Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИ ВлГУ)

Отделение среднего профессионального образования

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
«_19_» ___05___ 2026 г.

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Разработка и интеграция модулей программного обеспечения

для специальности 09.02.11 Разработка и управление программным обеспечением

Муром, 2026 г.

Рабочая программа профессионального модуля разработана на основе Федерального государственного образовательного стандарта (далее - ФГОС) по специальности среднего профессионального образования (далее - СПО) 09.02.11 Разработка и управление программным обеспечением №138 от 24 февраля 2025 года.

Кафедра-разработчик: информационных систем.

Рабочую программу составил: к.т.н., доцент Канунова Е.Е.

_____ от «05» мая 2026 г.
(подпись)

Рабочая программа рассмотрена и одобрена на заседании кафедры ИС.

Протокол № 21 от «05» мая 2026 г.

Заведующий кафедрой ИС *Андрианов Д.Е.* _____
(подпись)

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ 4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ 6
3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ 7
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ 15
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ 17

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Разработка и интеграция модулей программного обеспечения

1.1. Область применения примерной программы

Примерная программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением в части освоения основного вида профессиональной деятельности (ВПД): **Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности Программист, в том числе профессиональными (ПК) компетенциями** и соответствующих профессиональных компетенций (ПК):

- ПК 2.1. Проектировать модули программного обеспечения
- ПК 2.2. Разрабатывать модули программного обеспечения
- ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения
- ПК 2.4. Выполнять тестирование и отладку программного обеспечения
- ПК 2.5. Осуществлять документирование программных модулей программного обеспечения

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

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

иметь практический опыт:

- владения современными программными средствами, предназначенными для разработки программных модулей (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- владения современными программными средствами, предназначенными для интеграции модулей в приложение с графическим интерфейсом (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.).

уметь:

- Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Создавать программу по разработанному алгоритму как отдельный модуль. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Выполнять отладку и тестирование программы на уровне модуля. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Выполнять оптимизацию и рефакторинг программного кода (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Осуществлять разработку кода программного модуля на современных языках программирования (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Выполнять интеграцию программных модулей (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.).

знать:

- Основные этапы разработки модулей программного обеспечения. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Этапы проектирования модулей программного обеспечения (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Основные принципы отладки и тестирования программных продуктов. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Инструментарий отладки программных продуктов. (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);

- Основные виды и принципы тестирования программных продуктов (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Методы организации рефакторинга и оптимизации кода (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Основные подходы к интегрированию программных модулей (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Основы верификации программного обеспечения (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Современные технологии и инструменты интеграции (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Основные методы отладки (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.);
- Подходы к документированию программных модулей программного обеспечения (ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.).

1.3. Рекомендуемое количество часов на освоение программы профессионального модуля:

всего - 839 часов, в том числе:
максимальной учебной нагрузки обучающегося - 515 часов, включая:
обязательной аудиторной нагрузки обучающегося - 344 часа;
самостоятельной нагрузки обучающегося - 171 час;
учебной и производственной практики - 288 часов;
экзамен по модулю - 36 часов.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности 09.02.11 Разработка и управление программным обеспечением, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	**Наименование результата обучения**
ПК 2.1.	Проектировать модули программного обеспечения
ПК 2.2.	Разрабатывать модули программного обеспечения
ПК 2.3.	Выполнять интеграцию модулей и компонентов программного обеспечения
ПК 2.4.	Выполнять тестирование и отладку программного обеспечения
ПК 2.5.	Осуществлять документирование программных модулей программного обеспечения

3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

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, часов	Производственная (по профилю специальности), часов
			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		
1	2	3	4	5	6	7	8	9	10
ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.	**Раздел 1.** МДК.02.01 Разработка программных модулей	231	176	92	20	55			
ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.	**Раздел 2.** МДК.02.02 Осуществление интеграции программных модулей	131	64	32		67			
ПК 2.1., ПК 2.2., ПК 2.3., ПК 2.4., ПК 2.5.	**Раздел 3.** МДК.02.03 Поддержка и тестирование программных модулей	153	104	68		49			
	Учебная практика.Разработка и интеграция модулей программного обеспечения	144						144	
	Производственная практика. Разработка и интеграция модулей программного обеспечения	144							144
	Квалификационный экзамен	36							
	Всего:	**839**	**344**	**192**		**171**		**144**	**144**

3.2. Содержание обучения по профессиональному модулю

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
МДК.02.01 Разработка программных модулей		**211**	
	3 семестр		
Раздел 1	**Разработка программных модулей**		
Тема 1.1 Методология проектирования программных продуктов	**Содержание учебного материала**		
	Лекционные занятия. Знакомство с Qt. Структура проекта. Основные типы.	4	1
	Практические занятия. Знакомство со средой Qt Creator и методами создания	4	2

	приложений. Знакомство со средой Qt Creator и методами создания приложений с графическим интерфейсом.		
	Самостоятельная работа обучающихся. Стандарты С++ 17 и С++20. Концепции ООП.	28	3
Тема 1.2 Основы Qt	**Содержание учебного материала**		
	Лекционные занятия. Модули Qt. Механизм сигналов и слотов. Qt Designer. Быстрая разработка прототипов. Контейнерные классы в Qt. Алгоритмы в Qt:сортировка, поиск, сравнение, подсчет значений. Строки в Qt. Регулярные выражения. Произвольный тип QVariant. Списки строк QStringList.	14	1
	Практические занятия. Работа с контейнерами в среде Qt Creator: QVector. Использование контейнера QVector для решения задач обработки одномерных массивов. Работа с контейнерами в среде Qt Creator: QVector для обработки многомерных массивов. Использование контейнера QVector для решения задач обработки многомерных массивов. Работа с контейнерами в среде Qt Creator: QList. Работа с контейнерами в среде Qt Creator: QStack, QQueue.	12	2
Тема 1.3 Элементы управления	**Содержание учебного материала**		
	Лекционные занятия. Класс QWidget. Управление автоматическим размещением элементов. Элементы отображения. Кнопки, флажки, переключатели. Элементы ввода. Элементы выбора. Модель-представление.	14	1
	Лабораторные работы. Создание простейшего графического приложения. Табулирование функции. Разработка приложения с графическим интерфейсом для обработки одномерных массивов. Разработка приложения с графическим интерфейсом для обработки строковых данных. Разработка приложения с графическим интерфейсом для обработки файлов.	32	3
Тема 1.4 Графика и звук	**Содержание учебного материала**		
	Самостоятельная работа обучающихся. Графинское представление. Анимация.	6	3
	4 семестр		
Тема 1.5 Основы Qt	**Содержание учебного материала**		
	Практические занятия. Сигналы и слоты. Работа с контейнерами в среде Qt Creator: QMap. Работа с контейнерами в среде Qt Creator: QSet.	6	2
Тема 1.6 Элементы управления	**Содержание учебного материала**		
	Лабораторные работы. Разработка приложения "Калькулятор". Разработка приложения "Текстовый редактор". Создание собственных элементов управления.	12	3

Тема 1.7 События и взаимодействия с пользователем	**Содержание учебного материала**		
	Лекционные занятия. События в Qt. События клавиатуры. События мыши. Класс QPaintEvent. События таймера. События перетаскивания. Фильтры событий.	8	1
	Лабораторные работы. Создание приложений с обработкой событий от пользователя.	4	3
Тема 1.8 Графика и звук	**Содержание учебного материала**		
	Лекционные занятия. Введение в компьютерную графику. Класс QPainter. Растровые изображения. Классы QImage, QPixmap, QBitmap. Работа со шрифтами. Элементы со стилем. Создание собственных элементов управления.	8	1
	Лабораторные работы. Разработка мультимедийного приложения.	4	3
	Самостоятельная работа обучающихся. Класс QSound. QMediaPalyer. Буфер обмена и перетаскивание drag-n-drop.	4	3
Тема 1.9 Создание приложений	**Содержание учебного материала**		
	Лекционные занятия. Сохранение настроек приложения. Интернационализация приложения. Qt Linguist. Создание меню. Диалоговые окна. Класс QDialog. Главное окно, создание SDI- и MDI-приложений.	10	1
	Практические занятия. Диалоговые окна. Работа с файлом ресурсов в Qt. Сохранение настроек приложения.	6	2
	Лабораторные работы. Интернационализация приложения. Qt Linguist. Создание SDI и MDI-приложений.	8	3
	Самостоятельная работа обучающихся. Создание help-файлов.	2	3
Тема 1.10 Особые возможности Qt	**Содержание учебного материала**		
	Лекционные занятия. Работа с файлами, каталогами и потоками ввода-вывода. Дата, время и таймер. Процессы и потоки.	6	1
	Лабораторные работы. Разработка многопоточного приложения.	4	3
	Самостоятельная работа обучающихся. Работа с XML. Динамические библиотеки и система расширений. Совместное использование Qt с платформозависимыми API. Проведение тестов.	15	3
МДК.02.02 Осуществление интеграции программных модулей		**131**	
	3 семестр		
Раздел 1	**Осуществление интеграции программных модулей**		
Тема 1.1 Основные понятия	**Содержание учебного материала**		
	Лекционные занятия. Введение. Задача проектирования программных систем.	6	1

	Понятия программный продукт, программное средство (ПС). Стратегии разработки программных средств и систем: базовые стратегии разработки ПС; каскадная стратегия разработки;. Стратегии разработки программных средств и систем: инкрементная стратегия; эволюционная стратегия.		
	Практические занятия. Инструментальные средства разработки программного обеспечения. Основные понятия и стандартизация требований к программному обеспечению. Разработка и оформление технического задания.	6	2
	Самостоятельная работа обучающихся. Технология конструирования программного обеспечения. Основы UML. Работы и процессы жизненного цикла ПО. Анализ осуществимости. Поставка. Приобретение. Разработка. Эксплуатация и сопровождение. Управление конфигурацией, средства конфигурационного управления. Управление проектом. Техническое и организационное обеспечение проекта. Аудит. Обеспечение качества. Управление разработкой ПО. Назначение управления разработкой и его основные процессы. Задачи руководства проекта. Организаций процесса разработки ПО. Надзор за разработкой. Подбор и обучение кадров. Бюджет. Планирование работ. Выбор руководителя разработкой.	43	3
Тема 1.2 Организация процесса проектирования программного обеспечения	**Содержание учебного материала**		
	Лекционные занятия. Жизненный цикл ПС (ЖЦ.ПС.) Модели ЖЦ ПС (каскадная, спиральная, экстремальная). Методологии и технологии проектирования ПС. Модель по ГОСТ Р ИСО/МЭК ТО 15271-2002; V-образная модель. Модели ЖЦ, реализующие эволюционную стратегию разработки ПС. Классификация проектов по разработке ПС. Процедура выбора модели ЖЦ ПС. Адаптация модели ЖЦ разработки ПС к условиям конкретного проекта. Методы восходящего проектирования. Методы расширения ядра. Метод Джексона. Оценка структурного разбиения ПС. Организация процесса проектирования программного обеспечения (ПО). Стандарты организации процесса проектирования ПО (ГОСТ 19, ГОСТ 34, ISO/ISE 12207). Общие принципы, обеспечивающие качество ПС. Основные примитивы качества ПС сложность, корректность, надежность, трудоемкость…. Подходы к обеспечению основных критериев качества. Измерения и оценка качества. Модели быстрой разработки ПО: базовая модель RAD; RAD-модель, основанная на моделировании предметной области; RAD-модель параллельной разработки ПО. Формирование требований. Анализ требований. Язык документирования требований. Управление процессом выработки требований. Спецификация качества. Функциональная спецификация. Контроль внешнего описания. Проектирование	26	1

	ПС. Уровни проектирования. Методы проектирования структуры. Аналитические и синтетические подходы к проектированию. Декомпозиция и абстракция при проектировании. Структурный подход к проектированию ПС. Методология функционального моделирования SADT. Функцио¬наль¬ной модель. Декомпозиция системы. Моделирование потоков данных (процессов). Внешние сущности. Системы и подсистемы. Процессы. Хранилища данных. Потоки данных.		
	Практические занятия. Разработка проекта программного обеспечения. Проектирование автоматизированной системы. Определение образа и границ проекта. Моделирование требований. Диаграммы вариантов использования. Моделирование требований. Диаграммы потоков данных. Диаграммы «сущность-связь». Диаграмма переходов состояний.	8	2
	Самостоятельная работа обучающихся. Документирование ПС. Документирование; создание и использование в процессе разработки ПС. Пользовательская документация ПС. Документация по сопровождению ПС. Другие виды документации.	4	3
Тема 1.3 Декомпозиция, абстракция, спецификация	**Содержание учебного материала**		
	Практические занятия. Разработка структурного алгоритма.	2	2
	Лабораторные работы. Спецификация и процедурная абстракция.	4	3
	Самостоятельная работа обучающихся. Основы объектно-ориентированного анализа и проектирования. Декомпозиция и абстракция. Абстракция. Виды абстракций. Процедурная абстракция. Спецификации процедурных абстракций. Реализация процедур. Создание процедурных абстракций. Абстракция данных. Спецификации абстракций данных. Реализация абстракции данных. Функция абстракции. Свойства абстракции данных (изменяемость, полнота, анализ типов).	12	3
Тема 1.4 Управление разработкой ПО	**Содержание учебного материала**		
	Лабораторные работы. Автономное тестирование программного модуля. Самодокументирование программ.	8	3
	Самостоятельная работа обучающихся. роектирование интерфейса с пользователем. Человеческие факторы спецификаций по инженерной психологии (эргономике). Проектирование интерфейса. Методы защиты программ и данных. Понятие защищённой системы (определение, свойства). Методы создания безопасных систем обработки информации.	8	3
Тема 1.5 Обеспечение качества ПС	**Содержание учебного материала**		
	Лабораторные работы. Системы управления версиями ПО.	4	3

МДК.02.03 Поддержка и тестирование программных модулей		**153**	
	4 семестр		
Раздел 1	**Тестирование программных модулей**		
Тема 1.1 Тестирование на этапах жизненного цикла проекта	**Содержание учебного материала**		
	Лекционные занятия. Основные понятия и определения тестирования. Тестирование на этапах жизненного цикла проекта. Автоматические тесты при разработке модулей программного обеспечения.	4	1
	Практические занятия. Организация тестирования в команде разработчиков.	4	2
	Лабораторные работы. Интеграционное тестирование. Приемочное тестирование. Тестирование юзабилити. Инструмент Postman.	16	3
	Самостоятельная работа обучающихся. Стандарты, регламентирующие процесс тестирования. Пирамида тестирования. Тестирование REST API. Организация тестирования в команде разработчиков. Инструменты тестировщика ПО (QA-инженера). Требования к тестам. Рассмотрение примеров хорошего и плохого написания модульных тестов. Тестирование с использованием искусственного интеллекта.	34	3
Тема 1.2 Виды ошибок	**Содержание учебного материала**		
	Лекционные занятия. Виды ошибок. Методы отладки.	4	1
	Практические занятия. Основы тестирования и классификация ошибок. Интуитивное тестирование.	4	2
Тема 1.3 Отчет об ошибке	**Содержание учебного материала**		
	Лекционные занятия. Отчет об ошибке.	2	1
	Практические занятия. Составление отчета об ошибке.	4	2
Тема 1.4 Проектирование и разработка тестов	**Содержание учебного материала**		
	Лекционные занятия. Классификация тестирования по уровням. Характеристики хорошего теста. V-модель разработки ПО. Позитивные и негативные тесты. Тестирование производительности программного обеспечения. Методы разработки тестов. Тестирование "белым ящиком". Методы разработки тестов. Тестирование "черным ящиком". Модульное тестирование. Интеграционное тестирование. Регрессионное тестирование. Рефакторинг кода. Оптимизация программ. Инструментальные средства для отладки и оптимизации программного обеспечения.	26	1
	Практические занятия. Тестирование "белым ящиком". Тестирование "черным ящиком". Автоматизация процесса тестирования программного продукта.	24	2

	Создание unit тестов. Автоматическое тестирование Selenium. Проведение нагрузочного тестирования.		
	Лабораторные работы. Модульное тестирование. Нефункциональное тестирование. Ручное тестирование. Тестирование API.	16	3
	Самостоятельная работа обучающихся. Методики разработки тестов. Автоматизация процесса тестирования программного продукта. Инструменты для тестирования юзабилити.	15	3
Учебная практика **Виды работ:** работа с документацией, анализ предметной области. составление плана управления проектом, оценка трудозатрат проекта ознакомление с инструментальными средствами разработки ПО оформление технических документов		144	
Производственная практика **Виды работ:** разработка спецификаций классов, функций, алгоритмов, разработка программных модулей разработка информационно-логических моделей и алгоритмов программного проекта анализ и обоснование выбора виртуальной среды реализации программ работа в средах визуального программирования согласно индивидуальному заданию разработка руководства пользователя, справочной системы оформление отчета		144	
Примерная тематика курсовых работ / индивидуальных проектов: 1. Программа сравнения методов сортировки 2. Решение систем уравнений матричным методом 3. Калькулятор комплексных чисел 4. Перевод из римской системы счисления в десятичную 5. Решение систем уравнений методом Крамера 6. Шифрование текста методом замены, шифром Цезаря 7. Инженерный калькулятор 8. Калькулятор над числами в различных системах счисления 9. Построитель графиков функций 10. Логический калькулятор 11. Построение частотного словаря по тексту 12. Цифровые часы с будильником		20	

13. Стрелочные часы с будильником 14. Словарь терминов 15. Морской бой 16. Поиск экстремумов функции 17. Игра Пятнашки 18. Автоматическое составление характеристик на студентов 19. Подсчет материалов для ремонта 20. Формирование открыток 21. Графический редактор 22. Поиск синонимов слов 23. Калькулятор комплексных чисел 24. Игра крестики-нолики 25. Матричный калькулятор 26. Проверка орфографии 27. Игра Память 28. Клавиатурный тренажер 29. Система анкетирования студентов 30. Расчет интегралов функции 31.Разработка приложения «Расписание студента» 32. Разработка приложения для парикмахерской 33. Разработка приложения для кредитной организации 34. Разработка приложения для управляющей компании ЖКХ 35. Разработка приложения для автосалона 36. Разработка приложения для автомастерской 37. Разработка приложения для компании, занимающейся ИТ для службы поддержки 38. Разработка приложения для компании по оказанию бытовых услуг 39. Разработка приложения для социального работника 40. Разработка приложения «Органайзер»			
Квалификационный экзамен		36	
	Всего:	819	

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Требования к минимальному материально–техническому обеспечению

Лекционная аудитория

Техническими средствами не оборудована

Лаборатория распределенных систем

12 персональных компьютеров; проектор Nec V300X; экран настенный Lumien Master Picture

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

Mozilla Firefox (MPL)

FireBird (Initial Developer's Public License и InterBase Public Licence)

Pycharm Community Edition (проприетарная лицензия и Apache License 2.0)

IntelliJ IDEA (Apache License 2.0)

РЕД ОС (Соглашение №140/05-21У от 18.05.2021 года о сотрудничестве в области науки, развития инновационной деятельности)

РЕД "База данных" (Соглашение №140/05-21У от 18.05.2021 года о сотрудничестве в области науки, развития инновационной деятельности)

Foxit Reader (Foxit EULA)

Java Development Kit (dk.java.net Terms of Use)

QT Creator ((L)GPL)

4.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, интернет–ресурсов, дополнительной литературы.

Основные источники:

1. Зубкова Т.М. Технология разработки программного обеспечения [Электронный ресурс]: учебное пособие для СПО/ Зубкова Т.М.— Электрон. текстовые данные.— Саратов: Профобразование, 2019.— 468 c.. http://www.iprbookshop.ru/86208.html
2. Хабаров, А. Н. Разработка программных приложений : учебник / А. Н. Хабаров, А. Н. Ермакова. — Ставрополь : АГРУС, 2025. — 208 c. . https://www.iprbookshop.ru/156614.html
3. Николюкин, М. С. Тестирование программного обеспечения : учебное пособие / М. С. Николюкин, В. В. Конкина, К. И. Патутин. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2025. — 80 c. https://www.iprbookshop.ru/154963.html
4. Котляров, В. П. Основы тестирования программного обеспечения : учебное пособие для СПО / В. П. Котляров. — 2-е изд. — Саратов : Профобразование, 2025. — 336 c. . https://www.iprbookshop.ru/153351.html
5. Моренкова, О. И. Программирование на C++ в среде разработки Qt Creator : практикум для СПО / О. И. Моренкова, А. Ю. Голошубов. — Саратов : Профобразование, 2024. — 89 c. . https://www.iprbookshop.ru/133498.html
6. Лебеденко, Л. Ф. Программирование на C++ : учебное пособие для СПО / Л. Ф. Лебеденко, О. И. Моренкова. — Саратов : Профобразование, 2024. — 199 c.. https://www.iprbookshop.ru/133497.html
7. Кузнецов, А. С. Системное программирование : учебное пособие / А. С. Кузнецов, И. А. Якимов, П. В. Пересунько. — Красноярск : Сибирский федеральный университет, 2018. — 170 c. . https://www.iprbookshop.ru/84121.html

Дополнительные источники:

1. Карпович, Е. Е. Методы тестирования и отладки программного обеспечения : учебник / Е. Е. Карпович. — Москва : Издательский Дом МИСиС, 2020. — 136 с.. https://www.iprbookshop.ru/106722.html
2. Астапчук, В. А. Технологии и методы программирования : учебное пособие / В. А. Астапчук, Е. Н. Павенко. — Новосибирск : Новосибирский государственный технический университет, 2024. — 132 с.. https://www.iprbookshop.ru/155765.html
3. Новые информационные технологии : учебное пособие / В. П. Дьяконов, И. В. Абраменкова, А. А. Пеньков [и др.] ; под редакцией В. П. Дьяконова. — Москва : СОЛОН-Пресс, 2021. — 640 с. . https://www.iprbookshop.ru/141997.html
4. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 350 с. . https://www.iprbookshop.ru/142310.html
5. Белева, Л. Ф. Программирование на языке С++ : учебное пособие / Л. Ф. Белева. — Москва : Ай Пи Ар Медиа, 2024. — 81 с. . https://www.iprbookshop.ru/134887.html
6. Сергеев, М. Ю. Программирование задач с применением структурированных данных : лабораторный практикум / М. Ю. Сергеев, Н. И. Гребенникова, Т. И. Сергеева. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2023. — 165 с.. https://www.iprbookshop.ru/131026.html
7. Мерзлякова, Е. Ю. Визуальное программирование и человеко-машинное взаимодействие / Е. Ю. Мерзлякова. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2022. — 49 с. . https://www.iprbookshop.ru/125262.html

Интернет-ресурсы:

1. Документация по C++ - https://learn.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170
2. Документация по Qt Creator - https://doc.qt.io/qtcreator/index.html

4.3. Общие требования к организации образовательного процесса

Освоению модуля «Разработка и интеграция модулей программного обеспечения» должно предшествовать изучение дисциплин "Основы алгоритмизации и программирования", "Язык программирования Python", "Технологии объектно-ориентированного программирования". При подготовке к квалификационному экзамену обучающимся оказываются консультации. Во время самостоятельной подготовки учащимся должен быть предоставлен доступ в Интернет. Требования к учебно-методической документации: наличие рекомендаций к выполнению лабораторных, практических и самостоятельных работ.

4.4. Кадровое обеспечение образовательного процесса

Требования к квалификации педагогических (инженерно-педагогических) кадров, обеспечивающих обучение междисциплинарному курсу (курсам): наличие высшего профессионального образования, соответствующего профилю модуля «Разработка и интеграция модулей программного обеспечения».

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результаты обучения (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ПК 2.1. Проектировать модули программного обеспечения	знать: Основные этапы разработки модулей программного обеспечения.; знать: Этапы проектирования модулей программного обеспечения ; знать: Основные принципы отладки и тестирования программных продуктов. ; знать: Инструментарий отладки программных продуктов.; знать: Основные виды и принципы тестирования программных продуктов; знать: Методы организации рефакторинга и оптимизации кода; знать: Основные подходы к интегрированию программных модулей; знать: Основы верификации программного обеспечения; знать: Современные технологии и инструменты интеграции; знать: Основные методы отладки; знать: Подходы к документированию программных модулей программного обеспечения; уметь: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.; уметь: Создавать программу по разработанному алгоритму как отдельный модуль.; уметь: Выполнять отладку и тестирование программы на уровне модуля.; уметь: Выполнять оптимизацию и рефакторинг программного кода; уметь: Осуществлять разработку кода программного модуля на современных языках программирования; уметь: Выполнять интеграцию программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для разработки программных	тест, защита курсовой работы; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест; тест; тест; тест; тест; тест; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест, защита курсовой работы;

	модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для интеграции модулей в приложение с графическим интерфейсом;	
ПК 2.2. Разрабатывать модули программного обеспечения	знать: Основные этапы разработки модулей программного обеспечения.; знать: Этапы проектирования модулей программного обеспечения ; знать: Основные принципы отладки и тестирования программных продуктов. ; знать: Инструментарий отладки программных продуктов.; знать: Основные виды и принципы тестирования программных продуктов; знать: Методы организации рефакторинга и оптимизации кода; знать: Основные подходы к интегрированию программных модулей; знать: Основы верификации программного обеспечения; знать: Современные технологии и инструменты интеграции; знать: Основные методы отладки; знать: Подходы к документированию программных модулей программного обеспечения; уметь: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.; уметь: Создавать программу по разработанному алгоритму как отдельный модуль.; уметь: Выполнять отладку и тестирование программы на уровне модуля.; уметь: Выполнять оптимизацию и рефакторинг программного кода; уметь: Осуществлять разработку кода программного модуля на современных языках программирования; уметь: Выполнять интеграцию программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для разработки программных модулей ; иметь практический	тест, защита курсовой работы; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест; тест; тест; тест; тест; тест; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест, защита курсовой работы;

	опыт: владения современными программными средствами, предназначенными для интеграции модулей в приложение с графическим интерфейсом;	
ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения	знать: Основные этапы разработки модулей программного обеспечения.; знать: Этапы проектирования модулей программного обеспечения ; знать: Основные принципы отладки и тестирования программных продуктов. ; знать: Инструментарий отладки программных продуктов.; знать: Основные виды и принципы тестирования программных продуктов; знать: Методы организации рефакторинга и оптимизации кода; знать: Основные подходы к интегрированию программных модулей; знать: Основы верификации программного обеспечения; знать: Современные технологии и инструменты интеграции; знать: Основные методы отладки; знать: Подходы к документированию программных модулей программного обеспечения; уметь: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.; уметь: Создавать программу по разработанному алгоритму как отдельный модуль.; уметь: Выполнять отладку и тестирование программы на уровне модуля.; уметь: Выполнять оптимизацию и рефакторинг программного кода; уметь: Осуществлять разработку кода программного модуля на современных языках программирования; уметь: Выполнять интеграцию программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для разработки программных модулей ; иметь практический опыт: владения современными	тест, защита курсовой работы; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест; тест; тест; тест; тест; тест; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест, защита курсовой работы;

	программными средствами, предназначенными для интеграции модулей в приложение с графическим интерфейсом;	
ПК 2.4. Выполнять тестирование и отладку программного обеспечения	знать: Основные этапы разработки модулей программного обеспечения.; знать: Этапы проектирования модулей программного обеспечения ; знать: Основные принципы отладки и тестирования программных продуктов. ; знать: Инструментарий отладки программных продуктов.; знать: Основные виды и принципы тестирования программных продуктов; знать: Методы организации рефакторинга и оптимизации кода; знать: Основные подходы к интегрированию программных модулей; знать: Основы верификации программного обеспечения; знать: Современные технологии и инструменты интеграции; знать: Основные методы отладки; знать: Подходы к документированию программных модулей программного обеспечения; уметь: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.; уметь: Создавать программу по разработанному алгоритму как отдельный модуль.; уметь: Выполнять отладку и тестирование программы на уровне модуля.; уметь: Выполнять оптимизацию и рефакторинг программного кода; уметь: Осуществлять разработку кода программного модуля на современных языках программирования; уметь: Выполнять интеграцию программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для разработки программных модулей ; иметь практический опыт: владения современными программными средствами,	тест, защита курсовой работы; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест; тест; тест; тест; тест; тест; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест, защита курсовой работы;

	предназначенными для интеграции модулей в приложение с графическим интерфейсом;	
ПК 2.5. Осуществлять документирование программных модулей программного обеспечения	знать: Основные этапы разработки модулей программного обеспечения.; знать: Этапы проектирования модулей программного обеспечения ; знать: Основные принципы отладки и тестирования программных продуктов. ; знать: Инструментарий отладки программных продуктов.; знать: Основные виды и принципы тестирования программных продуктов; знать: Методы организации рефакторинга и оптимизации кода; знать: Основные подходы к интегрированию программных модулей; знать: Основы верификации программного обеспечения; знать: Современные технологии и инструменты интеграции; знать: Основные методы отладки; знать: Подходы к документированию программных модулей программного обеспечения; уметь: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.; уметь: Создавать программу по разработанному алгоритму как отдельный модуль.; уметь: Выполнять отладку и тестирование программы на уровне модуля.; уметь: Выполнять оптимизацию и рефакторинг программного кода; уметь: Осуществлять разработку кода программного модуля на современных языках программирования; уметь: Выполнять интеграцию программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для разработки программных модулей ; иметь практический опыт: владения современными программными средствами, предназначенными для интеграции	тест, защита курсовой работы; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест; тест; тест; тест; тест; тест; тест, защита курсовой работы; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест; тест, защита курсовой работы; тест, защита курсовой работы;

	модулей в приложение с графическим интерфейсом;	

Приложение

Фонд оценочных материалов (средств) по дисциплине

Разработка и интеграция модулей программного обеспечения

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Дисциплина "Разработка программных модулей"

1.QStack Опишите стек целых чисел. Заполните его длинами строк, считанных из файла. Распечатайте на экране содержимое стека. Укажите номер и длину последней самой короткой строки файла

2. QStack Дается строка s состоящая из следующих символов '(', ')', '{', '}', '[' и ']', определить, является ли входная строка допустимой.

Входная строка действительна, если:

1) Открытые скобки должны быть закрыты однотипными скобками.

2) Открытые скобки должны быть закрыты в правильном порядке.

3) Каждой закрывающей скобке соответствует открытая скобка того же типа

3. Матрицы Найти наибольшие элементы каждой строки матрицы Z(16,16) и поместить их на главную диагональ. Вывести полученную матрицу

4. QString Дано предложение, в котором имеются одна буква с и одна буква Т. Определить, какая из них встречается раньше (при просмотре слова слева направо).

5. QString Дано предложение, в котором имеются одна буква с и одна буква Т. Определить, какая из них встречается раньше (при просмотре слова слева направо).

6. Получить все четырехзначные счастливые номера. Счастливым называется номер, у которого сумма первых двух цифр номера равна сумме последних двух цифр. Определить функцию для расчета суммы цифр двухзначного числа

7. QVector Разработать программу поиска в векторе всех элементов со значением большим или равным среднему арифметическому между минимальным и максимальным элементами вектора и удаления найденных элементов

8. QMap Разработать программу «Телефонный справочник», где будет храниться словарь из имен и прикрепленных к ним номеров. Реализовать поиск записей по ключу или значению.

9. QSet Заданы имена девочек. Определить, какие из этих имен встречаются во всех

классах данной параллели, какие есть только в некоторых классах, какие из этих

имен не встречаются ни в одном классе.

10. QSet Разработать программу заполнения и вывода множества простых чисел из диапазона, задаваемого пользователем.

11. Добавить в приложение «Калькулятор» возможность вычисления тригонометрических функций - sin, cos, tan.

12. Добавить в приложение «Калькулятор» возможность работы с «памятью» (сохранение промежуточных результатов во временной ячейке памяти)

13. Добавить в приложение «Калькулятор» возможность работы с двоичными числами

14. Добавить в приложение «Калькулятор» возможность работы с восьмеричными числами

15. Добавить в приложение «Калькулятор» возможность работы с шестнадцатеричными цифрами

16. Добавить в приложение «Калькулятор» возможность вычисления факториала

17. Добавить в приложение «Калькулятор» возможность вычисления логарифма десятичного и натурального

18. Добавить в приложение «Калькулятор» возможность вычисления степени с произвольным основанием и произвольным показателем.

19. Добавить в приложение «Калькулятор» возможность вычисления корня произвольной степени. Предусмотреть контроль ввода отрицательного аргумента

20. Добавить в приложение «Калькулятор» возможность округления чисел с плавающей запятой до заданного числа знаков после запятой

Регламент проведения защиты и оценивание курсовой работы / индивидуального проекта

Защита курсовой работы проводится на заседании Комиссии. На защите кроме руководителя курсовой работы присутствует один или два члена комиссии.

Может присутствовать научный руководитель курсовой работы.

Защита курсовой работы осуществляется в устной форме с показом презентации на проекторе.

Продолжительность защиты, как правило, не превышает 20 минут.

Для доклада основных положений курсовой работы, обоснования выводов и предложений студенту предоставляется не более 5-7 минут.

После доклада студент должен ответить на замечания научного руководителя, а также на заданные членами Комиссии вопросы по теме курсовой работы.

По результатам защиты выставляется оценка, определяемый оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценка курсовой работы даётся членами Комиссии по защите курсовых работ на её закрытом заседании. Комиссией принимается во внимание содержание работы, обоснованность выводов и предложений, содержание доклада студента, отзыв научного руководителя, уровень теоретической и практической подготовки

студента, а также соблюдение требований по порядку оформления работы. Отзыв научного руководителя является важным, но не доминирующим фактором в определении оценки.

Оценки курсовых работ объявляются в тот же день после оформления в установленном порядке протокола заседания Комиссии. После защиты

курсовая работа хранится на профильной кафедре в течение 3 (трех) лет. Если защита курсовой работы признана неудовлетворительной, то

Комиссия устанавливает, может ли студент представить к повторной защите ту же работу с соответствующей доработкой, или должен написать

новую.

При оценке курсовой работы учитываются следующие факторы: степень самостоятельности выполнения работы, новизна выводов и конструктивность предложений, качество используемого материала, уровень грамотности (общий и специальный), а также порядок оформления.

Общими критериями оценки качества курсовой работы являются: соответствие содержания курсовой работы, по которой она выносится на защиту;

соответствие требованиям, предъявляемым к форме и содержанию; уровень защиты курсовой работы.

Оценка «отлично» выставляется за курсовую работу, которая имеет грамотно изложенный теоретический раздел, характеризуется логичным и последовательным изложением материала с соответствующими выводами и обоснованными предложениями по практическому применению результатов исследования. Такая работа должна иметь положительный отзыв научного руководителя. При её защите студент показывает глубокие знания вопросов темы, свободно оперирует данными исследования, вносит обоснованные предложения по практическому применению результатов исследования, четко отвечает на

поставленные вопросы.

Оценка «хорошо» выставляется за работу, которая, имеет грамотно изложенный теоретический раздел, характеризуется логичным и последовательным изложением материала, однако имеет не вполне обоснованные выводы и не имеет предложений по практическому применению результатов исследования.

Работа имеет в целом положительный отзыв научного руководителя, но содержит ряд незначительных замечаний. При её защите студент показывает знание вопросов темы, оперирует данными исследования, без особых затруднений отвечает на поставленные вопросы.

Оценка «удовлетворительно» выставляется за работу, которая характеризуется непоследовательностью в изложения материала.

Представленные выводы автора необоснованны. В отзыве научного руководителя имеются серьёзные замечания по содержанию работы и

методике анализа. При её защите студент проявляет неуверенность, показывает слабое знание вопросов темы, не даёт полного аргументированного ответа на заданные вопросы.

Оценка «неудовлетворительно» выставляется за работу, которая не отвечает требованиям, изложенным в учебно-методических указаниях по выполнению курсовых работ. В курсовой работе нет выводов, либо они носят декларативный характер. В отзыве научного руководителя имеются серьезные замечания. При защите курсовой работы студент затрудняется отвечать на поставленные вопросы по теме, не знает теории вопроса, при ответе допускает существенные ошибки. К защите не подготовлена презентация.

Оценка «неудовлетворительно» может быть также выставлена студенту, представившему на защиту чужую курсовую работу, написанную и уже защищенную в другом вузе или на другой кафедре. Подобные работы вообще не принимаются к рассмотрению членами комиссии, а студент обязан разработать новую тему, которая определяется профильной кафедрой.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Лабораторные и практические работы	20
Рейтинг-контроль 2	Лабораторные и практические работы	20
Рейтинг-контроль 3	Лабораторные и практические работы	20
Посещение занятий студентом		10
Дополнительные баллы (бонусы)		10
Выполнение семестрового плана самостоятельной работы		20

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой. Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Оценочные средства для промежуточной аттестации приведены в приложении 1.

Методические материалы, характеризующих процедуры оценивания

1. Qt. Интерфейс Qt Creator. Модули Qt. Состав Qt
2. Структура проекта Qt с графическим интерфейсом. Файл проекта (*.pro)
3. Работа с текстовыми строками в Qt.

Класс QString. Основные функции для работы со строками. Пример

4. Механизм сигналов и слотов

5. Преимущества использования механизма сигналов и слотов. Недостатки
6. Сигналы. Пример определения сигнала. Ключевое слово emit
7. Слоты. Пример реализации слота. Метод connect. Пример
8. Библиотека Qt. Класс QWidget
9. Управление автоматическим размещением элементов. Классы компоновки виджетов
10. Работа с файлами в Qt Creator. Класс QFile. Методы класса QFile. Пример
11. Элементы отображения в Qt. Надписи, индикатор процесса, электронный индикатор
12. Элементы ввода в Qt. Класс QLineEdit. Слоты, сигналы и открытые методы QLineEdit
13. Элементы ввода даты и времени в Qt. QDateTimeEdit. Слоты, сигналы, методы QDateTimeEdit.
14. Классы QDateTime, QDate, QTime.
15. Редактор текста QTextEdit. Методы, слоты и сигналы QTextEdit.
16. Кнопки. Класс QAbstractButton. Сигналы класса QAbstractButton
17. Класс QPushButton. Ползунок QSlider. Виджеты QCheckBox и QRadioButton. Методы классов
18. Работа с диалогами. Стандартные диалоговые окна. Примеры
19. Управление событиями. События клавиатуры QKeyEvent
20. События мыши Класс QMouseEvent
21. Дайте определение тестированию. В чем заключается основная задача тестирования?
22. Что такое отладка ПО?
23. Назовите этапы тестирования.
24. Назовите основные принципы тестирования.
25. Что такое верификация и валидация?
26. Дайте определение дефекту (багу)? Как определить – дефект перед нами или нет?
27. Назовите 3 основных вида ошибок.
28. Назовите методы отладки.
29. Дайте определение отчета об ошибке и его цели.
30. Серьезность и приоритет багов.
31. Жизненный цикл бага.
32. Позитивное и негативное тестирование.
33. Основные модели разработки ПО. Расскажите об одной модели разработки ПО.
34. Тестирование «белым ящиком» - определение, достоинства, недостатки.
35. Тестирование «черным ящиком» - определение, достоинства, недостатки.
36. Модульное тестирование - определение, достоинства, недостатки.
37. Интеграционное тестирование - определение, достоинства, недостатки.
38. Регрессионное тестирования - определение, достоинства, недостатки.
39. Рефакторинг.
40. Оптимизация программ.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки	***Высокий уровень***

		работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. Что не является обязательным элементом объектной модели:
a) абстрагирование;
b) инкапсуляция;
c) устойчивость;
d) модульность;

2. время исполнения проекта – это:
a) это время, необходимое для создания рабочей версии программы, т.е. версии, готовой для работы в производственных условиях и выполняющей предусмотренные функции;
b) это время, необходимое для создания базы данных, готовой для работы в производственных условиях и выполняющей предусмотренные функции;
c) это время, необходимое для создания рабочей версии программы, т.е. версии, готовой для работы в производственных условиях , но не выполняющей предусмотренные функции;
d) нет правильного варианта;

3. технология программирования не включает:
a) указание последовательности выполнения технологических операций;
b) перечисление условий, при которых выполняется та или иная операция;
c) описания самих операций, где для каждой операции определены исходные данные, результаты, а также инструкции, нормативы, стандарты, критерии и методы оценки и т. п;
d) время выполнения операций;

4. какой из элементов не является элементом объектной модели:
a) модульность;
b) параллелизм;
c) последовательность;
d) иерархия;

5. Программа-это:
a) последовательность действий, направленных на выполнение их некоторым исполнителем.;
b) это точно определенная последовательность простых действий, обеспечивающих решение любой задачи из некоторого класса;
c) комплекс программ, которые обеспечивают управление компонентами компьютерной системы;
d) указание последовательности выполнения технологических операций;

6. Что не включает в себя качество ПО:
a) эффективность;
b) мобильность;
c) практичность;
d) управление доступом;

7. Массив-это:
a) упорядоченная последовательность переменных разных типов;
b) упорядоченная последовательность констант одного типа;
c) упорядоченная последовательность переменных одного типа;
d) нет верного ответа;

8. К структурированным типам относятся
1) вещественный, 2) файл, 3) множество,
4) символьный, 5) диапазон, 6) целый

9. Что такое массив?
a) Ограниченная упорядоченная совокупность однотипных величин;
b) Ограниченная совокупность различных элементов;
c) Совокупность ограниченного числа логически связанных компонент, принадлежащих к разным типам.

10. В операторе цикла с предусловием тело цикла …
a) обязательно выполняется более 1 раза;
b) выполняется по крайней мере 1 раз;
c) может не выполниться вообще;
d) выполняется строго определенное число раз.

11. В операторе цикла с постусловием тело цикла
a) обязательно выполняется более 1 раза;

b) выполняется по крайней мере 1 раз;
c) может не выполниться вообще;
d) выполняется строго определенное число раз.

12. Алгоритм-это:
a) последовательность действий, направленных на выполнение их некоторым исполнителем;
b) Комплекс программ, которые обеспечивают управление компонентами компьютерной системы;
c) это точно определенная последовательность простых действий, обеспечивающих решение любой задачи из некоторого класса;
d) все ответы верны

13. Какой фактор не учитывается при выборе языка программирования?
a) Иерархичность;
b) Поддержка и сообщество;
c) Время исполнения проекта;
d) Целевая платформа;

14. Технологией программирования называют:
a) последовательность действий, направленных на выполнение их некоторым исполнителем;
b) совокупность методов и средств, используемых в процессе разработки программного обеспечения;
c) Комплекс программ, которые обеспечивают управление компонентами компьютерной системы;
d) Нет правильного ответа;

15. Декомпозицией называют:
a) выделение существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов;
b) деятельность по предотвращению утечки защищаемой информации;
c) научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач;
d) все ответы верны;

16. Абстрагирование-это:
a) выделение существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов;
b) научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач;
c) неконтролируемое распространение конфиденциальной информации путём её разглашения или несанкционированного доступа к ней;
d) все верно;

17. Инкапсуляция-это:
a) это точно определенная последовательность простых действий, обеспечивающих решение любой задачи из некоторого класса;
b) свойство системы, позволяющее объединить данные и методы, работающие с ними в классе;
c) совокупность методов и средств, используемых в процессе разработки программного обеспечения;
d) все ответы верны;

18. Перевести число 95 из десятичной системы счисления в восьмеричную:
a) 116;
b) 1871;
c) 137;
d) A16

19. По каким характеристикам можно определить качество ПО:
a) Функциональность;
b) открытость;
c) надежность;
d) наследственность;

20. Идентификатор - это:
a) последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
b) неизменяемые объекты языка (константы)
c) последовательность латинских и русских букв
d) способ кодирования, допустимые преобразования над значением данной переменной

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/course/view.php?id=4565

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.