

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Романчук Иван Сергеевич  
Должность: Ректор  
Дата подписания: 16.02.2023 08:57:46  
Уникальный программный ключ:  
e68634da050325a92342840b0b4c0181e1d

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «Тюменский государственный университет»  
Тюменский педагогический институт им. Д.И.Менделеева (филиал)  
Тюменского государственного университета



УТВЕРЖДАЮ  
Заместитель директора филиала  
Шитиков П.М.  
«23» 02 2023 год

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ**  
**КОМПЬЮТЕРНЫХ СИСТЕМ**  
для обучающихся по программе подготовки специалистов среднего звена  
09.02.07 Информационные системы и программирование  
форма обучения очная

Маковийчук Лилия Фриятулловна ПМ.01 Разработка модулей программного обеспечения для компьютерных систем. Фонд оценочных средств профессионального модуля для обучающихся по программе подготовки специалистов среднего звена 09.02.07 Информационные системы и программирование. Форма обучения – очная. Тобольск, 2023.

Фонд оценочных средств профессионального модуля разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от 9 декабря 2016 г. № 1547.

Фонд оценочных средств профессионального модуля опубликован на сайте ТюмГУ [электронный ресурс] / Режим доступа: <https://www.utmn.ru/sveden/education/#>

© Тобольский педагогический институт им. Д.И. Менделеева (филиал) Тюменского государственного университета, 2023

© Маковийчук Лилия Фриятулловна, 2023

## Содержание

1. Общая характеристика фондов оценочных средств.....	3
2. Паспорт фонда оценочных средств.....	8
3. Типовые задания для промежуточной аттестации .....	9

## 1. Общая характеристика фондов оценочных средств

### 1.1. Область применения программы

Фонд оценочных средств профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

### 1.2. Место профессионального модуля в структуре программы подготовки специалистов среднего звена

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем. входит в профессиональный цикл.

### 1.3. Цели и задачи профессионального модуля – требования к результатам освоения.

В результате освоения профессионального модуля обучающийся должен обладать следующими компетенциями:

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

ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	<p><b>Умения:</b> определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</p> <p><b>Знания:</b> номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</p>
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	<p><b>Умения:</b> определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования</p> <p><b>Знания:</b> содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования</p>
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде	<p><b>Умения:</b> организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности</p> <p><b>Знания:</b> психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности</p>
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	<p><b>Умения:</b> грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе</p> <p><b>Знания:</b> особенности социального и культурного контекста; правила оформления документов и построения устных сообщений</p>
ОК 06	Проявлять гражданско-	<b>Умения:</b> описывать значимость своей специальности

	патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	<b>Знания:</b> сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	<p><b>Умения:</b> соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности</p> <p><b>Знания:</b> правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения</p>
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	<p><b>Умения:</b> использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения, характерными для данной специальности</p> <p><b>Знания:</b> роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения</p>
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	<b>Умения:</b> понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

		<b>Знания:</b> правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности
--	--	---

Основные виды деятельности	Код и формулировка компетенции	Показатели освоения компетенции
Разработка модулей программного обеспечения для компьютерных систем	ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Практический опыт: Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
		Умения: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценка сложности алгоритма.
		Знания: Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов.
	ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	Практический опыт: Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля. Разрабатывать мобильные приложения.
		Умения: Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней, в том числе для мобильных платформ.
		Знания: Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем.
ПК.1.3. Выполнять отладку	Практический опыт: Использовать инструментальные средства на этапе отладки программного продукта.	

программных модулей с использованием специализированных программных средств	Проводить тестирование программного модуля по определенному сценарию.
	Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства. Применять инструментальные средства отладки программного обеспечения.
	Знания: Основные принципы отладки и тестирования программных продуктов. Инструментарий отладки программных продуктов.
ПК 1.4. Выполнять тестирование программных модулей	Практический опыт: Проводить тестирование программного модуля по определенному сценарию. Использовать инструментальные средства на этапе тестирования программного продукта.
	Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.
	Знания: Основные виды и принципы тестирования программных продуктов.
ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода	Практический опыт: Анализировать алгоритмы, в том числе с применением инструментальных средств. Осуществлять рефакторинг и оптимизацию программного кода.
	Умения: Выполнять оптимизацию и рефакторинг программного кода. Работать с системой контроля версий.
	Знания: Способы оптимизации и приемы рефакторинга. Инструментальные средства анализа алгоритма. Методы организации рефакторинга и оптимизации кода. Принципы работы с системой контроля версий.
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ	Практический опыт: Разрабатывать мобильные приложения.
	Умения: Осуществлять разработку кода программного модуля на современных языках программирования. Оформлять документацию на программные средства.
	Знания: Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования.

<p>Осуществление интеграции программных модулей</p>	<p>ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент</p>	<p>Практический опыт: Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p> <p>Умения: Анализировать проектную и техническую документацию. Использовать специализированные графические средства построения и анализа архитектуры программных продуктов. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Определять источники и приемники данных. Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace). Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выявлять ошибки в системных компонентах на основе спецификаций.</p> <p>Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и варианты интеграционных решений. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы отладочных классов. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Графические средства проектирования архитектуры программных продуктов. Методы организации работы в команде разработчиков.</p>
<p>Разработка, администриров</p>	<p>ПК 11.1. Осуществлять</p>	<p>Практический опыт: Выполнять сбор, обработку и анализ информации для</p>

ание и защита баз данных	сбор, обработку и анализ информации для проектирования баз данных.	проектирования баз данных.
		Умения: Работать с документами отраслевой направленности. Собирать, обрабатывать и анализировать информацию на предпроектной стадии.
		Знания: Методы описания схем баз данных в современных системах управления базами данных. Основные положения теории баз данных, хранилищ данных, баз знаний. Основные принципы структуризации и нормализации базы данных. Основные принципы построения концептуальной, логической и физической модели данных.

## 2. Паспорт фонда оценочных средств

п/п	Темы дисциплины, МДК, разделы (этапы) практики, в ходе текущего контроля, вид промежуточной аттестации с указанием семестра	Код контролируемой компетенции (или её части), знаний, умений	Наименование оценочного средства (с указанием количество вариантов, заданий и т.п.)
1.	МДК.01.01 Разработка программных модулей 3-5 семестр контрольная работа, 4 семестр курсовое проектирование, 6 семестр дифференцированный зачет	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Коллоквиум (4 вопроса) Контрольная работа (10 задач) Примерная тематика курсового проектирования (31 тема) Вопросы к дифференцированному зачету (155 вопросов)
2.	МДК.01.02 Поддержка и тестирование программных модулей 4-5 семестр контрольная работа, 6 семестр дифференцированный зачет	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Тестирование (38 вопросов) Контрольная работа (18 вопросов) Вопросы к дифференцированному зачету (38 вопросов)
3.	МДК.01.03. Разработка мобильных приложений 4-5 семестр контрольная работа, 6 семестр дифференцированный зачет	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Контрольная работа (4 вопроса) Контрольная работа (5 задач) Вопросы к дифференцированному зачету (20 вопросов)
4.	МДК.01.04. Системное программирование 5 семестр контрольная работа,	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Тестирование (28 вопросов) Контрольная работа

	6 семестр дифференцированный зачет		(30 задач) Вопросы к дифференцированному зачету (32 вопроса)
5.	УП.01.01 Учебная практика 5 семестр - зачет	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Отчет по практике
6.	ПП.01.01 Производственная практика 6 семестр - зачет	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Отчет по практике
1. 7.	ПМ.01 Разработка модулей программного обеспечения для компьютерных систем Экзамен по модулю в 6 семестре	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1	Практические задания

### 3. Типовые задания для промежуточной аттестации

МДК.01.01 Разработка программных модулей	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1
---	--

#### Вопросы к коллоквиуму

1. Понятие жизненного цикла программного обеспечения. Этапы ЖЦ ПО.
2. Технология структурного программирования.
3. Инструментальные средства оформления и документирования алгоритмов программ.
4. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи.

#### Вопросы к коллоквиуму

1. Основные принципы объектно-ориентированного программирования. Классы: основные понятия
2. Назначение и виды паттернов.
3. Классификация паттернов.
4. Основные шаблоны.

### Контрольная работа №1

#### Решение задач

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)

### Контрольная работа №2

1. 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)
2. 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)
3. 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
4. Посчитать сумму элементов побочной диагонали квадратной матрицы
5. Найти среднее арифметическое положительных элементов одномерного массива. Количество элементов массива вводится пользователем, массив заполняется случайными числами
6. Посчитать в строке количество цифр. Строка вводится пользователем с клавиатуры
7. Создать форму Windows. Программа должна предоставлять возможность менять цвет фона формы
8. Создать Windows-приложение, демонстрирующее работу с модальными и немодальными окнами
9. Написать программу на языке C++, которая считает сумму элементов одномерного массива, не используя операцию индексирования [].
10. Написать программу на языке C++, демонстрирующую способы передачи параметров в функции

### Контрольная работа №3

1. Написать рекурсивную функцию для нахождения факториала числа
2. Написать программу на языке C++ для подсчета суммы элементов одномерного массива. Размерность массива вводится пользователем с клавиатуры
3. Написать программу, демонстрирующую возможность создания односвязного списка и добавления элемента в список

4. Написать программу, демонстрирующую возможность создания стека и добавления элемента в любое место стека
5. Создать класс «Прямоугольник», который имеет два конструктора. Первый конструктор принимает в качестве параметров длину и ширину. Второй конструктор создает квадрат с заданной стороной
6. Написать программу, демонстрирующую работу с абстрактными классами
7. Создать функцию на языке C++, подсчитывающую сумму элементов массива. Массив передается как параметр в функцию
8. Написать программу для нахождения максимально элемента матрицы.
9. Написать программу для сортировки массива методом пузырька.
10. Написать программу для нахождения корней квадратного уравнения  $ax^2+bx+c=0$ . Коэффициенты a, b, c вводятся пользователем

### **Примерная тематика курсового проектирования 4 семестр**

1. Разработка электронного словаря с возможностью подключения к базе данных.
2. Создание программы для построения графиков математических функций.
3. Создание программы для общения по локальной сети с выделенным сервером.
4. Разработка простейшего почтового клиента.
5. Разработка калькулятора логических функций.
6. Программа для распознавания графических образов
- 7.Создание интерактивного приложения для выстраивания пользователем последовательностей из однотипных объектов в виде кристаллов.
8. Разработка компьютерного учебника.
9. Разработка интерактивной логической игры "Квест".
10. Разработка программного обеспечения туристического агентства.
11. Создание интерактивной развивающей игры для детей "ПАЗЛ".
12. Разработка управляемой модели исполнителя "Батискаф", выполняющего сбор ресурсов в среде с заданными ограничениями.
13. Разработка программы "Слайд шоу".
14. Моделирование поведения тел, брошенных под углом к горизонту.
15. Разработка интерактивной развивающей игры "Найди отличия".
16. Разработка программного обеспечения для домашней библиотеки.
17. Создание интерактивной развивающей игры для детей "Угадай мелодию".
18. Создание интерактивной развивающей игры для детей "Повтори мелодию".
19. Разработка логической игры "Пазл простой".
20. Разработка игрового квеста "Я ищу".
21. Создание интерактивного развивающего приложения "Лабиринт".
22. Создание интерактивной развивающей игры для детей "Нотная грамота".
23. Создание модели музыкального инструмента "Симулятор фортепиано".
24. Разработка графического редактора со сменными фонами "Раскраска".
25. Разработка игрового приложения «Составь слово».
26. Разработка игрового приложения «Мозаика».
27. Разработка игрового приложения «Угадай число».
28. Разработка приложения по технологии WinApi
29. Кроссплатформенное клиент-серверное приложение "Эхо-сервер" с использованием библиотеки Qt.
- 30.Кроссплатформенное клиентское приложение для работы с сервером по протоколу FTP с использованием библиотеки Qt.
31. Кроссплатформенный текстовый редактор с использованием библиотеки Widgets.

### **Вопросы к дифференцированному зачету**

1. Трехслойная 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 предусловием
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

МДК.01.02 Поддержка и тестирование программных модулей	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1
--	--

#### **Контрольная работа в виде тестирования**

1. Какие существуют особенности управляющего графа программы в случае объектно-ориентированного программирования (ООП)?
  - а) она становится неприменимой
  - б) она требует адаптации по обработке сообщений в) она требует описания поведения программы
  - г) она требует описания не только структуры, но и поведения программы
2. Какие существуют особенности интеграционного тестирования для ООП?
  - а) тестирование дерева классов
  - б) тестирование последовательностей прямых вызовов методов с помощью Р-путей
  - в) тестирование последовательностей обработки сообщений с помощью ММ-путей
  - г) тестирование исключительно цепочек вызовов процедур программ
3. Какие этапы включает методика ООП при тестировании программного комплекса?
  - а) тестирование методов каждого класса программного комплекса б) тестирование отношений между классами с помощью тестов на основе Р-путей или ММ-путей
  - в) тестирование взаимодействия модулей по всей иерархии комплекса
4. Какие этапы методики тестирования используются в ООП?
  - а) тестирование методов каждого класса программы
  - б) тестирование методов класса, входящих в его контекст в) тестирование дерева классов программного проекта, включающего оттестированный класс
5. Какие этапы методов тестирования класса используются в ООП?
  - а) тестирование класса как модуля по выбранному критерию

- б) тестирование класса как иерархической структуры
- в) тестирование классов, входящих в модель проекта
6. Что такое управляющий граф программы (УГП)? а) множество операторов программы
- б) граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов
- в) множество операторов управления
7. Что такое путь в УГП?
- а) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной б) последовательность ветвей управляющего графа программы с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
- в) множество связанных дуг управляющего графа программы
8. Что такое ветвь управляющего графа программы?
- а) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной, которые кодируют либо условные операторы, либо первый и последний операторы управляющего графа программы
- соответственно
- б) часть пути, в котором все внутренние вершины кодируют линейные операторы
- в) начальная и конечная вершина пути
9. Можно ли гарантировать остановку программы на любом тесте? а) в общем случае нет
- б) возможно в частных случаях
- в) задача в общей постановке алгоритмически неразрешима
10. Какие задачи у модульного тестирования?
- а) выявление локальных ошибок реализации алгоритмов модулей б) выявление ошибок при вызове модулей
- в) выявление ошибок взаимодействия модуля с окружением
11. На основе каких принципов строятся тесты для модульного тестирования? а) анализ потоков управления модуля
- б) анализ потоков данных модуля
- в) анализ покрытия в соответствии с заданными критериями C0, C1, C2
12. Каковы фазы процесса построения тестовых путей? а) построение управляющего графа программы б) выбор тестовых путей
- в) генерация тестов, соответствующих выбранным тестовым путям
13. Какие существуют методы построения тестовых путей?
- а) статические
- б) динамические
- в) методы реализуемых путей
14. Как реализуются динамические методы построения тестовых путей? а) наращивание начальных отрезков реализованных путей продолжающимися их фрагментами, чтобы увеличить покрытие б) построение пути методом удлинения за счет добавления дуг в) поиск всех реализуемых путей
15. Какие существуют разновидности интеграционного тестирования? а) монолитное тестирование
- б) нисходящее

тестирование в) восходящее

тестирование

г) Регрессионное тестирование

16. Каковы особенности нисходящего

тестирования? а) необходимость разработки заглушек

б) необходимость разработки среды управления очередностью вызовов модулей

в) параллельная разработка эффективных модулей 17.Каковы особенности восходящего тестирования?

а) минимизация разработки заглушек

б) запаздывание в проверке функциональности реализуемого приложения

в) необходимость разработки среды управления очередностью вызовов модулей

18. В чем заключаются особенности интеграционного тестирования для процедурного программирования?

а) тестирование программных комплексов, заданных в виде иерархических структур модулей

б) использование диаграмм потока управления в качестве модели тестируемого комплекса

в) контроль соответствия спецификациям параметров модулей и межмодульных связей

г) контроль наследования

19. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена.К какой группе относится тест, покрывающий только функцию А?

а) тесты, пригодные для повторного использования б)

тесты, требующие повторного запуска

в) устаревшие

тесты г) новые тесты

20. Возможно ли тестирование программы на всех допустимых значениях параметров?

а)

никогда б) да,

всегда

в) возможно в отдельных случаях

21. Каковы особенности системного тестирования?

а) тестированию подлежит система в целом

б) тесты оперируют пользовательским или другими внешними интерфейсами

в) структура проекта тестируется на уровне подсистем

г) тестирование осуществляется по методу «черного ящика» 22.Какие задачи решаются на уровне системного тестирования?

а) выявление дефектов в функционировании приложения или в работе с ним

б) выявление дефектов использования ресурсов в) выявление несовместимости с окружением

г) выявление непредусмотренных сценариев применения д) использования непредусмотренных комбинаций данных

23. Какие категории тестов разрабатываются для системного тестирования? а) тесты для проверки полноты функциональности

- б) тесты для проверки корректности использования ресурсов
  - в) тесты для проверки стрессовых режимов и оценки производительности
  - г) тесты для проверки защиты от искаженных данных и некорректных действий
  - д) тесты для проверки инсталляции и конфигурации для платформ, предусмотренных спецификацией
  - е) тесты для проверки корректности пользовательской документации
24. Каковы особенности регрессионного тестирования?
- а) регрессионное тестирование является подмножеством системного тестирования
  - б) выбор между полным и частичным перетестированием и пополнением тестовых наборов
  - в) перетестирование предусматривает только контроль частей приложения, связанных с изменениями
25. Какие задачи решает тестировщик, проверяя изменения, внесенные разработчиком в код?
- а) проверка и подтверждение исправления дефекта
  - б) обеспечение гарантий, что качество исправленного кода не ухудшилось
  - в) перетестирование последствий исправлений на предмет обнаружения индуцированных ошибок
  - г) проверка возможности воспроизведения ошибки каким-либо другим способом
26. Какие типы дефектов выявляются при системном или при регрессионном тестировании?
- а) отсутствующая или некорректная функциональность
  - б) некорректность проектной документации
  - в) непредусмотренные данные или неподдерживаемые сценарии использования
  - г) ошибки переносимости на другие платформы
  - д) ошибки пользовательской документации
  - е) ошибки инсталляции и конфигурирования
27. Можно ли гарантировать безопасность метода регрессионного тестирования при отсутствии информации об изменениях в программе?
- а) нет
  - б) да
28. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?
- а) метод повторного прогона всех тестов
  - б) случайные методы
  - в) безопасные методы
  - г) методы минимизации
  - д) методы, основанные на покрытии кода
29. Какими свойствами обладает метод повторного прогона всех тестов?
- а) полнота
  - б) точность
  - в) эффективность
  - г) универсальность
30. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий?
- а) тесты, пригодные для повторного использования

- б) тесты, требующие повторного запуска в) устаревшие тесты
  - г) новые тесты
31. Какие из перечисленных методов тестирования наиболее затратны
- а) системное тестирование с моделируемым окружением
  - б) статические методы тестирования
  - в) интеграционное тестирование
  - г) системное тестирование в реальном окружении и реальном времени
  - д) модульное тестирование
32. Какие из перечисленных методов тестирования дают наиболее надежные результаты?
- а) статические методы
  - б) модульное тестирование
  - в) интеграционное тестирование
  - г) системное тестирование с моделируемым окружением
  - д) системное тестирование в реальном окружении и реальном времени
33. Что влияет на обеспечение качества программного продукта?
- а) тестирование
  - б) анализ дизайна
  - в) обзоры кода
  - г) аудиты процесса разработки
  - д) ревьюирование проектной документации
34. Какие основные уровни выделены в процессе тестирования?
- а) модульное тестирование
  - б) Регрессионное тестирование
  - в) интеграционное тестирование
  - г) системное тестирование
35. Какие методы регрессионного тестирования применяются в условиях высоких требований к программному продукту?
- а) метод повторного прогона всех тестов
  - б) случайные методы
  - в) безопасные методы
  - г) методы, основанные на покрытии кода
  - д) методы минимизации
36. В среднем, метод выборочного регрессионного тестирования...
- а) обнаруживает столько же ошибок, сколько и метод повторного прогона всех тестов
  - б) обнаруживает больше ошибок, чем метод повторного прогона всех тестов
  - в) обнаруживает меньше ошибок, чем метод повторного прогона всех тестов
  - г) может обнаруживать больше или меньше ошибок, чем метод повторного прогона всех тестов
37. На каком этапе регрессионного тестирования удаляются устаревшие тесты?
- а) предсказание целесообразности
  - б) идентификация изменений
  - в) отбор тестов
  - г) выполнение тестов
  - д) создание дополнительных тестов
  - е) обновление базы данных
38. На каком этапе регрессионного тестирования проводится

упорядочение тестов?

- а) предсказание целесообразности
- б) идентификация изменений
  - в) отбор тестов
  - г) выполнение тестов
  - д) создание дополнительных тестов
- е) обновление базы данных

### Вопросы для подготовки к контрольной работе

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

#### Критерии оценки:

Оценка «отлично» выставляется, если студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы:

- обнаруживает всестороннее системное и глубокое знание материала;
- обстоятельно раскрывает соответствующие теоретические положения;
- демонстрирует знание современной учебной и научной литературы;
- владеет понятийным аппаратом;
- демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;
- подтверждает теоретические постулаты примерами из юридической практики;
- способен творчески применять знание теории к решению профессиональных задач;

- имеет собственную оценочную позицию и умеет аргументировано и убедительно ее раскрыть;
- четко излагает материал в логической последовательности.

Оценка **«хорошо»** выставляется, если студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:

- обнаруживает при этом твёрдое знание материала;
- допускает несущественные ошибки и неточности в изложении теоретического материала; исправленные после дополнительного вопроса;
- опирается при построении ответа только на обязательную литературу;
- подтверждает теоретические постулаты отдельными примерами из юридической практики;
- способен применять знание теории к решению задач профессионального характера;
- наблюдается незначительное нарушение логики изложения материала.

Оценка **«удовлетворительно»** выставляется, если студент в основном знает программный материал в объёме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:

- допускает существенные ошибки и неточности в изложении теоретического материала;
- в целом усвоил основную литературу;
- обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;
- требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);
- испытывает существенные трудности при определении собственной оценочной позиции;
- наблюдается нарушение логики изложения материала.

Оценка **«неудовлетворительно»** выставляется, если студент обнаруживает незнание или непонимание большей или наиболее существенной части содержания учебного материала:

- не способен применять знание теории к решению задач профессионального характера;
- не умеет определить собственную оценочную позицию;
- допускает грубое нарушение логики изложения материала;
- допускает принципиальные ошибки в ответе на вопросы;
- не может исправить ошибки с помощью наводящих вопросов.

### **Вопросы для дифференцированного зачета**

1. Концепция тестирования. Тестирование как способ обеспечения качества программного обеспечения.
2. Тестирование, верификация и валидация – различия в понятиях.
3. Цели и задачи процесса тестирования.
4. Жизненный цикл тестирования.
5. Процессы тестирования при разработке программного обеспечения.
6. Техники тестирования требований.
7. Чек-листы, тест – кейсы, наборы тест – кейсов.
8. Анализ и тестирование требований

9. Организация тестирования. Фазы тестирования, основные проблемы тестирования.
10. Тестирование на основе потока управления.
11. Тестирование на основе потока данных. Тестовые сценарии.
12. Критерии выбора тестов. Требования к идеальному критерию тестирования и классы частых критериев.
13. Тестирование программного продукта по структурным критериям
14. Мутационный критерий тестирования.
15. Стохастический критерий тестирования.
16. Тестирование по методу «белого ящика»
17. Тестирование по методу «черного ящика»
18. Модульное и интеграционное тестирование. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных. Организация модульного тестирования.
19. Динамические и статические методы при структурном и объектном подходах
20. Тестирование по функциональному критерию
21. Интеграционное тестирование
22. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования.
23. Оценка оттестированности проекта: метрики и методика оценки.
24. Системное тестирование
25. Регрессионное тестирование: особенности и виды регрессионного тестирования, методы отбора тестов, оценка эффективности.
25. Особенности регрессионного тестирования для ООП
26. Документирование процесса тестирования. Тестовые процедуры программного продукта.
27. Документирование и жизненный цикл дефекта.
28. Отчеты о дефектах.
30. Ошибки, дефекты, сбои, отказы.
31. Тест – план.
32. Метрики покрытия требований и метрики покрытия кода.
33. Составление отчета о тестировании.
34. Отчет о дефектах и его жизненный цикл. Атрибуты (поля) отчета о дефектах. Тестовое покрытие

### **Критерии оценки:**

Оценка «**отлично**» выставляется, если студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы:

- обнаруживает всестороннее системное и глубокое знание материала;
- обстоятельно раскрывает соответствующие теоретические положения;
- демонстрирует знание современной учебной и научной литературы;
- владеет понятийным аппаратом;
- демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;
- подтверждает теоретические постулаты примерами из юридической практики;
- способен творчески применять знание теории к

- решению профессиональных задач;
- имеет собственную оценочную позицию и умеет аргументировано и убедительно ее раскрыть;
  - четко излагает материал в логической последовательности.

Оценка **«хорошо»** выставляется, если студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:

- обнаруживает при этом твёрдое знание материала;
- допускает несущественные ошибки и неточности в изложении теоретического материала; исправленные после дополнительного вопроса;
- опирается при построении ответа только на обязательную литературу;
- подтверждает теоретические постулаты отдельными примерами из юридической практики;
- способен применять знание теории к решению задач профессионального характера;
- наблюдается незначительное нарушение логики изложения материала.

Оценка **«удовлетворительно»** выставляется, если студент в основном знает программный материал в объёме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:

- допускает существенные ошибки и неточности в изложении теоретического материала;
- в целом усвоил основную литературу;
- обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;
- требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);
- испытывает существенные трудности при определении собственной оценочной позиции;
- наблюдается нарушение логики изложения материала.

Оценка **«неудовлетворительно»** выставляется, если студент обнаруживает незнание или непонимание большей или наиболее существенной части содержания учебного материала:

- не способен применять знание теории к решению задач профессионального характера;
- не умеет определить собственную оценочную позицию;
- допускает грубое нарушение логики изложения материала;
- допускает принципиальные ошибки в ответе на вопросы;
- не может исправить ошибки с помощью наводящих вопросов.

МДК.01.03. Разработка мобильных приложений	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1
--	--

### Контрольная работа «№1

1. Основные языки для разработки мобильных приложений (Java, ObjectiveC и др.)
2. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения
3. Ресурсы в Android-приложениях
4. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования

## Контрольная работа №2

### Решение задач

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

### Вопросы для дифференцированного зачета

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

### **Критерии оценки:**

Оценка **«отлично»** выставляется, если студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы:

- обнаруживает всестороннее системное и глубокое знание материала;
- обстоятельно раскрывает соответствующие теоретические положения;
- демонстрирует знание современной учебной и научной литературы;
- владеет понятийным аппаратом;
- демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;
- подтверждает теоретические постулаты примерами из юридической практики;
- способен творчески применять знание теории к решению профессиональных задач;
- имеет собственную оценочную позицию и умеет аргументировано и убедительно ее раскрыть;
- четко излагает материал в логической последовательности.

Оценка **«хорошо»** выставляется, если студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:

- обнаруживает при этом твердое знание материала;
- допускает несущественные ошибки и неточности в изложении теоретического материала; исправленные после дополнительного вопроса;
- опирается при построении ответа только на обязательную литературу;
- подтверждает теоретические постулаты отдельными примерами из юридической практики;
- способен применять знание теории к решению задач профессионального характера;
- наблюдается незначительное нарушение логики изложения материала.

Оценка **«удовлетворительно»** выставляется, если студент в основном знает программный материал в объеме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:

- допускает существенные ошибки и неточности в изложении теоретического материала;
- в целом усвоил основную литературу;
- обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;
- требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);
- испытывает существенные трудности при определении собственной оценочной позиции;
- наблюдается нарушение логики изложения материала.

Оценка «неудовлетворительно» выставляется, если студент обнаруживает незнание или непонимание большей или наиболее существенной части содержания учебного материала:

- не способен применять знание теории к решению задач профессионального характера;
- не умеет определить собственную оценочную позицию;
- допускает грубое нарушение логики изложения материала;
- допускает принципиальные ошибки в ответе на вопросы;
- не может исправить ошибки с помощью наводящих вопросов.

МДК.01.04. Системное программирование	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1
---------------------------------------	--

**Контрольная работа №1**  
**Тестирование**  
**Системное программирование**

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

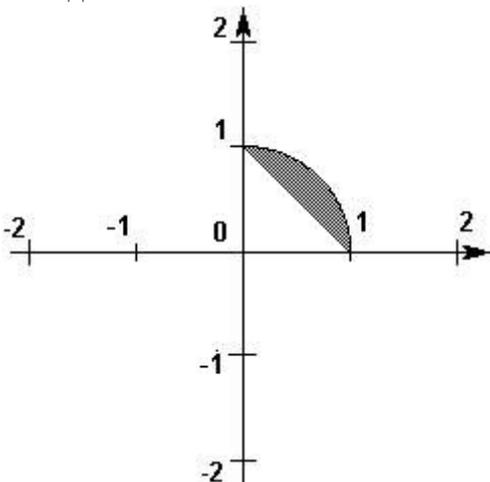
- a) **программы для организации совместной работы устройств компьютера как единой системы**
  - b) программы для организации удобной системы размещения программ на диске
  - c) набор программ для работы устройства системного блока компьютера
9. Сервисные (обслуживающие) программы:
- a) программы сервисных 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) procedurecreate;
  - b) constructorcreate;
  - c) functioncreate;
  - d) **functionconstructor;**
17. Как описывается деструктор объекта
- a) procedurefree;
  - b) **destructorfree;**

- c) free;
  - d) functionfree;
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

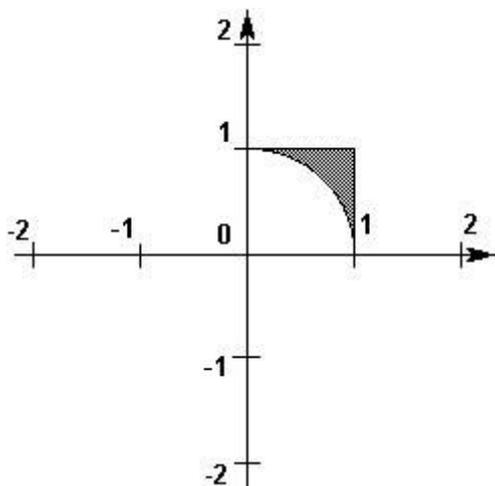
- a) Грамматики
  - b) Транслятора**
  - c) Интерпретатора
  - d) Редактора связей (компоновщика)
27. Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется
- a) строителем кода
  - b) компилятор**
  - c) переводчиком
  - 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

### Контрольная работа Решение задач

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 трем 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

### Вопросы для дифференцированного зачета

1. Что такое системное программирование?
2. Инструментальные средства для создания и запуска программ на языке С

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

### **Критерии оценки:**

Оценка «**отлично**» выставляется, если студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы:

- обнаруживает всестороннее системное и глубокое знание материала;
- обстоятельно раскрывает соответствующие теоретические положения;
- демонстрирует знание современной учебной и научной литературы;
- владеет понятийным аппаратом;
- демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;
- подтверждает теоретические постулаты примерами из юридической практики;
- способен творчески применять знание теории к решению профессиональных задач;
- имеет собственную оценочную позицию и умеет аргументировано и

убедительно ее раскрыть;

- четко излагает материал в логической последовательности.

Оценка **«хорошо»** выставляется, если студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:

- обнаруживает при этом твёрдое знание материала;
- допускает несущественные ошибки и неточности в изложении теоретического материала; исправленные после дополнительного вопроса;
- опирается при построении ответа только на обязательную литературу;
- подтверждает теоретические постулаты отдельными примерами из юридической практики;
- способен применять знание теории к решению задач профессионального характера;
- наблюдается незначительное нарушение логики изложения материала.

Оценка **«удовлетворительно»** выставляется, если студент в основном знает программный материал в объёме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:

- допускает существенные ошибки и неточности в изложении теоретического материала;
- в целом усвоил основную литературу;
- обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;
- требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);
- испытывает существенные трудности при определении собственной оценочной позиции;
- наблюдается нарушение логики изложения материала.

Оценка **«неудовлетворительно»** выставляется, если студент обнаруживает незнание или непонимание большей или наиболее существенной части содержания учебного материала:

- не способен применять знание теории к решению задач профессионального характера;
- не умеет определить собственную оценочную позицию;
- допускает грубое нарушение логики изложения материала;
- допускает принципиальные ошибки в ответе на вопросы;
- не может исправить ошибки с помощью наводящих вопросов.

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем	ОК.01-09, ПК.1.1 – ПК.1.6. ПК.11.1, ПК.2.1
---	--

Итоговая аттестация по ПМ – экзамен по модулю проводится как процедура внешнего оценивания с участием представителей работодателя – носителей профессионального контекста. Экзамен по модулю представляет собой форму выполнения практических заданий на ПК

Процедура экзамена по модулю включает следующие задания:

- создание программного продукта или модуля;
- самооценка результата практической работы;
- ответы на вопросы экспертов.

#### **Требования к процедуре экзамена по модулю**

Количество заданий для экзаменуемого: 3

Время выполнения каждого задания и максимальное время на экзамен по модулю:

Задание № 1 90 мин.

Задание № 2 180 мин

Задание № 3 90 мин.

Всего на экзамен 360 мин.

#### **Критерии оценки**

Эксперт оценивает экзаменуемого по каждому показателю оценки результата в баллах:

2 – показатель проявляется полностью.

1 – показатель проявляется частично;

0 – показатель не проявляется.

Выставленные баллы заносятся по каждому показателю в ведомость оценки эксперта. По каждому экзаменуемому баллы суммируются по всем показателям. Если полученная сумма баллов равна или превышает 65%, что составляет 8 баллов из 12 максимально возможных, то эксперт выставляет оценку «ВПД освоен/оценка». Оценка определяется исходя из суммы набранных баллов. Если студент набрал:

7-8 баллов (65-75% от максимально возможного балла) - оценка 3 «удовлетворительно»;

9 -10 баллов (75-90% от максимально возможного балла) - оценка 4 «хорошо»;

11-12 балла (90-100% от максимально возможного балла) - оценка 5 «отлично».

Если средний балл меньше 7 баллов, то выставляется оценка «ВПД не освоен».

#### **Типовой вариант задания:**

#### **ЗАДАНИЕ ДЛЯ ЭКЗАМЕНУЮЩЕГОСЯ**

**Вариант № \_\_\_\_\_**

#### **Инструкция**

1. Внимательно прочитайте задание.
2. Включите компьютер.
3. Задание № 1. Выполните вход в среду программирования Visual Studio. Оформите интерфейс будущего программного продукта. Составьте алгоритм кодовой части программы. Разработайте кодовую часть программы.
4. Задание № 2. Выполните задание, выполните при необходимости отладку.
5. Задание 3. Разработайте мобильное приложение.

Количество заданий для экзаменуемого - 3

Время выполнения каждого задания и максимальное время на экзамен по модулю:

Задание №1 - 90 мин

Задание №2 - 180 мин

Задание №3 - 90 мин

Ответы на вопросы экспертов для уточнения отдельных моментов по выполнению студентами заданий - 10 мин

Всего на экзамен 370 мин.

Вы можете воспользоваться отчетами по практическим работам

### Типовые практические задания для экзамена по модулю

1. Создать приложение, возводящее целое положительной число  $A$  в целую положительную степень  $N$ .
2. Создать приложение, находящее сумму цифр положительного целого числа. Ограничить ввод числа шестью знаками.
3. По выданному программному модулю составить техническую документацию. Использовать необходимое инструментальное программное обеспечение.
4. На языке блок - схем составить алгоритм решения задачи: возвести целое положительной число  $A$  в целую положительную степень  $N$ . Использовать необходимое инструментальное программное обеспечение.
5. Выполните тестирование и отладку готового программного модуля, используя инструментальные средства отладки языка ООП. Определите и исправьте имеющиеся ошибки. Зафиксируйте имеющиеся ошибки.
6. Создать приложение, находящее максимальный элемент двумерного целочисленного массива, введенного в компонент отображения информации в табличном виде и вывести его значение в текстовый компонент.
7. Создать приложение, определяющее является ли введенное в текстовый компонент число палиндромом.
8. На языке блок - схем составить алгоритм решения задачи: определить максимальный элемент двумерного целочисленного массива, введенного в компонент отображения информации в табличном виде и вывести его значение в текстовый компонент.
9. Выполните тестирование и отладку готового программного модуля, используя инструментальные средства отладки языка ООП. Определите и исправьте имеющиеся ошибки. Зафиксируйте имеющиеся ошибки.
10. По выданному алгоритму составить спецификацию программного продукта. Использовать необходимое инструментальное программное обеспечение.
11. Создать приложение, выводящее на графический компонент график функции  $Y=\cos(X)$ . Начальное и конечное значение аргумента вводятся в текстовый компонент.
12. Создать приложение, работающее с текстовыми файлами формата \*.txt (элементарный текстовый редактор).
13. По выданному программному модулю составить техническую документацию. Использовать необходимое инструментальное программное обеспечение.
14. На языке блок - схем составить алгоритм решения задачи: Найти минимальный элемент одномерного целочисленного массива, заполненного случайным образом и вывести его в текстовый компонент. Использовать необходимое инструментальное программное обеспечение.
15. Выполните тестирование и отладку готового программного модуля, используя инструментальные средства отладки языка ООП. Определите и исправьте имеющиеся ошибки. Зафиксируйте имеющиеся ошибки.
16. Создать приложение, рисующее на компоненте отображения графики «карандашом».
17. Создать приложение «элементарный арифметический калькулятор». Количество и типы компонентов определите самостоятельно.
18. На языке блок - схем составить алгоритм решения задачи: Определить количество слов, введенных в текстовый компонент. Использовать необходимое инструментальное программное обеспечение.

19. Выполните тестирование и отладку готового программного модуля, используя инструментальные средства отладки языка ООП. Определите и исправьте имеющиеся ошибки. Зафиксируйте имеющиеся ошибки.
20. По выданному программному модулю составить техническую документацию. Использовать необходимое инструментальное программное обеспечение.
21. Создать приложение, предназначенное для просмотра графических файлов в формате \*.bmp.
22. Создать приложение, предназначенное для проигрывания звуковых файлов.
23. На языке блок - схем составить алгоритм решения задачи: Найти максимальный элемент двумерного целочисленного массива, заполненного случайным образом и вывести его в текстовый компонент.
24. Выполните тестирование и отладку готового программного модуля, используя инструментальные средства отладки языка ООП. Определите и исправьте имеющиеся ошибки. Зафиксируйте имеющиеся ошибки.
25. По выданному программному модулю составить техническую документацию. Использовать необходимое инструментальное программное обеспечение.
26. По выданному алгоритму составить программный модуль.
27. По выданному алгоритму составить программный модуль