
- d) 8fa

4. Чему будет равен результат сложение двоичных чисел 1001 и 1000

- a) 10001**
- b) 11001
- c) 10011
- d) 10101

5. Выполните деление в двоичной системе счисления  $1110:10=$

- a) 101
- b) 011
- c) 110
- d) 111**

6. Переведите двоичное число 1010101 в восьмеричную систему счисления

- a) 125**
- b) 124
- c) 126
- d) 123

7. Установите порядок, в котором осуществляется перевод целого числа в системе счисления с основанием  $q=2^n$ . Укажите порядок следования всех 3 вариантов ответа: 1. данное двоичное число разбить справа налево на группы по  $n$  цифр в каждой; 2. рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основа-

нием  $q = 2p$ ; 3. если в последней левой группе окажется меньше разрядов, то ее надо дополнить слева нулями до нужного числа разрядов

a) **1 3 2**

b) 1 2 3

c) 3 2 1

d) 2 1 3

8. Переведите шестнадцатеричное число FACC в двоичную систему счисления

a) 1111001011011001

b) **1111101011001100**

c) 1011010111111100

d) 1101100111011010

9. Выполните вычисление в двоичной системе счисления  $110101-101=$

a) 110100

b) 110010

c) **110000**

d) 110001

10. В саду 100 фруктовых деревьев - 14 яблонь и 42 груши. В какой системе счисления посчитаны деревья?

a) **В шестеричной**

b) В шестнадцатеричной

c) В двоичной

d) В восьмеричной

11. Умножьте в двоичной системе счисления  $1111*11=$

a) 1111111

b) 101111

c) 1010101

d) **101101**

12. Установите соответствие. Укажите соответствие для всех 4 вариантов ответа: 1) 4      2) 5    3) 6      4) 7

a)  $4=101$   $5=110$   $6=100$   $7=111$

b)  **$5=101$   $6=110$   $4=100$   $7=111$**

c)  $6=101$   $5=110$   $4=100$   $7=111$

d)  $4=101$   $7=110$   $6=100$   $5=111$

13. Выполните умножение в системе счисления  $10101*101=$

a) 110101011

b) **1101001**

c) 111001001

d) 1010101

14. Установите соответствие. Укажите соответствие для всех 4 вариантов ответа:

a) A=1010 B=1111 F=1110 E=1011

b) **A=1010 F=1111 E=1110 B=1011**

c) A=1010 C=1111 F=1110 E=1011

d) A=1010 B=1111 F=1110 C=1011

15. В какой системе счисления будет верным равенство  $7+8=16$ ?

a) 12

b) 6

c) **9**

d) 16

16.  $FFFF+1=$

a) 1FFF

b) **10000**

- c) FFFE
  - d) FFF1
17. Выполните вычитание в двоичной системе счисления 1111-101
- a) 1010**
  - b) 1110
  - c) 1100
  - d) 1001
18. Результат деления в двоичной системе счисления числа 110 на 11 равен
- a) 11
  - b) 10**
  - c) 01
  - d) 101
19. Перевести смешанное двоичное 111100000,101 число в восьмеричную систему счисления
- a) 610,4
  - b) 740,5**
  - c) 170,5
  - d) 1E0,A
20. Переведите восьмеричное число 0,25 в двоичную систему счисления
- a) 0,010101**
  - b) 0,101010
  - c) 0,110111
  - d) 0,101011
21. Алгоритм-это:
- a) Указание на выполнение действий,
  - b) Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи,**
  - c) Процесс выполнения вычислений, приводящих к решению задачи
22. Свойство алгоритма – дискретность, выражает, что:
- a) Команды должны следовать последовательно друг за другом,
  - b) Каждая команда должна быть описана в расчете на конкретного исполнителя,
  - c) Разбиение алгоритма на конечное число команд**
23. Формальное исполнение алгоритма – это:
- a) Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,
  - b) Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,
  - c) Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем ав- томатически**
  - d) Исполнение алгоритма осуществляется исполнителем на уровне его знаний
24. Самое важное свойство алгоритма:
- a) Визуальность,
  - b) Массовость,**
  - c) Дискретность,
  - d) Аудиальность,
25. Какой алгоритм называется линейным:
- a) Выполнение операций зависит от условия,
  - b) Операции выполняются друг за другом,**
  - c) Одни и те же операции выполняются многократно
  - d) Присутствие всех возможных операций в одном алгоритме
26. Графическое задание алгоритма – это:
- a) Способ представления алгоритма с помощью геометрических фигур,**
  - b) Представление алгоритма в форме таблиц и расчетных формул,
  - c) Система обозначений и правил для единообразной и точной записи алгоритмов и их испол-

нения

27. В расчете на кого должен строиться алгоритм:
- a) В расчете на ЭВМ,
  - b) В расчете на умственные способности товарища,
  - c) В расчете на конкретного исполнителя**
28. Какое из перечисленных свойств относится к свойствам алгоритма:
- a) Визуальность,
  - b) Совокупность,
  - c) Аудиальность,
  - d) Понятность**
29. Псевдокоманда DB означает
- a) определить байт;**
  - b) определить слово (2 байта);
  - c) определить двойное слово (4 байта);
  - d) определить 6 байт
  - e) определить учетверенное слово (8 байт);
30. Псевдокоманда DW означает
- a) определить байт;
  - b) определить слово (2 байта);**
  - c) определить двойное слово (4 байта);
  - d) определить 6 байт
  - e) определить учетверенное слово (8 байт);
31. Псевдокоманда DD означает
- a) определить байт;
  - b) определить слово (2 байта);
  - c) определить двойное слово (4 байта);**
  - d) определить 6 байт
  - e) определить учетверенное слово (8 байт);
32. К сегментным регистрам относят
- a) CS, DS, SS и ES.**
  - b) AX, BX, CX и DX
  - c) SP и BP.
  - d) SI и DI
33. К индексным регистрам относят
- a) CS, DS, SS и ES.
  - b) AX, BX, CX и DX
  - c) SP и BP.
  - d) SI и DI**
34. Устанавливает бит результата в 1, если оба бита, бит источника и бит приемника установлены в 1.
- a) AND
  - b) OR**
  - c) XOR (НЕ ИЛИ)
  - d) NOT
35. устанавливает бит результата в 1, если бит источника отличается от бита приемника.
- a) AND**
  - b) OR
  - c) XOR
  - d) NOT
36. Запись регистра в память осуществляется при помощи команды
- a) MOV**



- b) XCHG
  - c) XLAT
37. Порядковый номер элемента в массиве, который всегда начинается с нуля
- a) Индекс**
  - b) Команда
  - c) Псевдокоманда
  - d) Макрокоманда
38. Позволяет сохранять длину исходного текста программы
- a) Индекс
  - b) Команда
  - c) Псевдокоманда
  - d) Макрокоманда**

### ***Критерии оценки:***

**90-100 баллов «отлично»** заслуживает студент, показавший всестороннее систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой; как правило, оценка «отлично» выставляется студентам, усвоившим взаимосвязь основных понятий междисциплинарного курса и их значение для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

**80-90 баллов «хорошо»** заслуживает студент, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе; как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

**60-80 баллов «удовлетворительно»** заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой; как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешности в ответе на зачете, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

**Менее 60 баллов «неудовлетворительно»** выставляется студенту, обнаружившему проблемы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий; как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжать обучение или приступить к профессиональной деятельности по окончании учебного заведения без дополнительных занятий по соответствующему междисциплинарному курсу.

**Отметка «2»:** при ответе обнаружено непонимание обучающимся основного содержания учебного материала или допущены существенные ошибки, которые он не смог исправить при наводящих вопросах преподавателя.

## Кейс-задачи

по МДК 01.04. Системное программирование

### **Задания:**

1. Создание простых программ на языке Ассемблер. Разбор ошибок и отладка программ.
2. Создание программ с разветвлением на Ассемблере. Решение проблем компиляции.
4. Создание программ с использованием циклов. Решение проблем отладки и компиляции.
5. Создание программ с использованием логических операций. Решение проблем логичной структуры алгоритмов программы.
6. Создание программ с использованием циклов и логических операций.
7. Создание программ с использованием команд сдвигов. Разбор ошибок логических операций.
8. Составление программ с использованием ввода-вывода на экран. Разбор ошибок ввода-вывода.

### **Критерии оценки:**

**отметка «5»:** Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

**отметка «4»:** Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

**отметка «3»:** Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

**отметка «2»:** Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

## Ситуационные задачи для учебной практики

по ПМ01 «Разработка модулей программного обеспечения для компьютерных систем»

### 1. Задания по первому разделу: Разработка программных

#### *модулей* **Задание 1.**

Использование окон ввода. Дано двузначное число. Вывести вначале его левую цифру (десятки), а затем - его правую цифру (единицы).

#### **Задание 2.**

С начала суток прошло  $N$  секунд ( $N$  — целое). Найти количество полных минут, прошедших с начала суток.

#### **Задание 3.**

Дано трехзначное число. Проверить истинность высказывания: «Все цифры данного числа различны».

#### **Задание 4**

Дана непустая строка. Вывести коды ее первого и последнего символа. Даны строки  $S$  и  $S_0$ . Найти количество вхождений строки  $S_0$  в строку  $S$ .

**Задание 5.** Дано натуральное число  $n$ , массивы  $A$  и  $B$  размерность  $n$ .

Вести с клавиатуры элементы массивов и вычислить сумму элементов массива  $A$ , количество положительных элементов массива  $B$  и его максимальный элемент.

#### **Задание 6.**

Дан массив  $A$  размерностью 10. Удалить минимальный элемент массива.

#### **Задание 7.**

Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$ , равные соответственно суммам положительных элементов строк. Отсортировать массив  $b$  по возрастанию.

#### **Задание 8.**

Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк, все элементы которых меньше по модулю первого.

### ***Задание 9.***

Дан массив данных об учениках класса: Фамилия, имя, отчество, адрес (улица, дом, квартира) и домашний телефон (если есть). Вывести на экран фамилию, имя и адрес тех учеников, до которых нельзя дозвониться.

## ***2. Задания по второму разделу: Поддержка и тестирование программ-ных модулей***

### ***Задание 1***

Протестировать тестовую базу данных методом тестирования программного обеспечения «черный ящик».

### ***Задание 2***

Протестировать тестовую базу данных методом тестирования программного обеспечения «белый ящик».

### ***Задание 3***

Протестировать тестовую базу данных модульным методом тестирования программного обеспечения.

### ***Задание 4***

Протестировать тестовую базу данных интегрированным методом тестирования программного обеспечения.

## ***3. Задания по третьему разделу: Разработка мобильных***

### ***приложений Задание 1***

Создание проекта приложения в Android Studio. Создание приложения с одним экраном.

### ***Задание 2***

Разработка мобильного приложения из четырех окон средствами Android Studio.

### ***Задание 3***

Разработка программы «Будильник» для ОС Android средствами Android Studio.

## **Задание 4**

Разработка программы «Калькулятор» для ОС Android средствами Android Studio.

### **4. Задания по четвертому разделу: Системное**

#### **программирование Задание 1**

Создание простых программ на языке Ассемблер.

#### **Задание 2**

Создание программ с разветвлением на Ассемблер.

#### **Задание 3**

Создание программ с использованием циклов на Ассемблер.

#### **Задание 4**

Создание программ с использованием логических операций на Ассемблер.

#### **Задание 5**

Создание программ с использованием циклов и логических операций на Ассемблер.

#### **Задание 6**

Создание программ с использованием команд сдвигов. Составление программ с использованием ввода-вывода на экран.

### **Критерии оценки:**

**отметка «5»:** Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

**отметка «4»:** Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

**отметка «3»:** Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

**отметка «2»:** Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

# Ситуационные задачи для производственной практики

по ПМ01 «Разработка модулей программного обеспечения для компьютерных систем»

## 1. Задания по первому разделу: Разработка программных

### *модулей Разработать на Lazarus:*

- Программу «Обычный калькулятор».
- Программу «Строковый калькулятор».
- Программу для нахождения индекса в массиве случайных чисел.
- Программу для нахождения минимального и максимального числа в массиве.
- Программу «Текущее время и текущая дата».
- Программу «Электронные часы».
- Программу «Олимпийский флаг».
- Программу «Градусник»
- Игру «Тетрис».

## 2. Задания по третьему разделу: Разработка мобильных

### *приложений Задание 1*

Напишите фрагмент класса (группу команд), который будет присваивать значения 5 и 9 двум переменным, а затем третья переменная будет получать значение, равное сумме значений первых двух переменных.

### *Задание 2*

Напишите фрагмент класса, который будет присваивать значение 7.5 переменной *x*, а затем будет присваивать переменным *a* и *b* значения вдвое и второе (соответственно) большие, чем значение переменной *x*.

### *Задание 3*

Напишите фрагмент класса, в котором переменным *a* и *b* будут присваиваться числовые значения из диапазона «однозначное положительное число». Затем следует присвоить переменной *c* значение, составленное следующим образом: значение переменной *a* является числом десятков, значение переменной *b* является числом единиц. Например, если переменной *a* присваивается значение 4, а переменной *b* - значение 7, то переменная *c* получает значение 47.

#### ***Задание 4***

Напишите фрагмент класса, который присваивает переменной  $x$  целочисленное значение, а затем присваивает переменной  $y$  треть от значения переменной  $x$ .

### ***3. Задания по четвертому разделу: Системное***

#### ***программирование Задание 1***

Ассемблер. Ввести в регистры произвольные данные и посмотреть их содержи- мое.

#### ***Задание 2***

Ассемблер. Сложить три числа.

#### ***Задание 3***

Ассемблер. Найти произведение трех чисел.

#### ***Задание 4***

Ассемблер. Найти разность двух чисел

#### ***Задание 5***

Ассемблер. Найти частное от двух чисел

#### ***Задание 6***

Ассемблер. Удвоить второе число, если оно больше первого.

#### ***Задание 7***

Ассемблер. Найти сумму первых пяти натуральных чисел (циклом).

#### ***Задание 8***

Ассемблер. Найти сумму первых пяти натуральных чисел (через стек).

#### ***Задание 9***

Ассемблер вывести содержимое последнего регистра стека.

#### ***Задание 10***

Ассемблер. Записать в стек какое-либо значение.

### ***Критерии оценки:***

**отметка «5»:** Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

**отметка «4»:** Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

**отметка «3»:** Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

**отметка «2»:** Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.



**5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ  
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ  
ДЕЯТЕЛЬНОСТИ)**

<b>Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля</b>	<b>Критерии оценки</b>	<b>Методы оценки</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. <b>выполнена оценка сложности алгоритма</b></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	<p>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.</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> -</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 Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

	<p>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>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<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 Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.4 Выполнять тестирование программных модулей</p>	<p>Оценка <b>«отлично»</b> - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами.</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>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля.</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в</p>

<p>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Оценка <b>«отлично»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>Оценка <b>«хорошо»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p> <p>Оценка <b>«удовлетворительно»</b> - определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p>	<p>процессе практики</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>
<p>ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.</p>	<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> -</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</p>

	разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.	процессе практики
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 задач 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 Российской Федерации с учетом особенностей социального и	Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	

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;	- соблюдение норм поведения во время 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 физической подготовленности;	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.;	- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;	