



**ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ГУМАНИТАРНО-МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ Г.КИЗИЛЮРТ**

Российская Федерация
Республика Дагестан,
368124, г. Кизилюрт,
ул. Вишневого, 170.

Тел.: +7(989) 476-00-15
E- mail: gmk.kizilurt@yandex.ru

ОДОБРЕНО

На заседании педагогического совета
ПОАНО «ГМК» г.Кизилюрт
Протокол № 6
от «06» сентябрь 2021г.

УТВЕРЖДАЮ

Директор
ПОАНО «ГМК» г.Кизилюрт

О.М.Гасанов
Приказ № 1-У
от «01» ноябрь 2021 г.

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

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

по специальности 09.02.07 «Информационные системы и программирование»
по программе базовой подготовки
на базе основного общего образования;
форма обучения – очная
Квалификация выпускника – программист

г.Кизилюрт 2021г.

Рабочая программа профессионального модуля «ПМ.01 Разработка модулей программного обеспечения для компьютерных систем» разработана на основе Федерального государственного образовательного стандарта и программы ППССЗ ПОАНО «Гуманитарно-многопрофильный колледж» по специальности среднего профессионального образования 09.02.07 «Информационные системы и программирование»

.

Организация-разработчик: ПОАНО «Гуманитарно-многопрофильный колледж» г.Кизилюрт.

Разработчик: преподаватель отделения Амиргамзаев Амиргамза Алисултанович

СОГЛАСОВАНО

Работодатель

ГБУ РД «Кизилюртовская ЦГБ»
368120, РД, г. Кизилюрт,
ул. Аскерханова 11
Главный врач

_____ П.Д.Шабанова
«___»_____ 2021 г.

СОГЛАСОВАНО

Работодатель

ГКУ РД СРЦН В МО «г.Кизилюрт»
368122, РД, г. Кизилюрт
ул. Садовая, д. 9
Директор

_____ С.И.Алибекова
«___»_____ 2021 г.

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	7
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	33
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)	34
5. АДАПТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ПРИ ОБУЧЕНИИ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	

1. ПАСПОРТ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ 01. Разработка модулей программного обеспечения для компьютерных систем

1.1. Требования к результатам освоения профессионального модуля

Результатом освоения программы профессионального модуля является овладение обучающимися видом деятельности «Разработка модулей программного обеспечения для компьютерных систем», в том числе профессиональными (ПК) и общими (ОК):

Код	Наименование результата обучения
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 11	Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

1.2. Цели и задачи профессионального модуля.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

иметь практический опыт:

– проведения профилактических мероприятий при осуществлении сестринского ухода.

уметь:

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;
- оформлять документацию на программные средства

знать:

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов

1.3. Рекомендуемое количество часов на освоение примерной программы профессионального модуля:

максимальной учебной нагрузки обучающегося – 779 часов,
обязательной аудиторной учебной нагрузки обучающегося – 516 часов,
самостоятельной работы обучающегося – 274 часа,
учебной практики – 72 часа,
производственной практики (по профилю специальности) – 216 часов.

3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ 01. Разработка модулей программного обеспечения

3.1. Тематический план профессионального модуля

Коды профессиональных компетенций	Наименования разделов профессионального модуля*	Всего часов (макс. учебная нагрузка и практики)	Объем времени, отведенный на освоение междисциплинарного курса (курсов)					Практика	
			Обязательная аудиторная учебная нагрузка обучающегося			Самостоятельная работа обучающегося		Учебная практика, часов	Производственная практика (по профилю специальности), часов (если предусмотрена рассредоточенная практика)
			Всего, часов	в т.ч. лабораторные занятия и практические занятия, часов	в т.ч. курсовая работа (проект), часов	Всего, часов	в т.ч., курсовая работа (проект), часов		
1	2	3	4	5	6	7	8	9	10
ПК 1.1 ПК 1.2 ОК 1 – ОК 10	МДК. 01.01. Разработка модулей программного обеспечения	142	96	44	-	46	-	-	-
ПК1.3, ПК 1.4, ПК 1.5 ОК 1 – ОК 10	МДК. 01.02. Поддержка и тестирование программных модулей	158	106	42	-	52	-	-	-
ПК 1.2, ПК 1.6 ОК 1 – ОК 10	МДК. 01.03. Разработка мобильных приложений	148	96	32	-	63	-	-	-
ПК 1.2, ПК 1.3 ОК 1 – ОК 10	МДК. 01.04. Системное программирование	331	218	40	-	113	-	-	-
ПК 1.1 – ПК 1.6 ОК 1 - ОК 11	Учебная практика	72	-	-	-	-	-	-	-
ПК 1.1 – ПК 1.6 ОК 1 – ОК 11	Производственная практика (по профилю специальности), часов	216	-	-	-	-	-	72	216
	Всего:	1067	516	158	-	274	-	72	216

**3.2. Содержание обучения по профессиональному модулю
ПМ 01. Разработка модулей программного обеспечения**

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов (теор/практ)	Уровень освоения
МДК. 01.01. Разработка программных модулей		96 (44/52)	
Тема 1.1.1 Жизненный цикл ПО	Содержание учебного материала (лекции) Жизненный цикл программного обеспечения Этапы разработки программного обеспечения Стратегии и модели жизненного цикла	2	1
	Практическое занятие. Практическое занятие 1 «Оценка сложности алгоритмов сортировки» Практическое занятие 2 «Оценка сложности алгоритмов поиска» Практическое занятие 3 «Оценка сложности рекурсивных алгоритмов» Практическое занятие 4 «Оценка сложности эвристических алгоритмов»	8	2
	Самостоятельная работа. Закрепление умений составления алгоритмов	6	3
<i>Тема 1.1.2 Структурное программирование</i>	Содержание учебного материала (лекции) Переменные и типы данных. Переменные. Ключевые слова. Константы. Преобразование значений типов (кастинг). Арифметические операторы. Области видимости. Использование ключевых слов как идентификаторов. Проверка и запрет проверки переполнения. Сцепление строк. Форматированный вывод. Флаги форматирования. Неявно типизированные переменные. Сравнение значений разных типов. Условные конструкции. Условная конструкция if. Условная конструкция if – else. Тернарный (третичный) оператор. Условная конструкция switch – case. Логика. Логические операции. Конъюнкция. Дизъюнкция Исключающее или. Отрицание. Битовые логические операции. Побитовое отрицание. Двоичная арифметика	4	1

	Примеры использования логических операций. Побитовые логические операции. Работа с портами (конъюнкция и дизъюнкция). Исключающее ИЛИ (пример использования). Логические операции. Операции сдвига. Короткозамкнутые вычисления. Теоремы Де Моргана Циклические конструкции. Схема работы циклов. Оператор безусловного перехода (goto). Цикл с предусловием (while). Использование циклической конструкции while. Цикл с постусловием (do-while) Примеры использования цикла do-while. Цикл со счетчиком (for). Использование циклической конструкции for. Вложенный цикл for. Алгоритм Дейкстры (Цикл Дейкстры. Цикл Паук. Упрощенный цикл Паук. Бесконечные циклы Методы. Понятие подпрограмм. Создание методов. Функции и процедуры. Примеры создания методов Пример правильного множественного возврата из метода. Использование сторожевых операторов Методы с изменяемыми параметрами. Методы с выходными параметрами. Перегрузка. Аргументы (параметры). Использование именованных параметров. Перегрузка методов. Именованные параметры Использование опциональных параметров..Метод Main () Перегрузка метода Main (). Рекурсия. Сложная рекурсия. Примеры использования рекурсии Массивы. Индекс массива. Использование одномерных массивов. Создание одномерных массивов. Двумерные массивы. Использование двумерных массивов. Массивы, состоящие из одного элемента. Трехмерные массивы. Использование трехмерных массивов. Четырехмерные массивы. Зубчатые массивы. Использование зубчатых массивов. Ключевое слово params		
	Практическое занятие. Практическая работа №5. Условные конструкции Практическая работа №6. Циклические конструкции Практическая работа №7 Методы Практическая работа №8. Массивы Практическая работа №9. Классы Практическая работа №10 Абстрактные классы Практическая работа №11. Статические классы Практическая работа №12. Структуры Практическая работа №13. События (events) Практическая работа №14. Обработка исключений Практическая работа №15. Перегрузка операторов	16	2

	Практическая работа №16. Обработка текстовой информации Практическая работа №17. Файлы конфигурации. Работа с реестром Практическая работа №18. Рефлексия Практическая работа №19. Управление памятью. Практическая работа №20. Полиморфизм. Версионность Практическая работа №21. Синхронизация при помощи объектов ядра		
	Самостоятельная работа. Закрепление умений по использованию массивов, создание методов	6	3
Тема 1.1.3 Объектно-ориентированное программирование	Содержание учебного материала (лекции) Введение в ООП, классы и объекты. Понятие класса. Объект и экземпляры. Примеры создания классов Использование методов доступа к закрытым полям. Использование свойств для доступа к закрытым полям. Свойства с одним методом доступа. Конструктор. Конструкторы, вызывающие другие конструкторы Передача экземпляра класса в качестве аргумента. Автоматически реализуемые свойства. Создание экземпляра по слабой ссылке. Стили использования классов. Инкапсуляция. Классы и объекты. Диаграммы классов. Частичные классы. Частичные методы. Работа с инкапсуляцией Поля только для чтения. Язык UML. Диаграммы классов. Ассоциация. Разновидности UML диаграмм Самоассоциация. Использование техники фабричных методов Наследование и полиморфизм. Наследование. Модификаторы доступа. Связанность и связность Работа с конструктором. Приведение к базовому типу. Полиморфизм. Замещение метода базового класса. Переопределение метода базового класса. Операторы is и as. Герметизированные классы. Герметизированные методы. Подмена объектов Абстракция. Абстрактные классы и интерфейсы. Понятие абстракции. Абстрактные классы. Ключевое слово abstract. Использование абстрактных классов и методов. Абстрактные методы. Интерфейсы Примеры использования интерфейсов. Множественное наследование абстракции. Наследование интерфейса от интерфейса. Наследование от интерфейсов. Отличие между классом и типом. Абстракция. Абстрактные классы и интерфейсы. Понятие абстракции. Абстрактные классы. Ключевое слово abstract. Использование абстрактных	8	1

	классов и методов. Абстрактные методы. Интерфейсы Примеры использования интерфейсов. Множественное наследование абстракции. Наследование интерфейса от интерфейса. Наследование от интерфейсов. Отличие между классом и типом Статические и вложенные классы. Статические члены. Статические члены в нестатических классах Константы. Статические поля и свойства. Статические конструкторы. Статические члены в абстрактных классах. Статические классы. Паттерн проектирования Singleton. Расширяющие методы. Вложенные классы. Техника Делегирования Структуры и их разновидности. Структура .Примеры работы со структурами. Конструкторы в структурах Пример плохого и хорошего дизайна .Стек..Куча. Расположение структур в стеке. Расположение структур на куче. Наследование в структурах. Вложенные структуры и классы Перечисления (enum). Вступление. Структуры. Упаковка и распаковка. Структура DateTime. Перечисления. Получение информации о типе элементов. Форматированный вывод элементов перечисления Поиск элемента перечисления по имени константы. Получение информации об элементах. Сравнение элементов перечисления Универсальные шаблоны (generics). Создание универсальных шаблоновПараметризованный методПараметризованный делегатКовариантность обобщенийКонтрвариантность обобщенийЧастичные классы и методыТип NullableОперация поглощения Ограничения универсальных шаблонов . Ограничения параметров типа. Ограничения параметров типа – «naked». Работа с коллекциями. Словари. Представление рефлексора События (events) Понятие событий.Работа с событиями. Абстрактные и виртуальные события. Анонимные методы в событиях. Использование событий. Шаблон MVP Потоки (threads). Многозадачность. Пространство имен System.Threading.Работа с потоками. Использование одного метода в двух потоках. Передача данных в поток. Анонимные методы. Основные и фоновые потоки. Техника синхронизации доступа к ресурсу. Критическая секция Коллекции. Понятие коллекций. Работа с коллекциями. Цикл foreach. Оператор yield. Восстановление кода из под рефлексора Обработка исключений (exception)		
--	---	--	--

	Обработка исключительных ситуаций. Таблица исключений. Конструкция try-catch. Ключевое слово throw. Разработка своих пользовательских исключений. Конструкция try-catch-finally. Обработка внутренних исключений. Порядок обработки исключений Перегрузка операторов Базовый класс Object. Метод ToString. Метод GetHashCode. Метод Equals. Метод ReferenceEquals. Метод GetType. Техника клонирования. Клонирование графов наследования. Клонирование ассоциаций. Паттерн «Prototype». Операторы языка C#. Перегрузка операторов Анонимные и динамические типы. LINQ Анонимные типы. Вложенные анонимные типы Язык LINQ. Примеры использования языка LINQ. Вложенные операторы from. Операция объединения join Локальный идентификатор let. Использование оператора group. Оператор into. Динамические типы данных Наследование динамических типов Пространства имен. Директивы препроцессора. Пространства имен. Вложенные пространства имен. Видимость во вложенных пространствах имен. DLL библиотеки. Создание DLL библиотек. Модификаторы доступа internal и internal protected. Директивы препроцессора Пользовательские коллекции Коллекции. Интерфейсы IEnumerable, IEnumerator. Работа цикла foreach. Оператор yield Интерфейсы ICollection и ICollection. Создание простой коллекции с помощью реализации интерфейса IList Рассмотрение коллекции созданной с использованием паттерна Iterator Системные коллекции Использование ArrayList и сбор элементов данных. Предназначение и использование интерфейса IComparerРабота с последовательными списками (Queue, Stack)Работа со словарями (Hashtable, SortedList, ListDictionary, HybridDictionary, OrderedDictionary)Роль интерфейса IEqualityComparerПрименение специализированных наборовРабота с двоичными значениями с помощью BitArray и BitVector32Использование методов CreateMask и CreateSectionХранение строк в наборах (StringCollection, StringDictionary, NameValueCollection)Обобщенные наборы Программирование ввода-вывода Навигация по файловой системе. Чтение и запись файлов. Работа с потоками.		
--	---	--	--

	Работа с изолированным хранилищем Работа с текстом. Регулярные выражения Работа со строками. Форматирование вывода. Локализация и глобализация. Кодирование-декодирование строк. Использование класса Encoding. Использование класса StringBuilder. Регулярные выражения и пространство System.Text.RegularExpressions. Метасимволы и Квантификаторы. Метод Regex.Replace и использование внутришаблонных переменных. Организация поиска по шаблону. Применение Match Построение регулярных выражений. Чтение и запись файлов в разных кодировках XML. Файлы конфигурации. Реестр Загрузка и чтение из XML файла. Применение XmlTextReader. Загрузка содержимого XML по URL Загрузка содержимого XML из строки. Поэлементное чтение XML файла. Использование XmlTextWriter Запись комментариев в XML файл. Запись пространства имен и префиксов в XML файл. Запись в XML файл со специальным форматированием. Выборка из XML с помощью XPath. (Запросы XPath). Вычисление min и max с помощью XPath. Создание XPathDocument из строки. Создание файлов конфигурации приложения. Чтение и запись настроек приложения. Работа с реестром Windows. Использование реестра для хранения настроек приложения Рефлексия Общее понятие рефлексии и отражения. Метаданные, манифест сборки, объект Type. Позднее связывание. Класс Assembly. Динамическая генерация кода. Класс Activator. Генерация кода во время выполнения программы Атрибуты Общее понятие атрибутов. Класс System.Attribute. Позиционные и именованные параметры атрибутов Создание атрибутов. Использование атрибутов. Определение атрибутов при помощи рефлексии. Системные атрибуты Сериализация Общее понятие сериализации. Процесс сериализации и десериализации. Совместимость версий. Управление сериализацией. Форматы сериализации. XML сериализация. Binary сериализация. SOAP сериализация Собственные методы сериализации Управление памятью. Сборщик мусора Время жизни объекта в памяти. Сборщик мусора. Поколения объектов.		
--	--	--	--

	Деструкторы и финализаторы Освобождение неуправляемых ресурсов. Интерфейс IDisposable Полиморфизм. Версионность Шаблон NVI. Классический полиморфизм. Ad hoc полиморфизм. Замещение, переопределение и перекрытие методов Потоки Многозадачность. Многозадачность на основе процессов. Многозадачность на основе потоков Работа с потоками в .NET Framework: пространство имен System.Threading, класс Thread. Создание потоков. Состояния потоков. Завершение потоков. Определение момента окончания потока. Передача аргументов потоку. Использование IsBackground. Приоритеты потоков. Классы Interlocked, Monitor. Синхронизация потоков. Сообщение между потоками. Методы Pulse(), Wait(), PulseAll () Синхронизация при помощи объектов ядра Конструкций пользовательского режима. Волатильные конструкции (volatile). Взаимозапирающие конструкции (interlocked). Конструкции синхронизации режима ядра. События AutoResetEvent и ManualResetEvent. Семафоры Semaphore. Мьютексы Mutex Асинхронная модель Программирования Пул потоков. Шаблон асинхронного вызова методов. Асинхронная природа делегатов Интерфейс IAsyncResult. Синхронизация вызывающего потока Делегат AsyncCallback. Класс AsyncResult. Передача и прием специальных данных состояния		
	Практическое занятие. Практическая работа №22. Обработка данных в табличном виде Практическая работа №23. Разработка приложения Браузер Практическая работа №24. Сохранение/чтение настроек приложения, используя Ini файлы. Практическая работа №25. Сохранение/чтение настроек приложения, используя реестр Windows. Практическая работа №26. Обработка строковой информации Практическая работа №27. Обработка графической информации Практическая работа №28. Технология Drag-And-Drop при разработки программных продуктов Практическая работа №29. Работа с файловым каталогом	6	2

	Практическая работа №30. Работа с приложениями Microsoft Office Практическая работа №31. Разработка почтового клиента Практическая работа №32. Форма и ее свойства Практическая работа №33. Диалоговые окна и сообщения Практическая работа №34. Однострочные текстовые редакторы Практическая работа №35. Элементы переключателей и выбора		
	Самостоятельная работа. Построение регулярных выражений. Чтение и запись файлов в разных кодировках	6	3
Тема 1.1.4 Паттерны проектирования	Содержание учебного материала (лекции) Назначение и виды паттернов. Шаблоны и их использование в объективной реальности. Паттерны в спорте. Фигуры логики и модусы силлогизмов. Шаблоны в музыке. Порождающие шаблоны в музыке Структурные шаблоны в музыке. Поведенческие шаблоны в музыке. Шаблоны в программировании Определение Паттерна. Формат описания паттернов проектирования. Каталог паттернов проектирования Техника ООП. Фасад Подсистемы. Принципы организации каталога. Моделирование и подходы проектирования. Язык моделирования UML. Набор используемых диаграмм UML. Работа с Visual Studio Создание простейших диаграмм. Класс (характеристики). Пакеты (характеристики). Наследование. Связи отношений между классами. Все основные связи отношений. Package import Диаграммы последовательностей. Самостоятельное Моделирование. Прямое и обратное проектирование	2	1
	Практическое занятие. «Использование основных шаблонов» Практическая работа 36 «Использование порождающих шаблонов» Практическая работа 37 «Использование структурных шаблонов» Практическая работа 38 «Использование структурных шаблонов»	4	2
	Самостоятельная работа: Создание простейших диаграмм	6	3
Тема 1.1.5. Событийно-управляемое программирование	Содержание учебного материала (лекции) Основы разработки оконных приложений. Принципы визуального программирования Инструментальные средства, их функциональное назначение и состав Содержание файлов проекта создаваемых при проектировании и при	6	1

	компиляции Структура кода основного файла приложения, модуля общего назначения и файла формы Настройки выполняемого файла. Работа с формами. Принцип построения и состав библиотеки визуальных компонент Обзор основных компонентов Способы размещения компонентов Правила формирования имен объектов Основные настройки (свойства) Методы компонент События и коды обработчиков Разъяснения типовых событий каждого источника Классификация источников событий Объединение форм Дочерние формы Диалоговые сообщения. Сообщения и их классификация Достоинства и недостатки каждого вида сообщений Подпрограммы реализующие диалоговые окна Односторонние текстовые редакторы. Однострочные текстовые редакторы. Преобразование типов данных Маска ввода. Защита от ошибок ввода данных. Обработка входных данных Элементы переключателей. Элементы переключателей. Компоненты выбора. Контейнеры для хранения объектов Вкладки. Компоненты выбора цвета Обработка табличной информации. Обработка данных в табличном виде. Способы обращения к таблицам Главное и контекстное меню. Главное и контекстное меню. Основные свойства и события Командные кнопки, кнопки с растровыми изображениями и быстрого доступа Объекты для создания браузера Ini-файлы и реестры. Инициализация и запуск приложения. Настройки параметров всплывающей подсказки. Ini файлы. Сохранение данных. Работа с реестром Заголовок секции, параметры, хранимые в секциях. Чтение и сохранение		
--	---	--	--

	параметров Многострочные текстовые редакторы. Многострочные редакторы. Обработка строковой информации. Методы для работы со строками. Контроль вводимых символов Диалоговые окна Работа с изображением. Компоненты для работы с графикой. Обработка графической информации Использование технологии перетаскивания и встраивание объектов. Drag&Drop. Методы и свойства для перемещения объектов. События управления манипулятора "Мышь" Работа с файловым каталогом. Диалоговые окна. Работа с файловым каталогом. Объекты для работы с проводником Работа с внешними приложениями. Запуск внешних приложений Особенности, методы использования и параметры. Организация работы с приложениями Microsoft Office Организация работы с электронной почтой. Сетевые протоколы прикладного уровня. Методы и свойства для работы и передачи данных по сети		
	Практическое занятие. Практическая работа 39 «Разработка приложения с использованием текстовых компонентов» Практическая работа 40 «Разработка приложения с несколькими формами» Практическая работа 41 «Разработка графических объектов» Практическая работа 42 «Работа с графическим редактором» Практическая работа 43 «Разработка приложения с невизуальными компонентами» Практическая работа 44 «Разработка игрового приложения» Практическая работа 45 «Разработка приложения с анимацией» Практическая работа 46 «Воспроизведение анимации различных форматов»	6	2
	Самостоятельная работа: Разработка приложения и графических объектов	6	3
Тема 1.1.6 Оптимизация и рефакторинг кода	Содержание учебного материала (лекции) Введение в рефакторинг. Рефакторинг. Дублирование кода. Длинные методы. Большой класс Длинный список параметров. Расходящиеся модификации	4	1

	Стрельба дробью. Завистливые функции. Группы данных Одержимость элементарными типами. Операторы типа switch Параллельные иерархии наследования. Ленивый класс. Теоретическая общность. Временное поле. Цепочки сообщений. Посредник. Неуместная близость. Альтернативные классы с разными интерфейсами. Неполнота библиотечного класса. Класы данных. Отказ от наследства. Комментарии Составление методов. Рассматриваемые техники рефакторинга. Выделение метода. Встраивание метода. Встраивание временной переменной. Замена встраиваемой переменной вызовом метода. Введение поясняющей переменной. Расщепление временной переменной. Замена метода объектом методов. Перемещение метода. Перемещение поля. Выделение класса. Соккрытие делегирования. Удаление посредника Организация данных. Замена значения данных объектом. Замена значений ссылкой. Замена однонаправленной связи двунаправленной. Инкапсуляция коллекции. Замена кода типа подклассом. Замена кода типа состоянием/стратегией. Замена подкласса полями Упрощение условных выражений. Декомпозиция условного оператора. Консолидация условного выражения Консолидация дублирующихся условных фрагментов. Удаление управляющего флага. Замена условного оператора полиморфизмом. Введение объекта Null Решение задач обобщения. Подъем поля. Подъем метода. Спуск метода/поля. Выделение подкласса. Выделение родительского класса. Выделение интерфейса. Замена наследования делегированием. Замена делегирования наследованием		
	Практическое занятие. Практическая работа 47 «Оптимизация и рефакторинг кода» Практическая работа 48 «Применение рефакторинга кода»	2	2
	Самостоятельная работа. Изучение возможностей графического редактора и программирование графических 2D объектов	6	3
	Тема 1.1.7 Разработка пользовательского интерфейса Содержание учебного материала (лекции) Правила разработки интерфейсов пользователя. Разработка интерфейса пользователя	2	1

	Практическое занятие. Практическая работа 49 «Планирование и разработка макета интерфейса программного комплекса» Практическая работа 50 «Разработка интерфейса программного приложения»	2	2
	Самостоятельная работа Решение тестовых заданий. Составление опорного конспекта	6	3
Тема 1.1.8 Основы ADO.Net	Содержание учебного материала (лекции) Архитектура технологии ADO.NET. Рассмотрение архитектуры ADO.NET. Рассмотрение «Создание подключения». Пример. Подключение к БД. Создание БД и зарегистрированного пользователя. Подключение зарегистрированного пользователя. Подключение с SqlConnectionStringBuilder. Рассмотрение «Пул соединений». Использование файлов конфигурации. Использование поставщика данных .NET OleDb. Получение данных. Объект SqlCommand. Создание команд с использованием SqlCommand. Получение данных.. Объект SqlDataReader. Выполнение пакетных запросов. Примеры: Создание SqlCommand, выполнение команд. Пример. Выполнение вставки и удаления. врат данных в табличном представлении. троготипизированные средства доступа. Получение данных с помощью индексатора. Асинхронное выполнение команд. Пример. Работа со значениями NULLРассмотрение «Транзакции». Создание параметризованных запросов. Использование коллекции Parameters. Пример. Вызов хранимой процедуры. Процедура, принимающая параметры. Возвращающие значение Автономная часть архитектуры. Объекты DataColumn, DataTable. Таблицы ADO.NET. Объект Data Column и DataRow.Data. Reader для создания схемы DataTable Создание экземпляров DataColumn. Создание строки для таблицы. Создание новой таблицы. Получение информации о схеме таблицы. Проверка данных Пример. Использование свойства Readonly. Использование свойства AllowDBNull Пример. Использование свойства MaxLength. Пример. Использование свойства Unique. Использование свойства UniqueConstraint. Ограничение на уникальность PrimaryKey Классы DataRow, DataSet и DataAdapter. Рассмотрение «Строк(DataRow)». Использование свойства RowState объекта	6	1

	DataRow. Пример. Значение Unchanged свойства RowState. Пример. Значение Delete свойства RowState. Пример. Значение Modified свойства RowState. Пример. Метод SetAdded, Метод SetModified. Пример. EditingRow. Применение перечисления DataRowVersion. Пример. Proposed версия строки. Данные строки, подготовленной к удалению. Рассмотрение «Адаптера данных». Пример. Использование «маппинга». Пример. MissingSchema. Получение данных по частям (GetPartOfRows). Получение данных по частям в страницах Работа с реляционными данными. Объект DataRelation. Получение связанных данных. Пример. Создание отношений между таблицами. Просмотр дочерних строк. Метод GetChildRows. Пример. Метод GetParentRow. Пример. Получение данных при связи сам к себе. Получение данных при связи сам к себе, рекурсивный метод. Пример. Запись схемы и данных БД в XML файл. Чтение схемы и данных БД. Метод выводящий данные. Пример. Создание рассчитываемого столбца. Правила удаления и обновления родительских строк. Создание правил для удаления и обновления данных Связанная строка, подготовленная к удалению Фильтрация изменений и сортировка данных. Рассмотрение объекта «DataView». Пример. Поиск данных с помощью метода Find. Пример. Поиск данных по указанному фильтру. Пример. Поиск данных по указанному шаблону. Поиск данных с использованием метода select. Пример. Сортировка с помощью метода select. Пример. Создание DataView. Пример. Метод FindMethod.. Пример. Создание таблиц с помощью представления. Пример. Создание LINQ запросов для объектов DataTable. Пример. Работа с методом AsDataView. Перекрестные запросы между таблицами. Пример. Работа с внутренним объединением. Пример. Реализация Left outer join в LINQ Объект DataSet со строгим контролем типов. Создание строго типизированного объекта DataSet. Создание DataSet и добавление строк. Пример. Работа со строками stroged DataSet. Поиск данных в DataSet. Пример. Работа со значениями NULL. Метод Merge для слияния таблиц. Работа с реляционными данными. Пример. Получение данных Пример. Использование метода GetChanges. Работа с адаптерами таблиц Метод GetData. Вставка данных помощью метода Insert Передача обновлений. Добавление (вставка) данных с формы в таблицу Изменение данных в таблице Удаление данных с таблицы Добавление, изменение и удаление данных Метод Updates использование SqlDataAdapter Рассмотрение		
--	---	--	--

	«Обновление данных»Метод Updates использование SqlDataAdapter		
	Практическое занятие Практическая работа 51 «Создание приложения с базами данных» Практическая работа 52 «Создание запросов к базам данных» Практическая работа 53 «Создание хранимых процедур» Практическая работа 54 «Создание фильтров и триггеров»	2	2
	Самостоятельная работа Разработка приложения базы данных. Создание пользовательских функций. Создание диаграмм и их редакция	4	3
МДК 01.02 Поддержка и тестирование программных модулей		106 (42/64)	
<i>Тема 1.2.1 Отладка и тестирование программного обеспечения</i>	Содержание учебного материала (лекции) Введение в тестирование программного обеспечения Понятие качества ПО. Стандарты качества ПО. Атрибуты и характеристики качества ПО. Основные определения тестирования. Цели и задачи процесса тестирования. Полный цикл тестирования. Фазы тестирования. Методы и виды тестирования. Анализ требований к ПО Методы и виды тестирования. Общий обзор. Критерии покрытия тестирования. Требования к ПО. Анализ требований с точки зрения пригодности к тестированию. Тестовая документация. Тест-план, тест-дизайн Документы, создаваемые в процессе тестирования. Тест план. Связь тестовых планов с другими типами документов. Тест – дизайн. Возможные формы подготовки тест-дизайна. Тестовая документация. Test Case. Отчет о прохождении тестов Определение Test Case. Правила написания, степень детализации, независимость. Правила описания дефектов, понятие важности, приоритета. Ведение системы отслеживания дефектов. Составление отчетов по результатам тестирования. Учебный проект: создание test cases для учебной программы. Методы Покрытие входных данных. Допустимые и недопустимые данные. Эквивалентное разбиение. Анализ граничных значений. Парное комбинирование. Предположение ошибок. Техники тестирования	14	1

	Техники, базирующиеся на интуиции и опыте инженера. Техники, базирующиеся на спецификации. Техники, ориентированные на код Тестирование, ориентированное на дефекты. Техники, базирующиеся на условиях использования. Техники, базирующиеся на природе приложения Выбор и комбинация различных техник Уровни тестирования. Критерии покрытия кода программы тестами Модульное тестирование. Драйверы. Заглушки. Интеграционное тестирование. Способы интеграционного тестирования. Системное тестирование. Понятие покрытия кода тестами. Критерии покрытия. Метрика покрытия. Анализ покрытия. Виды тестирования: функциональное и нефункциональное тестирование Функциональные виды тестирования. Тестирование безопасности, тестирование взаимодействия. Нефункциональные виды тестирования. Тестирование производительности. Нагрузочное тестирование. Тестирование пользовательского интерфейса (GUI). Тестирование web-приложений Задачи и цели тестирования пользовательского интерфейса. Функциональное тестирование пользовательского интерфейса. Тестирование удобства пользовательского интерфейса. Тестирование web-приложений Регрессионное тестирование Регрессионное тестирование. Подходы к составлению набора test cases. Жизненный цикл ПО. Каскадный, спиральный жизненные циклы. Методологии разработки ПО. MSF, RUP, Экстремальное программирование. Команда тестирования. Роли.		
	Практическое занятие Практическая работа №1 «Тестирование программ методом «Белого ящика». Способ тестирования базового пути» Практическая работа №2 «Тестирование условий» Практическая работа №3 «Тестирование циклов» Практическая работа №4 «Тестирование ветвей и операторов отношений» Практическая работа №5 «Тестирование потоков данных» Практическая работа №6 «Тестирование программ методом «Чёрного ящика». Способ анализа граничных условий» Практическая работа №7 «Тестирование программ методом «Чёрного ящика». Способ разбиения по эквивалентности» Практическая работа №8 «Тестирование программ методом «Чёрного ящика». Способ диаграмм причин-следствий»	20	2

	Самостоятельная работа. Подготовить сообщения по темам: «Виды тестирования», «Регрессионное тестирование», «Драйверы», «Покрытие входных данных», «Уровни тестирования», «Пользовательский интерфейс»	17	3
Тема 1.2.2 Командное тестирование	Содержание учебного материала (лекции) Процессы тестирования Место тестирования в процессе разработки ПО. Тестирование и качество. Оценка качества продукта. Метрики. Базовые принципы тестирования Планирование работ по тестированию. Виды деятельности, осуществляемые при составлении плана тестирования. Артефакты тестирования Модели зрелости тестирования ПО (TMMi). Инструментарий тест менеджера Риски тестирования. Команда тестирования Создание условий работы в команде. Стили управления. Выстраивание отношений. Коммуникации. Постановка задач. SMART. Хронофаги –поглотители времени Оперативное планирование. Приоритет задач. Матрица Эйзенхауэра. Работа с почтой Оценка трудозатрат на тестирование. Определение задач, которые должны быть выполнены. Оценка трудоемкости задач. Эмпирическое правило Брукса. Практические соображения. Метод анализа видов ошибок и их влияния (FMEA). Упрощенный вариант.	14	1
	Практическое занятие. Практическая работа №9 «Модульное тестирование» Практическая работа №10 «Интеграционное тестирование» Практическая работа № 11 «Проведение автоматизированного тестирования» Практическая работа № 12 «Выявление несоответствия результата выполнения модуля его спецификации» Практическая работа № 13 «Реализация процедуры поиска некачественного кода программного продукта» Практическая работа № 14 «Составление тестовых сценариев» Практическая работа № 15 «Разработка системы тестов на основе потока управления» Практическая работа № 16«Разработка системы тестов на основе потока данных» Практическая работа № 17 «Тестирование программного продукта по ранее определенному сценарию» Практическая работа № 18 «Тестирование безопасности программных продуктов» Практическая работа № 19 «Регрессионное тестирование программного продукта» Практическая работа № 20 «Тестирование производительности программного продукта» Практическая работа № 21 «Ручное тестирование. Генерация тестов»	34	2

	Практическая работа № 22 «Документирование результата тестирования программного продукта» Практическая работа №23 «Автоматизированное тестирование программного продукта» Практическая работа №24 «Тестирование пользовательского интерфейса» Практическая работа №25 «Отладка программного продукта»		
	Самостоятельная работа. Построение матриц	18	3
<i>Тема 1.2.3 Документирование</i>	Содержание учебного материала (лекции) Unit тестирование. Использование MSTest. Понятие Unit Test Типы тестирования. Интеграционное тестирование Верификация и валидация. Свойства хорошего Unit теста Расположение Unit тестов. Именованые проектов. Именованые методов и классов. Какой код тестировать Unit Test Frameworks. Подход AAA. Пример создания простого Unit теста Более сложный пример со значением определяющим сложность пароля Работа атрибутов TestInitialize и ClassInitialize. Класс Assert Свойство TextContext. Data-driven test. Ordered и Gineric тесты Атрибут DeploymentItem. Как заставить себя писать Unit тесты. Польза Unit тестирования. Test Driven Development Использование Stub объектов. Виды Unit тестов Виды тестовых объектов. Inversion of Control (IOC) Шаблон Service Locator. Шаблон Dependency Injection. Constructor Injection Property Injection. Interface Injection. Local Factory Method. Extract and Override Factory Injection. Проблемы инкапсуляции Использование Mock объектов. Moq Framework Mock-объекты. Разница между Mock и Stub. Библиотека moq. Пример с библиотекой Logging Library. Настройка Mock-объектов когда мы используем Moq framework out и ref параметры. Как проверять аргументы передаваемые в mock-объект. Как можно контролировать выполнение приложения Исключения. Как создавать mock'и для свойств. Как создавать stub'ы для свойств. Как создавать mock-объекты которые инициализируют события Использование фабрики mock'ов и режимы создания mock'ов	14	1
	Практическое занятие. Практическая работа №26 «Оформление документации на программный модуль с использованием инструментальных средств»	10	2

	Самостоятельная работа. Создание проектов	17	3
МДК 01.03 Разработка мобильных приложений		96 (32/64)	
Тема 1.3.1 Основные платформы и языки разработки мобильных приложений	Содержание учебного материала (лекции) Введение в Java технологии Особенности языка программирования Java. Описание Java технологий. Использование интегрированной среды разработки. Введение в язык программирования Java Языковые лексемы Java.. Введение в систему типов языка Java.. Работа с примитивными типами и константами.. Операции языка Java.. Преобразование простых типов. Методы и операторы Java Создание и вызов методов.. Перегрузка и методы с переменным числом аргументов.. Операторы ветвления.. Операторы циклов. Создание и использование массивов Java Одномерные массивы. Многомерные массивы. Работа с массивами и класс Arrays. Работа со строками в Java Основные методы класса String. Сложение и преобразование строк. Классы динамических строк. Ввод данных с клавиатуры и класс Scanner. Регулярные выражения. Техники тестирования Индекс массива. Использование одномерных массивов. Создание одномерных массивов. Двумерные массивы. Использование двумерных массивов. Массивы, состоящие из одного элемента. Трехмерные массивы. Использование трехмерных массивов. Четырехмерные массивы. Зубчатые массивы Использование зубчатых массивов. Ключевое слово params Разработка классов на Java Обзор основных концепций ООП.. Объявление класса. Члены класса и создание объектов класса. Модификаторы доступа. Модификаторы final & static.. Использование пакетов, директив импорта и переменной среды CLASSPATH Наследование и полиморфизм Наследование как механизм повторного использования кода. Конструктор при наследовании. Преобразование типов и операция instanceof. Виртуальные методы и позднее связывание. Абстрактные классы и методы.	16	1

	Пакет java. lang Класс Object и переопределение его методов. Метаданные и рефлексия. Классы System и Math. Обработка ошибок в Java Концепция исключений в Java. Использование операторов try, catch и finally. Проверяемые и непроверяемые исключения. Создание своих классов исключений. Оператор try для освобождения ресурсов. Потоки данных в Java Обзор классов потоков. Работа с байтовыми потоками. Работа с потоками символов. Использование класса java.io.File. Сжатие файлов. Сериализация объектов в Java. Работа с файловой системой в NIO 2 Использование интерфейса Path.. Работа с атрибутами файлов. Основные возможности класса Files. Использование класса Files для обхода дерева каталогов. Мониторинг изменений в файловой системе. Пакет java. util Форматирование данных. Работа с датой и временем. Класс Locale и глобализация кода. Локализация и класс ResourceBundle. Генерация псевдослучайных чисел Коллекции в Java Иерархия классов коллекций. Концепция параметризованных типов данных. Работа с параметризованным методов и интерфейсом. Обзор возможностей списков, множеств и словарей в Java. Вложенные классы в Java Внутренние классы. Вложенные классы. Анонимные классы. Перечисления в Java. Лямбда-выражения Синтаксис лямбда-выражений. Ссылки на методы. Функциональные интерфейсы. Паттерны проектирования Обзор паттернов. Паттерн одиночка. Паттерн композиция. Паттерн наблюдатель.		
	Практическое занятие. Практическая работа №1 «Установка инструментария и настройка среды для разработки мобильных приложений» Практическая работа №2 «Установка среды разработки мобильных приложений с применением виртуальной машины»	14	2

	Практическая работа № 3 «Инструменты разработки Windows Phone7-приложений. Пример простейших программ Windows Phone 7-приложения. Запуск приложения на эмуляторе». Практическая работа №4 «Microsoft Silverlight. Основы работы с сенсорным вводом. Обработка нескольких касаний» Практическая работа №5 «Акселерометр и служба определения местоположения. Использование pivot и panorama».		
	Самостоятельная работа. Работа с файловыми системами	31	3
<i>Тема 1.3.2 Создание и тестирование модулей для мобильных приложений</i>	Содержание учебного материала (лекции) Обзор платформы Android История возникновения Android. Преимущества Android. Архитектура Android. Особенности платформы Android. Основные компоненты Android. Безопасность и полномочия (Permissions). Установка и настройка компонентов среды разработки Активности и ресурсы Активности (Activity) в Android. Создание Активности. Жизненный цикл Активности. Стеки Активностей. Состояния Активностей. Отслеживание изменений состояния Активности. Ресурсы. Отделение ресурсов от кода программы. Создание ресурсов. Простые значения. Визуальные стили и темы Изображения. Разметка. Анимация. Меню Приложения и пользовательский интерфейс Использование внешних ресурсов в коде приложения. Использование ресурсов внутри ресурсов. Локализация приложения с помощью внешних ресурсов Класс Application. Обработка событий жизненного цикла приложения Понятие контекста. Пользовательский интерфейс. Представления (View) Разметка (Layout) Намерения, меню и работа с данными Адаптеры в Android. Использование Адаптеров для привязки данных Намерения в Android. Использование Намерений для запуска Активностей Неявные намерения. Сохранение состояния и настроек приложения Общие Настройки (Shared Preferences). Работа с файлами Использование статических файлов как ресурсов. Меню в Android Дочерние и контекстные меню. Описание меню с помощью XML СУБД, контент-провайдеры и использование сетевых сервисов Работа с базами данных в Android. Курсоры (Cursor) и ContentValues. Работа с СУБД SQLite. Работа с СУБД без адаптера. Особенности работы с БД в Android	16	1

	Выполнение запросов для доступа к данным. Изменение данных в БД Использование SimpleCursorAdapter. Контент-провайдеры. Использование контент-провайдеров. Создание контент-провайдеров. Использование интернет-сервисов Диалоги в Android Виды Диалогов. Рекомендации по дизайну Диалогов. Создание и удаление Диалогов. Обработка событий Широковещательные приемники (Broadcast Receivers) Применение Широковещательных Приемников. Жизненный цикл Приемника Регистрация Приемника. Использование Ordered Broadcast. Использование PendingIntent Извещения (Notifications) в Android Взаимодействие с Извещениями. Управление Извещениями. Создание Извещений. Обновление Извещений Фрагменты (Fragments) Создание Фрагментов. Добавление пользовательского интерфейса. Добавление фрагментов к Активностям. Управление Фрагментами. Транзакции с Фрагментами. Взаимодействие Фрагментов и Активностей. Жизненный цикл Фрагментов Процессы и потоки (Threads) Жизненный цикл процесса. Потоки. Фоновые потоки. Использование AsyncTask Сервисы (Services) Описание Сервисов в Манифесте приложения. Запуск Сервисов. Остановка Сервисов. Связанные Сервисы. Сервисы и Извещения. Сервисы переднего плана (Foreground Services). Жизненный цикл Сервисов Виджеты (Widgets) Описание Виджетов в Манифесте приложения. Создание разметки Виджета Класс AppWidgetProvider. Создание Виджета. Использование Конфигурационной Активности. Использование Preview Image. Обновление Виджетов Работа картами SD и внутренним хранилищем устройства Проверка доступности носителя. Доступ к файлам. Совместно используемые файлы и стандартные каталоги. Файлы кэша приложений Загрузчики (Loaders) Обзор API Загрузчиков. Применение Загрузчиков. Запуск и перезапуск Загрузчиков. Использование LoaderManager. Использование LoaderCursor		
--	--	--	--

	Беспроводные соединения Проверка сетевых соединений. Отслеживание состояния соединений ConnectivityManager и NetworkInfo. Эффективное использование сетевых соединений Использование AlarmManager и AlarmClock Типы будильников в Android. Однократные и повторяющиеся события Области применения AlarmManager и альтернативы (Timer и Handler) Использование AlarmClock Сенсоры в Android Обзор сенсоров. Типы сенсоров и получение информации об их доступности Sensor Framework. Мониторинг состояния сенсоров. Лучшие практики при работе с сенсорами Телефония и SMS Совершение звонков из приложения. Определение состояния и параметров телефона. Мониторинг состояния телефонного модуля. Использование SMS. Отправка SMS. Получение SMS Собственные View Использование Canvas. Использование SurfaceView. Использование Drawable Shape Drawable и 2D графика. Модификация существующих View. Создание собственных View Звук и камера в Android Запись и воспроизведение звука. Основы работы с камерой в Android Использование имеющихся приложений работы с камерой. Прямое управление камерой. Съемка и сохранение фото и видео		
	Практическое занятие. Практическая работа №6 «Создание эмуляторов и подключение у «Создание нового проекта» Практическая работа №7 «Изменение элементов дизайна» Практическая работа №8 «Обработка событий: цветовая индикация» Практическая работа №9 «Обработка событий: переключение между экранами» Практическая работа №10 «Тестирование и оптимизация мобильного приложения» Практическая работа №11 «Инструменты разработки Android-приложений. Пример простейших программ Android приложения. Запуск приложения на эмуляторе. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS)». Практическая работа №12 «Использование изображений». Практическая работа №13 «Игровая физика» Практическая работа №14 «Искусственный интеллект в играх» Практическая работа №15 «Профилировщик Windows Phone OS 7.1»	50	2

	Практическая работа №16 «Создание и изменение баз данных SQLite через Android-приложение». Практическая работа №17 «Публикация Android-приложения на Google Play». Практическая работа №18 «Инструменты разработки Windows Phone7-приложений. Пример простейших программы». Практическая работа №19 «Microsoft Silverlight. Основы работы с сенсорным вводом. Обработка нескольких касаний». Практическая работа №20 «Акселерометр и служба определения местоположения. Использование pivot и panorama». Практическая работа №21 «Создание эмуляторов и подключение устройств. Игровая физика» Практическая работа №22 «Создание нового проекта. Изменение элементов дизайна. Использование изображений» Практическая работа №23 «Обработка событий: цветовая индикация. Обработка событий: переключение между экранами. Тестирование и оптимизация мобильного приложения» Практическая работа №24 «Инструменты разработки Android-приложений. Пример простейших программ Android приложения. Запуск приложения на эмуляторе. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS). Публикация Android-приложения на Google Play».		
	Самостоятельная работа. Решение вариативных задач	32	3
Дифференцированный зачёт			
МДК. 01.04 Системное программирование		218 (70/142)	
<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	Содержание учебного материала (лекции) Ознакомление со спецификой сборки ПО в GNU/Linux (Ubuntu) Ознакомление с консольными текстовыми редакторами (vi,nano,mcedit) Ручная и автоматическая сборка ПО (Makefile). Модель Клиент-Интерфейс-Сервер (КИС). Статическая сборка библиотек. Совместно используемые библиотеки. Работа с переменными окружения Низкоуровневый ввод-вывод и файловые операции Обзор механизмов ввода-вывода в Linux (Ubuntu). Файловые дескрипторы Системные вызовы: open, close, write, read и lseek. Типы файлов. Индексные дескрипторы и жесткие ссылки. Права доступа к файлу Файловая система rroc. Два способа прочесть содержимое директории Разреженные файлы и специфика их применения. Блокировка областей файла Межпроцессное взаимодействие	36	1

	Механизмы межпроцессного взаимодействия Linux (Ubuntu). Неименованные каналы (pipes). Именованные каналы (named pipes). Сообщения (message queue) Разделяемая память (shared memory). Семафоры (semaphores) Сокеты Сокеты в файловом пространстве имен (UNIX-сокеты). Парные сокеты (pair sockets). Сетевые сокеты (sockets) Сигналы Знакомство с сигналами (signals). Отличие сигналов от других механизмов межпроцессного взаимодействия. Специфика обработки сигналов (signal handling) Процессы Клонирование процессов — fork(). Замена исполняемого процесса — exec() Зомби (zombies) — причины возникновения и способы их устранения Потоки Потоки и процессы. Специфика построения многопоточных приложений (multithreading). Досрочное завершение потока. Создание обработчика завершения потока. Средства синхронизации потоков (synchronize primitives) Атрибуты потоков Демоны (службы) Отличие демона от консольной утилиты. Специфика разработки демонов (daemons). Создание демона использующего сетевые сокеты. Консольный ввод-вывод Специфика разработки консольных приложений. Предотвращение перенаправления вывода. Управление терминалом. Скрытие пароля пользователя при аутентификации. Отображаемая память Отображение обычного файла. Совместный доступ к файлу. Частные отображения. Другие применения mmap Программирование графического интерфейса с помощью GTK+ Специфика разработки 64-битных приложений. Использование библиотеки ncurses. Программирование графического интерфейса с помощью GTK+		
	Практическое занятие. Практическая работа №1 «Обработка символьной информации» Практическая работа №2 «Создание макросов» Практическая работа №3 «Разработка многомодульной программы» Практическая работа №4 «Обработка файлов» Практическая работа №5 «Взаимодействие Ассемблера и языков высокого уровня»	12	2

	Самостоятельная работа. Разработка приложений на языках низкого уровня	57	3
<i>Тема 1.4.2 Разработка драйверов устройств в Linux</i>	Содержание учебного материала (лекции) Ядро Linux Архитектура ядра linux. Отличия разработки драйверов от прикладного ПО Потенциальные проблемы с безопасностью. Лицензирование модулей ядра Модули ядра Linux Простейший модуль ядра. Знакомство с printk(). Сборка модулей ядра Макроопределения __init и __exit. Вопросы документирования модулей Передача модулю параметров командной строки. Модули, состоящие из нескольких файлов. Сборка модулей под существующее ядро Модули ядра, пространства и устройства Модули ядра и прикладные программы. Функции, которые доступны из модулей. Пространство пользователя и пространство ядра. Пространство имен. Адресное пространство. Старший и младший номер устройства Символьные устройства Файлы символьных устройств. Структура file_operations Структура file. Регистрация устройства. Отключение устройства Пример драйвера символьного устройства с доступом только на чтение Специфика поддержки записи для символьных устройств. Пример драйвера символьного устройства с доступом на чтение и запись Файловая система /proc Особенность файловой системы /proc. Пример драйвера, создающего файл в /proc с доступом на чтение. Пример драйвера, создающего файл в /proc с доступом на чтение. Блокировка процессов при конкурентном доступе к устройству. Пример драйвера, приостанавливающего работу процессов Управление памятью в ядре Linux Специфика управления памятью в ядре Linux. Преобразование адреса в ядре Отображение файла в память ядра. Управление отображением в память Блочные устройства Блочные устройства. Регистрация блочного устройства. Операции, поддерживаемые для блочных устройств. Обработка запросов к блочному устройству. Детали реализации блочного устройства. Пример драйвера блочного устройства Семафоры Конкуренция и ситуация гонок. Семафоры и мьютексы. Реализация семафора в	34	1

	ядре Linux. Семафоры на чтение и запись. Взаимоблокировки. Пример драйвера, использующего семафоры Данные Типы данных и структуры ядра. Специфика использования стандартных типов данных. Назначение типам данных явного размера. Специфичные типы данных Аспекты, связанные с совместимостью. Измерение временных интервалов Размер страницы. Порядок следования байт в словах. Выравнивание данных Размер указателя		
--	--	--	--

	Практическое занятие. Практическая работа №6 «Borland C++ как инструмент системного программирования. Работа символьными строками» Практическая работа №7 «Примеры программирования, закрепляющие основные особенности конструкций Borland C++» Практическая работа №8 «Создание программ представления в памяти массивов и матриц» Практическая работа №9 «Алгоритм создания приложения Windows. Использование IDE Borland C++ . Создание и запуск простейшего приложения Windows» Практическая работа №10 «Создание приложений с обработкой сообщений. Практическая работа №11 Использование def-файла» Практическая работа №12 «Создание приложение с выводом информации в окно» Практическая работа №13 «Примеры использования классов C++ при создании приложений Windows. Стили окон». Практическая работа №14 «Примеры приложений, обрабатывающих клавиатурные сообщения, сообщения от драйвера «мыши» и таймера» Практическая работа №15 «Ресурсы в приложениях Windows. Таблицы текстовых строк. Пиктограммы. Курсоры «мыши». Bitmap образы» Практическая работа №16 «Примеры приложений, использующих меню. Шаблоны меню в файле ресурсов. Создание плавающего меню» Практическая работа №17 «Организация органов управления ОС Windows» Практическая работа №18 «Создание приложений, использующих диалоговые панели ОС Windows» Практическая работа №19 «Проверка оборудования» Практическая работа №20 «Управление клавиатурой» Практическая работа №21 «Управление таймером» Практическая работа №22 «Управление видеоадаптером» Практическая работа №23 «Дисковые структуры данных» Практическая работа №24 «Управление программами» Практическая работа №25 «Генерация и оптимизация объектного кода» Практическая работа №26 «Программирование пользовательского интерфейса»	48	2
	Самостоятельная работа. Разработка драйверов	56	3
Для характеристики уровня освоения учебного материала используются следующие обозначения: 1 – ознакомительный (узнавание ранее изученных объектов, свойств); 2 – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством); 3 – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач).			

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Требования к минимальному материально-техническому обеспечению

Наименование учебных кабинетов, лабораторий, полигонов	Оснащенность учебных кабинетов, лабораторий, полигонов	Перечень лицензионного программного обеспечения
Учебный кабинет «Разработка программных модулей», «Поддержка и тестирование программных модулей», «Разработка мобильных приложений», «Системное программирование» (лекционные, практические и лабораторные занятия)	Шкаф для хранения учебно-наглядных пособий, приборов, раздаточного материала Скелет, муляжи. Аудиторная доска Стол для преподавателя Стул для преподавателя Столы для студентов Стулья для студентов Шкафы для муляжей и моделей; приборы и оборудование необходимое для изучения дисциплины и овладения профессиональными знаниями и компетенциями.	Consultant+ Операционная система MS Windows 7 Pro, Операционная система MS Windows XP SP3. MS Office. Kaspersky Endpoint Security. 1С, Google Chrome, OpenOffice, LibreOffice
Учебный кабинет «Компьютерный класс» (практические занятия с использованием персональных компьютеров).	Специализированная мебель, технические средства обучения (персональные компьютеры) с возможностью подключения к телекоммуникационной сети «Интернет» и доступу к электронно-библиотечной системе	

При изучении профессионального модуля в целях реализации компетентного подхода использованы активные и интерактивные формы обучения: лекция – конференция, лекция – проблема, решение ситуационных задач, групповые дискуссии и иные тренинги.

3.2. Информационное обеспечение обучения

Основная литература:

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие для спо / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 312 с. — ISBN 978-5-8114-9817-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200390> (дата обращения: 30.10.2022). — Режим доступа: для авториз. пользователей.
2. Соколова, В. В. Разработка мобильных приложений : учебное пособие / В. В. Соколова. — Томск : ТПУ, 2014. — 176 с. — ISBN 978-5-4387-0369-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/82830> (дата обращения: 30.10.2022). — Режим доступа: для авториз. пользователей.
3. Кузнецов, А. С. Системное программирование : учебное пособие / А. С. Кузнецов, И. А. Якимов, П. В. Пересунько. — Красноярск : СФУ, 2018. — 170 с. — ISBN 978-5-7638-3885-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL:

<https://e.lanbook.com/book/157574> (дата обращения: 30.10.2022). — Режим доступа: для авториз. пользователей.

Дополнительная учебная литература:

1. Основы разработки приложений для мобильных телефонов смартфонов : учебно-методическое пособие / М. Р. Богданов, И. Н. Думчикова, Л. В. Миниярова, А. Р. Мухамедьянов. — Уфа : БГПУ имени М. Акмуллы, 2012. — 312 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/43380> (дата обращения: 30.10.2022). — Режим доступа: для авториз. пользователей.
2. Головнин, О. К. Введение в системное программирование и основы жизненного цикла системных программ : учебное пособие / О. К. Головнин, А. А. Столбова. — Самара : Самарский университет, 2021. — 172 с. — ISBN 978-5-7883-1695-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/257132> (дата обращения: 30.10.2022). — Режим доступа: для авториз. пользователей.

Профессиональные базы данных и справочные системы

- Федеральная служба государственной статистики - <https://rosstat.gov.ru/>
- Научометрическая и реферативная база данных SCOPUS - <https://www.scopus.com>
- Информационно-справочная система "КонсультантПлюс"

3.3 Методические указания для обучающихся по освоению дисциплины

Методические указания для подготовки к лекционным занятиям

В ходе-лекций преподаватель излагает и разъясняет основные, наиболее сложные для понимания темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на семинарское занятие и указания на самостоятельную работу.

В ходе лекционных занятий необходимо:

-вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

-задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

-дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой -в ходе подготовки к семинарам изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях журналах, газетах и т.д. При этом учесть рекомендации преподавателя и требования учебной программы.

-подготовить тезисы для выступлений по всем учебным вопросам, выносимым на семинар. Готовясь к докладу или реферативному сообщению, обращаться за методической помощью к преподавателю, составить план-

конспект своего выступления, продумать примеры с целью обеспечения тесной связи изучаемой теории с реальной жизнью.

-своевременное и качественное выполнение самостоятельной работы базируется на соблюдении настоящих рекомендаций и изучении рекомендованной литературы. Студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при подготовке практических работ.

Методические указания для подготовки к практическим (семинарским) занятиям

Начиная подготовку к семинарскому занятию, необходимо, прежде всего, обратить внимание на конспект лекций, разделы учебников и учебных пособий, которые способствуют общему представлению о месте и значении темы в изучаемом курсе. Затем следует поработать с дополнительной литературой, сделать записи по рекомендованным источникам. Подготовка к семинарскому занятию включает 2 этапа:

- 1й этап - организационный;
- 2й этап - закрепление и углубление теоретических знаний. На первом этапе студент планирует свою самостоятельную работу, которая включает:
 - уяснение задания, выданного на самостоятельную работу;
 - подбор рекомендованной литературы;
 - составление плана работы, в котором определяются основные пункты предстоящей подготовки.

Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку студента к занятию. Начинать надо с изучения рекомендованной литературы. Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная её часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретает практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Готовясь к консультации, необходимо хорошо продумать вопросы, которые требуют разъяснения.

В начале занятия студенты под руководством преподавателя более глубоко осмысливают теоретические положения по теме занятия, раскрывают и объясняют основные положения выступления.

Записи имеют первостепенное значение для самостоятельной работы обучающихся. Они помогают понять построение изучаемого материала, выделить основные положения и проследить их логику. Ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у студента, систематически ведущего записи, создается свой индивидуальный фонд подсобных материалов для быстрого повторения прочитанного, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе. Важно развивать умение сопоставлять источники, продумывать изучаемый материал.

Большое значение имеет совершенствование навыков конспектирования. Преподаватель может рекомендовать студентам следующие основные формы записи план(простой и развернутый), выписки, тезисы. Результаты конспектирования могут быть представлены в различных формах.

План - это схема прочитанного материала, краткий (или подробный) перечень вопросов, отражающих структуру и последовательность материала. Подробно составленный план вполне заменяет конспект.

Конспект - это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов.

План-конспект - это развернутый детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении.

Текстуальный конспект - это воспроизведение наиболее важных положений и фактов источника.

Свободный конспект - это четко и кратко сформулированные (изложенные) основные положения в результате глубокого осмысливания материала. В нем могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом.

Тематический конспект составляется на основе изучения ряда источников и дает более или менее исчерпывающий ответ по какой-то схеме (вопросу).

Ввиду трудоемкости подготовки к семинару следует продумать алгоритм действий, еще раз внимательно прочитать записи лекций и уже готовый конспект по теме семинара, тщательно продумать свое устное выступление.

На семинаре каждый его участник должен быть готовым к выступлению по всем поставленным в плане вопросам, проявлять максимальную активность при их рассмотрении. Выступление должно строиться свободно, убедительно и аргументировано. Необходимо следить, чтобы выступление не сводилось к репродуктивному уровню (простому воспроизведению текста), не допускать и простое чтение конспекта. Необходимо, чтобы выступающий проявлял собственное отношение к тому, о чем он говорит, высказывал свое личное мнение, понимание, обосновывал его и мог сделать правильные выводы из сказанного.

Выступления других обучающихся необходимо внимательно и критически слушать, подмечать особенное в суждениях обучающихся, улавливать недостатки и ошибки. При этом обратить внимание на то, что еще не было сказано, или поддержать и развить интересную мысль, высказанную выступающим студентом. Изучение студентами фактического, материала по теме практического занятия должно осуществляться заблаговременно. Под фактическим материалом следует понимать специальную литературу по теме занятия, а также по рассматриваемым проблемам. Особое внимание следует обратить на дискуссионные -теоретические вопросы в системе изучаемого вопроса: изучить различные точки зрения ведущих ученых, обозначить противоречия современного законодательства. Для систематизации основных положений по теме занятия рекомендуется составление конспектов.

Обратить внимание на:

- составление списка нормативных правовых актов и учебной и научной литературы по изучаемой теме;
- изучение и анализ выбранных источников;
- изучение и анализ практики по данной теме, представленной в информационно-справочных правовых электронных системах и др.;
- выполнение предусмотренных программой заданий в соответствии с тематическим планом;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями:
- на их еженедельных консультациях;
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний решения, представленных в учебно-методических материалах.

Методические указания по выполнению лабораторных работ

Подготовку к лабораторной работе рекомендуется проводить в следующей последовательности:

- уяснить тему и цель, предстоящей лабораторной работы;
- изучить теоретический материал в соответствии с темой лабораторной работы (рекомендуется использовать рекомендованную литературу, конспект лекций, учебное пособие (практикум по лабораторным работам);
- ознакомиться с оборудованием и материалами, используемыми на лабораторной работе (при использовании специализированного оборудования необходимо изучить порядок и правила его использования).

Вопросы, вынесенные для собеседования при защите лабораторных работ дисциплины, представлены в ФОС.

При выполнении лабораторной работы студенты должны строго соблюдать, установленные правила охраны труда.

При выполнении лабораторной работы студентам рекомендуется:

- уяснить цель, выполняемых заданий и способы их решения;

- задания, указанные в лабораторной работе выполнять в той последовательности, в которой они указаны в лабораторном практикуме;
- при выполнении практического задания и изучении теоретического материала использовать помощь преподавателя;
- оформить отчет по лабораторной работе;
- ответить на контрольные вопросы.

При подготовке к защите лабораторной работы студентам рекомендуется:

- подготовить отчет по лабораторной работе;
- подготовить обоснование, сделанных выводов;
- закрепить знания теоретического материала по теме лабораторной работы (рекомендуется использовать контрольные вопросы);
- знать порядок проведения расчетов (проводимых исследований);
- уметь показать и пояснить порядок исследований при использовании специализированного оборудования.

Методические указания для обучающихся по освоению дисциплины для самостоятельной работы

Методика организации самостоятельной работы студентов зависит от структур; характера и особенностей изучаемой дисциплины, объема часов на ее изучение, I заданий для самостоятельной работы студентов, индивидуальных особенностей студентов и условий учебной деятельности.

При этом преподаватель назначает студентам варианты выполнения самостоятельной работы, осуществляет систематический контроль выполнения студентами графика самостоятельной работы, проводит анализ и дает оценку выполненной работы.

Самостоятельная работа обучающихся осуществляется в аудиторной внеаудиторной формах. Самостоятельная работа обучающихся в аудиторное время может включать:

- конспектирование (составление тезисов) лекций, выполнение контрольных работ
- работу со справочной и методической литературой;
- работу с нормативными правовыми актами;
- выступления с докладами, сообщениями на семинарских занятиях;
- защиту выполненных практических работ;
- участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
- участие в беседах, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
- участие в тестировании и др.

Самостоятельная работа обучающихся во внеаудиторное время может состоять

- повторения лекционного материала;
- подготовки к семинарам (практическим занятиям);
- изучения учебной и научной литературы;

- выполнения практических заданий;
- подготовки к контрольным работам, тестированию и т.д.;
- подготовки к семинарам устных докладов (сообщений);
- подготовки рефератов, эссе и иных индивидуальных письменных работ заданию преподавателя;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на еженедельных консультациях;
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний.

Методическое указание по применению электронного обучения и дистанционных технологий при освоении дисциплины.

Дистанционные образовательные технологии применяются при изучении дисциплин в очно, очно-заочной и заочной формах обучения.

Освоение учебной дисциплины в очной форме обучения с применением дистанционных образовательных технологий применяется в случае установления карантинных мер, в связи со сложившимся сложной санитарно-эпидемиологической обстановкой или иным основанием в виду обстоятельств неопределенной силы. Занятия лекционного типа проводятся с использованием открытых онлайн-курсов, лекций в режиме онлайн конференции с контрольными вопросами для самостоятельной работы.

Практические занятия проводятся с использованием видео уроков, презентаций и виртуальных аналогов приборов, оборудования, иных средств обучения используемых в соответствии с содержанием учебного материала.

Семинарские занятия проводятся в режиме видео-конференции с использованием контрольных заданий, контрольных работ, позволяющих закрепить полученные теоретические знания.

Лабораторные занятия проводятся с использованием открытых онлайн-курсов и виртуальных аналогов приборов, оборудования и иных средств обучения позволяющих изучить теоретический материал и практические навыки с помощью экспериментального подтверждения.

Для материально-технического обеспечения освоения учебной дисциплины с применением электронного обучения и дистанционных образовательных технологий используется ZOOM, WatsApp, Discord, образовательных платформ «Система дистанционного обучения SDO.poanonic.ru », базы данных ЭБС «Лань», «IPRbooks»

При использовании дистанционных образовательных технологий обучающиеся переводятся на обучения по индивидуальному учебному плану в котором указаны трудоемкость, последовательность изучения дисциплин (модулей), виды учебной деятельности (лекции, практические занятия, семинарские занятия, лабораторные занятия, самостоятельная работа), формы промежуточной аттестации, определяющие порядок освоения основной образовательной программы с использованием дистанционных образовательных технологий.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
<i>ПК 1.1</i> Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Оценка «отлично» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. Оценка «хорошо» - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры. Оценка «удовлетворительно» - алгоритм разработан и соответствует заданию.	Экзамен/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам
<i>ПК 1.2</i> Разрабатывать программные модули в соответствии с техническим заданием	Оценка «отлично» - программный модуль разработан 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. Оценка «хорошо» - программный модуль разработан 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. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/структурного	Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

	программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.	
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	Оценка «отлично» - выполнена отладка модуля; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки. Оценка «хорошо» - выполнена отладка модуля; сохранены и представлены результаты отладки. Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.	Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 1.4 Выполнять тестирование программных модулей	Оценка «отлично» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия, сделан вывод о достаточности тестового пакета. Оценка «хорошо» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия. Оценка «удовлетворительно» - выполнено тестирование модуля и оформлены результаты тестирования. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия с некоторыми погрешностями.	Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля. оценке тестового покрытия. Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

<i>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</i>	<i>Оценка «отлично» - 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.</i> <i>Оценка «хорошо» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</i> <i>Оценка «удовлетворительно» - 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.</i>	<i>Экзамен/зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</i> <i>Защита отчетов по практическим и лабораторным работам</i> <i>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</i>
<i>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</i>	<i>Оценка «отлично» - программный модуль разработан 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.</i> <i>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и практически соответствует техническому</i>	<i>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</i> <i>Защита отчетов по практическим и лабораторным работам</i> <i>Интерпретация результатов</i>

	заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов	наблюдений за деятельностью обучающегося в процессе практики
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.	Оценка «отлично» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации. 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. 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.	Экзамен/зачет в форме собеседования: практическое задание по созданию модуля для заданного мобильного устройства на основе спецификации Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного	Экзамен/зачет в форме собеседования: практическое задание по

	программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «хорошо» - программный модуль разработан 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. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.	разработке программного модуля в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	Оценка «отлично» - выполнена отладка модуля (с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки. Оценка «хорошо» - выполнена отладка модуля; сохранены и представлены результаты отладки. Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.	Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

Результаты (освоенные общие компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОК 02. 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	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.	- демонстрация ответственности за принятые решения - обоснованность самоанализа и коррекция результатов собственной работы;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	- взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.	- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,	

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	- эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	
ОК 09. Использовать информационные технологии в профессиональной деятельности.	- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;	
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.	

5. АДАПТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ПРИ ОБУЧЕНИИ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Адаптация рабочей программы дисциплины ПМ 01. Разработка модулей программного обеспечения проводится при реализации адаптивной образовательной программы - программы подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование» в целях обеспечения права инвалидов и лиц с ограниченными возможностями здоровья на получение профессионального образования, создания необходимых для получения среднего профессионального образования условий, а также обеспечения достижения обучающимися инвалидами и лицами с ограниченными возможностями здоровья результатов формирования практического опыта.

Оборудование кабинета для обучающихся с различными видами ограничения здоровья

Оснащение кабинета должно отвечать особым образовательным потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья. Кабинеты должны быть оснащены оборудованием и учебными местами с техническими средствами обучения для обучающихся с различными видами ограничений здоровья.

Кабинет, в котором обучаются лица с нарушением слуха, должен быть оборудован радиоклассом, компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

Для слабовидящих обучающихся в кабинете предусматриваются просмотр удаленных объектов при помощи видеоувеличителей для удаленного просмотра. Использование Брайлевской компьютерной техники, электронных луп, программ невизуального доступа к информации, технических средств приема-передачи учебной информации в доступных формах.

Для обучающихся с нарушением опорно-двигательного аппарата кабинет должен быть оборудован передвижными регулируемые партами с источником питания.

Вышеуказанное оснащение устанавливается в кабинете при наличии обучающихся по адаптированной образовательной программе с учетом имеющегося типа нарушений здоровья у обучающегося.

Информационное и методическое обеспечение обучающихся

Доступ к информационным и библиографическим ресурсам, указанным в п.3.2 рабочей программы, должен быть представлен в формах, адаптированных к ограничениям здоровья обучающихся инвалидов и лиц с ограниченными возможностями здоровья:

Для лиц с нарушениями зрения (не менее двух видов):

- в печатной форме увеличенным шрифтом;

- в форме электронного документа;
- в форме аудиофайла;
- в печатной форме на языке

Брайля. Для лиц с нарушениями слуха:

- в печатной форме;
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата (не менее двух видов):

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

Для лиц с нервно-психическими нарушениями (расстройство аутистического спектра, нарушение психического развития):

- использование текста с иллюстрациями;
- мультимедийные материалы.

Во время самостоятельной подготовки обучающиеся инвалиды и лица с ограниченными возможностями здоровья должны быть обеспечены доступом к сети Интернет.

Формы и методы контроля и оценки результатов обучения

Формы и методы контроля проводятся с учетом ограничения здоровья обучающихся. Целью текущего контроля является своевременное выявление затруднений и отставания, обучающегося с ограниченными возможностями здоровья и внесение коррективов в учебную деятельность.

Форма промежуточной аттестации устанавливается с учетом индивидуальных психофизических особенностей обучающихся инвалидов и лиц с ограниченными возможностями здоровья (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.).

При проведении промежуточной аттестации обучающемуся предоставляется время на подготовку к ответу, увеличенное не более чем в три раза, установленного для подготовки к ответу обучающимся, не имеющим ограничений в состоянии здоровья.