



Министерство образования и науки Республики Татарстан
Государственное автономное профессиональное образовательное учреждение
«ЕЛАБУЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»


Рассмотрено
на заседании ЦМК ОПД


О.Н. Голованова
«27» августа 2022 г.

Согласовано
Заместитель директора по УМР


О.С. Шарборина
«29» августа 2022 г.

Согласовано
Заместитель директора по УТР


А.В. Шмухаметова
«29» августа 2022 г.

Согласовано
Руководитель службы
администрирования
образовательного кластера
АО «ОЭЗ ИИТ «Алабуга»


Э.М. Фомина

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ 01. Разработка модулей программного обеспечения для

компьютерных систем

Специальность: 09.02.07 Информационные системы и

программирование

квалификация – программист

г. Елабуга, 2022 г.

Рабочая программа профессионального модуля разработана с учетом требований:

- Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации №1547 от 09 декабря 2016 года (Зарегистрировано в Минюсте России 26.12.2016 N 44936)

- Федерального закона 31 июля 2020 г. №304-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания обучающихся»;

- Приказа Министерства просвещения Российской Федерации от 28.08.2020 г. №441 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по образовательным программам среднего профессионального образования, утвержденный приказом Министерства образования и науки Российской Федерации от 14 июня 2013 г. № 464».

Организация-разработчик: ГАПОУ «Елабужский политехнический колледж»

Разработчик: Симонов А.Н. – преподаватель информатики

Иванова Е.М. – преподаватель информатики

Шараборина О.С. – заместитель директора по учебно-методической работе

СОДЕРЖАНИЕ

1. ПАСПОРТ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	8
3. СТРУКТУРА И ПРИМЕРНОЕ СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	10
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	24
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	31

1. ПАСПОРТ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ.01.Разработка модулей программного обеспечения для компьютерных систем

1.1. Область применения программы:

Рабочая программа профессионального модуля является частью основной профессиональной образовательной программы по специальности среднего профессионального образования в соответствии с федеральным государственным образовательным стандартом среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, в части освоения основного вида профессиональной деятельности (ВПД): ВПД.1 Разработка модулей программного обеспечения для компьютерных систем (ПК):

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.

Рабочая программа профессионального модуля может быть использована в дополнительном профессиональном образовании и профессиональной подготовке работников.

1.2. Цели и задачи профессионального модуля – требования к результатам освоения профессионального модуля:

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

иметь практический опыт:

- Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
- Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.
- Разрабатывать мобильные приложения.
- Использовать инструментальные средства на этапе отладки программного продукта.
- Проводить тестирование программного модуля по определенному сценарию.
- Проводить тестирование программного модуля по определенному сценарию.
- Использовать инструментальные средства на этапе тестирования программного продукта.
- Анализировать алгоритмы, в том числе с применением инструментальных средств.
- Осуществлять рефакторинг и оптимизацию программного кода.
- Разрабатывать мобильные приложения.

уметь:

- Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
- Оформлять документацию на программные средства.
- Оценка сложности алгоритма.
- Создавать программу по разработанному алгоритму как отдельный модуль.
- Оформлять документацию на программные средства.
- Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.
- Выполнять отладку и тестирование программы на уровне модуля.
- Оформлять документацию на программные средства.
- Применять инструментальные средства отладки программного обеспечения.
- Выполнять отладку и тестирование программы на уровне модуля.
- Оформлять документацию на программные средства.

- Выполнять оптимизацию и рефакторинг программного кода.
- Работать с системой контроля версий.
- Осуществлять разработку кода программного модуля на современных языках программирования.
- Оформлять документацию на программные средства.

знать:

- Основные этапы разработки программного обеспечения.
- Основные принципы технологии структурного и объектно-ориентированного программирования.
- Актуальная нормативно-правовая база в области документирования алгоритмов.
- Основные этапы разработки программного обеспечения.
- Основные принципы технологии структурного и объектно-ориентированного программирования.
- Знание API современных мобильных операционных систем.
- Основные принципы отладки и тестирования программных продуктов.
- Инструментарий отладки программных продуктов.
- Основные виды и принципы тестирования программных продуктов.
- Способы оптимизации и приемы рефакторинга.
- Инструментальные средства анализа алгоритма.
- Методы организации рефакторинга и оптимизации кода.
- Принципы работы с системой контроля версий.
- Основные этапы разработки программного обеспечения.
- Основные принципы технологии структурного и объектно-ориентированного программирования.

1.3. Формирование личностных результатов воспитательной работы обучающихся:

Л 10 - Заботиться о защите окружающей среды, собственной и чужой безопасности, в том числе цифровой.

Л 13 - Демонстрация умения эффективно взаимодействовать в команде, вести диалог, в том числе с использованием средств коммуникации.

Л 14 - Демонстрация навыков анализа и интерпретации информации из различных источников с учетом нормативно-правовых норм.

Л 15 - Демонстрация готовности и способности к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности.

1.4. Рекомендуемое количество часов на освоение программы профессионального модуля:

максимальная учебная нагрузка обучающихся – **1109** часов, включающая:

- обязательную аудиторную учебную нагрузку обучающегося – **805** часов (теоретическое обучение – **262** часа, лабораторно-практические занятия – **513** часов, курсовое проектирование – **30** часов);

- внеаудиторную самостоятельную работу обучающихся - **40** часов;

- квалификационный экзамен по профессиональному модулю – **12** часов;

- учебную практику – **108** часов;

- производственную практику – **144** часа.

В т.ч. часов из вариативной части ФГОС СПО по специальности 09.02.07 Информационные системы и программирование направленные на углубление профессиональных компетенций в соответствии с требованиями производств АО «ОЭЗ ППТ «Алабуга».

Индекс	Наименование циклов (разделов, дисциплин, МДК), требования к знаниям, умениям, практическому опыту	Обязательная учебная нагрузка, час.
МДК.01.01	Разработка программных модулей	53
МДК.01.02	Поддержка и тестирование программных модулей	60
МДК.01.03	Разработка мобильных приложений	60
МДК.01.04	Системное программирование	60
УП.01	Учебная практика	33
ПП.01	Производственная практика (по специальности)	44
КЭ	Квалификационный экзамен	12

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности (ВПД) ВПД.1 Разработка модулей программного обеспечения для компьютерных систем, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование общих компетенций
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	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 5	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
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	Планировать предпринимательскую деятельность в профессиональной сфере
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

3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Тематический план профессионального модуля ПМ.01.Разработка модулей программного обеспечения для компьютерных систем

Код 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, часов	Производственная (по профилю специальности), часов (если предусмотрена рассредоточенная практика)
			Всего, часов	в т.ч. лабораторные работы и практические занятия, часов	в т.ч., курсовая работа (проект), часов	Всего, часов	в т.ч., курсовая работа (проект), часов		
ПК 1.1 - 1.6 ОК 1-11	МДК 01.01 Разработка программных модулей	275	265	161	30	10	-	-	-
ПК 1.1 - 1.6 ОК 1-11	МДК 01.02 Поддержка и тестирование программных модулей	170	160	112	-	10	-	-	-
ПК 1.1 - 1.6 ОК 1-11	МДК 01.03 Разработка мобильных приложений	200	190	120	-	10	-	-	-
ПК 1.1 - 1.6 ОК 1-11	МДК 01.04 Системное программирование	200	190	120	-	10	-	-	-
ПК 1.1 - 1.6 ОК 1-11	Учебная практика	108						108	-
ПК 1.1 - 1.6 ОК 1-11	Производственная практика (по профилю специальности)	144						-	144
ПК 1.1 - 1.6 ОК 1-11	Квалификационный экзамен	12						-	-
	Всего:	1109	805	513	30	40		108	144

3.2. Содержание обучения по профессиональному модулю ПМ.01.Разработка модулей программного обеспечения для компьютерных систем

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовой проект	Объем часов	Уровень освоения
МДК 01.01 Разработка программных модулей		275	
Тема 1. Жизненный цикл ПО	1. Жизненный цикл программного обеспечения. Жизненный цикл программного обеспечения Этапы разработки программного обеспечения Стратегии и модели жизненного цикла	2	1
Тема 2. Структурное программирование	1. 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 строк Неявно типизированные переменные Сравнение значений разных типов	2	1
	2. 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 Примеры использования логических операций	2	1

	Побитовые логические операции Работа с портами (конъюнкция и дизъюнкция) Исключающее ИЛИ (пример использования) Логические операции Операции сдвига Короткозамкнутые вычисления Теоремы Де Моргана		
	3. 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	2	1
	4. 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	2	1
	5. Массивы Индекс массива Использование одномерных массивов	2	1

	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 <code>ragams</code>		
	Практическая работа №1. Условные конструкции	4	2
	Практическая работа №2. Циклические конструкции	4	2
	Практическая работа №3. Методы	4	2
	Практическая работа №4. Массивы	4	2
Тема 3. Объектно-ориентированное программирование	1. 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	2	1
	2. 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	2	1
	3. Наследование и полиморфизм Наследование Модификаторы доступа Связанность и связность	1	1

<p>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</p>		
<p>4. 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</p>	1	1
<p>5. 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</p>	1	1
<p>6. 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</p>	1	1

	7. 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	1	1
	8. Универсальные шаблоны (generics) Создание универсальных шаблонов Параметризованный метод Параметризованный делегат Ковариантность обобщений Контрвариантность обобщений Частичные классы и методы Тип Nullable Операция поглощения	1	1
	9. Ограничения универсальных шаблонов Ограничения параметров типа Ограничения параметров типа – naked Работа с коллекциями Словари Представление рефлектора	1	1
	10. События (events) Понятие событий Работа с событиями Абстрактные и виртуальные события Анонимные методы в событиях Использование событий Шаблон MVP	1	1
	11. 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	1	1
	12. Коллекции	1	1

Понятие коллекций Работа с коллекциями Цикл foreach Оператор yield Восстановление кода из под рефлятора		
13. Обработка исключений (exception) Обработка исключительных ситуаций Таблица исключений Конструкция try-catch Ключевое слово throw Разработка своих пользовательских исключений Конструкция try-catch-finaly Обработка внутренних исключений Порядок обработки исключений	1	1
14. 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	1	1
15. 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	1	1
16. Пространства имен. Директивы. Препроцессора Пространства имен Вложенные пространства имен Видимость во вложенных пространствах имен	1	1

<p>DLL библиотеки Создание DLL библиотек Модификаторы доступа</p>		
<p>17. 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</p>	1	1
<p>18. Системные коллекции Использование ArrayList и сбор элементов данных. Предназначение и использование интерфейса IComparer Работа с последовательными списками (Queue, Stack) Работа со словарями (Hashtable, SortedList, ListDictionary, HybridDictionary, OrderedDictionary) Роль интерфейса IEqualityComparer Применение специализированных наборов Работа с двоичными значениями с помощью BitArray и BitVector32 Использование методов CreateMask и CreateSection Хранение строк в наборах (StringCollection, StringDictionary, NameValueCollection) Обобщенные наборы</p>	1	1
<p>19. Программирование ввода-вывода Навигация по файловой системе Чтение и запись файлов Работа с потоками Работа с изолированным хранилищем</p>	1	1
<p>20. 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</p>	1	1
<p>21. XML. Файлы конфигурации. Реестр</p>	1	1

<p>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</p>		
<p>22. 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</p>	1	1
<p>23. Атрибуты Общее понятие атрибутов Класс System.Attribute Позиционные и именованные параметры атрибутов Создание атрибутов Использование атрибутов Определение атрибутов при помощи рефлексии Системные атрибуты</p>	1	1
<p>24. Сериализация Общее понятие сериализации Процесс сериализации и десериализации. Совместимость версий Управление сериализацией Форматы сериализации XML сериализация Binary сериализация SOAP сериализация Собственные методы сериализации</p>	1	1
<p>25. Управление памятью. Сборщик Мусора Шаблон NVI Классический полиморфизм Ad hoc полиморфизм Замещение, переопределение и перекрытие методов</p>	1	1

	26. 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()	1	1
	27. 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	1	1
	28. 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	1	1
	Практическая работа №5. Классы	4	2
	Практическая работа №6. Абстрактные классы	4	2
	Практическая работа №7. Статические классы	4	2
	Практическая работа №8. Структуры	4	2
	Практическая работа №9. События (events)	4	2
	Практическая работа №10. Обработка исключений	4	2
	Практическая работа №11. Перегрузка операторов	4	2
	Практическая работа №12. Обработка текстовой информации	4	2
	Практическая работа №13. Файлы конфигурации. Работа с реестром	4	2
	Практическая работа №14. Рефлексия	4	2
	Практическая работа №15. Управление памятью.	4	2
	Практическая работа №16. Полиморфизм. Версионность	4	2
	Практическая работа №17. Синхронизация при помощи объектов ядра	4	2
Тема 4. Паттерны проектирования	1. Назначение и виды паттернов Шаблоны и их использование в объективной реальности	1	1

	<p>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</p>		
	<p>2. Язык UML 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 Разновидности конечных автоматов Конечный автомат мили Граф соответствий (представление)</p>	1	1

	Табличное представление Конечный автомат мура Представление автомата мура		
	3. Шаблоны проектирования Grammar Abstract Factory Builder Factory Method Prototype Singleton Adapter Bridge Composite Decorator Facade Flyweight Proxy Chain of Responsibility Command Interpreter Iterator Mediator Memento Observer State Strategy Template Method	1	1
Тема 5. Событийно-управляемое программирование	1. Основы разработки оконных приложений Принципы визуального программирования Инструментальные средства, их функциональное назначение и состав Содержание файлов проекта создаваемых при проектировании и при компиляции Структура кода основного файла приложения, модуля общего назначения и файла формы Настройки выполняемого файла.	1	1
	2. 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	1	1

<p>События и коды обработчиков Разъяснения типовых событий каждого источника Классификация источников событий Объединение форм Дочерние формы</p>		
<p>3. Диалоговые сообщения Сообщения и их классификация Достоинства и недостатки каждого вида сообщений Подпрограммы реализующие диалоговые окна</p>	1	1
<p>4. Однострочные текстовые редакторы Однострочные текстовые редакторы Преобразование типов данных Маска ввода Защита от ошибок ввода данных. Обработка входных данных</p>	1	1
<p>5. Элементы переключателей Элементы переключателей Компоненты выбора Контейнеры для хранения объектов Вкладки Компоненты выбора цвета</p>	1	1
<p>6. Обработка табличной информации Обработка данных в табличном виде Способы обращения к таблицам</p>	1	1
<p>7. 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</p>	1	1
<p>8. 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</p>	1	1
<p>9. Многострочные текстовые редакторы Многострочные редакторы Обработка строковой информации Методы для работы со строками. Контроль вводимых символов Диалоговые окна</p>	1	1

	10. Работа с изображением Компоненты для работы с графикой Обработка графической информации	1	1
	11. Использование технологии перетаскивания и встраивание объектов Drag&Drop Методы и свойства для перемещения объектов События управления манипулятора "Мышь"	1	1
	12. Работа с файловым каталогом Диалоговые окна Работа с файловым каталогом Объекты для работы с проводником	1	1
	13. Работа с внешними приложениями Запуск внешних приложений Особенности, методы использования и параметры Организация работы с приложениями Microsoft Office	1	1
	14. Организация работы с электронной почтой Сетевые протоколы прикладного уровня Методы и свойства для работы и передачи данных по сети	1	1
	Практическая работа №18. Форма и ее свойства	4	2
	Практическая работа №19. Диалоговые окна и сообщения	4	2
	Практическая работа №20. Однострочные текстовые редакторы	4	2
	Практическая работа №21. Элементы переключателей и выбора	4	2
	Практическая работа №22. Обработка данных в табличном виде	4	2
	Практическая работа №23. Разработка приложения Браузер	4	2
	Практическая работа №24. Сохранение/чтение настроек приложения, используя Ini файлы.	4	2
	Практическая работа №25. Сохранение/чтение настроек приложения, используя реестр Windows.	4	2
	Практическая работа №26. Обработка строковой информации	4	2
	Практическая работа №27. Обработка графической информации	4	2
	Практическая работа №28. Технология Drag-And-Drop при разработке программных продуктов	4	2
	Практическая работа №29. Работа с файловым каталогом	4	2
	Практическая работа №30. Работа с приложениями Microsoft Office	4	2
	Практическая работа №31. Разработка почтового клиента	4	2
Тема 6. Оптимизация и рефакторинг кода	1. Введение в рефакторинг Рефакторинг Дублирование кода Длинные методы Большой класс Длинный список параметров Расходящиеся модификации Стрельба дробью Завистливые функции	1	1

<p>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 Классы данных Отказ от наследства Комментарии</p>		
<p>2. 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 Сокращение делегирования Удаление посредника</p>	1	1
<p>3. 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</p>	1	1
<p>4. Упрощение условных выражений Декомпозиция условного оператора Консолидация условного выражения Консолидация дублирующихся условных фрагментов Удаление управляющего флага Замена условного оператора полиморфизмом</p>	1	1

	Введение объекта Null		
	5. Решение задач обобщения Подъем поля Подъем метода Спуск метода/поля Выделение подкласса Выделение родительского класса Выделение интерфейса Замена наследования делегированием Замена делегирования наследованием	1	1
Тема .8. Основы ADO.Net	1. 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 sqlConnectionString Builder Рассмотрение Пул соединений Использование файлов конфигурации Использование поставщика данных .NET OleDb	1	1
	2. 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 Возврат данных в табличном представлении Строготипизированные средства доступа Получение данных с помощью индексатора Асинхронное выполнение команд Пример. Работа со значениями NULL Рассмотрение Транзакции Создание параметризованных запросов Использование коллекции Parameters Пример. Вызов хранимой процедуры Процедура, принимающая параметры Возвращающие значение	1	1
	3. Автономная часть архитектуры. Объекты DataColumn, DataTable Таблицы ADO.NET. Объект Data Column и DataRow	1	1

<p>DataReader 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</p>		
<p>4. Классы DataRow, DataSet и DataAdapter Рассмотрение Строк(DataRow) Использование свойства RowState объекта 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</p>	1	1
<p>5. 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</p>	1	1
<p>6. Фильтрация изменений и сортировка данных</p>	1	1

	<p>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</p>		
	<p>7. 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</p>	1	1
	<p>8. 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</p>	1	1
	Практическая работа №32. Разработка приложения базы данных	37	2
Курсовое проектирование		30	
	1.Введение	2	2
	2.Структура типичного приложения	2	2
	3.Реализация графического интерфейса	2	2
	4.Требования к дизайну пользовательского интерфейса	2	2

	5.Функциональные требования к программному модулю.	2	2
	6.Структура пользовательского интерфейса	2	2
	7.Анализ полей класса	2	2
	8.Интерфейс пользователя	2	2
	9.Сценарий работы пользователя	2	2
	10.Архитектура программного модуля.	2	2
	11.Архитектура формы «имя формы»	2	2
	12.Описание алгоритма разработки модуля	2	2
	13.Алгоритм метода «имя метода»	2	2
	14.Тестирование и отладка модуля	2	2
	15.Разработка документа «Руководство оператора»	2	2
Примерная тематика курсовых проектов:			
1.Разработка программного модуля «Дорожно-транспортные происшествия».			
2. Разработка программного модуля «Телефоны и абоненты».			
3. Разработка программного модуля «Сведения о книжном фонде библиотеки».			
4. Разработка программного модуля «Продажа авиабилетов».			
5. Разработка программного модуля «Обувной магазин».			
6. Разработка программного модуля Кафедра».			
7. Разработка программного модуля «Интернет магазин».			
8. Разработка программного модуля «Преподаватели».			
9. Разработка программного модуля «Отдел кадров предприятия».			
10. Разработка программного модуля «Отдел снабжения предприятия».			
11. Разработка программного модуля «Ремонтная мастерская».			
12. Разработка программного модуля «Магазин бытовой техники».			
13. Разработка программного модуля «Аукционы».			
14. Разработка программного модуля «Кинотеатры (Афиша)».			
15. Разработка программного модуля «Кинотеатры (Размещение и сеансы)».			
16. Разработка программного модуля «Ресторан».			
17. Разработка программного модуля «Бюро знакомств».			
18. Разработка программного модуля «Продажа жилья».			
19. Разработка программного модуля «Клиент отеля».			
20. Разработка программного модуля «Нарушители правил дорожного движения».			
21. Разработка программного модуля «Путевой лист для перевозки груза».			
22. Разработка программного модуля «Расписание электричек».			
23. Разработка программного модуля «Продажа автомобилей».			
24. Разработка программного модуля «Туристические путевки».			
25. Разработка программного модуля «Трудоустройство».			
26. Разработка программного модуля «Выставка собак».			
27. Разработка программного модуля «Научные труды сотрудников».			
28. Разработка программного модуля «Пассажирское судоходство».			
29. Разработка программного модуля «Приемные экзамены».			
30. Разработка программного модуля «Ипподром».			
31. Разработка программного модуля «Справочник филателиста».			

32. Разработка программного модуля «Расписание занятий студента».		
33. Разработка программного модуля «Пассажир поезда дальнего следования».		
34. Разработка программного модуля «Цех предприятия».		
35. Разработка программного модуля «Читатели студенческой библиотеки».		
36. Разработка программного модуля «Записная книжка».		
37. Разработка программного модуля «Телепрограмма».		
38. Разработка программного модуля «Промышленное рыболовство».		
39. Разработка программного модуля «Страховые иски».		
40. Разработка программного модуля «Учет успеваемости в колледже».		
41. Разработка программного модуля «Автобусные маршруты».		
42. Разработка программного модуля «Обработка заказов».		
43. Разработка программного модуля «Спортивные рекорды».		
Самостоятельная работа	10	
1. Разработка электронного словаря с возможностью подключения к базе данных.		
2. Создание программы для построения графиков математических функций.		
3. Разработка простейшего почтового клиента.		
4. Разработка калькулятора логических функций.		
5. Разработка инженерного калькулятора.		
6. Разработка калькулятора программиста.		
7. Разработка компьютерной игры «Шарики».		
8. Разработка компьютерной игры «Шашки».		
9. Разработка интерактивной логической игры «Квест».		
10. Разработка компьютерной игры «Найди пару!».		
11. Разработка графического редактора со сменными фонами «Раскраска».		
12. Создание интерактивной развивающей игры для детей «ПАЗЛ».		
13. Разработка программы «Слайд шоу».		
14. Разработка интерактивного медиаплеера.		
15. Разработка интерактивной развивающей игры «Найди отличия».		
16. Разработка игрового приложения «Мэмор» со звуком.		
17. Разработка компьютерной игры «Тэтрис».		
18. Разработка аналога программы Блокнот.		
19. Разработка аналога программы WordPad.		
20. Разработка игрового квеста «Я ищу». 2		
1. Создание интерактивного развивающего приложения.		
Дифференцированный зачет в 5 семестре	2	
Промежуточная аттестация в форме экзамена в 4 и 6 семестрах		
Всего	275	
МДК 01.02 Поддержка и тестирование программных модулей	170	
Тема 1.		
Отладка и тестирование программного обеспечения	4	1
1. Введение в тестирование программного обеспечения		
Понятие качества ПО. Стандарты качества ПО.		
Атрибуты и характеристики качества ПО.		

<p>Основные определения тестирования. Цели и задачи процесса тестирования. Полный цикл тестирования. Фазы тестирования.</p>		
<p>2. 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.</p>	4	1
<p>3. 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.</p>	4	1
<p>4. Тестовая документация. TestCase. 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.</p>	4	1
<p>5. Методы Покрытие входных данных. Допустимые и недопустимые данные. Эквивалентное разбиение. Анализ граничных значений. Попарное комбинирование. Предположение ошибок.</p>	4	1
<p>6. Техники тестирования 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</p>	4	1
<p>7. Уровни тестирования. Критерии покрытия кода программы тестами Модульное тестирование. Драйверы. Заглушки. Интеграционное тестирование. Способы интеграционного тестирования. Системное тестирование.</p>	2	1

	<p>Понятие покрытия кода тестами. Критерии покрытия. Метрика покрытия. Анализ покрытия.</p>		
	<p>8. Виды тестирования: функциональное и нефункциональное тестирование Функциональные виды тестирования. Тестирование безопасности, тестирование взаимодействия. Нефункциональные виды тестирования. Тестирование производительности. Нагрузочное тестирование.</p>	2	1
	<p>9. Тестирование пользовательского интерфейса (GUI). Тестирование web-приложений Задачи и цели тестирования пользовательского интерфейса. Функциональное тестирование пользовательского интерфейса. Тестирование удобства пользовательского интерфейса. Тестирование web-приложений</p>	2	1
	<p>10. 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.</p>	2	1
	<p>Практическая работа №1. Тестирование требований к учебной программе.</p>	12	2
	<p>Практическая работа №2. Составление плана тестирования учебной программы.</p>	12	2
	<p>Практическая работа №3. Составление набора входных данных для тестирования учебной программы.</p>	12	2
	<p>Практическая работа №4. Инсталляционное тестирование учебной программы.</p>	12	2
	<p>Практическая работа №5. Функциональное тестирование GUI.</p>	12	2
<p>Тема 2. Командное тестирование</p>	<p>1. 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</p>	2	1
	<p>2. Риски тестирования. Команда тестирования Создание условий работы в команде. Стили управления. Выстраивание отношений. Коммуникации. Постановка задач. SMART.</p>	2	1

	Хронофаги –поглотители времени Оперативное планирование. Приоритет задач. Матрица Эйзенхауэра.		
	3. 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.	2	1
	Практическая работа №6. Составление списка задач по тестированию и создание оперативного плана.	12	2
Тема 3. Документирование	1. Unit тестирование. Использование MSTest Понятие Unit Test Типы тестирования Интеграционное тестирование Верификация и валидация Свойства хорошего Unit теста Расположение Unit тестов Именованние проектов Именованние 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	2	1
	2. Использование Stub объектов Виды Unit тестов Виды тестовых объектов Inversion of Control (IOC) Шаблон Service Locator Шаблон Dependency Injection Constructor Injection	2	1

	Property Injection Interface Injection Local Factory Method Extract and Override Factory Injection Проблемы инкапсуляции		
	3. 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	2	1
	Практическая работа №7. Unit тестирование	12	2
	Практическая работа №8. Создание сценариев тестов	16	2
	Практическая работа №9. Описание результатов тестирования	12	2
Самостоятельная работа Примерная тематика рефератов: 1. Тестирование как часть процесса верификации программного обеспечения; 2. Виды ошибок; 3. Методы отладки; 4. Методы тестирования; 5. Классификация тестирования по уровням; 6. Тестирование производительности; 7. Регрессионное тестирование; 8. Тестирование «белым ящиком»; 9. Тестирование «черным ящиком»; 10. Модульное тестирование; 11. Регрессионное тестирование; 12. Оценка сложности алгоритмов сортировки; 13. Оценка сложности алгоритмов поиска; 14. Оформление документации на программные средства и использованием инструментальных средств. 15. Средства разработки технической документации; 16. Технологии разработки документов; 17. Документирование программного обеспечения в соответствии с Единой системой программной документации; 18. Автоматизация разработки технической документации;		10	

19. Автоматизированные средства оформления документации;			
20. Оформление документации на программные средства с использованием инструментальных средств.			
Дифференцированный зачет в 3 семестре		2	
Дифференцированный зачет в 5 семестре		2	
Всего		170	
МДК 01.03 Разработка мобильных приложений		200	
Тема 1. Основные платформы и языки разработки мобильных приложений	1. Введение в Java технологии Особенности языка программирования Java. Описание Java технологий. Использование интегрированной среды разработки.	2	1
	2. 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.	2	1
	3. Методы и операторы Java Создание и вызов методов. Перегрузка и методы с переменным числом аргументов. Операторы ветвления. Операторы циклов.	2	1
	4. Создание и использование массивов Java Одномерные массивы. Многомерные массивы. Работа с массивами и класс Arrays.	2	1
	5. Работа со строками в Java Основные методы класса String. Сложение и преобразование строк. Классы динамических строк. Ввод данных с клавиатуры и класс Scanner.	2	1
	6. 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	2	1

<p>Четырехмерные массивы Зубчатые массивы Использование зубчатых массивов Ключевое слово <code>params</code></p>		
<p>7. Разработка классов на Java Обзор основных концепций ООП. Объявление класса. Члены класса и создание объектов класса. Модификаторы доступа. Модификаторы <code>final</code> & <code>static</code>. Использование пакетов, директив импорта и переменной среды <code>CLASSPATH</code></p>	2	1
<p>8. Наследование и полиморфизм Наследование как механизм повторного использования кода. Конструктор при наследовании Преобразование типов и операция <code>instanceof</code>. Виртуальные методы и позднее связывание. Абстрактные классы и методы.</p>	2	1
<p>9. Пакет <code>java.lang</code> Класс <code>Object</code> и переопределение его методов. Метаданные и рефлексия. Классы <code>System</code> и <code>Math</code>.</p>	2	1
<p>10. Обработка ошибок в Java Концепция исключений в Java. Использование операторов <code>try</code>, <code>catch</code> и <code>finally</code>. Проверяемые и непроверяемые исключения. Создание своих классов исключений. Оператор <code>try</code> для освобождения ресурсов.</p>	2	1
<p>11. Потоки данных в Java Обзор классов потоков. Работа с байтовыми потоками. Работа с потоками символов. Использование класса <code>java.io.File</code>. Сжатие файлов. Сериализация объектов в Java.</p>	2	1
<p>12. Работа с файловой системой в NIO 2 Использование интерфейса <code>Path</code>. Работа с атрибутами файлов. Основные возможности класса <code>Files</code>. Использование класса <code>Files</code> для обхода дерева каталогов. Мониторинг изменений в файловой системе.</p>	2	1
<p>13. Пакет <code>java.util</code></p>	2	1

	<p>Форматирование данных. Работа с датой и временем. Класс Locale и глобализация кода. Локализация и класс ResourceBundle. Генерация псевдослучайных чисел</p>		
	<p>14. Коллекции в Java Иерархия классов коллекций. Концепция параметризованных типов данных. Работа с параметризованным методов и интерфейсом. Обзор возможностей списков, множеств и словарей в Java.</p>	2	1
	<p>15. Вложенные классы в Java Внутренние классы. Вложенные классы. Анонимные классы. Перечисления в Java.</p>	2	1
	<p>16. Лямбда-выражения Синтаксис лямбда-выражений. Ссылки на методы. Функциональные интерфейсы.</p>	2	1
	<p>17. Паттерны проектирования Обзор паттернов. Паттерн одиночка. Паттерн композиция. Паттерн наблюдатель.</p>	2	1
	Практическая работа №1. Установка JDK/IDE, настройка параметров среды	6	2
	Практическая работа №2. Activity, создание графического приложения	6	2
	Практическая работа №3. Разработка интерфейса мобильного приложения программно в коде Java	6	2
	Практическая работа №4. Разработка интерфейса мобильного приложения в XML	6	2
	Практическая работа №5. Контейнеры и позиционирование элементов в мобильных приложениях	6	2
	Практическая работа №6. Ресурсы	6	2
	Практическая работа №7. Работа с изображениями	6	2
	Практическая работа №8. Адаптеры	6	2
<p>Тема 2. Создание и тестирование модулей для мобильных приложений</p>	<p>1. 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</p>	2	1
	2. Активности и ресурсы	2	1

	<p>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</p>		
	<p>3. 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)</p>	2	1
	<p>4. 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</p>	2	1
	<p>5. 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</p>	2	1

<p>Особенности работы с БД в Android Выполнение запросов для доступа к данным Изменение данных в БД Использование SimpleCursorAdapter Контент-провайдеры Использование контент-провайдеров Создание контент-провайдеров Использование интернет-сервисов</p>		
<p>6. Диалоги в Android Виды Диалогов Рекомендации по дизайну Диалогов Создание и удаление Диалогов Обработка событий</p>	2	1
<p>7. Широковещательные приемники (Broadcast Receivers) Применение Широковещательных Приемников Жизненный цикл Приемника Регистрация Приемника Использование Ordered Broadcast Использование PendingIntent</p>	2	1
<p>8. Извещения (Notifications) в Android Взаимодействие с Извещениями Управление Извещениями Создание Извещений Обновление Извещений</p>	2	1
<p>9. Фрагменты (Fragments) Создание Фрагментов Добавление пользовательского интерфейса Добавление фрагментов к Активностям Управление Фрагментами Транзакции с Фрагментами Взаимодействие Фрагментов и Активностей Жизненный цикл Фрагментов</p>	2	1
<p>10. Процессы и потоки (Threads) Жизненный цикл процесса Потоки Фоновые потоки Использование AsyncTask</p>	2	1
<p>11. Сервисы (Services) Описание Сервисов в Манифесте приложения Запуск Сервисов Остановка Сервисов Связанные Сервисы Сервисы и Извещения</p>	2	1

Сервисы переднего плана (Foreground Services) Жизненный цикл Сервисов		
12. Виджеты (Widgets) Описание Виджетов в Манифесте приложения Создание разметки Виджета Класс AppWidgetProvider Создание Виджета Использование Конфигурационной Активности Использование Preview Image Обновление Виджетов	2	1
13. Работа картами SD и внутренним хранилищем устройства Проверка доступности носителя Доступ к файлам Совместно используемые файлы и стандартные каталоги Файлы кэша приложений	2	1
14. Загрузчики (Loaders) Обзор API Загрузчиков Применение Загрузчиков Запуск и перезапуск Загрузчиков Использование LoaderManager Использование LoaderCursor	2	1
15. Беспроводные соединения Проверка сетевых соединений Отслеживание состояния соединений ConnectivityManager и NetworkInfo Эффективное использование сетевых соединений	1	1
16. Использование AlarmManager и AlarmClock Типы будильников в Android Однократные и повторяющиеся события Области применения AlarmManager и альтернативы (Timer и Handler) Использование AlarmClock	1	1
17. Сенсоры в Android Обзор сенсоров Типы сенсоров и получение информации об их доступности Sensor Framework Мониторинг состояния сенсоров Лучшие практики при работе с сенсорами	1	1
18. Телефония и SMS Совершение звонков из приложения Определение состояния и параметров телефона Мониторинг состояния телефонного модуля Использование SMS Отправка SMS	1	1

	Получение SMS		
	19. Собственные View Использование Canvas Использование SurfaceView Использование Drawable Shape Drawable и 2D графика Модификация существующих View Создание собственных View	1	1
	20. 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	1	1
	Практическая работа №9. Стили и темы	8	2
	Практическая работа №10. Меню	8	2
	Практическая работа №11. Фрагменты	8	2
	Практическая работа №12. Многопоточность и асинхронность	8	2
	Практическая работа №13. Работа с мультимедиа	8	2
	Практическая работа №14. Настройки и состояние приложения	8	2
	Практическая работа №15. Работа с файловой системой	8	2
	Практическая работа №16. Работа с базами данных SQLite	8	2
	Практическая работа №17. Провайдеры контента	8	2
	Самостоятельная работа Примерная тематика рефератов: 1. Понятие мобильного устройства. Основные типы мобильных устройств. 2. Понятие коммуникационных технологий. 3. Технология GSM: 1G, 2G. 4. Технология GSM: 3G, 4G. 5. Технология GSM: 5G. 6. Технология WiFi. 7. Технология Bluetooth. 8. Виды организации беспроводных сетей. 9. Понятие программной платформы. 10. Платформа Android. 11. Платформа iOS. 12. Платформа WindowsPhone. 13. Язык программирования Java. 14. Типы мобильных приложений. 15. Архитектура мобильных приложений. 16. Типы соединений. 17. Конфигурации и профили J2ME. Профиль Foundation. 18. Конфигурации и профили J2ME. Профиль Personal.	10	

19. Конфигурации и профили J2ME. Профиль RMI.		
20. Конфигурации и профили J2ME. Профиль Mobile Information Device.		
Дифференцированный зачет в 6 семестре	2	
Промежуточная аттестация в форме экзамена в 5 семестре		
Всего	200	
МДК 01.04 Системное программирование	200	
Тема 1. Программирование на языке низкого уровня		
1. 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	4	1
2. 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 proc Два способа прочесть содержимое директории Разреженные файлы и специфика их применения Блокировка областей файла	4	1
3. Межпроцессное взаимодействие Механизмы межпроцессного взаимодействия Linux (Ubuntu) Неименованные каналы (pipes) Именованные каналы (named pipes) Сообщения (message queue) Разделяемая память (shared memory) Семафоры (semaphores)	4	1
4. Сокеты Сокеты в файловом пространстве имен (UNIX-сокеты) Парные сокеты (pair sockets) Сетевые сокеты (sockets)	4	1
5. Сигналы Знакомство с сигналами (signals) Отличие сигналов от других механизмов межпроцессного взаимодействия Специфика обработки сигналов (signal handling)	4	1
6. Процессы Клонирование процессов — fork()	4	1

	Замена исполняемого процесса — <code>exec()</code> Зомби (zombies) — причины возникновения и способы их устранения		
	7. Потоки Потоки и процессы Специфика построения многопоточных приложений (multithreading) Досрочное завершение потока Создание обработчика завершения потока Средства синхронизации потоков (synchronize primitives) Атрибуты потоков	4	1
	8. Демоны (службы) Отличие демона от консольной утилиты Специфика разработки демонов (daemons) Создание демона использующего сетевые сокеты	4	1
	9. Консольный ввод-вывод Специфика разработки консольных приложений Предотвращение перенаправления вывода Управление терминалом Соккрытие пароля пользователя при аутентификации	4	1
	10. Отображаемая память Отображение обычного файла Совместный доступ к файлу Частные отображения Другие применения mmap	4	1
	11. Программирование графического интерфейса с помощью GTK+ Специфика разработки 64-битных приложений Использование библиотеки ncurses Программирование графического интерфейса с помощью GTK+	4	1
	Практическая работа №1. Переменные окружений	6	2
	Практическая работа №2. Низкоуровневый ввод-вывод и файловые операции	6	2
	Практическая работа №3. Межпроцессное взаимодействие	6	2
	Практическая работа №4. Сокеты	6	2
	Практическая работа №5. Сигналы	8	2
	Практическая работа №6. Процессы	8	2
	Практическая работа №7. Потоки	8	2
	Практическая работа №8. Демоны (службы)	8	2
	Практическая работа №9. Консольный ввод-вывод	8	2
	Практическая работа №10. Отображаемая память	8	2
	Практическая работа №11. Программирование графического интерфейса с помощью GTK+	8	2
Тема 2. Разработка драйверов устройств в Linux	1. Ядро Linux Архитектура ядра linux Отличия разработки драйверов от прикладного ПО Потенциальные проблемы с безопасностью	4	1

Лицензирование модулей ядра		
2. 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	4	1
3. 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	4	1
4. 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	2	1
5. 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	2	1
6. Управление памятью в ядре Linux Специфика управления памятью в ядре Linux Преобразование адреса в ядре Отображение файла в память ядра Управление отображением в память	2	1
7. Блочные устройства Блочные устройства Регистрация блочного устройства Операции, поддерживаемые для блочных устройств Обработка запросов к блочному устройству Детали реализации блочного устройства	2	1

	Пример драйвера блочного устройства		
	8. Семафоры Конкуренция и ситуация гонок Семафоры и мьютексы Реализация семафора в ядре Linux Семафоры на чтение и запись Взаимоблокировки Пример драйвера, использующего семафоры	2	1
	9. 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	2	1
	Практическая работа №12. Модули ядра, пространства и устройства	8	2
	Практическая работа №13. Символьные устройства	8	2
	Практическая работа №14. Файловая система /proc	8	2
	Практическая работа №15. Управление памятью в ядре Linux	8	2
	Практическая работа №16. Блочные устройства	8	2
	Самостоятельная работа Примерная тематика рефератов: 1. Основные понятия и их определения; расположение системного программного обеспечения в общей структуре вычислительной системы. 2. Организация взаимодействия между аппаратурой ЭВМ, системным и прикладным программным обеспечением. 3. Классификация и структура системного программного обеспечения. 4. Концепция процессов и потоков. 5. Процессы, потоки, волокна. 6. Мультипрограммирование. 7. Мультизадачность. 8. Мультипроцессирование. 9. Алгоритмы планирования потоков. 10. Понятие фазы компиляции. Общая схема работы компиляторы. 11. Фазы компиляции их особенности. Однопроходные и многопроходные компиляторы. 12. Таблицы идентификаторов (состав, особенности построения.) 13. Методы организации таблиц идентификаторов: логарифмический поиск. 14. Алгоритм «бинарное дерево». 15. Метод хэширования. 16. Метод цепочек.	10	

17. Конечные автоматы. 18. Преобразование конечных автоматов. 19. Статическая, стековая, динамическая память: особенности организации, принципы работы и хранения переменных. 20. Два типа приложений на языке программирования C операционной системе Windows: консольные и оконные. 21. Структура консольных приложений. 22. Структура программы на языке Ассемблера. 23. Выполнение арифметических вычислений на ассемблере. 24. Операции с двоичными, восьмеричными и шестнадцатеричными числами. 25. Представление вещественных чисел в памяти ЭВМ. 26. Вставка в проект исходного модуля на языке ассемблера. 27. Использование директив в ассемблере. 28. Применение макросов. 29. Структура процессора. 30. Основные регистры процессора.		
Дифференцированный зачет в 5 семестре	2	
Промежуточная аттестация в форме экзамена в 6 семестре		
Всего	200	
Учебная практика Виды работ: 1. Разработка алгоритма решения поставленной задачи. 2. Создавать программу по разработанному алгоритму как отдельный модуль. 3. Разработка кода программного продукта на основе спецификации на уровне модуля. 4. Разработка интерфейса мобильного приложения и определение компонентов для приложения. 5. Программирование с использованием нескольких активностей. 6. Разработка мобильного приложения. 7. Проведение тестирования программного модуля по определенному сценарию. 8. Использование инструментальных средств на этапе тестирования программного продукта. 9. Осуществлении рефакторинга и оптимизации программного кода	108	
Производственная практика Виды работ: 1. Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов; – применение стандартных алгоритмов в соответствующих областях. 2. Создание программного кода в соответствии с техническим заданием (готовыми спецификациями). 3. Наличие соответствия результатов выполнения ожидаемым значениям. 4. Установление соответствия разработанных тестов используемой методике тестирования. 5. Проверка полноты группы тестов. 6. Проверка отсутствия избыточной вложенности циклов. 7. Проверка отсутствия дублирующих действий. 8. Установление разумного баланса по используемой памяти и быстродействию. 9. Выбор платформы разработки для среды выполнения. 10. Создание программного кода в соответствии с техническим заданием. 11. Тестирование приложений с использованием эмулятора.	144	

Квалификационный экзамен в 6 семестре	12	
Всего	1109	

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).

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Для реализации программы профессионального модуля предусмотрены следующие специальные помещения:

Реализация программы модуля предполагает наличие лаборатории **«Программного обеспечения и сопровождения компьютерных систем»** имеющие следующее оснащение:

- Автоматизированные рабочие места на 12-15 обучающихся (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;) или аналоги;
- Автоматизированное рабочее место преподавателя (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;)или аналоги;
- Проектор и экран;
- Маркерная доска;
- Программное обеспечение общего и профессионального назначения

4.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Мусаева Т.В. Разработка дизайна веб-приложений (1-е изд.). М.: Издательский центр «Академия», 2020 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

Перлова О.Н. Соадминистрирование баз данных и серверов (3-е изд.). М.: Издательский центр «Академия», 2022 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

3. Федорова Г.Н. Разработка, администрирование и защита баз данных (5-е изд.). М.: Издательский центр «Академия», 2021 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

4. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем (4-е изд.). М.: Издательский центр «Академия», 2020 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

5. Семакин И.Г. Основы алгоритмизации и программирования (5-е изд.). М.: Издательский центр «Академия», 2021 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

6. Перлова О.Н. Проектирование и разработка информационных систем (4-е изд.). М.: Издательский центр «Академия», 2022 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

7. Федорова Г.Н. Осуществление интеграции программных модулей (5-е изд.). М.: Издательский центр «Академия», 2021 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

8. Федорова Г.Н. Сопровождение информационных систем (2-е изд.). М.: Издательский центр «Академия», 2021 г. [электронный ресурс] – режим доступа <https://www.academia-library.ru/>

Дополнительные источники:

1. Батаев А.В. Операционные системы и среды (6-е изд.). М.: Издательский центр «Академия», 2021 г.

2. Баринов В.В. Компьютерные сети (5-е изд.). М.: Издательский центр «Академия», 2021 г.

3. Гребенюк Е.И. Технические средства информатизации (5-е изд.). М.: Издательский центр «Академия», 2021 г.

4.3. Общие требования к организации образовательного процесса:

Освоение профессионального модуля ПМ.01.Разработка модулей программного обеспечения для компьютерных систем производится в соответствии с учебным планом по специальности 09.02.07 Информационные системы и программирование и графиком учебно-воспитательного процесса колледжа, утвержденным директором колледжа.

Освоению ПМ.01.Разработка модулей программного обеспечения для компьютерных систем предшествует обязательное изучение учебных дисциплин ОП.01 Операционные системы и среды, ОП.02 Архитектура аппаратных средств, ОП.04 Основы алгоритмизации и программирования, Оп.08 Основы проектирования баз данных и др.

Лабораторные работы проводятся в специально оборудованных лабораториях.

Результатом освоения ПМ выступают ПК, оценка которых представляет собой создание и сбор свидетельств деятельности на основе заранее определенных критериев.

С целью оказания помощи студентам при освоении теоретического и практического материала, выполнения самостоятельной работы разрабатываются учебно-методические комплексы (кейсы студентов).

С целью методического обеспечения прохождения учебной и производственной практики, выполнения курсового проекта разрабатываются методические рекомендации для студентов.

Обязательным условием допуска к производственной практике в рамках профессионального модуля ПМ.01.Разработка модулей программного обеспечения для компьютерных систем является освоение учебной практики для получения первичных профессиональных навыков.

Учебная практика направлена на формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта и проводится в лабораториях образовательного учреждения преподавателями технологий.

Производственная практика (по профилю специальности) направлена на формирование у студентов общих компетенций, а также профессиональных компетенций, приобретение практического опыта. Производственная практика (по профилю специальности) проводится на предприятиях АО «ОЭЗ ПИТ «Алабуга», оснащённых современной техникой, применяющих новейшие технологии, современные программные продукты и современную организацию труда, располагающих высококвалифицированным персоналом и реальными возможностями организации производственного обучения студентов, а так же в колледже под руководством преподавателя спец.дисциплин.

Организацию и руководство практикой по профилю специальности осуществляет руководитель практики от образовательного учреждения.

По итогам производственной практики (по профилю специальности) студенты представляют:

- дневник практики
- отзыв - характеристику
- отчет о работе

Каждый этап производственной (профессиональной) практики завершается оценкой освоенных компетенций.

4.4. Кадровое обеспечение образовательного процесса:

Требования к квалификации педагогических (инженерно-педагогических) кадров, обеспечивающих обучение по ПМ 01:

- наличие высшего профессионального образования, соответствующего профилю преподаваемого модуля;
- опыт деятельности в организациях соответствующей профессиональной сферы, стажировка в профильных организациях не реже 1 раза в 3 года.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<p>Оценка «отлично» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. Дополнительно для квалификаций "Программист" и "Технический писатель": указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма</p> <p>Оценка «хорошо» - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры. Дополнительно для квалификаций "Программист" и "Технический писатель": выполнена оценка сложности алгоритма</p> <p>Оценка «удовлетворительно» - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p>
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	<p>Оценка «отлично» - программный модуль (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

	по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" на указанном языке программирования) методами объектно-ориентированного/ структурного 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	<p>Оценка «отлично» - выполнена отладка модуля (Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования); с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка «хорошо» - выполнена отладка модуля (Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования); сохранены и представлены результаты отладки.</p> <p>Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.4 Выполнять тестирование программных модулей	<p>Оценка «отлично» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами. Дополнительно для квалификации "Специалист по тестированию в области информационных технологий": выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия, сделан вывод о достаточности тестового пакета.</p> <p>Оценка «хорошо» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования. Дополнительно для квалификации "Специалист по тестированию в области информационных технологий": выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия.</p> <p>Оценка «удовлетворительно» - выполнено тестирование модуля и оформлены результаты тестирования. Дополнительно для квалификации "Специалист по тестированию в области информационных технологий": выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия с некоторыми погрешностями.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля. Дополнительно для квалификации "Специалист по тестированию в области информационных технологий": оценке тестового покрытия.</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода	<p>Оценка «отлично» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов,</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля,</p>

	<p>алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>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.</p> <p>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.</p>	<p>поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</p> <p>Защита отчетов по практическим и лабораторным работам 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 «отлично» - программный модуль (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.6 Разрабатывать модули программного</p>	<p>Оценка «отлично» - разработан модуль для заданного мобильного устройства с</p>	<p>Экзамен/зачет в форме собеседования:</p>

<p>обеспечения для мобильных платформ.</p>	<p>соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.</p> <p>Оценка «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>практическое задание по созданию модуля для заданного мобильного устройства на основе спецификации</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный модуль (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" 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 (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки (Дополнительно для квалификаций "Программист" на указанном языке программирования) методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных</p>	<p>Экзамен/зачет в форме собеседования:</p> <p>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>

	отклонений от стандартов.	
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	<p>Оценка «отлично» - выполнена отладка модуля (Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования); с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка «хорошо» - выполнена отладка модуля (Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий": с использованием инструментария среды проектирования); сохранены и представлены результаты отладки.</p> <p>Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.</p>	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

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся не только сформированность профессиональных компетенций, но и развитие общих компетенций и обеспечивающих их умений.

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.	<ul style="list-style-type: none"> - 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
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	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.	<ul style="list-style-type: none"> - 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.	<ul style="list-style-type: none"> - 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-	- соблюдение норм поведения во время	

патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	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.	

134
Прошито, пронумеровано и скреплено печатью
55 (12816996689 12817) листов
Заместитель директора по учебно-методической работе
Шараборина О.С.

«29» августа 2022 г.

МП

