

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ ЛЕНИНГРАДСКОЙ ОБЛАСТИ

«ЛЕНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ А.С. ПУШКИНА»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для промежуточной аттестации по профессиональному модулю

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

---

Специальность 09.02.07 Информационные системы и программирование

Квалификация выпускника Программист

Форма обучения очная

Санкт-Петербург  
2025

## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

### **Разделы фонда оценочных средств**

1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП СПО.
2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения ОПОП СПО.
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Фонд оценочных средств составлен в соответствии с требованиями ФГОС по специальности.

Фонд оценочных средств разработали: Петров Юрий Николаевич

## **1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОПОП СПО**

Целью освоения профессионального модуля является достижение следующих результатов обучения: ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6.

Этап профессионального модуля в формировании компетенцией соответствует 4 – 8 семестрам.

Этап формирования компетенций в процессе освоения образовательной программы определяется на основе общей характеристики и соответствует порядку изучения дисциплин/профессиональных модулей/практик в учебном плане.

Основными этапами формирования указанных компетенций при изучении обучающимися профессионального модуля является последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий.

## 2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показателями оценивания компетенции(-й) являются следующие результаты обучения:

Код компетенции	Планируемые результаты обучения
ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	<p><u>Иметь практический опыт:</u></p> <ul style="list-style-type: none"> <li>– разработки кода программного продукта на основе готовой спецификации на уровне модуля;</li> <li>– разработке кода программного продукта на основе готовой спецификации на уровне модуля;</li> <li>– проведении тестирования программного модуля по определенному сценарию;</li> <li>– использовании инструментальных средств на этапе отладки программного продукта;</li> <li>– разработке мобильных приложений.</li> </ul> <p><u>Уметь:</u></p> <ul style="list-style-type: none"> <li>– осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</li> <li>– создавать программу по разработанному алгоритму как отдельный модуль;</li> <li>– выполнять отладку и тестирование программы на уровне модуля;</li> <li>– осуществлять разработку кода программного модуля на современных языках программирования;</li> <li>– уметь выполнять оптимизацию и рефакторинг программного кода;</li> <li>– оформлять документацию на программные средства.</li> </ul> <p><u>Знать:</u></p> <ul style="list-style-type: none"> <li>– основные этапы разработки программного обеспечения;</li> <li>– основные принципы технологии структурного и объектно-ориентированного программирования;</li> <li>– способы оптимизации и приемы рефакторинга;</li> <li>– основные принципы отладки и тестирования программных продуктов.</li> </ul>

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся сформированность профессиональных компетенций.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
<b>Раздел 1. Разработка программных модулей</b>		
ПК 1.1 Формировать алгоритмы	Оценка «отлично» - техническое задание проанализировано, алгоритм разработан,	Экзамен/зачет в форме

<p>разработки программных модулей в соответствии с техническим заданием</p>	<p>соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры, указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма.</p> <p>Оценка «<b>хорошо</b>» - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры; выполнена оценка сложности алгоритма.</p> <p>Оценка «<b>удовлетворительно</b>» - алгоритм разработан и соответствует заданию.</p>	<p>собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по лабораторным работам</p> <p>Тестирование Контрольная работа</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «<b>отлично</b>» - программный модуль разработан на указанном языке программирования по имеющемуся алгоритму в среде разработки (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 на указанном языке программирования по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «<b>удовлетворительно</b>» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по лабораторным работам</p> <p>Тестирование Контрольная работа</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

Раздел 2. Поддержка и тестирование программных модулей		
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	<p>Оценка <b>«отлично»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«хорошо»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по лабораторным работам</p> <p>Тестирование</p> <p>Контрольная работа</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.4 Выполнять тестирование программных модулей	<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 лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода	<p>Оценка <b>«отлично»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>Оценка <b>«хорошо»</b> - определены качественные</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного</p>

	<p>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>программного кода, его анализу, оптимизации методами рефакторинга. Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<b>Раздел 3. Разработка программных модулей</b>		
<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<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>» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.6 Разрабатывать модули программного</p>	<p>Оценка «<b>отлично</b>» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков</p>	<p>Экзамен/зачет в форме собеседования: практическое</p>

<p>обеспечения для мобильных платформ.</p>	<p>программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.</p> <p>Оценка «хорошо» - 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 «удовлетворительно» - 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.</p>	<p>задание по созданию модуля для заданного мобильного устройства на основе спецификации</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
--	--	---

**Раздел 4. Системное программирование**

<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
--	---	---

	модуль оформлена без существенных отклонений от стандартов.	
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	<p>Оценка <b>«отлично»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«хорошо»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся сформированность общих компетенций и обеспечивающих их умений.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<ul style="list-style-type: none"> <li>– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач;</li> <li>– адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач.</li> </ul>	Экспертное наблюдение за выполнением работ
ОП 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	– использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач.	

<p>ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<ul style="list-style-type: none"> <li>– демонстрация ответственности за принятые решения;</li> <li>– обоснованность самоанализа и коррекция результатов собственной работы.</li> </ul>	
<p>ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p>	<ul style="list-style-type: none"> <li>– взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик;</li> <li>– обоснованность анализа работы членов команды (подчиненных).</li> </ul>	
<p>ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p>	<ul style="list-style-type: none"> <li>– демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей.</li> </ul>	
<p>ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.</p>	<ul style="list-style-type: none"> <li>– соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик.</li> </ul>	
<p>ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.</p>	<ul style="list-style-type: none"> <li>– эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик;</li> <li>– демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности.</li> </ul>	
<p>ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной</p>	<ul style="list-style-type: none"> <li>– эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.</li> </ul>	

деятельности и поддержания необходимого уровня физической подготовленности.		
ОК 09. Использовать информационные технологии в профессиональной деятельности.	– эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту.	

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов профессионального модуля:

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
1.	<b>Раздел 1. Разработка программных модулей</b> МДК.01.01. Разработка программных модулей	ПК 1.1, ПК 1.2	<u>Тема 1.1.1</u> Знать понятие ЖЦ ПО. Знать этапы ЖЦ ПО. <u>Тема 1.1.2</u> Знать основные парадигмы программирования. Знать состав языка C++. Дать определение понятию массив. Уметь работать с указателями и адресами. Дать определение понятию функции, её перегрузки и шаблонизации. Уметь работать со структурами, файлами и строками. <u>Тема 1.1.3</u> Знать основные парадигмы в объектно-ориентированном программировании. Дать определение понятию класс. Дать определение понятию инкапсуляции. Уметь работать с объектами. Выполнять включение классов. Дать определение понятию наследование. Дать определение понятию	Лабораторная работа Тест	Отлично Хорошо Удовлетворительно Неудовлетворительно

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>полиморфизм. Уметь описывать и использовать полиморфные иерархии классов. Знать механизмы раннего и позднего связывания. Уметь работать с шаблонами классов. Знать механизм обработки исключительных ситуаций. Уметь использовать механизм обработки исключительных ситуаций</p> <p>Знать правила и методы преобразования типов данных. Знать иерархию потоковых классов. Уметь использовать файловые, строковые и стандартные потоки и выполнять форматирование данных в них. Знать типы итераторов и правила их описания. Знать типы стандартных контейнерных классов. Уметь использовать алгоритмы стандартной библиотеки C++.</p> <p><u>Тема 1.1.4</u> Знать назначение и виды паттернов. Знать основные шаблоны. Знать порождающие шаблоны. Знать поведенческие шаблоны.</p> <p><u>Тема 1.1.5</u> Знать основы разработки оконных приложений. Уметь работать с формами; диалоговыми сообщениями.</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>Знать работу с изображением.</p> <p>Знать работу с внешними приложениями.</p> <p><u>Тема 1.1.6</u></p> <p>Знать понятие рефакторинг</p> <p>Знать порядок составление методов и организации данных</p> <p>Уметь упрощать условные выражения</p> <p>Уметь решать задачи обобщения</p> <p><u>Тема 1.1.7</u></p> <p>Знать правила разработки интерфейсов пользователя.</p> <p><u>Тема 1.1.8</u></p> <p>Знать архитектуру технологии ADO.NET; получение данных; объект SqlCommand.</p> <p>Знать автономную часть архитектуры; объекты DataColumn, DataTable; классы DataRow, DataSet и DataAdapter</p> <p>Уметь работать с реляционными данными и объектом DataRelation</p> <p><u>Тема 1.1.9</u></p> <p>Знать основные понятия языка.</p> <p>Объяснить проблемы кроссбраузерности.</p> <p>Перечислить структурные теги.</p> <p>Уметь оформлять текстовые блоки.</p> <p>Демонстрировать создание таблиц.</p> <p>Использовать внедрение графических изображений и анимации. Применять использование гиперссылок.</p> <p>Демонстрировать создание и</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>назначение HTML-форм. Использовать плавающие фреймы. Владеть новыми возможностями редакции HTML5</p> <p>Знать основные понятия таблиц стилей. Объяснить проблемы кроссбраузерности. Использовать правила оформления текста. Демонстрировать понятие и использование псевдоклассов. Дать определение боксовой модели HTML-объектов. Применять на практике позиционирование элементов: абсолютное, относительное, фиксированное. Описать классификацию и идентификацию объектов. Владеть новыми возможностями редакции CSS3. Дать определение основным понятиям языка web-программирования на клиентской стороне Javascript. Описать типы данных и их преобразование. Применять на практике основные операторы языка: арифметические, сравнения, логические, ветвления, циклы. Объяснить проблемы кроссбраузерности. Рассказать о функциях, рекурсиях и области видимости переменных. Создавать пользовательские</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>объекты, их свойства и методы. Использовать методы и свойства глобального объекта window. Применять методы и свойства объектов ВОР. Применять методы и свойства объектов DOM. Создавать динамически построенные web-страницы. Демонстрировать управление атрибутами и стилями HTML-объектов. Применять встроенные библиотеки Javascript: объекты Date, Math, массивы, строковые объекты. Дать определения событиям Javascript: события мыши, клавиатуры, форм и загрузки объектов. Владеть возможностями технологии AJAX. Дать определение основным понятиям языка web-программирования на серверной стороне PHP. Описать типы данных и их преобразование. Применять на практике основные операторы языка: арифметические, сравнения, логические, ветвления, циклы. Объяснить проблемы кроссбраузерности. Рассказать о функциях, рекурсиях и области видимости переменных. Демонстрировать способы передачи данных с клиентской стороны и их</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>обработку. Владеть навыками работы с файловой системой сервера: записью и считыванием данных из файлов, использованием каталогов, загрузкой файлов с клиентской стороны. Применять на практике механизмы суперглобальных данных: session и cookies. Обеспечивать безопасность с помощью регулярных выражений. Владеть навыками работы с сервером баз данных. Применять встроенные библиотеки PHP: функции работы с массивами, строками, датой и временем.</p>		
2.	<p><b>Раздел 2. Поддержка и тестирование программных модулей</b> МДК.01.02 Поддержка и тестирование программных модулей</p>	<p>ПК 1.3, ПК 1.4, ПК 1.5</p>	<p>Дать определение верификации и аттестации программного обеспечения. Знать о методах контроля качества ПО. Рассказать о проведении инспектирования ПО на разных этапах жизненного цикла. Рассказать о роли тестирования в процессе верификации программного обеспечения. Перечислить методы тестирования ПО, рассказать о каждом из них. Перечислить методы отладки ПО. Знать основные понятия и методы метрологии. Сравнить отечественные и международные стандарты по метрологии и качеству ПО.</p>	<p>Тест Лабораторная работа</p>	<p>Отлично Хорошо Удовлетворительно Неудовлетворительно</p>

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>Продемонстрировать знания о понятии качества программного продукта. Перечислить основные показатели и характеристики качества программного продукта. Сформулировать критерии качества разработки и использования программного продукта.</p>		
3.	<p><b>Раздел 3. Разработка мобильных приложений</b> МДК.01.03 Разработка мобильных приложений</p>	ПК 1.2, ПК 1.6	<p>Уметь создавать одно оконные и многооконные шаблоны для мобильных приложений. Уметь отделять логику управления от графического интерфейса. Знать пересекающиеся операции управления графическими элементами и уметь реализовывать события графических элементов управления. Уметь создавать асинхронные потоки для обработки фоновых процессов приложения. Знать особенности работы стандартных контейнеров языка Java и уметь применять принципы ООП в сфере декомпозиции кода мобильного приложения. Уметь создавать службы и реализовывать систему сообщений между приложением и службой. Уметь работать с файловой системой мобильного устройства. Знать разрешения доступа и применять их в разрабатываемых программах. Уметь передавать и принимать широковещательные</p>	Тест Лабораторная работа	Отлично Хорошо Удовлетворительно Неудовлетворительно

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			сообщения от служб. Уметь работать с периферийным оборудованием по средствам намерений и широковещательных сообщений. Знать особенности работы мобильных устройств и требований к программам для подобных устройств. Уметь выполнять автоматическую обфускацию кода и настраивать систему сборки программ. Уметь добавлять в программы нативные библиотеки используя механизм JNI..		
4.	<b>Раздел 4. Системное программирование</b> МДК.01.04 Системное программирование	ПК 1.2, ПК 1.3	Понимать принципы компиляции и сборки программ. Уметь комбинировать команды компилятора для получения исполняемой программы или библиотеки. Уметь использовать адреса памяти для обращения к данным. Уметь распределять память внутри программы и запрашивать ее у ОС. Уметь дизассемблировать программы. Определять машинные команды по кодам операций или шестнадцатеричному коду. Знать шаблоны стандартных алгоритмических конструкций на машинном уровне. Находить в дизассемблированной программе логику и транслировать ее в алгоритмический язык. Строить иерархию стека вызовов. Уметь определять высоконагруженные узлы программы. Встраивать машинный код в текст программы на	Лабораторная работа Контрольная работа	Отлично Хорошо Удовлетворительно Неудовлетворительно

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>алгоритмическом языке. Определять условия перекрытия машинных инструкций. Выполнять декомпозицию программы. Создавать архитектуру загружаемой программы или группы программ. Уметь создавать загрузочную область и программу загрузчик. Уметь переводить микропроцессор в защищенный или реальный режим работы и управлять памятью в соответствующем режиме. Уметь переводить микропроцессор в защищенный или реальный режим работы и управлять памятью в соответствующем режиме. Программировать взаимодействие с видео памятью через линейный буфер кадров или общую память. Уметь обращаться к микропрограммам и создавать многозадачную среду. Определять необходимость использования векторных или скалярных вычислений при работе с данными. Обращаться к регистрах MMX, SSE. Реализовывать вычисления путем взаимодействия с векторными операциями над данными. Создавать отдельные библиотеки обработки данных векторным способом и внедрять такие библиотеки в языки высокого и сверхвысокого уровня.</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			<p>Уметь выполнять профилирование кода программы и находить высоконагруженные узлы. Транслировать код высоконагруженных узлов программы в совместимый машинный код. Добавлять расширенные операции микропроцессора для увеличения производительности высоконагруженных узлов программы. Знать принципы параллельной обработки данных с использованием многоядерных вычислительных систем. Уметь создавать многопоточные программы и реализовывать связь между потоками в рамках одной или нескольких программ. Понимать механизм межпроцессного взаимодействия на уровне ОС. Знать отличительные особенности сетевых протоколов передачи данных. Уметь настраивать маски для сетевых адресов. Знать стек протоколов TCP/IP. Уметь создавать сокеты и назначать порты для входящих и исходящих сетевых соединений. Уметь передавать данные между процессами на различных компьютерах используя ЛВС. Уметь передавать в контекст программы</p>		

№ п/п	Наименование раздела ПМ	Компетенции (части компетенций)	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
			контекст видеопамати и реализовывать передачу данных между программой и видео фрейм буфером. Знать особенности архитектуры современных видеоподсистем и видеопамати. Знать особенности адресации видеопамати.		
Итого:		ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	<b>Форма контроля</b>	<b>Оценочные средства промежуточной аттестации</b>	<b>Шкала оценивания</b>
			Письменный дифференцированный зачет	Устно-практический дифференцированный зачет – перечень вопросов, заданий	Зачтено (отлично) Зачтено (хорошо) Зачтено (удовлетворительно) Не зачтено (неудовлетворительно)
			Зачет	устный зачет	Отлично Хорошо Удовлетворительно Неудовлетворительно
			Экзамен	Устно-практический экзамен – перечень вопросов, заданий	Отлично Хорошо Удовлетворительно Неудовлетворительно
			Квалификационный экзамен по профессиональному модулю	Устный экзамен	Зачтено (основной вид деятельности освоен) Не зачтено (основной вид деятельности не освоен)

### 3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И(ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОПОП СПО

Контроль успеваемости по ПМ осуществляется с помощью следующих оценочных средств:

#### ЛАБОРАТОРНАЯ РАБОТА

№ п/п	Номер раздела (темы) ПМ	Наименование лабораторной работы	Трудоемкость, часов
1.	1 (1.1.2)	Линейные программы	2
2.	1 (1.1.2)	Разветвляющиеся программы. Циклы	2
3.	1 (1.1.2)	Одномерные и двумерные массивы	2
4.	1 (1.1.2)	Типы данных, определяемые пользователем	2
5.	1 (1.1.2)	Оценка сложности алгоритмов сортировки	4
6.	1 (1.1.2)	Оценка сложности алгоритмов поиска	4
7.	1 (1.1.3)	Классы и объекты: полиморфизм	2
8.	1 (1.1.3)	Шаблоны классов	2
9.	1 (1.1.3)	Обработка исключительных ситуаций	2
10.	1 (1.1.3)	События (events)	4
11.	1 (1.1.3)	Файлы конфигурации.	4
12.	1 (1.1.3)	Стандартная библиотека шаблонов: контейнеры	2
13.	1 (1.1.3)	Стандартная библиотека шаблонов: алгоритмы	2
14.	1 (1.1.4)	Использование основных шаблонов	4
15.	1 (1.1.4)	Использование порождающих шаблонов	4
16.	1 (1.1.4)	Использование структурных шаблонов	4
17.	1 (1.1.4)	Использование поведенческих шаблонов	4
18.	1 (1.1.5)	Форма и ее свойства	4
19.	1 (1.1.5)	Диалоговые окна и сообщения	4
20.	1 (1.1.5)	Обработка графической информации	4
21.	1 (1.1.5)	Работа с приложениями Microsoft Office	4
22.	1 (1.1.6)	Подъем поля или метода	4
23.	1 (1.1.6)	Спуск поля или метода	4
24.	1 (1.1.6)	Выделение интерфейсов	4
25.	1 (1.1.6)	Выделение подклассов и родительских классов	4
26.	1 (1.1.6)	Замена наследования делегированием	4
27.	1 (1.1.6)	Замена делегирования наследованием	4
28.	1 (1.1.8)	Разработка приложения базы данных	22
29.	1 (1.1.9)	Способы оформления текстовых блоков	2
30.	1 (1.1.9)	Использование гиперссылок	2
31.	1 (1.1.9)	Создание и назначение HTML-форм	2
32.	1 (1.1.9)	Правила оформления текста	2
33.	1 (1.1.9)	Понятие и использование псевдоклассов	2
34.	1 (1.1.9)	Позиционирование элементов: абсолютное, относительное, фиксированное	2
35.	1 (1.1.9)	Новые возможности редакции CSS3 – плавные переходы значений, анимация keyframes	2

36.	1 (1.1.9)	Основные операторы языка: арифметические, сравнения, логические, ветвления, циклы	4
37.	1 (1.1.9)	Пользовательские объекты, их свойства и методы. Методы и свойства глобального объекта window	4
38.	1 (1.1.9)	Ветка объектов ВОМ. Ветка объектов DOM	3
39.	1 (1.1.9)	Управление атрибутами тегов	2
40.	1 (1.1.9)	Управление стилями, примененными к объектам	2
41.	1 (1.1.9)	Встроенные библиотеки Javascript: объекты Date, Math, массивы, строковые объекты	2
42.	1 (1.1.9)	События Javascript: события мыши, клавиатуры, форм и загрузки объектов	2
43.	1 (1.1.9)	Использование возможностей технологии AJAX	2
44.	1 (1.1.9)	Основные операторы языка: арифметические, сравнения, логические, ветвления, циклы	2
45.	1 (1.1.9)	Функции, рекурсии и области видимости переменных	2
46.	1 (1.1.9)	Способы передачи данных с клиентской стороны и их обработка	2
47.	1 (1.1.9)	Суперглобальные данные: механизмы session и cookies	2
48.	1 (1.1.9)	Обеспечение безопасности с помощью регулярных выражений	2
49.	1 (1.1.9)	Работа с сервером баз данных	2
50.	1 (1.1.9)	Встроенные библиотеки PHP: функции работы с массивами, строками, датой и временем	2
51.	2 (1.2.1)	Оформление документации на программные средства с использованием инструментальных средств.	8
52.	2 (1.2.1)	Документирование результатов тестирования.	4
53.	2 (1.2.1)	Автоматизация разработки технической документации.	4
54.	2 (1.2.2)	Тестирование «белым ящиком»	6
55.	2 (1.2.2)	Тестирование «черным ящиком»	6
56.	2 (1.2.2)	Модульное тестирование	6
57.	2 (1.2.2)	Интеграционное тестирование	6
58.	3 (1.3.1)	Установка инструментария и настройка среды для разработки мобильных приложений	6
59.	3 (1.3.1)	Установка среды разработки мобильных приложений с применением виртуальной машины	6
60.	3 (1.3.2)	Реализация декомпозиции в программе на Java	4
61.	3 (1.3.2)	Изучение распределения динамической памяти в программах на Java	6
62.	3 (1.3.2)	Динамические структуры данных в программе на Java	4
63.	3 (1.3.2)	Операции файлового ввода/вывода	4
64.	3 (1.3.2)	Многопоточное программирование	6
65.	3 (1.3.2)	Изучение жизненного цикла приложения для ОС Android	6
66.	3 (1.3.2)	Создание одноконного приложения и реализация событий элементов управления	6
67.	3 (1.3.2)	Менеджмент компоновки элементов управления окна	4
68.	3 (1.3.2)	Реализация асинхронных задач и многопоточных приложений	4
69.	3 (1.3.2)	Реализация широковещательных сообщений и	4

		обработка событий аппаратного обеспечения	
70.	3 (1.3.2)	Реализация компьютерной графики и система событий	6
71.	3 (1.3.2)	Реализация нативного кода при низкоуровневой декомпозиции приложения на Java	4
72.	4 (1.4.1)	Написание, дизассемблирование и восстановление исходного кода программы	6
73.	4 (1.4.1)	Реализация заданного алгоритма в рамках динамической и статической библиотеки	8
74.	4 (1.4.1)	Разработка программы загрузчика и программы оболочки	16
75.	4 (1.4.1)	Разработка программы загрузчика с возможностью перехода в защищенный режим и программы оболочки	10
76.	4 (1.4.1)	Реализация программы обработки больших объемов данных с использованием технологий MMX/SSE	6
77.	4 (1.4.1)	Обработка данных с использованием векторных вычислений	4
78.	4 (1.4.2)	Разработка многопоточных программ и обработка больших объемов данных	6
79.	4 (1.4.2)	Обмен данными между программами в сетевом режиме и в локальном режиме	8
80.	4 (1.4.2)	Программирование графического ускорителя с использованием видео фрейм буфера	6
<b>Итого:</b>			<b>312</b>

## МДК 01.01 Разработка программных модулей

### Тема 1.1.2 Структурное программирование

Пример задания лабораторной работы

#### ЛАБОРАТОРНАЯ РАБОТА № 4

#### ТИПЫ ДАННЫХ, ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ

**Цель работы:** изучение ссылочных типов данных, работа со связными списками.

**Практическое задание:**

Сформировать двухсвязный список и выполнить задание по своему варианту.

**Варианты заданий:**

1. Структура содержит фамилию и 4 оценки. Удалить из списка неуспевающих.
2. Структура содержит фамилию и 4 оценки. Удалить из списка имеющих 2, 3.
3. Структура содержит название книги, автора, год издания. Удалить издания с годом меньше заданного.
4. Структура содержит название книги, автора, год издания. Удалить книги заданного автора.
5. Структура содержит название, цену, количество товара. Удалить из списка заданный товар.
6. Структура содержит название, цену, количество товара. Удалить из списка партии товара, превышающие заданную стоимость.

7. Структура содержит фамилию, год рождения. Добавлять новые записи так, чтобы список был упорядочен по алфавиту.
8. Структура содержит фамилию, год рождения. Добавлять новые записи так, чтобы список был упорядочен по возрасту.
9. Структура содержит название издания, газета или журнал, цена экземпляра. Добавлять новые записи так, чтобы сначала располагались журналы, затем газеты.
10. Структура содержит название издания, газета или журнал, цена экземпляра. Добавлять новые издания так, чтобы названия были упорядочены по алфавиту.
11. Структура содержит фамилию спортсмена, вид спорта, количество очков. Добавлять новые записи так, чтобы информация по каждому виду спорта располагалась последовательно.
12. Структура содержит фамилию спортсмена, вид спорта, количество очков. Добавлять новые записи так, чтобы они были упорядочены по убыванию очков.

### Контрольные вопросы:

1. Понятие статической и динамической памяти.
2. Как создаётся и просматривается односвязный список?
3. Как создаётся и просматривается двухсвязный список?
4. Как удалить структуру из списка?
5. Как добавить структуру в список?

### Содержание отчета:

1. Титульный лист (по образцу).
2. Цель работы.
3. Практическое задание.
4. Индивидуальное задание.
5. Блок-схема алгоритма работы программы/диаграмма классов.
6. Листинг.
7. Протокол выполнения.

### Отчет по лабораторной работе

По результатам лабораторной работы обучающиеся составляют отчет. Отчет по лабораторной работе представляется в электронном виде в формате, предусмотренном шаблоном отчета по лабораторной работе в установленное преподавателем время.

### Отчет по лабораторной работе № \_\_\_\_\_

« \_\_\_\_\_ »

1. Цель и задачи лабораторной работы: \_\_\_\_\_
2. Индивидуальное задание: \_\_\_\_\_
3. Технология выполнения лабораторной работы: \_\_\_\_\_
4. Результаты выполнения: \_\_\_\_\_

Выводы: \_\_\_\_\_

Защита отчета проходит в форме доклада обучающегося по выполненной работе, представления программы и ответов на вопросы преподавателя.

Основаниями для снижения количества баллов в диапазоне от **max** до **min** являются:

- небрежное выполнение,

- низкое качество графического материала (небрежное представление графиков и диаграмм),
- выполнение практического задания не в полном объеме;
- некорректные результаты программы.

Отчет не может быть принят и подлежит доработке в случае:

- несоответствие результатов работы индивидуальному практическому заданию,
- отсутствия необходимых разделов,
- отсутствия необходимого графического материала,
- некорректных результатов моделируемых объектов (менее чем на 60%).

Шкала оценивания и критерии оценки:

Оценка	Минимальное количество баллов	Максимальное количество баллов	Критерий
«5» (отлично)	9	10	выполнены все задания лабораторной работы, обучающийся четко и без ошибок ответил на все контрольные вопросы, представлен отчет в соответствии с требованиями.
«4» (хорошо)	7	8	выполнены все задания лабораторной работы; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«3» (удовлетворительно)	5	6	выполнены все задания лабораторной работы с замечаниями; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«2» (неудовлетворительно)	0	4	обучающийся не выполнил или выполнил неправильно задания лабораторной работы; обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, не представлен отчет по выполнению лабораторной работы или предоставлен, но не в соответствии с требованиями.

## Тема 1.1.2 Веб-программирование

Пример задания лабораторной работы

### ЛАБОРАТОРНАЯ РАБОТА №5

#### ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ С ПОМОЩЬЮ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

**Цель работы:** овладеть практическими навыками проверки строковых данных на основе шаблонов регулярных выражений и механизмом суперглобальных данных.

#### Индивидуальное задание:

Организовать на серверной стороне проверку входящих пользовательских данных присланной анкеты с помощью регулярных выражений.

Клиентская сторона отправляет следующие данные:

1. ФИО – только латинские символы и пробелы..
2. Логин - только латинские символы, цифры, знак нижнего подчеркивания, дефис.
3. Дата рождения – в формате поля INPUT.
4. Паспортные данные – серия пробел номер.

При некорректно введенных данных отображается та же самая страница анкеты, и напротив неверно заполненных пунктов отображается сообщение об этом.

При корректно введенных данных – сообщение об этом (без отображения анкеты).

#### Этапы выполнения работы:

1. Проанализировать предметную область согласно варианту задания.
2. Реализовать передачу пользовательских данных с клиентской стороны к веб-серверу
3. Создать программный модуль на серверной стороне для проверки пользовательских данных регулярными выражениями и размещения их в суперглобальных массивах.
4. Подготовить отчет о проделанной работе в распечатанном виде.

#### Отчет по лабораторной работе

По результатам выполнения всех лабораторных работ обучающиеся составляют отчет. Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе.

#### Отчет по лабораторной работе № \_\_\_\_\_

« \_\_\_\_\_ »  
(название лабораторной работы)

5. Цель и задачи работы: \_\_\_\_\_
6. Индивидуальное задание: \_\_\_\_\_
7. Этапы выполнения работы: \_\_\_\_\_
8. Программный код модуля клиентской стороны разработки: \_\_\_\_\_
9. Программный код модуля серверной стороны разработки: \_\_\_\_\_
10. Выводы: \_\_\_\_\_

Защита лабораторной работы проходит в форме доклада обучающегося по выполненной работе, представления программы и ответов на вопросы преподавателя.

Отчет не может быть принят и подлежит доработке в случае:

- ошибки в программном коде в отчете;
- отсутствия необходимых разделов.

#### Шкала оценивания и критерии оценки:

Оценка	Минимальное количество баллов	Максимальное количество баллов	Критерий
«5» (отлично)	9	10	выполнены все задания лабораторной работы, обучающийся четко и без ошибок ответил на все контрольные вопросы, представлен отчет в соответствии с требованиями.
«4» (хорошо)	7	8	выполнены все задания лабораторной работы; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«3» (удовлетворительно)	5	6	выполнены все задания лабораторной работы с замечаниями; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«2» (неудовлетворительно)	0	4	обучающийся не выполнил или выполнил неправильно задания лабораторной работы;

			обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, не представлен отчет по выполнению лабораторной работы или предоставлен, но не в соответствие с требованиями.
--	--	--	--

## МДК.01.02. Поддержка и тестирование программных модулей

Пример задания лабораторной работы

### ЛАБОРАТОРНАЯ РАБОТА №6

#### МОДУЛЬНОЕ ТЕСТИРОВАНИЕ

**Цель лабораторной работы:** ознакомиться с unit-тестированием и составить тестовые наборы по типу чёрного ящика.

**Индивидуальное задание:** Создать приложение в виде простого калькулятора со следующим функционалом:

1. Простейшие арифметические функций (сложение, вычитание, умножение, деление);
2. Управление памятью (запоминание числа, очистка памяти, прибавление в память, вычитание из памяти).

Затем создать и обосновать выбор тестовых наборов для этого приложения.

#### **Технология выполнения лабораторной работы:**

1. Создать приложение Калькулятор с необходимыми требованиями.
2. Выбрать класс, для которого будет создаваться unit-тест.
3. Выписать все функции этого класса с передаваемыми и возвращаемыми типами данных.
4. Если функция ничего не возвращает, следовательно она должна менять какое-либо поле класса, и оно считается возвращаемым значением.
5. Если функция ничего не принимает, следовательно она должна работать с каким-то полем класса, и оно считается принимаемым значением.
6. Для каждого типа данных выбрать значения для тестирования согласно методике чёрного ящика.
7. Возможно уменьшение тестовых наборов согласно методике серого ящика.

#### **Выводы.**

#### **Отчет по лабораторной работе**

По результатам выполнения всех лабораторных работ обучающиеся составляют отчет. Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе.

#### **Отчет по лабораторной работе №\_\_\_\_\_**

«\_\_\_\_\_»

1. Цель и задачи лабораторной работы: \_\_\_\_\_
2. Технология выполнения лабораторной работы: \_\_\_\_\_
3. Исполняемый файл приложения 9 (листинг): \_\_\_\_\_
4. Выводы: \_\_\_\_\_

Защита отчета проходит в форме доклада обучающегося по выполненной работе, представления программы и ответов на вопросы преподавателя.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- некорректных результатов.

Шкала оценивания и критерии оценки:

Оценка	Минимальное количество баллов	Максимальное количество баллов	Критерий
«5» (отлично)	9	10	выполнены все задания лабораторной работы, обучающийся четко и без ошибок ответил на все контрольные вопросы, представлен отчет в соответствии с требованиями.
«4» (хорошо)	7	8	выполнены все задания лабораторной работы; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«3» (удовлетворительно)	5	6	выполнены все задания лабораторной работы с замечаниями; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«2» (неудовлетворительно)	0	4	обучающийся не выполнил или выполнил неправильно задания лабораторной работы; обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, не представлен отчет по выполнению лабораторной работы или предоставлен, но не в соответствии с требованиями.

### МДК.01.03 Разработка мобильных приложений

Пример задания лабораторной работы

#### ЛАБОРАТОРНАЯ РАБОТА № 7

#### СОЗДАНИЕ ОДНОКОННОГО ПРИЛОЖЕНИЯ И РЕАЛИЗАЦИЯ СОБЫТИЙ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ

**Цель лабораторной работы:** Создание одноконного приложения и реализация событий элементов управления

**Индивидуальное задание:** Реализовать приложение с кнопками добавления данных в элемент список и сортировкой этих данных по убыванию и возрастанию. Предусмотреть отправку выбранного элемента списка через намерение.

**Технология выполнения лабораторной работы:** Изучить полученное задание, реализовать макет одноконного приложения, создать элементы управления с использованием встроенного графического редактора или с использованием текстового XML редактора, создать список событий для элементов управления.

**Выводы:**

#### Отчет по лабораторной работе

По результатам выполнения всех лабораторных работ обучающиеся составляют отчет. Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе.

## Отчет по лабораторной работе № \_\_\_\_\_

« \_\_\_\_\_ »

1. Цель лабораторной работы: \_\_\_\_\_
2. Индивидуальное задание: \_\_\_\_\_
3. Листинг программы: \_\_\_\_\_
4. Выводы: \_\_\_\_\_

Защита отчета проходит в форме доклада обучающегося по выполненной работе, представления программы и ответов на вопросы преподавателя.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- некорректных результатов.

Шкала оценивания и критерии оценки:

Оценка	Минимальное количество баллов	Максимальное количество баллов	Критерий
«5» (отлично)	9	10	выполнены все задания лабораторной работы, обучающийся четко и без ошибок ответил на все контрольные вопросы, представлен отчет в соответствии с требованиями.
«4» (хорошо)	7	8	выполнены все задания лабораторной работы; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«3» (удовлетворительно)	5	6	выполнены все задания лабораторной работы с замечаниями; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«2» (неудовлетворительно)	0	4	обучающийся не выполнил или выполнил неправильно задания лабораторной работы; обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, не представлен отчет по выполнению лабораторной работы или предоставлен, но не в соответствии с требованиями.

### МДК.01.04 Системное программирование»

Пример задания лабораторной работы

#### ЛАБОРАТОРНАЯ РАБОТА №8

#### РАЗРАБОТКА ПРОГРАММЫ ЗАГРУЗЧИКА С ВОЗМОЖНОСТЬЮ ПЕРЕХОДА В ЗАЩИЩЕННЫЙ РЕЖИМ И ПРОГРАММЫ ОБОЛОЧКИ

**Цель работы:** получение навыков в области разработки простейших операционных систем, программ загрузки и взаимодействия с оборудованием в защищенном режиме.

**Практическое задание:** Написать программу по индивидуальному заданию и, реализовать внешние и внутренние процедуры соответствующие заданию.

**Порядок выполнения работы:** В качестве названий процедур использовать названия, наиболее хорошо описывающие профиль разрабатываемой процедуры. Реализовать программу

загрузчик на ассемблере и загружаемый модуль на языке Си или ином. При необходимости использовать ассемблерные вставки или линковку с уже готовыми модулями на ассемблере.

**Варианты индивидуального задания:**

1. Разработать программу загрузчик с возможностью загрузки программы в память и передачи управления последней. Программа должна удовлетворять в обязательном порядке следующим функциям: (1) Наличие анимированной заставки, (2) Возможность завершения работы (3) Наличие способа взаимодействия с пользователем.

2. Реализовать интерфейс пользователя для выполнения арифметических операций над числами (калькулятор) ввод должен происходить при нажатии цифровых клавиш на клавиатуре, отображение должно происходить на экране в специальном поле.

**Результаты:**

**Выводы:**

**Отчет по лабораторной работе**

По результатам лабораторной работы обучающиеся составляют отчет. Отчет по лабораторной работе представляется в электронном виде в формате, предусмотренном шаблоном отчета по лабораторной работе в установленном преподавателем время.

**Отчет по лабораторной работе № \_\_\_\_\_**

« \_\_\_\_\_ »

- 1. Цель работы: \_\_\_\_\_
- 2. Задание: \_\_\_\_\_
- 3. Исходный текст на Assembler: \_\_\_\_\_
- 4. Исходный текст на алгоритмическом языке: \_\_\_\_\_
- 5. Параметры сборки проекта: \_\_\_\_\_

Защита отчета проходит в форме доклада обучающегося по выполненной работе, представления программы и ответов на вопросы преподавателя.

Основаниями для снижения количества баллов в диапазоне от **max** до **min** являются:

- небрежное выполнение,
- низкое качество графического материала (неверный выбор масштаба чертежей, отсутствие указания единиц измерения на графиках),
- подмена (специально или по неосторожности) задания,
- наличие любого электронного устройства с дополнительной информацией или средствами решения способствующими правильным ответам на вопросы преподавателя по выполненной работе.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- отсутствия необходимого графического материала,
- некорректной обработки результатов измерений,
- подмены (специально или по неосторожности) задания.

**Шкала оценивания и критерии оценки:**

Оценка	Минимальное количество баллов	Максимальное количество баллов	Критерий
«5» (отлично)	9	10	выполнены все задания лабораторной работы,

			обучающийся четко и без ошибок ответил на все контрольные вопросы, представлен отчет в соответствии с требованиями.
«4» (хорошо)	7	8	выполнены все задания лабораторной работы; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«3» (удовлетворительно)	5	6	выполнены все задания лабораторной работы с замечаниями; обучающийся ответил на все контрольные вопросы с замечаниями, представлен отчет в соответствии с требованиями.
«2» (неудовлетворительно)	0	4	обучающийся не выполнил или выполнил неправильно задания лабораторной работы; обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, не представлен отчет по выполнению лабораторной работы или предоставлен, но не в соответствии с требованиями.

## **ТЕСТ**

Тест являются формой рубежного контроля и содержат теоретические задания. Каждое задание теста имеет в зависимости от вида теста определенный вес (в промежуточных баллах). Промежуточные итоговые баллы за выполнение теста переводятся в баллы по шкале оценивания для теста.

При проведении тестирования тест выполняется индивидуально, в письменной форме на бланке, выданном преподавателем. Бланки содержат вопросы теста с вариантами ответов. Бланки должны удовлетворять следующим требованиям: в работе указывается ФИО студента, номер группы и выделенный ответ (ы). На выполнение отводится 2 академических часа.

### **Примеры тестовых заданий**

#### **МДК.01.02 Разработка программных модулей**

**Тема 1.1.3 Объектно-ориентированное программирование»** (каждое задание теста имеет вес 1 балл)

#### **Тест 1 (3 семестр, рубежный)**

- 1) Точкой входа в программу на языке C++ (из перечисленных) является функция
  - a) begin( )
  - b) start( )
  - c) main( )
- 2) В стандартном языке C++ для хранения текстовых строк используется
  - a) массив переменных типа char;
  - b) специальный строковый тип данных string;
  - c) массив переменных типа string;
- 3) В результате выполнения программы  

```
int x, y;  
x = 0;  
y = 0;  
if (x)  
{  
    y = 1;  
}
```

переменная y получит значение
  - a) 0
  - b) 1
  - c) -1
- 4) В языке C++ существует специальный оператор прерывания циклов continue. Он служит для того, чтобы
  - a) досрочно прекратить выполнение содержащего его ближайшего цикла while, do ... while или for или условного оператора switch.
  - b) досрочно прекратить выполнение текущей итерации содержащего его ближайшего цикла while, do ... while или for.
  - c) досрочно завершить программу.
- 5) Прототипом функции называется
  - a) словесное описание действий функции
  - b) перечень переменных, объявленных в функции
  - c) заголовок функции без её тела, оканчивающийся символом ‘;’
- б) Перегруженные функции применяются тогда, когда

- a) необходимо смоделировать вложенность функций
  - b) функция должна выполнять различные действия в зависимости от типа и количества её параметров
  - c) нужно объявить глобальную функцию
- 7) Для того, чтобы иметь возможность вызывать функции, описанные в одном модуле программы на языке C++, из другого модуля нужно
- a) создать заголовочный файл, поместить в него прототипы функций и подключить этот заголовочный файл к вызывающему модулю
  - b) создать заголовочный файл, поместить в него локальные переменные функций и подключить этот заголовочный файл к вызывающему модулю
  - c) скопировать описание функций из одного модуля в другой
- 8) Основное отличие динамического размещения данных от статического состоит в том, что
- a) статические данные размещаются на диске, динамические – в памяти машины
  - b) статические данные размещаются в момент старта программы, динамические – при её выполнении
  - c) динамические данные размещаются в момент старта программы, статические - при её выполнении
- 9) Элементы объединений располагаются в памяти компьютера
- a) последовательно, один за другим
  - b) начинаясь с одного адреса памяти, перекрывая друг друга
  - c) только на жёстком диске
- 10) К элементам структур можно обратиться
- a) только из функций, описанных в том же модуле, что и структура
  - b) только из функции main( )
  - c) из любой функции программы

## Тест 2 (4 семестр, рубежный)

- 1) Членами класса могут быть:
- a) как переменные, так и функции, могут быть объявлены как private и как public
  - b) только переменные, объявленные как private
  - c) только функции, объявленные как private
  - d) только переменные и функции, объявленные как private
  - e) только переменные и функции, объявленные как public
- 2) Что называется конструктором?
- a) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
  - b) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
  - c) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
  - d) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса
- 3) Выберите правильные утверждения
- a) у конструктора могут быть параметры
  - b) конструктор наследуется, но должен быть перегружен
  - c) конструктор должен явно вызываться всегда перед объявлением объекта
  - d) конструктор вызывается автоматически при объявлении объекта
  - e) объявление каждого класса должно содержать свой конструктор
  - f) если конструктор не создан, компилятор создаст его автоматически
- 4) Что называется наследованием?

- a) это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
  - b) это механизм переопределения методов базового класса
  - c) это механизм, посредством которого производный класс получает все поля базового класса
  - d) это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить
- 5) Выберите правильное объявление производного класса
- a) `class MoreDetails:: Details;`
  - b) `class MoreDetails: public class Details;`
  - c) `class MoreDetails: public Details;`
  - d) `class MoreDetails: class(Details);`
- 6) Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового
- a) ключ доступа - `public`; в базовом классе: `private`; права доступа в производном классе - `protected`
  - b) ключ доступа - любой; в базовом классе: `private`; права доступа в производном классе - нет прав
  - c) ключ доступа - `protected` или `public` ; в базовом классе: `protected`; права доступа в производном классе - `protected`
  - d) ключ доступа - `private`; в базовом классе: `public`; права доступа в производном классе - `public`
  - e) ключ доступа – любой; в базовом классе: `public`; права доступа в производном классе – такие же, как ключ доступа
- 7) Дружественная функция - это
- a) функция другого класса, среди аргументов которой есть элементы данного класса
  - b) функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;
  - c) функция, являющаяся членом класса и объявленная с атрибутом `friend`;
  - d) функция, которая в другом классе объявлена как дружественная данному
- 8) Переопределение операций имеет вид:
- a) имя\_класса, ключевое слово `operation`, символ операции
  - b) имя\_класса, ключевое слово `operator`, символ операции, в круглых скобках могут быть указаны аргументы
  - c) имя\_класса, ключевое слово `operator`, список аргументов
  - d) имя\_класса, два двоеточия, ключевое слово `operator`, символ операции
- 9) Виртуальными называются функции:
- a) функции базового класса, которые могут быть переопределены в производном классе
  - b) функции базового класса, которые не используются в производном классе;
  - c) функции базового класса, которые не могут быть переопределены в базовом классе;
  - d) функции производного класса, переопределенные относительно базового класса
- 10) Полиморфизм в объектно-ориентированном программировании реализуется:
- a) через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны
  - b) через механизмы перегрузки (функций и операций) и шаблоны;
  - c) через виртуальные функции и шаблоны;
  - d) через механизмы перегрузки (функций и операций) и виртуальные функции

Шкала оценивания и критерии оценки:

Критерий	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 6 баллов и более		6	10
<b>Итого:</b>		<b>6</b>	<b>10</b>

### Соответствие баллов шкале оценивания:

Количество баллов	Оценка обучающегося
10	отлично
8-9	хорошо
6-7	удовлетворительно
менее 6	неудовлетворительно

### Тема 1.1.2 Веб-программирование

#### Перечень тестовых заданий по теме 1.2:

Тест 1 «Язык разметки web-страниц HTML».

Тест 2 «Таблицы стилей оформления объектов CSS».

Тест 3 «Язык web-программирования на клиентской стороне Javascript».

Тест 4 «Язык web-программирования на серверной стороне PHP».

**Пример теста 3 «Язык web-программирования на клиентской стороне Javascript»** (каждый правильный ответ имеет вес 1 балл)

1. Отметить типы данных, которыми оперирует Javascript (отметить правильные варианты):
  - строковые;
  - потоковые;
  - видеоданные;
  - вещественные;
  - логические;
  - ряды баз данных.
2. Отличие оператора инкремента от преинкремента в том, что:
  - Преинкремент вначале возвращает значение, а затем производит увеличение на единицу;
  - Инкремент вначале возвращает значение, а затем производит увеличение на единицу;
  - Отличий нет.
3. Назначение языка программирования Javascript:
  - организация доступа к файловой системе клиентской стороны;
  - организация работы с удаленными базами данных;
  - создание динамического интерфейса пользователя на клиентской стороне
  - создание модулей обработки пользовательских данных на серверной стороне.
4. Javascript является языком программирования:
  - со строгой типизацией данных;
  - со слабой типизацией данных;
  - типизация зависит от настроек web-клиента.
5. Отметить контексты исполнения, существующие в Javascript (отметить правильные варианты):
  - Глобальный контекст
  - Локальный контекст
  - Суперглобальный контекст
6. Переменные, продекларированные в глобальном контексте:
  - по умолчанию видны в пределах локального контекста функций;
  - видны в пределах локального контекста функций только при передаче их в функцию при вызове с помощью оператора global;

- не видны в локальном контексте функций.
7. Разница между способами декларации функций Function Declaration и Function Expression:
- только в синтаксисе;
  - в том, что декларация при первом способе происходит до исполнения основного контекста, а при втором — во время исполнения;
  - в том, что декларация при первом способе происходит до исполнения основного контекста, а при втором — во время исполнения. Однако в некоторых web-клиентах это правило не соблюдается.
8. Ветка объектов ВОМ позволяет (отметить правильные варианты):
- получать сведения о web-клиенте;
  - работать с файловой системой клиента;
  - управлять адресной строкой клиента;
  - отправлять запросы к удаленным базам данных;
9. Технология AJAX позволяет:
- упростить построение динамического интерфейса за счет дополнительных библиотек;
  - отправлять HTTP-запросы и обрабатывать HTTP-ответы с помощью сценариев Javascript;
  - использовать графический интерфейс для создания сценариев Javascript.
10. Следующие утверждения верны:
- массивы в Javascript индексированные. Роль ассоциативных массивов выполняют объекты;
  - индексация массивов непрерывна. При удалении элементов индексы пересчитываются;
  - свойства объектов как и элементы массивов можно перебрать циклом for...in
  - размер массива нельзя изменять через изменение свойства length
11. Примитивное значение времени в Javascript это:
- запись в упрощенном формате день-месяц-год;
  - количество миллисекунд, прошедших с полуночи 1 января 1970 года;
  - в Javascript нет такого понятия.

### Шкала оценивания и критерии оценки:

Критерий	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 6 баллов и более		6	11
<b>Итого:</b>		<b>6</b>	<b>11</b>

### Соответствие баллов шкале оценивания:

Количество баллов	Оценка обучающегося
10-11	отлично
8-9	хорошо
6-7	удовлетворительно
менее 6	неудовлетворительно

### МДК.01.02 Поддержка и тестирование программных модулей

Тест по теме 1.2.2 Отладка и тестирование программного обеспечения («Тестовые наборы ПО») (каждый правильный ответ имеет вес 1 балл)

**Вопрос 1.** Стресс-тестирование предназначено для:

Выберите один или несколько ответов:

1. Проверка системы на корректность нагрузки в длительное время работы
2. Проверка системы на корректность работы под 100% нагрузкой
3. Проверка системы на корректность работы под нагрузкой в 2-4 раза превышающей запланированную
4. Тестирования производительности
5. Проверка системы на оптимальную нагрузку системы

**Вопрос 2.** Возможно ли проведение автоматизацию тестирования локализации?

Выберите один или несколько ответов:

1. Возможно
2. Частично
3. Нет
4. Да
5. Не знаю

**Вопрос 3.** Возможно ли автоматизировать инспектирование?

Выберите один или несколько ответов:

1. Не знаю
2. Частично да
3. Нет
4. Да

**Вопрос 4.** Укажите принципы тестирования белого ящика (White-box).

Выберите один или несколько ответов:

1. Известен только примерный алгоритм работы программы
2. Входные данные подбираются методом декартово произведения.
3. При выполнении теста должен быть достигнут каждый оператор минимум один раз
4. Отработавшие некорректные входные данные удаляются из наборов тестирования
5. Исходный код программы изначально недоступен

**Вопрос 5.** Для альфа-версии программного обеспечения характерны следующие принципы:

Выберите один или несколько ответов:

1. Заморозка всего функционала
2. Возможность тестирования сторонними тестировщиками
3. Большая нестабильность работы
4. Заморозка основного функционала
5. Возможность изменения ядра программы

**Вопрос 6.** Отметьте признаки хороших требований

Выберите один или несколько ответов:

1. Осуществимость
2. Проверяемость
3. Абстрактность
4. Атомарность
5. Четкость
6. Недвусмысленность
7. Понятность
8. Точность

**Вопрос 7.** Дайте корректное и полное определение производительности.

Выберите один или несколько ответов:

1. Возможность выполнения операций за отведённое время на ограниченных ресурсах
2. Возможность выполнения операций только за отведённое время
3. Возможность выполнения операций только на определённых ресурсах
4. Возможность работы со всеми ресурсами за нужное время

**Вопрос 8.** Если тестирование завершилось без ошибок, найденных в процессе тестирования, это значит:

Выберите один или несколько ответов:

1. Тестирование прошло неудачно
2. Программа работает без ошибок
3. Входные данные подобраны верно, но ошибок в программе нет
4. На текущих входных данных ошибок не возникло
5. Тест мог быть составлен некорректно

**Вопрос 9.** Дайте правильную классификацию unit-тестированию.

Выберите один или несколько ответов:

1. Модульное, функциональное, автоматизированное, альфа, белого ящика, негативное
2. Модульное, производительности, полу-автоматизированное, бета, не зависит от степени знания системы или позитивности
3. Модульное, функциональное, автоматизированное, альфа, чёрного ящика, позитивное
4. Модульное, функциональное, автоматизированное, альфа, не зависит от степени знания системы или позитивности
5. Интеграционное, функциональное, автоматизированное, дымовое, не зависит от степени знания системы или позитивности
6. Надёжности, функциональное, полу-автоматизированное, бета, не зависит от степени знания системы или позитивности
7. Производительности, функциональное, автоматизированное, альфа, не зависит от степени знания системы или позитивности

**Вопрос 10.** Какие преимущества имеет инспектирование исходного кода перед тестированием готового приложения?

Выберите один или несколько ответов:

1. Требуются высокооплачиваемые профессионалы для проведения тестирования
2. Преимуществ не имеет вовсе
3. Находит опечатки кода, но корректные с точки зрения синтаксиса языка
4. Инспектирование в пустую тратит время
5. Обнаруживает явные логические ошибки с точки зрения алгоритма

#### **Шкала оценивания и критерии оценки:**

<b>Критерий</b>	<b>Баллы обучающегося</b>	<b>Минимальное количество баллов</b>	<b>Максимальное количество баллов</b>
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 6 баллов и более		6	10
<b>Итого:</b>		<b>6</b>	<b>10</b>

**Соответствие баллов шкале оценивания:**

<b>Количество баллов</b>	<b>Оценка обучающегося</b>
10	Отлично
8-9	хорошо
6-7	удовлетворительно
менее 6	неудовлетворительно

**Тест по теме 1.2.2 Отладка и тестирование программного обеспечения («Unit-тестирование»)** (каждый правильный ответ имеет вес 2 балла)

**Вопрос 1.** Верно ли утверждение: "Невозможно провести unit-тестирование интерфейса."

Выберите один ответ:

1. Верно
2. Неверно

**Вопрос 2.** По какому методу (по степени знания системы) рекомендуется проведение unit-тестирования?

Выберите один или несколько ответов:

1. Метод белого ящика.
2. Не регламентируется.
3. Метод чёрного ящика.
4. Метод серого ящика.
5. Комбинирование методов.

**Вопрос 3.** Верно ли, что для Unit-теста в Qt необходимо создавать отдельный проект?

Выберите один ответ:

1. Верно
2. Неверно

**Вопрос 4.** По объекту тестирования unit-тест относится к виду?

Выберите один или несколько ответов:

1. Тестирование безопасности
2. Функциональное тестирование
3. Тестирование совместимости
4. Конфигурационное тестирование
5. Тестирование производительности

**Вопрос 5.** Выберите верное утверждение для unit-теста при проведении тестирования.

Выберите один или несколько ответов:

1. Один тестовый сценарий тестирует только одну функцию
2. Разрешается использование сложения и сравнение в неограниченных количествах
3. Все функции в тестируемом классе по умолчанию не содержат ошибки и являются доверенными
4. После прохождения одной функции тестов её можно использовать для тестирования другой функции
5. Unit-тест не должен содержать никаких сложных вычислений

**Вопрос 6.** Выберите корректные высказывания, характеризующие тестирование чёрного ящика.

Выберите один или несколько ответов:

1. Известен примерный алгоритм работы

2. Известны входные данные: их количество и типы
3. Исходный код тестируемой функции при составлении тестовых сценариев доступен
4. Если ошибочное данное успешно отсеяно, все наборы с ним удаляются из тестов
5. Каждый оператор функции должен быть достигнут при тестировании хотя бы 1 раз

**Вопрос 7.** Для тестовых слотов (сценариев) в Qt, в которых не используется дополнительный слот для подготовки данных к тестированию, верны следующие утверждения:

Выберите один или несколько ответов:

1. Каждый набор тестовых данных выводится как отдельный запуск в консоли вывода
2. При первой ошибке тестирование прекращается
3. Отображаются все ошибки найденные во время тестирования
4. Тестовый слот принимает значения тестируемых переменных в соответствии с тестируемой функцией
5. Название тестового слота должно совпадать с названием тестируемой функции

### Шкала оценивания и критерии оценки:

Критерий	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 7 баллов и более		8	14
<b>Итого:</b>		<b>8</b>	<b>14</b>

### Соответствие баллов шкале оценивания:

Количество баллов	Оценка обучающегося
13-14	Отлично
10-12	хорошо
8-9	удовлетворительно
менее 8	неудовлетворительно

### МДК.01.03 Разработка мобильных приложений

**Тест по теме 1.3.2 «Создание и тестирование модулей для мобильных приложений»**  
(каждый правильный ответ имеет вес 1 балл)

1. Что такое JNI?
  - Виртуальная машина исполнения
  - интерфейс для связи с нативным кодом
  - интерфейс для связи с другим приложением
2. Как выполняется дополнительный поток по отношению к главному?
  - синхронно с главным
  - после главного
  - асинхронно к главному
3. Что такое фоновая служба?
  - программа работающая без окна приложения
  - программа работающая без окна приложения но имеющая связь с приложением
  - программа в виде функции выделенной в поток

- программа имеющая наименьший приоритет
4. Динамическая память дает?
    - возможность во время работы программы выделять ее для новых ресурсов
    - возможность использовать виртуальную память в рамках своего приложения
    - возможность загружать быстро запоминать данные размещая их в кэш памяти процессора
  5. Что такое канва?
    - область экрана для расположения элементов управления
    - область памяти для расположения в ней изображения
    - область экрана для отрисовки графических примитивов
  6. Что такое зиготный процесс?
    - любая задача которая только начала исполняться
    - любая задача которая порождает исключение
    - задача-шаблон со всеми базовыми классами
  7. Что дает увеличение скорости обработки компьютерной графики?
    - Использование канвы для отрисовки примитивов
    - Использование аппаратного ускорения видео подсистемы
    - Использование специализированного фреймворка
  8. Широковещательное сообщение дает возможность:
    - Отправить всем приемникам одни и тем же данные
    - Отправить данные сразу всем классам текущего проекта
    - Отправить всем приложениям одни и те же данные
  9. Как заставить дополнительный поток выполняться синхронно?
    - Установить поток в таймер
    - Используя метод join присоединить его в очередь исполнения после последнего
    - Он в любом случае будет синхронным и будет исполняться в порядке очереди
  10. Контейнер для расположения элементов управления и установки им правил вывода на экран.
    - канва
    - менеджер компоновки
    - нет верного ответа

**Шкала оценивания и критерии оценки:**

Критерий	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 6 баллов и более		6	10
<b>Итого:</b>		<b>6</b>	<b>10</b>

**Соответствие баллов шкале оценивания:**

Количество баллов	Оценка обучающегося
10	отлично
8-9	хорошо
6-7	удовлетворительно
менее 6	неудовлетворительно

## МДК.01.04 Системное программирование

### Тест по теме 1.4.1 Микропрограммирование

1. Динамическая разделяемая библиотека...
  - компилируется в программу и становится ее частью
  - вызывается из программы в начале загрузки или при необходимости
  - это любая библиотека, которая расположена в ОЗУ
2. Стек-фрейм – это...
  - область стека, ограниченная специализированными идентификаторами его начала и конца
  - область стека в любой программе, при условии что речь идет о многозадачной ОС
  - область стека любой функции, к которой помещены локальные переменные
3. Дизассемблирование – это...
  - написание программ на ассемблере
  - преобразование машинного кода к ассемблеру
  - преобразование ассемблера к машинному коду
4. Атомарная операция – это...
  - Операция, которая оперирует не выровненными в памяти данными
  - операция, которая выполняется за несколько проходов исполнения
  - операция, которая выполняется за один проход исполнения и не имеет деления на субоперации
5. Загрузочная область жесткого диска в формате MBR состоит из...
  - расположенной в начале микропрограммы загрузки размером 512 байтов и расположенным в конце идентификатором 0x55aa
  - расположенным в начале двухбайтовым идентификатором 0x55aa и микропрограммой загрузки
  - расположенной в начале микропрограммой загрузки таблицей разделов и расположенным в конце идентификатором 0x55aa
6. Векторная операция – это...
  - любая операция над данными в разных регистрах, которая сохраняет результат в память
  - любая операция над данными в разных регистрах, которая выполняется над каждым блоком данных в этих регистрах
  - любая операция над двумя линейными множествами в памяти, которая выполняется над каждым блоком данных в этих множествах
7. Фрейм буфер видеоподсистемы – это...
  - сегментная область памяти размером 64кб внутри видеопамати видеоподсистемы
  - линейная область памяти размером, равным размеру всей видеопамати или ее части
  - линейная область оперативной памяти размером, равным разрешению экрана
8. В защищенном режиме область памяти программы описывается с помощью...
  - Дескриптора, содержащего ограничения памяти и флаги
  - никак не описывается, т.к. адресация возможна в пределах разрядности регистра счетчика команд
  - сегментной организации программы, где каждый сегмент будет иметь размер 64кб и начальный адрес в сегментном регистре
9. Какие особенности вносит ассемблерная вставка?
  - отсутствие аппаратной переносимости на иные наборы команд, но при этом высокую скорость работы
  - высокая скорость выполнения кода и небольшой размер программы
  - отсутствие аппаратной переносимости на иные наборы команд и небольшой размер программы

10. Передача данных в память устройства наиболее быстра в случае, если...

- передача происходит через программное прерывание
- передача происходит через порт устройства
- передача происходит через общую память

**Шкала оценивания и критерии оценки:**

<b>Критерий</b>	<b>Баллы обучающегося</b>	<b>Минимальное количество баллов</b>	<b>Максимальное количество баллов</b>
Количество правильных ответов на вопросы теста при общем количестве правильных ответов не менее, чем на 6 баллов и более		6	10
<b>Итого:</b>		<b>6</b>	<b>10</b>

**Соответствие баллов шкале оценивания:**

<b>Количество баллов</b>	<b>Оценка обучающегося</b>
10	отлично
8-9	хорошо
6-7	удовлетворительно
менее 6	неудовлетворительно

## **КОНТРОЛЬНАЯ РАБОТА**

Контрольная работа (КР) являются формой текущего контроля и содержат практические задания. Контрольные работа выполняется обучающимися во время аудиторных занятий. На выполнение отводится 2 академических часа. Работы выполняются индивидуально, представляются в письменной форме и должны удовлетворять следующим требованиям: в работе указывается ФИО студента, номер группы, условие каждого задания, выполнение. Каждое задание КР имеет в зависимости от вида задачи определенный вес (в промежуточных баллах).

### **Комплект заданий контрольной работы**

#### **Тема 1.4.1 «Системное программирование» (Дизассемблирование программ и стандартные алгоритмические конструкции)**

Задание 1. Дизассемблировать программы:

```
int i,j,a;
For(i=0;i<16;i++)
For(j=7;j<5;j++)
a=8;
int a=5,b=7,c=0;
if(a>b)c=1;else c=-1;
```

Задание 2. Способы хранения данных в программе с плоской моделью памяти.

Задание 3. Особенности работы стек фрейма в 32 разрядных вычислительных системах. Способы вызова функций в 32 и 64 разрядных операционных системах.

Задание 4. Использование встроенного ассемблера на примере оптимизации функции поиска вхождения точки в окружность с известными параметрами. Оптимизировать с использованием математического сопроцессора.

#### **Тема 1.4.1 «Микропрограммирование» (Загрузочные программы и микро ОС)**

##### **Вариант 1**

Задание 1. Загрузочная область. Требования к загрузочным программам и их реализации.

Задание 2. Особенности перехода и работы в защищенном режиме без поддержки со стороны ОС. Дескрипторы и распределение памяти.

Задание 3. Управление буфером кадров видеоподсистемы в защищенном режиме.

Задание 4. Использование векторных операций над данными в высоконагруженных программах.

##### **Шкала оценивания и критерии оценки:**

**Минимальное количество баллов — 4 балла**

**Максимальное количество баллов — 8 баллов**

- правильное решение менее 2 задач – 0 баллов,
- каждая правильно выполненное задание при общем количестве выполненных более 2 оценивается в 2 балла.

Основаниями для снижения количества баллов за одну задачу в диапазоне от 0.25 до 1 балла являются:

- небрежное выполнение,
- низкое качество графического материала,
- подмены (специально или по неосторожности) задания,
- наличие любого электронного устройства с дополнительной информацией или средствами решения способствующими выполнению заданий или написанию ответов на вопросы.

**Соответствие баллов шкале оценивания:**

<b>Количество баллов</b>	<b>Оценка обучающегося</b>
7,5-8	отлично
5.8-7.4	хорошо
4-5.7	удовлетворительно
менее 4	неудовлетворительно

## УСТНО-ПРАКТИЧЕСКИЙ ЭКЗАМЕН (МДК.01.01), 6 сем.)

Экзамен по МДК.01.01 проводится в 6 семестре в комбинированной форме и представляет собой ответы обучающихся на теоретические контрольные вопросы по изученным темам междисциплинарного курса и решение практической задачи. В экзаменационный билет включается два теоретических вопроса и одно практическое задание. Для подготовки студенты отводятся не более 2 академических часов.

Ответ должен содержать определение понятий, входящих в вопрос; при необходимости интерпретацию понятий (иллюстрации и (или) схемы); назначение команд, примеры использования; изложение методов, указание границ их применимости; примеры практического применения понятий, методов, команд.

### Примерный перечень вопросов для подготовки к экзамену:

1. Этапы жизненного цикла программного обеспечения.
2. Элементы языка C++. Алфавит. Константы. Идентификаторы. Ключевые слова. Знаки операций. Использование комментариев в тексте программы. Примеры.
3. Синтаксис языка C/C++. Типы данных.
4. Запись констант в языке C/C++. Основные операции. Запись выражений.
5. Структура программы на языке C/C++. Описание переменных и именованных констант.
6. Простые операторы языка C/C++. Примеры.
7. Типы данных и их объявление. Основные типы данных. Спецификаторы типов данных. Описание, объявление и определение переменных. Инициализация данных. Выражения и переменные. Константы. Примеры.
8. Операнды и операции. Арифметические операции. Операции присваивания. Инкремент, декремент. Операция определения размера sizeof. Операции отношения. Логические операции. Операции отрицания. Операции сдвига. Поразрядные операции. Условная операция ? : . Приоритет операций и порядок вычислений. Примеры.
9. Операторы. Оператор выражение. Пустой оператор. Составной оператор. Примеры.
10. Операторы ветвления. Оператор if. Оператор switch. Примеры.
11. Операторы цикла. Оператор for. Оператор while. Оператор do while. Примеры.
12. Операторы передачи управления. Оператор continue. Оператор break. Оператор return. Оператор goto. Примеры.
13. Классы памяти в языке C/C++.
14. Указатели. Инициализация указателей. Операции над указателями. Пустой указатель (NULL-указатель). Ссылки. Примеры.
15. Массивы. Одномерные массивы. Многомерные массивы. Методы доступа к элементам массивов. Указатели на много мерные массивы. Массивы указателей. Примеры.
16. Строки. Внутреннее представление. Строковая библиотека. Операции над строками. Примеры.
17. Переименование типов. Перечисления. Структуры. Битовые поля. Объединения. Примеры.
18. Файловый ввод/вывод средствами языка Си (на верхнем уровне). Типы потоков.
19. Стандартные потоки. Основные функции для работы с файлами. Примеры.
20. Файловый и консольный ввод-вывод средствами языка C++. Иерархия потоковых классов.
21. Операции, перегружаемые в классах. Средства форматирования при файловом и консольном выводе. Примеры.
22. Побитовые операции языка C/C++. Примеры использования.
23. Символьные строки в языке Си. Описание и инициализация. Основные функции обработки символьных строк.

24. Строковый класс в языке C++. Описание и инициализация. Основные операции и методы, определенные в классе.
25. Динамические переменные. Средства для работы с динамической памятью, определенные в языках Си и C++. Создание одномерных и многомерных динамических массивов.
26. Структура программы. Функции. Объявление и определение функций. Инициализация глобальных и локальных переменных и возвращаемое значение. Области действия идентификаторов. Параметры функции. Примеры.
27. Функции, их описание. Вызов функции. Описание прототипа функции. Примеры.
28. Передача в функцию параметров различных типов по значению и по ссылке. Передача параметров со значениями по умолчанию.
29. Перегрузка функций. Способы возврата функцией одномерных и двумерных массивов. Примеры.
30. Рекурсия. Механизм рекурсивных вызовов. Рекурсивный спуск и подъем. Примеры.
31. Параметры со значением по умолчанию. Функции с переменным числом параметров. Рекурсивные функции. Перегрузка функций. Шаблоны функций. Примеры.
32. Директивы препроцессора. Директива `#include`. Директива `#define`. Директива `#undef`. Директивы условной компиляции. Примеры.
33. Организация списков и их обработка. Линейные списки. Методы организации и хранения линейных списков. Линейный список как массив данных. Операции со списками при последовательном хранении (односвязные списки). Организация двусвязных списков. Организация циклических связей в списках. Операции со списками. Примеры.
34. Последовательности, стеки и очереди. Последовательности, стеки и очереди как массивы данных. Операции с ними. Примеры.
35. Деревья. Обход дерева. Сравнение деревьев со списками и массивами. Определение глубины дерева. Поиск в дереве. Обход дерева. Двоичные деревья. Поиск и включение в двоичные деревья. Нумерация вершин в двоичном дереве.
36. Паттерны проектирования. Почему они полезны?
37. Порождающие паттерны. Singleton, Builder, Factory Method, Prototype.
38. Поведенческие паттерны. Iterator, Command, Template Method, State, Strategy.
39. Структурные паттерны. Adapter, Decorator, Proxy, Facade, Composite.
40. Паттерны Object Pool, Null Object, Servant. Модель MVC. Модель цикла обработки событий (Event Loop).
41. Работа с библиотеками в C++. Структурирование исходного кода в больших проектах.
42. Методы оптимизации кода.
43. Методы рефакторинга кода.
44. Правила разработки интерфейсов пользователя.

### **Типовые экзаменационные практические задания:**

1. Написать функцию `power()`, которая принимает два аргумента `x` и `n` и вычисляет  $x$  в степени  $n$ .
2. Написать функцию `sum_sum()`, которая вычисляет куммулятивную сумму для каждого элемента массива.
3. Написать функцию `reverse()`, которая разворачивает массив элементов, таким образом, все элементы следуют в обратном порядке.
4. Написать функцию `unique()`, которая подсчитывает число уникальных элементов в массиве.
5. Написать функцию `isort()`, которая упорядочивает элементы массива, используя алгоритм вставками.
6. Написать функцию `datedelta()`, которая вычисляет сколько дней прошло между двумя датами.

7. Написать функцию `timedelta`, которая находит разницу между двумя точками времени  $t_2 > t_1$ .
8. Написать функцию `bin_search()`, которая ищет указанный элемент `e` в массиве методом бинарного (двоичного) поиска.
9. Написать функцию `shift()`, которая циклически сдвигает все элементы массива на указанное число позиций `m`.
10. Написать две небольших функции, которые проверяют, что слово и число являются палиндромами.
11. Написать функцию `transpose()`, которая транспортирует матрицу (двумерный массив).
12. Написать функцию, которая проверяет сбалансированность скобок в строке (каждой открывающей должна соответствовать закрывающая), строка задана массивом символов.
13. Создать структуру `rational`, которая описывает рациональное число вида  $m/n$  и написать основные функции для работы с рациональными числами.

### Пример экзаменационного билета

#### Экзаменационный билет № 1

**Вопрос 1.** Синтаксис языка C/C++. Типы данных.

**Вопрос 2.** Классы памяти в языке C/C++.

**Задание 1.** Написать функцию `bin_search()`, которая ищет указанный элемент `e` в массиве методом бинарного (двоичного) поиска.

Шкала оценивания и критерии оценки:

Критерии оценки	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Уровень усвоения теоретического и практического материала, предусмотренного программой		5,5	8
Уровень знакомства с основной литературой, предусмотренной программой		1	2
Уровень раскрытия причинно-следственных связей		2	3
Уровень раскрытия междисциплинарных связей		0,5	1
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		1	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		1	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		1	2
<b>Итого баллов:</b>		<b>12</b>	<b>20</b>

**Соответствие баллов шкале оценивания:**

Количество баллов	Оценка обучающегося
18-20	отлично
15-17	хорошо
12-14	удовлетворительно
менее 12	неудовлетворительно

Знания, умения и навыки обучающихся при промежуточной аттестации **в форме экзамена** определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

«Отлично» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

«Хорошо» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

«Удовлетворительно» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

«Неудовлетворительно» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

## **УСТНО-ПРАКТИЧЕСКИЙ ЭКЗАМЕН (МДК.01.01 (1.1.9), 5 сем.)**

Экзамен по теме 1.1.2 МДК 01.01 проводится 5 семестре в комбинированной форме и представляет собой ответы обучающихся на теоретические контрольные вопросы по изученным темам междисциплинарного курса и решение практической задачи. В экзаменационный билет включается один теоретический вопрос и одно практическое задание. Для подготовки студенты отводится не более 2 академических часов.

Ответ должен содержать определение понятий, входящих в вопрос; при необходимости интерпретацию понятий (иллюстрации и (или) схемы); назначение команд, примеры использования; изложение методов, указание границ их применимости; примеры практического применения понятий, методов, команд.

### **Примерный перечень вопросов для подготовки к экзамену:**

1. Основные понятия языка web-программирования на серверной стороне PHP. Типы данных и их преобразование.
2. Функции, рекурсии и области видимости переменных. Способы передачи данных с клиентской стороны и их обработка.
3. Способы получения и обработки данных в PHP. Методы отправки данных GET и POST. Их различия, плюсы и минусы.
4. Работа с файловой системой сервера: запись и считывание данных из файлов, использование каталогов, загрузка файлов с клиентской стороны.
5. Работа с файловой системой сервера в PHP. Права доступа. Работа с каталогами.
6. Работа с сервером баз данных.
7. Запись и считывание данных из файла. Понятие дескриптора и смещения внутреннего указателя.
8. Суперглобальные данные в PHP. Использование механизмов session и cookie. Плюсы и минусы этих двух механизмов.
9. Регулярные выражения. Назначение, формат и использование в сценариях PHP.
10. Работа с базами данных в сценариях PHP. Подключение внешних файлов со сценариями.
11. Встроенные библиотеки PHP. Работа с массивами, строками, датой и временем.
12. Технология AJAX. Назначение и использование.

### **Пример экзаменационного билета**

#### **Экзаменационный билет № \_\_**

##### Часть I: теоретическая

1. Понятие суперглобальных данных в PHP и их назначение. Примеры использования. Два механизма реализации использования суперглобальных данных и их этапы. Проиллюстрировать диалог между клиентской и серверной стороной в контексте использования суперглобальных данных. Указать ключевое различие в этих механизмах, их плюсы и минусы.

##### 2. Часть II: практическая

1. Создать web-страницу, предоставляющую пользователю меню ресторана.

Напротив каждого списка меню отображается его цена и поле формы для установки маркера выбора.

При отметке маркером выбранного блюда его цена прибавляется к общей стоимости заказа. При снятии маркера – отнимается. Общая стоимость заказа отображается внизу заказа в пункте «Итого». При изменении заказа сразу же изменяется и значение этого пункта.

После формирования заказа пользователь нажимает кнопку «Сделать заказ» и данные о заказе сохраняются в Базе данных на серверной стороне.

При обновлении web-страницы отмеченные пункты сохраняются.

Условия проведения экзамена:

Задание 1 предполагает устный ответ обучающегося.

Задания 2 выполняется практически на ПК с использованием следующего программного обеспечения:

- веб-клиент (Chrome, Mozilla Firefox, и т.д.);
- текстовый редактор исходного кода (Notepad++, Sublime Text, и т.д.);

веб-сервер либо эмулятор веб-сервера (Denwer, Open Server, и т.д.)

Шкала оценивания и критерии оценки:

Критерии оценки	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Уровень усвоения теоретического и практического материала, предусмотренного программой		5,5	8
Уровень знакомства с основной литературой, предусмотренной программой		1	2
Уровень раскрытия причинно-следственных связей		2	3
Уровень раскрытия междисциплинарных связей		0,5	1
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		1	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		1	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		1	2
<b>Итого баллов:</b>		<b>12</b>	<b>20</b>

**Соответствие баллов шкале оценивания:**

Количество баллов	Оценка обучающегося
18-20	отлично
15-17	хорошо
12-14	удовлетворительно
менее 12	неудовлетворительно

Знания, умения и навыки обучающихся при промежуточной аттестации **в форме экзамена** определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

«Отлично» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

«Хорошо» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно

применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

«Удовлетворительно» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

«Неудовлетворительно» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

## **УСТНЫЙ ЭКЗАМЕН (МДК.01.02, 6 СЕМ.)**

Экзамен по МДК.01.02 проводится в 6 семестре в устной форме и представляет собой ответы обучающихся на теоретические контрольные вопросы по изученным темам междисциплинарного курса. В экзаменационный билет включается 2 теоретических вопроса. Для подготовки студенту отводится не более 1 академического часа.

Ответ должен содержать определение понятий, входящих в вопрос; при необходимости интерпретацию понятий (иллюстрации и (или) схемы); примеры использования; изложение методов, указание границ их применимости; примеры практического применения понятий, методов, стандартов.

### **Примерный перечень вопросов для подготовки к экзамену:**

1. Понятие верификации и аттестации ПО.
2. Методы обеспечения верификации ПО.
3. Методы контроля качества ПО.
4. Понятие инспектирования ПО
5. Процесс аудита ПО, процесс совместного анализа Пою
6. Проведение инспектирования ПО на различных стадиях ЖЦ.
7. Правила проведения инспектирования ПО.
8. Понятие тестирования ПО; тестовые наборы ПО.
9. Правила составления тестовых наборов ПО.
10. Виды тестирования: по объекту системы, по знанию системы, по степени автоматизации, по степени изолированности, по времени проведения, по признаку позитивности, по подготовленности к тестированию.
11. Основные понятия и термины метрологии ПО.
12. Стандарты в области метрологии ПО: Метрология ПО как основа повышения качества ПО
13. Основные понятия и ключевые слова: сложность проектирования ПО, трудоемкость, вычислительная сложность, производительность, эффективность, качество, метрика, измерительный монитор.
14. Отечественные ГОСТы и международные стандарты по метрологии и качеству ПО.
15. Понятие качества программного продукта (ПП).
16. Показатели и характеристики качества ПП: результаты разработки программного обеспечения: спецификация, проект, код, документация, тестовые наборы.
17. Понятие качества программного продукта (ПП).
18. Показатели, характеризующие качество разработки ПП. Характеристики качества собственно ПП: корректность, надежность, сложность, эффективность, удобство использования, сопровождаемость, мобильность.
19. Критерии качества разработки и использования программного продукта (ПП).
20. Виды метрик качества ПП.
21. Проблемы управления качеством: статический и динамический анализ качества ПП.
22. Критерии качества технологий проектирования ПО и критерии качества собственно ПП; функциональные и конструктивные критерии качества ПП;
23. Виды метрик для оценки качества ПП: номинальные, порядковые, ранжирующие.
24. Организация сбора метрик качества ПП.
25. Управление качеством ПП по результатам обработки метрик.
26. Стандарты качества программного обеспечения: документы и стандарты: ISO 9001, ISO 9000-3, IEEE.

## Пример экзаменационного билета

### Экзаменационный билет № \_\_

1. Показатели и характеристики качества ПП: результаты разработки программного обеспечения: спецификация, проект, код, документация, тестовые наборы.
2. Организация сбора метрик качества ПП.

Шкала и критерии оценивания (итоговое оценивание по МДК):

Критерии оценки	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Уровень усвоения теоретического и практического материала, предусмотренного программой		5,5	8
Уровень знакомства с основной литературой, предусмотренной программой		1	2
Уровень раскрытия причинно-следственных связей		2	3
Уровень раскрытия междисциплинарных связей		0,5	1
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		1	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		1	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		1	2
<b>Итого баллов:</b>		<b>12</b>	<b>20</b>

### Соответствие баллов шкале оценивания:

Количество баллов	Оценка обучающегося
18-20	отлично
15-17	хорошо
12-14	удовлетворительно
менее 12	неудовлетворительно

Знания, умения и навыки обучающихся при промежуточном контроле в форме экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

1. «Отлично» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

2. «Хорошо» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

3. «Удовлетворительно» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

4. «Неудовлетворительно» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, а также если обучающийся после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена (списывал, подсказывал и т.д.).

## **УСТНО-ПРАКТИЧЕСКИЙ ЭКЗАМЕН (МДК.01.03, 5сем.)**

Экзамен по МДК.01.03 проводится 5 семестре в комбинированной форме и представляет собой ответы обучающихся на теоретические контрольные вопросы по изученным темам междисциплинарного курса и решение практической задачи. В экзаменационный билет включается два теоретических вопроса и одно практическое задание. Для подготовки студенты отводятся не более 2 академических часов.

Ответ должен содержать определение понятий, входящих в вопрос; при необходимости интерпретацию понятий (иллюстрации и (или) схемы); назначение команд, примеры использования; изложение методов, указание границ их применимости; примеры практического применения понятий, методов, команд.

### **Примерный перечень вопросов для подготовки к экзамену:**

1. Типы данных и базовые конструкции языка Java
2. Динамические конструкции для хранения данных
3. ООП архитектура программы в рамках языка Java
4. Архитектура виртуальной машины исполнения Java
5. Архитектура виртуальной машины исполнения в рамках ОС Android
6. Архитектура ОС Android
7. Многопоточные программы на языке Java
8. Описание графического интерфейса путем использования XML
9. Система событий библиотеки графических элементов ОС Android
10. Функции работы с текстом и числовыми значениями языка Java
11. Асинхронные потоки в программах для ОС Android
12. Функции работы со звуком
13. Функции чтения и записи данных в языке Java
14. Функции работы с канвой экрана и вывод графических примитивов
15. Разработка нативных модулей
16. Вызов нативных функций и передача им аргументов
17. Работа с динамической памятью в рамках нативных функций
18. Способы реализации компьютерной графики в ОС Android
19. Сторонние Фреймворки реализации компьютерной графики и их особенности работы
20. Службы. Жизненный цикл и особенности работы
21. Жизненный цикл окна и особенности работы
22. Фоновые процессы и широковещательные сообщения
23. Обработка событий аппаратного обеспечения
24. Механизмы передачи данных между окнами в рамках приложения
25. Механизмы передачи данных между основным потоком и асинхронным потоком

### **Типовые экзаменационные практические задания:**

1. Написать программу для ОС Android с выводом в текстовый элемент управления значений массива больше 19 и меньше 26.
2. Написать программу для ОС Android с возможностью ввода текстовой строки в элемент управления и выводом этой текстовой строки при нажатии кнопки в заголовок программы.
3. Написать программу для ОС Android с тремя элементами управления типа кнопка. При нажатии только на одну из них она должна быть выделена красным цветом а все остальные серым.
4. Написать программу для ОС Android вывода на экран выбранной строки матрицы с текстом.

5. Написать программу для ОС Android вывода на экран трех элементов управления, среди которых: кнопка, кнопка, поле редактирования текста. Реализовать функцию адаптации перечисленных элементов управления под размер экрана и соотношение сторон.

6. Написать программу для ОС Android вывода в текстовый элемент управления текущего времени с обновлением раз в секунду.

7. Написать программу для ОС Android с элементом управления типа кнопка, при нажатии на которую должно быть открыто новое окно, в которое необходимо передать произвольный текст и установить его в виде заголовка окна.

8. Написать программу для ОС Android, устанавливающую в качестве текста заголовка выбранный текст из элемента управления типа раскрывающийся список.

9. Написать программу для ОС Android, выводящую поля класса с названием Name, Surname, Address, City, Phone в элемент управления типа список.

10. Написать программу для ОС Android, выводящую матрицу размером 4x6 в элемент управления типа решетка.

11. Написать программу для ОС Android с элементами управления типа кнопка и изображения. При нажатии на кнопку показывать по порядку одно из пяти изображений.

## Пример экзаменационного билета

### Экзаменационный билет № 1

**Вопрос 1.** Описание графического интерфейса путем использования XML. Многопоточные программы на языке Java.

**Вопрос 2.** Функции работы с канвой экрана и вывод графических примитивов. Система событий библиотеки графических элементов ОС Android

**Задание 1.** Написать программу для ОС Android, устанавливающую в качестве текста заголовка выбранный текст из элемента управления типа раскрывающийся список.

#### Условия проведения экзамена:

Задание 1 и 2 предполагают устный ответ обучающегося.

Задания 2 выполняется практически на ПК с использованием следующего программного обеспечения: среда разработки Android studio (свободная среда программирования для ОС Android. Лицензия Apache 2.0).

Шкала оценивания и критерии оценки:

Критерии оценки	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Уровень усвоения теоретического и практического материала, предусмотренного программой		5,5	8
Уровень знакомства с основной литературой, предусмотренной программой		1	2
Уровень раскрытия причинно-следственных связей		2	3
Уровень раскрытия междисциплинарных связей		0,5	1
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		1	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		1	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		1	2

<b>Итого баллов:</b>	<b>12</b>	<b>20</b>
----------------------	-----------	-----------

**Соответствие баллов шкале оценивания:**

<b>Количество баллов</b>	<b>Оценка обучающегося</b>
18-20	отлично
15-17	хорошо
12-14	удовлетворительно
менее 12	неудовлетворительно

Знания, умения и навыки обучающихся при промежуточной аттестации **в форме экзамена** определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

«Отлично» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

«Хорошо» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

«Удовлетворительно» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

«Неудовлетворительно» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

## **ПИСЬМЕННЫЙ ДИФФЕРЕНЦИРОВАННЫЙ ЗАЧЕТ (МДК.01.04), 8 семестр)**

Дифференцированный зачет по МДК.01.04 проводится в 8 семестре в письменной форме. Задание для дифференцированного зачета содержит два теоретических вопроса и одно практическое задание. На выполнение отводится 2 академических часа. Работы представляются в письменной форме на бланке и должны удовлетворять следующим требованиям: в работе указывается ФИО студента, номер группы, условие каждого задания, выполнение.

### **Перечень вопросов для подготовки к дифференцированному зачету:**

1. Назвать и описать техническое назначение восьми видов адресации и их подвидов.
2. Коды операций в 16-ти и 32-х разрядном режиме.
3. Управление памятью при помощи указателей.
4. Виртуальная память, страницы, кадры.
5. Реверс инжиниринг программ, дизассемблирование стандартных алгоритмических конструкций.
6. стек-фрейм.
7. Использование внешнего и встроенного ассемблера при оптимизации программ.
8. Комбинирование микропрограмм с языками высокого уровня.
9. Технологии MMX, SSE, команды и примеры использования.
10. Особенности разработки независимых от ОС программ.
11. Работа программы в защищенном режиме. Таблицы дескрипторов.
12. Программирование видеопамати в реальном и защищенном режиме.
13. Основы разработки ОС, загрузочная область, написание стартового кода, написание независимых от ОС программ и параметры их сборки.
14. Оптимизация операций копирования с использованием технологий MMX, SSE.
15. Низкоуровневая работа с жестким диском, способы адресации к данным.
16. Реализация микропрограмм в виде отдельных библиотек.
17. Способы оптимизации программ. Оптимизация кода по скорости выполнения, по работе с данными, по размеру, по уровню аппаратной переносимости.
18. Внедрение в код на ассемблере высокоуровневых операций языка Си, способы сборки таких программ и параметры вызовов операций.
19. Комбинирование микропрограмм с языками высокого уровня на примере C, C++.
20. Технология JNI.
21. Векторные команды SSE.
22. Векторные команды AVX.
23. Обработка данных с использованием команд SSE и обоснование выбора технологии на фоне команд ALU/MMX.
24. Архитектура виртуальной машины Java и представление кодов команд.
25. Способы реализации многозадачных систем.
26. Программная модель процессора при многопоточном программировании.
27. Низкоуровневая реализация потоков.
28. Реализация низкоуровневого управления памятью.
29. Архитектура процессоров MIPS.
30. Система команд процессоров MIPS.
31. Коды операций процессоров MIPS.
32. Программирование периферийных устройств через команды процессоров MIPS.
33. Организация системы памяти процессоров MIPS.
34. Многопоточные ОС. Реализация потоков на уровне ОС.
35. Использование нескольких ядер процессора для параллелизации исполнения программ.
36. Система обмена данными между потоками в рамках одного или нескольких процессов.
37. Межпроцессная передача данных в современных ОС.

38. Стек протоколов TCP/IP.
39. Что такое сокет, порт, маска, сетевой адрес.
40. Передача данных по LВС между программами.
41. Особенности архитектуры современных видеоподсистем.
42. Особенности адресации видеопамяти и видео фреймбуфера
43. Способы программирования компьютерной графики в современных ОС.

### **Типовые практические задания для дифференцированного зачета:**

1. Написать программу на ассемблере Intel перевода процессора в защищенный режим.
2. Написать модуль на ассемблере Intel/SSE копирования указанного диапазона данных из одной области памяти в другую. Реализацию написать на языке Си.
3. Написать программу предварительной загрузки на ассемблере Intel
4. Написать программу на языке Си с ассемблерными вставками (intel), выполняющую чтение указанного сектора в нулевой дорожке жесткого диска, вывести на экран первые два байта считанной области.
5. Написать программу на ассемблере Intel использующую LFB видеоадаптера.
6. Написать программу на языке Си без использования ассемблерных вставок, реализующую вывод на экран пикселей в режиме реального времени.
7. Написать программу на языке Си без использования ассемблерных вставок, реализующую вывод на экран текстовой строки в режиме реального времени.
8. Написать программу на ассемблере MIPS с процедурой поиска максимума во множестве статически заданных данных. Указать опции сборки.
9. Написать библиотеку на ассемблере Intel, фильтрующую переданное множество байтов на значения превышающие фильтр, вызывающий код на языке Си. Указать опции сборки.
10. Написать библиотеку на языке Си для сортировки полученных от вызывающей программы на Java данных. Указать опции сборки.
11. Написать библиотеку на ассемблере Intel, выполняющую математическую векторную операцию через MMX над переданными строками данных из вызывающего кода на Си. Указать опции сборки
12. Написать программу на C++ в виде класса, вызывающего низкоуровневый код в подставляемом виде. Методы класса: сложение, вычитание, поиск длины строки. Указать опции сборки.
13. Написать библиотеку на ассемблере Intel поиска указанного значения во множестве данных. Вызывающий код написать на языке Java. Указать опции сборки.
14. Написать библиотеку на ассемблере Intel записи данных в стековую область без применения стандартных операций push, pop и производных. Вызывающий код написать на языке Си. Указать опции сборки.
15. Написать программу на ассемблере MIPS вывода залитой квадратной области. Адрес видеопамяти произвольный, глубина цвета 2 байта.
16. Написать программу на ассемблере MIPS с процедурами записи и извлечения данных из стека.
17. Написать программу на ассемблере MIPS, выполняющую поиск единиц в главной диагонали матрицы, матрицу представить в виде одномерного множества.
18. Написать программу реализующую несколько потоков параллельной обработки данных целочисленного типа
19. Написать программу реализующую передачу текстовой строки в рамках одной программы между двумя потоками.
20. Написать программу передачи текстовой строки между двумя программами по средствам межпроцессного взаимодействия.

21. Написать программу создания сетевого сокета и передачи данных другой программе.

**Пример содержание задания для письменного дифференцированного зачета**

**Билет №** \_\_\_\_\_

1. Векторные команды SSE. Способы реализации многозадачных систем. Система команд процессоров MIPS
2. Комбинирование микропрограмм с языками высокого уровня на примере C, C++.
3. Написать программу на ассемблере MIPS вывода залитой квадратной области. Адрес видеопамати произвольный, глубина цвета 2 байта.

Шкалы оценивания и критерии оценки:

Критерии оценки	Баллы обучающегося	Минимальное количество баллов	Максимальное количество баллов
Уровень усвоения теоретического материала, предусмотренного программой		3	4
Умение выполнять практические задания, предусмотренные программой		3	4
Уровень знакомства с основной литературой, предусмотренной программой		1	2
Уровень знакомства с дополнительной литературой		0,5	1
Уровень раскрытия причинно-следственных связей		1	2
Уровень раскрытия междисциплинарных связей		1	2
Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)		1	2
Ответы на вопросы: полнота, аргументированность, убежденность, умение использовать ответы на вопросы для более полного раскрытия содержания вопроса		1	2
Деловые и волевые качества докладчика: ответственное отношение к работе, стремление к достижению высоких результатов, готовность к дискуссии, контактность		0,5	1
<b>Итого баллов:</b>		<b>12</b>	<b>20</b>

**Соответствие баллов шкале оценивания:**

Количество баллов	Оценка обучающегося
19-20	зачтено (отлично)
15-18	зачтено (хорошо)
12-14	зачтено (удовлетворительно)
менее 12	не зачтено (неудовлетворительно)

Знания, умения и навыки обучающихся при промежуточной аттестации **в форме экзамена** определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

«Отлично» – обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.

«Хорошо» – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно

применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.

«Удовлетворительно» – обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.

«Неудовлетворительно» – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

## КВАЛИФИКАЦИОННЫЙ ЭКЗАМЕН ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

Итоговая оценка результатов освоения профессионального модуля осуществляется в рамках комплексного квалификационного экзамена, в ходе которого экзаменационной комиссией анализируется освоение ПК и ОК в условиях, приближенных к трудовой деятельности. Сформированность профессиональных и общих компетенций определяется «зачтено» (основной вид деятельности освоен), «не зачтено» (основной вид деятельности не освоен).

Комплексный квалификационный экзамен проводится по окончании производственной практики (по профилю специальности) по профессиональным модулям ПМ.01, ПМ.02 и ПМ.04. Экзамен проводится в форме защиты результатов выполнения индивидуального проекта по проектированию и созданию прототипа программного продукта (в рамках производственной практики) в форме презентации (дополнительно предоставляется отчет по производственной практике, содержащий, обоснование, результаты моделирования предметной области, техническое задание, и программный прототип).

### Требования к содержанию и структуре презентации

1. Презентация должна быть выполнена в программе PowerPoint в формате ppt или pptx.
2. Объем презентации – не менее 10 слайдов.
3. На титульном слайде должны быть указаны название университета, тема работы, фамилия, имя, отчество автора(ов), номер учебной группы; фамилия, имя, отчество и ученая степень и ученое звание преподавателя.
4. В презентации необходимо отразить все этапы выполнения проектного задания.
5. На заключительном слайде должен быть представлен список использованных источников.
6. Представление в презентации материала по выбранной теме должно иметь четкую структуру и отражать наиболее важные аспекты темы.

Текст на слайдах должен быть тезисным, отражающим самые важные аспекты темы.

Возможно добавление необходимых пояснений и примеров в заметки к слайдам.

Предпоследний слайд презентации должен содержать авторские выводы по теме работы.

На доклад студенту отводится не более 7 минут.

### Шкала оценивания и критерии оценки презентации:

Критерии	Минимальный ответ, менее 60 баллов	Изложенный, раскрытый ответ, 60-74 балла	Законченный, полный ответ, 74-90 баллов	Образцовый, примерный, достойный подражания ответ, 90-100 баллов	Баллы обучающегося
Постановка задачи (обоснование проекта) ( <i>min количество баллов – 11, max количество баллов – 17</i> )	Проблема не раскрыта. Отсутствуют выводы.	Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы.	Проблема раскрыта. Проведен анализ проблемы без привлечения дополнительной литературы. Не все выводы сделаны и/или обоснованы.	Проблема раскрыта полностью. Проведен анализ проблемы с привлечением дополнительной литературы. Выводы обоснованы.	
Техническое задание	Техническое задание не	Техническое задание	Техническое задание	Техническое задание	

<i>(min количество баллов – 16, max количество баллов – 27)</i>	соответствует заявленной теме.	сформировано в соответствии с ГОСТ. Имеются существенные замечания.	сформировано в соответствии с ГОСТ. Имеются несущественные замечания.	сформировано в полном соответствии с ГОСТ.	
<i>Программная реализация (прототип программного продукта согласно индивидуальному заданию) (min количество баллов – 16, max количество баллов – 27)</i>	Реализованный программный продукт не соответствует требованиям функциональности согласно ТЗ.	Реализованный программный продукт имеет нарушения требований функциональности согласно ТЗ.	Реализованный программный продукт незначительно не соответствует требованиям функциональности согласно ТЗ.	Реализованный программный продукт соответствует всем требованиям функциональности согласно ТЗ.	
<i>Представление (min количество баллов – 6, max количество баллов – 11)</i>	Представляемая информация логически не связана. Не использованы профессиональные термины.	Представляемая информация не систематизирована и/или не последовательна. Использован 1-2 профессиональных термина.	Представляемая информация систематизирована и последовательна. Использовано более 2 профессиональных терминов.	Представляемая информация систематизирована, последовательна и логически связана. Использовано более 5 профессиональных терминов.	
<i>Оформление презентации (min количество баллов – 6, max количество баллов – 10)</i>	Не использованы информационные технологии (например, PowerPoint). Больше 4 ошибок в представляемой информации.	Использованы информационные технологии (например, PowerPoint). 3-4 ошибки в представляемой информации.	Использованы информационные технологии (например, PowerPoint). Не более 2 ошибок в представляемой информации.	Широко использованы возможности информационных технологий (например, PowerPoint). Отсутствуют ошибки в представляемой информации.	
<i>Ответы на вопросы (min количество баллов – 5, max количество баллов – 8)</i>	Ответы на вопросы не получены.	Только ответы на элементарные вопросы.	Ответы на вопросы полные и/или частично полные.	Ответы на вопросы полные с приведением примеров и/или пояснений.	
<b>Итого баллов:</b>					
<b>Минимальное количество баллов :</b>					<b>60</b>
<b>Максимальное количество баллов:</b>					<b>100</b>

**Соответствие баллов шкале оценивания:**

<b>Количество баллов</b>	<b>Оценка обучающегося</b>
90-100	отлично
75-89	хорошо
60-74	удовлетворительно
менее 60	неудовлетворительно

Сформированность профессиональных и общих компетенций по профессиональному модулю определяется «зачтено» (основной вид деятельности освоен), «не зачтено» (основной вид деятельности не освоен).

«Зачтено» – обучающийся усвоил весь программный материал, в целом выполнил индивидуальное задание, не затрудняется с ответом на дополнительные вопросы, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок, демонстрируя удовлетворительный уровень профессиональной компетентности по профессиональному модулю.

«Не зачтено» – обучающийся не знает значительной части программного материала, не выполнил индивидуальное задание, допускает принципиальные ошибки при ответах на дополнительные вопросы, демонстрируя неудовлетворительный уровень профессиональной компетентности по профессиональному модулю.

#### **4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ**

Во время проведения лекционных занятий учитывается посещаемость обучающихся, оценивается их познавательная активность на занятии.

Тестирование по разделам профессионального модуля проводится преподавателем. Баллы переводятся в систему оценок преподавателем в соответствии с утвержденной шкалой оценивания.

В случае невыполнения лабораторных работ и тестовых заданий в установленные сроки обучающемуся необходимо погасить задолженность по невыполненным заданиям до проведения экзамена или зачета. График погашения задолженности устанавливается преподавателем в индивидуальном порядке с учетом причин невыполнения.

Допуск обучающегося к защите лабораторной работы происходит при условии наличия у обучающегося печатной версии (или электронной) отчета по лабораторной работе. Защита проходит в форме показа результатов работы и ответов на вопросы преподавателя.

По окончании освоения междисциплинарного курса профессионального модуля проводится промежуточная аттестация в виде зачета и дифференцированного зачета по МДК или экзамена по МДК, что позволяет оценить достижение результатов обучения по профессиональному модулю.

Перечень вопросов и список учебной литературы для подготовки к зачетам и экзамену предоставляется в начале семестра.

Во время сдачи промежуточной аттестации в устной форме в аудитории может находиться одновременно не более 4-5 обучающихся. Во время сдачи промежуточной аттестации в письменной форме в аудитории может находиться одновременно вся учебная группа. При выполнении заданий на компьютере – по одному обучающемуся за персональным компьютером (не более 12 студентов).

По окончании освоения профессионального модуля после производственной практики проводится квалификационный экзамен, в ходе которого проверяется уровень сформированности общих и профессиональных компетенций в рамках ПМ. В комиссию по проведению экзамена включаются преподаватели и представители работодателя. Численный состав комиссии – не менее 3-х человек.

Во время проведения квалификационного экзамена допускается присутствие в аудитории всех студентов группы. Экзамен проводится в аудитории, оснащенной презентационной техникой.