

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
« 23 » 05 2023 г.

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Разработка модулей программного обеспечения для компьютерных систем

для специальности 09.02.07 Информационные системы и программирование

Муром, 2023 г.

Рабочая программа профессионального модуля разработана на основе Федерального государственного образовательного стандарта (далее - ФГОС) по специальности среднего профессионального образования (далее - СПО) 09.02.07 Информационные системы и программирование №1547 от 09 декабря 2016 года.

Кафедра-разработчик: информационных систем.

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

от «17» мая 2023 г.

(подпись)

Рабочая программа рассмотрена и одобрена на заседании кафедры ИС.

Протокол № 18

от «17» мая 2023 г.

Заведующий кафедрой ИС *Андреианов 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 модулей программного обеспечения для компьютерных систем

1.1. Область применения примерной программы

Примерная программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.07 Информационные системы и программирование в части освоения основного вида профессиональной деятельности (ВПД): **Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности Программист, в том числе профессиональными (ПК) и общими (ОК) компетенциями и соответствующих профессиональных компетенций (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
- ПК 2.3.. Выполнять отладку программного модуля с использованием специализированных программных средств
- ПК 2.4.. Осуществлять разработку тестовых наборов и тестовых сценариев для 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:

- разработки кода программного продукта на основе готовой спецификации на уровне модуля (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- использования инструментальных средств на этапе отладки программного продукта (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- проведения тестирования программного модуля по определенному сценарию (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- использования инструментальных средств на этапе отладки программного продукта (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- разработки мобильных приложений (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.).

уметь:

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);

- создавать программу по разработанному алгоритму как отдельный модуль (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- выполнять отладку и тестирование программы на уровне модуля (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- осуществлять разработку кода программного модуля на современных языках программирования (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- выполнять оптимизацию и рефакторинг программного кода (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.).

знать:

- основные этапы разработки программного обеспечения (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- основные принципы технологии структурного и объектно-ориентированного программирования (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- способы оптимизации и приемы рефакторинга (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.);
- основные принципы отладки и тестирования программных продуктов (ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.).

1.3. Рекомендуемое количество часов на освоение программы профессионального модуля:

всего - 493 часа, в том числе:

максимальной учебной нагрузки обучающегося - 493 часа, включая:

обязательной аудиторной нагрузки обучающегося - 410 часов;

самостоятельной нагрузки обучающегося - 83 часа;

учебной и производственной практики - 144 часа.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результатом освоения профессионального модуля является овладение обучающимися видом профессиональной деятельности 09.02.07 Информационные системы и программирование, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3.	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4.	Выполнять тестирование программных модулей
ПК 1.5.	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6.	Разрабатывать модули программного обеспечения для мобильных платформ
ПК 2.3.	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.4.	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения
ОК 01.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04.	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 06.	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 09.	Использовать информационные технологии в профессиональной деятельности
ОК 10.	Пользоваться профессиональной документацией на государственном и иностранном языке
ОК 11.	Планировать предпринимательскую деятельность в профессиональной сфере

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 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 2.3.	Раздел 1. МДК.01.01 Разработка программных модулей	162	132	132	0	30			
ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 2.3., ПК 2.4.	Раздел 2. МДК.01.02 Поддержка и тестирование программных модулей	84	70	70	0	14			
ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3.	Раздел 3. МДК.01.03 Разработка мобильных приложений	130	104	104	1	26			
ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5.	Раздел 4. МДК.01.04 Системное программирование	117	104	104	0	13			
	Учебная практика	108						108	
	Производственная практика	36							36
	Всего:	637	410	410	1	83		108	36

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.01.01 Разработка программных модулей		162	
	1 семестр		
Раздел 1	Основы		
Тема 1.1 Основные операторы в программировании	Содержание учебного материала		
	Лекционные занятия. Введение в программирование. Условия. Типы STL.	6	1
	Практические занятия. Переменные. Циклы.	4	2
	Самостоятельная работа обучающихся. Сложные условия. Двойные циклы.	10	3

	Изучение алгоритмов сортировки.		
Раздел 2	Работа с памятью		
Тема 2.1 Указатели	Содержание учебного материала		
	Лекционные занятия. Статические массивы. Динамические массивы.	4	1
Раздел 4	Продвинутая работа с языком		
Тема 4.1 Реализация структур данных	Содержание учебного материала		
	Лекционные занятия. Функции и процедуры.	2	1
	Практические занятия. Рекурсия.	2	2
Тема 4.2 Продвинутая работа с памятью	Содержание учебного материала		
	Практические занятия. Указатели.	2	2
Тема 4.3 Дополнительные операторы языка	Содержание учебного материала		
	Практические занятия. Структуры. Синтаксический сахар. Исключения.	6	2
Тема 4.4 Производительность и отладка	Содержание учебного материала		
	Лекционные занятия. Введение в ООП. Уровень доступа в классах.	4	1
	Практические занятия. Препроцессор и многофайловость.	2	2
	2 семестр		
Раздел 2	Работа с памятью		
Тема 2.1 Указатели	Содержание учебного материала		
	Лекционные занятия. Многомерные массивы. Динамические классы.	4	1
Раздел 3	ООП		
Тема 3.1 Классы	Содержание учебного материала		
	Лекционные занятия. Перегрузка операторов.	2	1
	Практические занятия. Абстрактные классы и интерфейсы. Стандартная библиотека шаблонов.	6	2
Раздел 4	Продвинутая работа с языком		
Тема 4.1 Реализация структур данных	Содержание учебного материала		
	Лекционные занятия. Структура "Дерево". Стек и очередь.	8	1
	Лабораторные работы. Реализация односвязного списка. Реализация двусвязного списка.	8	3
Тема 4.2 Продвинутая работа с памятью	Содержание учебного материала		
	Лекционные занятия. Умные указатели.	2	1
	Самостоятельная работа обучающихся. Написание классов для интерфейса парсера.	10	3

Тема 4.3 Реализация сложных алгоритмов	Содержание учебного материала		
	Практические занятия. Работа с файлами. Временные файлы. Простой сервер. Написание обёрток. Обработка изображений.	10	2
	Лабораторные работы. Реализация парсера и лексера текстового формата. Реализация парсера и лексера бинарного формата. Чтение форматов файлов.	16	3
Тема 4.4 Дополнительные операторы языка	Содержание учебного материала		
	Лекционные занятия. Статические и глобальные переменные. UNION, ENUM.	4	1
	Практические занятия. Макросы. Шаблоны. Регулярные выражения и псевдослучайные числа. Современный C++.	8	2
	Самостоятельная работа обучающихся. Изучение оператора goto.	2	3
Тема 4.5 Работа с байтами и битами	Содержание учебного материала		
	Лекционные занятия. Низкоуровневая работа с памятью. Сложные преобразования байтов.	4	1
	Практические занятия. Битовые операции. Защита памяти от изменения.	4	2
Тема 4.6 Производительность и отладка	Содержание учебного материала		
	Лекционные занятия. Наследование классов. Отладка сложных проектор. Производительность.	6	1
Тема 4.7 Сборка проектов	Содержание учебного материала		
	Лекционные занятия. Линковка.	2	1
	Практические занятия. Системы сборки. Тип сборки.	6	2
	Лабораторные работы. Использование внешних библиотек.	4	3
	Самостоятельная работа обучающихся. Создание проекта на makefile, make, qmake и premake.	8	3
Раздел 5	Сложные задачи		
Тема 5.1 Подход к разработке GUI приложений	Содержание учебного материала		
	Лекционные занятия. Подходы к написанию GUI приложений.	2	1
	Лабораторные работы. Написание GUI приложения.	4	3
МДК.01.02 Поддержка и тестирование программных модулей		84	
	2 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.	2	1

	Самостоятельная работа обучающихся. Стандарты, регламентирующие процесс тестирования.	4	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.	2	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. Создание unit тестов. Автоматическое тестирование Selenium. Проведение нагрузочного тестирования. Изучение систем отслеживания ошибок.	28	2
	Самостоятельная работа обучающихся. Методики разработки тестов. Автоматизация процесса тестирования программного продукта. Управление тестированием.	10	3
Тема 1.5 Структура документации тестирования	Содержание учебного материала		
	Лекционные занятия. Структура документации тестирования.	2	1
МДК.01.03 Разработка мобильных приложений		130	
	2 семестр		
Раздел 1	Разработка мобильных приложений		
Тема 1.1 Основные платформы и языки разработки мобильных приложений	Содержание учебного материала		
	Лекционные занятия. Основные платформы мобильных приложений, сравнительная характеристика. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения.	18	1

	Основные языки для разработки мобильных приложений (Java). Основные языки для разработки мобильных приложений (Objective-C). Основные языки для разработки мобильных приложений (Kotlin). Инструменты разработки мобильных приложений (JDK). Инструменты разработки мобильных приложений (Android Studio). Инструменты разработки мобильных приложений (WebView). Инструменты разработки мобильных приложений Phonegap).		
	Практические занятия. Экспорт Android-приложения, особенности создания сертификата. Публикация Android-приложения на GooglePlay.	1	2
	Лабораторные работы. Установка инструментария и настройка среды для разработки мобильных приложений. Установка среды разработки мобильных приложений с применением виртуальной машины.	8	3
	Самостоятельная работа обучающихся. Основные платформы и языки разработки мобильных приложений.	10	3
Тема 1.2 Создание и тестирование модулей для мобильных приложений	Содержание учебного материала		
	Лекционные занятия. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения. Элементы управления и контейнеры. Работа со списками. Способы хранения данных. Архитектура платформы Android. Уровень ядра. Уровень библиотек. Архитектура платформы Android. Dalvik Virtual Machine. Архитектура платформы Android. Уровень каркаса приложений. Уровень приложений. Среда разработки для Android. EclipseIDE. Плагин ADT. Android Virtual Device.	18	1
	Практические занятия. Создание нового модуля. Обработка событий: цветовая индикация. Подготовка стандартных модулей. Обработка событий: переключение между экранами. Передача данных между модулями. Тестирование и оптимизация мобильного приложения. Инструменты разработки Windows Phone 7-приложений. Пример простейших программ WindowsPhone 7-приложения. Запуск приложения на эмуляторе. Инструменты разработки Android-приложений. Пример простейших программ Android-приложения. Запуск Android-приложения на эмуляторе. Тестирование приложения с помощью DalvikDebugMonitorServer (DDMS). Службы в Android. Компонент Service. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования. Программный доступ к дисплею устройства. Менеджер окон. Параметры дисплея. Примеры добавления анимации в Android-приложение. Покадровая анимация. TweenAnimation и FrameAnimation. Описание анимации в XML и в коде программы.	36	2

	Лабораторные работы. Создание эмуляторов и подключение устройств. Настройка режима терминала. Создание нового проекта. Изучение и комментирование кода. Изменение элементов дизайна. Обработка событий: подсказки.	24	3
	Самостоятельная работа обучающихся. Создание и тестирование модулей для мобильных приложений.	16	3
МДК.01.04 Системное программирование		117	
	2 семестр		
Раздел 1	Системное программирование		
Тема 1.1 Методология проектирования программных продуктов	Содержание учебного материала		
	Лекционные занятия. Виды программного обеспечения: системное, прикладное и промежуточное (middleware) программное обеспечение. Тенденции развития современного программного обеспечения. Системное программирование – основные определения: языки программирования, алгоритм, свойства алгоритма.	6	1
	Практические занятия. Изучение сред проектирования программного обеспечения.	2	2
	Лабораторные работы. Файловый ввод/вывод высокого уровня. Указатели и основные операции над ними.	8	3
Тема 1.2 Базовые конструкции структурного программирования	Содержание учебного материала		
	Лекционные занятия. Основные элементы языка C++: алфавит, структура программы, переменные и константы, типы данных, операции и выражения. Структурированный тип данных – массив: понятие индекса, базовый тип, описание массива, способы описания, метод обработки одномерных массивов. Многомерные массивы. Строки. Типы данных, определяемые пользователем. Введение в C++. Понятие алгоритма. Виды алгоритмов.	10	1
	Практические занятия. Приоритеты операций в C++. Унарные и бинарные операции. Побитовые операции в C++. Осуществление разработки кода программного модуля. Работа с операциями ввода, вывода в C++. Форматированный ввод-вывод.	8	2
	Самостоятельная работа обучающихся. Создание диалоговых окон.	2	3
Тема 1.3 Системное программирование для различных ОС	Содержание учебного материала		
	Лекционные занятия. Стандартные типы данных C++. Классы памяти. Начало работы в приложении Qt Creator. Функции. Определение и вызов. inline-функции. Передача аргументов по значению и по ссылке. Переопределение функции. Шаблоны. Программная реализация разветвляющегося алгоритма. Оператор	12	1

	switch. Программная реализация циклических алгоритмов.		
	Практические занятия. Технология работы с файлами при создании модуля. Технология разработки модуля с использованием окон и элементов управления. Библиотека динамической компоновки DLL. Использование принципов структурного программирования. Растровая графика. Разработка модуля с элементами графики. Методы и средства разработки технической документации программного продукта. Структуры, как тип данных, определяемый пользователем. Объединения в C++. Применение объединений для решения прикладных задач.	14	2
	Лабораторные работы. Работа с памятью. Подготовка и вызов системных функций. Эмуляция нажатия клавиш. Перехват системных сообщений.	16	3
	Самостоятельная работа обучающихся. Файловый ввод/вывод низкого уровня. Обмен данными по сети.	3	3
Тема 1.4 Объектно-ориентированное программирование	Содержание учебного материала		
	Лекционные занятия. Понятие класса и объекта класса. Инкапсуляция. Наследование и полиморфизм.	4	1
	Практические занятия. Разработка объектно-ориентированной программы. Шаблоны классов.	4	2
	Лабораторные работы. Разработка динамических библиотек. Загрузка динамических библиотек.	8	3
	Самостоятельная работа обучающихся. Написание тестов для ПО.	2	3
Тема 1.5 Стандарты языка C++	Содержание учебного материала		
	Лекционные занятия. Список основных изменений в C++11. Список изменений для каждой версии стандарта C++.	4	1
	Практические занятия. Оформленное техническое задание на разработку программного обеспечения. Правила оформления кода. Первичная документация по разработке программного обеспечения. Документация для разработчика.	8	2
	Самостоятельная работа обучающихся. Стандарт C++11. Стандарт C++20. Лямбда-функции.	6	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		108	

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 мобильного приложения согласно индивидуальному заданию разработка руководства пользователя, справочной системы оформление отчета		36	
	Всего:	638	

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Требования к минимальному материально–техническому обеспечению

Лаборатория распределенных систем

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

Лекционная аудитория

Экран Lumien, проектор в комплекте Sanyo PDG-DSU20

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

7-Zip (GNU LGPL)

Mozilla Firefox (MPL)

Microsoft Visual Studio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

Notepad++ (GNU GPL 3)

IntelliJ IDEA (Apache License 2.0)

Foxit Reader (Foxit EULA)

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

QT Creator ((L)GPL)

Python 3 (PSF License Agreement)

4.2. Информационное обеспечение обучения

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

Основные источники:

1. Кузнецов, А. С. Системное программирование : учебное пособие / А. С. Кузнецов, И. А. Якимов, П. В. Пересунько. — Красноярск : Сибирский федеральный университет, 2018. — 170 с. .
<https://www.iprbookshop.ru/84121.html>
2. Журавлева, М. Г. Основы программирования. Введение в язык Си. Ч.1 : учебное пособие по курсам «Программирование», «Основы алгоритмизации и программирования» / М. Г. Журавлева, В. А. Алексеев, П. А. Домашнев. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2019. — 99 с.. <https://www.iprbookshop.ru/101463.html>
3. Пирская, Л. В. Разработка мобильных приложений в среде Android Studio : учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2019. — 123 с..
<https://www.iprbookshop.ru/100196.html>
4. Котляров, В. П. Основы тестирования программного обеспечения : учебное пособие для СПО / В. П. Котляров. — Саратов : Профобразование, 2019. — 335 с..
<https://www.iprbookshop.ru/86202.html>

Дополнительные источники:

1. Бабушкина, И. А. Практикум по объектно-ориентированному программированию / И. А. Бабушкина, С. М. Окулов. — 5-е изд. — Москва : Лаборатория знаний, 2020. — 367 с. — ISBN 978-5-00101-780-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/12254.html> . <http://www.iprbookshop.ru/12254.html>
2. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы / Д. М. Златопольский. — 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с.. <http://www.iprbookshop.ru/12264.html>

3. программирование на языке высокого уровня : методические указания и варианты заданий для студентов 1-го курса направления подготовки 09.03.01 Информатика и вычислительная техника / составители С. П. Зоткин. — Москва : Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2016. — 89 с. . <http://www.iprbookshop.ru/46060.html>

Интернет-ресурсы:

1. Справочник по Android Studio - <http://easyandroid.ru/>
2. Документация по C# - <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
3. Открытые интернет-курсы - www.intuit.ru/studies/courses
4. Открытая электронная библиотека «ИИТО ЮНЕ-СКО» по ИКТ в образовании - <http://ru.iite.unesco.org/publications>
5. <http://www.iprbookshop.ru>

4.3. Общие требования к организации образовательного процесса

Освоению модуля «Разработка модулей программного обеспечения для компьютерных систем» должно предшествовать изучение дисциплин «Информатика», «Введение в специальность», «Теория алгоритмов», «Основы 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 кадров, обеспечивающих обучение междисциплинарному курсу (курсам): наличие высшего профессионального образования, соответствующего профилю модуля «Разработка модулей программного обеспечения для компьютерных систем».

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результаты обучения (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт:</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>проведения тестирования программного модуля по определенному сценарию;</p> <p>иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ПК 1.4. Выполнять тестирование программных модулей</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	<p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p>
<p>ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</p>	<p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p> <p>тест; тест; тест;</p>

	<p>уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт:</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>
<p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

[illegible]

	разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей	знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования	тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;

	инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>
ОК 09. Использовать информационные технологии в профессиональной деятельности	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	<p>обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: разработки мобильных приложений;</p>	
<p>ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере</p>	<p>знать: основные этапы разработки программного обеспечения; знать: основные принципы технологии структурного и объектно-ориентированного программирования; знать: способы оптимизации и приемы рефакторинга; знать: основные принципы отладки и тестирования программных продуктов; уметь: осуществлять разработку кода программного модуля на языках низкого и высокого уровней; уметь: создавать программу по разработанному алгоритму как отдельный модуль; уметь: выполнять отладку и тестирование программы на уровне модуля; уметь: осуществлять разработку кода программного модуля на современных языках программирования; уметь: выполнять оптимизацию и рефакторинг программного кода; уметь: выполнять разработку тестовых наборов и тестовых сценариев для программного обеспечения; иметь практический опыт: разработки кода программного продукта на основе готовой спецификации на уровне модуля; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта; иметь практический опыт: проведения тестирования программного модуля по определенному сценарию; иметь практический опыт: использования инструментальных средств на этапе отладки программного продукта;</p>	<p>тест; тест; тест; тест; тест; тест; тест; тест; тест; тест;</p>

	иметь практический опыт: разработки мобильных приложений;	
--	---	--

Фонд оценочных материалов (средств) по дисциплине
Разработка модулей программного обеспечения для компьютерных систем

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

ПК 1.1, ПК 1.2

Блок 1 (знать)

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

21. Оператор INT в Си# применяется для: 1: переопределения диапазона целых чисел 2: преобразования переменной к целому типу 3: описания переменных целого типа 4: прибавления единицы к коду символа

22. Соккрытие информации и комбинирование данных и методов внутри объекта определяет понятие ...:

a) иерархии;

b) инкапсуляции;

c) полиморфизма;

d) наследования.

23. Присваивание действию общего имени, и каждый объект иерархии выполняет это действие способом, подходящим именно ему, определяет понятие ...:

- a) наследования;
- b) класса;
- c) инкапсуляции;
- d) полиморфизма.

24. В объектно-ориентированном программировании естественным средством структурирования являются ...:

- a) классы;
- b) параметры;
- c) методы;
- d) свойства.

25. Основой методологии объектно-ориентированного программирования является ...:

- a) вывод некоторого целевого утверждения;
- b) описание системы в терминах объектов и связей между ними;
- c) отказ от использования подпрограмм при реализации системы;
- d) описание системы в виде рекуррентных соотношений.

26. В объектно-ориентированном подходе структура системы описывается в терминах ...

- a) сообщений;
- b) методов, различной степени вложенности;
- c) объектов и связей;
- d) методов и функций.

27. Инкапсуляция характеризуется ...:

- a) посылкой сообщений объектам;
- b) сокрытием информации и комбинированием данных и методов внутри объекта;
- c) способностью объекта сохранять свойства и методы класса-родителя;
- d) возможностью задания в иерархии объектов различных действий в методе с одним именем.

28. Наследование характеризуется ...:

- a) посылкой сообщений объектам;
- b) способностью объекта сохранять свойства и методы класса-родителя;
- c) возможностью задания в иерархии объектов различных действий в методе с одним именем;
- d) сокрытием информации и комбинированием данных и методов внутри объекта.

29. Полиморфизм характеризуется ...:

- a) возможностью задания в иерархии объектов различных действий в методе с одним именем;
- b) сокрытием информации и комбинированием данных и методов внутри объекта;
- c) способностью объекта наследовать свойства и методы класса-родителя;
- d) посылкой сообщений объектам.

30. В объектно-ориентированном программировании правильной является фраза ...:

- a) объект – это то же самое, что и класс;
- b) класс и объект – не связанные между собой понятия;
- c) объект – это экземпляр класса;
- d) класс – это экземпляр объекта.

31. В объектно-ориентированном программировании структуру и функционирование множества объектов с подобными характеристиками, атрибутами и поведением описывает ...:

- a) класс;
- b) полиморфизм;
- c) событие;
- d) метод.

32. Порождение иерархии объектов происходит в процессе ...:

- a) наследования;
- b) полиморфизма;
- c) вызова соответствующего метода;
- d) инкапсуляции.

33. В процессе наследования возникает ...:

- a) новое поле в каждом классе;
- b) иерархия объектов;
- c) инкапсуляция данных и методов;
- d) новый метод в каждом классе.

34. В объектно-ориентированном программировании при создании объекта первым вызывается ...:

- a) конструктор;
- b) произвольный метод в зависимости от ситуации;
- c) первый описанный в программе метод;
- d) деструктор.

35. Операция модификации над объектом ...:

- a) даёт доступ к содержанию объекта по частям, в строго определенном порядке;
- b) разрушает объект и освобождает занимаемую им память;
- c) дает доступ к состоянию, но не изменяет его;
- d) изменяет состояние объекта.

36. В объектно-ориентированном программировании каждый объект по отношению к своему классу является...:

- a) экземпляром;
- b) кодом;
- c) свойством;
- d) методом.

37. В основе объектно-ориентированного программирования лежит:

- e) Понятие объекта;
- f) Понятие языка программирования;
- g) Понятие компонента;
- h) Понятие события.

38. Что из ниже перечисленного не является принципом ООП:

- a) инкапсуляция;
- b) порождение;

- с) наследование;
- d) полиморфизм.

39. Свойства определения объекта путем не только описания его структуры, но и всего множества операций с ним это...:

- a) инкапсуляция;
- b) порождение;
- с) наследование;
- d) полиморфизм.

40. Возможность построения иерархии порожденных объектов, начиная с некоторого простого первоначального объекта и кончая более сложными объектами:

- a) инкапсуляция;
- b) порождение;
- с) наследование;
- d) полиморфизм.

41. Возможность идентифицировать одним и тем же именем множество аналогичных операций (действий):

- a) инкапсуляция;
- b) порождение;
- с) наследование;
- d) полиморфизм.

ПК 1.3.

Блок 1 (знать)

1. Проект – это:

- a) набор программ и установленные отношения между ними;
- b) набор компонентов и установленные отношения между ними;
- с) набор классов и установленные отношения между ними;
- d) нет верного ответа;

2. Задача компилятора -:

- a) трансляция исходного кода программы в бинарное представление;
- b) чтение исходного кода программы;
- с) создание копии программы;
- d) нет верного ответа;

Блок 2 (уметь)

1. Укажите неправильно записанную операцию отношения:

- a) !=;
- b) <=
- с) >=;
- d) все операторы записаны правильно.

2. Структуры - это:

- a) это составной тип данных, построенный с использованием разнообразных типов
- b) упорядоченная последовательность констант одного типа;
- с) набор программ и установленные отношения между ними;
- d)нет верного ответа;

3. Перебор элементов массива характеризуется:

- a) количеством одновременно обрабатываемых элементов;

- b) количеством индексов;
- c) удалением массива;
- d) нет верного ответа;

4. Операции, применяемые к одному операнду, называются:

- a) бинарными;
- b) комплексными;
- c) унарными;
- d) нет верного ответа

5. Операция, возвращающая остаток от деления:

- a) ||;
- b) &;
- c) //;
- d) %;

6. Логическое «или»:

- a) &&;
- b) ||;
- c) !;
- d) <<;

7. Логическое «и»:

- a) >>;
- b) !;
- c) &&;
- d) ++;

8. Операции сравнения в результате дают значение типа: a) int;

- b) bool;
- c) real;
- d) double;

ПК 1.4

Блок 1 (знать)

1. Какой из представленных не является этапом разработки программы:

- a) Разработка или выбор алгоритма;
- b) Постановка задачи;
- c) Сертификация;
- d) Тестирование и отладка программы;

Блок 2 (уметь)

1. В каком случае можно не использовать фигурные скобочки в операторе выбора if?

- a) нет правильного ответа
- b) если в теле оператора if два и более операторов
- c) если в теле оператора if всего один оператор
- d) если в теле оператора if нет ни одного оператора

2. Какие преобразования типов данных не возможны без потери данных?

- a) int to float
- b) float to int
- c) все перечисленные преобразования не возможны

d) char to float

3. В каком случае лучше всего использовать приведение типов данных?

- a) чтобы разрешить программе использовать только целые числа
- b) во всех выше указанных случаях
- c) чтобы изменить тип возвращаемого значения функции
- d) при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой

4. Логическая операция с большим приоритетом выполнения. a) ||

- b) !
- c) &&

5. Почему приведение типов данных может быть не безопасно.

- a) Вы можете навсегда изменить значение переменной.
- b) Некоторые преобразования не определены компилятором, такие как - преобразование символа в целое.
- c) Вы можете временно потерять часть данных - таких, как отсечение десятичной части чисел с плавающей точкой.
- d) нет никаких опасностей.

6. Укажите неправильно записанную операцию отношения

- a) >=
- b) <=
- c) !=
- d) все операторы записаны правильно

7. Что из нижеперечисленного не является прототипом функции?

- a) void funct();
- b) double funct(char x)
- c) int funct(char x, char y);
- d) char x();

8. Какие из следующих функций являются встроенными?

- a) нет правильного ответа
- b) inline void foo() { }
- c) inline: void foo() { }
- d) void foo() inline { }

9. Для чего используются встроенные функции?

- a) Для увеличения скорости работы программы
- b) Чтобы уменьшить размер программы
- c) Для удаления ненужных функций
- d) Для упрощения файла с исходным кодом

10. Укажите тип возвращаемого значения следующей функции
int func(char x, float v, double t);

- a) float
- b) int
- c) char
- d) double

11. Какое значение будет содержать локальная переменная x, в конце main?
int x = 5;

```
int main(int argc, char** argv)
{int x = x;
return 0;}
```

12. Укажите правильный вызов функции, предполагается, что функция была объявлена ранее.

- a) funct();
- b) funct;
- c) funct x, y;
- d) int funct();;

13. Какой стандартный код используется для Символьных данных типа char в C++?

- a) Код ASCII
- b) Код UTF-8
- c) Код cp-1251
- d) Код ASCII

14. Какой порядковый номер последнего элемента массива, размер массива 19?

- a) 18
- b) порядковый номер определяется программистом
- c) 19
- d) 20

15. Каким символом завершается Си-строка?

- a) "
- b) "
- c) 'n'
- d) '.'

16. Укажите правильное объявление массива!

- a) array an array[10];
- b) int anarray[10];
- c) anarray{10};
- d) int anarray;

17. Вы собираетесь создать проект Windows Forms. Отметьте верные утверждения

a) В Visual Studio 2015 непосредственное создание проектов Windows Forms не поддерживается

b) Можно создавать проект Windows Forms без установки дополнений

c) Вначале вручную создаётся проект приложения с использованием Windows Forms .
Затем этот проект сохраняется в качестве шаблона Visual Studio.

18. Для ввода текста предназначен следующий визуальный компонент:

- a) label
- b) textbox
- c) button
- d) timer

19. Свойство text есть у этих компонентов:

- a) label
- b) textbox
- c) button
- d) timer

20. Для чего нужен элемент MessageBox?

- a) Вывод текстового сообщения.
- b) Ввод текста в соответствующем окне.
- c) Вывод текста на форму.

21. Укажите, какое из нижеследующих утверждений ложное.

- a) Тело каждой функции ограничивается левой и правой фигурными скобками ({ и }).
- b) Программа на C++, которая выводит три строки, должна содержать три оператора вывода, использующих cout.
- c) Элементы данных или элемент-функции, объявленные со спецификатором доступа private, доступны для элемент-функций класса, в котором они объявлены.
- d) Все высказывания истинные

22. Каждый оператор заканчивается _____.

- a) endl
- b) return
- c) точкой с запятой
- d) управляющей последовательностью

23. Оператор _____ используется для принятия решений

- a) cout
- b) if
- c) while
- d) for

24. Комментарии заключаются в скобки:

- a) { }
- b) /* */
- c) []
- d) /% %/

ПК 1.5

Блок 1 (Знать)

1. Возможность идентифицировать одним и тем же именем множество аналогичных операций (действий):

- a) инкапсуляция;
- b) порождение;
- c) наследование;
- d) полиморфизм.

2. Каким свойством не обладает объект?

- a)наследование;
- b)множественность;
- c)инкапсуляции;
- d)полиморфизма;

3. Наследование характеризуется ...:

- a) посылкой сообщений объектам;
- b) способностью объекта сохранять свойства и методы класса-родителя;
- c) возможностью задания в иерархии объектов различных действий в методе с одним именем;
- d) сокрытием информации и комбинированием данных и методов внутри объекта.

4. В процессе наследования возникает ...:

- a) новое поле в каждом классе;
- b) иерархия объектов;
- c) инкапсуляция данных и методов;
- d) новый метод в каждом классе.

5. Функция – это:

- a) программа, предназначенная для выполнения определенных пользовательских задач и рассчитанная на непосредственное взаимодействие с пользователем;
- b) последовательность действий, направленных на выполнение их некоторым исполнителем;
- c) это точно определенная последовательность простых действий, обеспечивающих решение любой задачи из некоторого класса;
- d) именованная последовательность описаний и операторов, выполняющая какое-либо законченное действие;

6. Функция начинает выполняться в момент:

- a) объявления;
- b) определения;
- c) вызова;
- d) компиляции;

7. Объявление функции не содержит:

- a) имя функции;
- b) список исключений функции;
- c) список передаваемых параметров;
- d) составлением инструментальных программ;

8. Выберите верное утверждение:

- a) Определение функции должно находиться после функции main();
- b) Все величины, описанные внутри функции, являются глобальными;
- c) Функция может входить в состав выражений, если тип возвращаемого ею значения – void;
- d) При вызове функции в стеке сохраняется содержимое регистров процессора на момент, предшествующий вызову функции;

9. При передаче переменной-аргумента по значению, в функции создается локальная переменная с именем аргумента, в которую записывается его значение. Внутри функции может измениться:

- a) значение локальной переменной и значение аргумента;
- b) только значение локальной переменной;
- c) только значение аргумента;
- d) ни значение локальной переменной, ни значение аргумента;

10. В случае передачи переменной-аргумента по ссылке функция получает:

- a) значение аргумента ;
- e) тип аргумента;
- f) адрес аргумента;
- g) ничего;

11. Выберите верное утверждение:

- a) В определении, в объявлении и при вызове одной и той же функции типы и порядок следования параметров могут не совпадать;
- b) Функция может принимать параметры и возвращать значение;
- c) Определений функции может быть несколько;
- d) Тип возвращаемого функцией значения может быть любым;

12. Список исключений, которые функция может прямо или косвенно порождать, задается в:: a) теле функции;

- b) описании переменных функции;
- c) заголовке функции;
- d) определении функции;

ПК 6.2.

Блок 3 (Иметь практический опыт)

1. Определить значения переменных *i*, *d* после выполнения операторов:

```
int i, d;
```

```
for (i=0, d=0; i< 10; i++, d =!d);
```

2. Определить значения переменных после выполнения действий:

```
double d1, d2, d3; d1 = 2.56; d2 = (int)d1 + 1.5; d3 = (int)(d1 + 1.5)
```

3. Определить значения переменных *d1*, *i* после выполнения действий:

```
double d1 = 2.56; int i; i = (d1 - (int)d1) * 10;
```

4. Определить значение переменной *i* после выполнения действий:

```
i=0; if (i++) i++;
```

5. Определить значение переменной *m* после выполнения действий:

```
int m, a, b; a=3; b=2; m = a > b ? a : b;
```

6. Определить значение переменной *m* после выполнения действий:

```
double a=0.3, b=-1; int m; m = (a * b) > 0;
```

7. Определить значения переменных *i*, *j* после выполнения операторов:

```
int i, k=2; for (i=0, ++k; 0; i++, --k);
```

8. Определить значения переменных *i*, *k* после выполнения операторов:

```
int i, k=-1; i=5; if (k--, i) i++;
```

9. Определить значения переменных *i*, *k* после выполнения операторов:

```
int i, k=3; for (i=0, k/=2; i>=0; i++);
```

10. Определить значения переменных *d1*, *d2*, *d3* после выполнения действий:

```
double d1,d2,d3;
```

```
d1 = 2.56;
```

```
d2 = (int)d1 + 1.5;
```

```
d3 = (int)(d1 + 1.5);
```

11. Определить значение переменной *s* после выполнения действий:

```
short i1=20000, i2=20000, s ; // sizeof(short) равен 2
```

```
long s1, s2; s1 = i1 + i2;
```

```
s2 = (long)i1 + i2;
```

```
if (s1 == s2) s=0;
```

```
else s=1;
```

12. Определить значения переменной *i* после выполнения действий:

i=0;

if (++*i*) *i*++;

13. Определить значения переменных *d1*, *i* после выполнения действий:

int *i*;

double *d1* = 2.86;

i = (*d1* - (int)*d1*) * 10; *d1*--;

14. Определить значения переменной *n* после выполнения операторов:

a = 5; *b* = 3; *c* = 1;

switch ((*a* > *b*) * 2 + (*b* > *c*)){

case 0: *n* = *c*; break;

case 1: *n* = *b*; break;

case 2: *n* = *c* > *a* ? *c* : *a*; break;

case 3: *n* = *a*; break; }

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)

ОК 01., ОК 02., ОК 03., ОК 04., ОК 05., ОК 06., ОК 07., ОК 09., ОК 10., ОК 11., ПК 1.1., ПК 1.2., ПК 1.3., ПК 1.4., ПК 1.5., ПК 1.6., ПК 2.3., ПК 2.4.

Блок 1 (знать)

1. Что не является обязательным элементом объектной модели:

а) абстрагирование;

б) инкапсуляция;

с) устойчивость;

д) модульность;

2. время исполнения проекта – это:

а) 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;

Блок 2 (уметь)

1. Массив-это:

- a) упорядоченная последовательность переменных разных типов;
- b) упорядоченная последовательность констант одного типа;
- c) упорядоченная последовательность переменных одного типа;
- d) нет верного ответа;

2. К структурированным типам относятся

- 1) вещественный, 2) файл, 3) множество,
- 4) символьный, 5) диапазон, 6) целый

3. 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

4. Оператор INT в Си++ применяется для: 1: переопределения диапазона целых чисел 2: преобразования переменной к целому типу 3: описания переменных целого типа 4: прибавления единицы к коду символа

5. Проект – это:

- a) набор программ и установленные отношения между ними;
- b) набор компонентов и установленные отношения между ними;
- c) набор классов и установленные отношения между ними;
- d) нет верного ответа;

6. Задача компилятора -:

- a) трансляция исходного кода программы в бинарное представление;
- b) чтение исходного кода программы;
- c) создание копии программы;
- d) нет верного ответа;

7. Укажите неправильно записанную операцию отношения:

- a) !=;
- b) <=;
- c) >=;
- d) все операторы записаны правильно.

8. Структуры - это:

- a) это составной тип данных, построенный с использованием разнообразных типов
- b) упорядоченная последовательность констант одного типа;
- c) набор программ и установленные отношения между ними;
- d) нет верного ответа;

9. Перебор элементов массива характеризуется:

- a) количеством одновременно обрабатываемых элементов;
- b) количеством индексов;
- c) удалением массива;
- d) нет верного ответа;

10. Операции, применяемые к одному операнду, называются:

- a) бинарными;
- b) комплексными;
- c) унарными;
- d) нет верного ответа

11. Операция, возвращающая остаток от деления:

- a) ||;
- b) &;
- c) //;
- d) %;

12. Логическое «или»:

- a) &&;
- b) ||;
- c) !;
- d) <<;

13. Логическое «и»:

- a) >>;

- b) !;
- c) &&;
- d) ++;

14. Операции сравнения в результате дают значение типа: a) int;
b) bool;
c) real;

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

Блок 3 (иметь практический опыт)

1. 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;

2. 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;

3. какой из элементов не является элементом объектной модели:

- a) модульность;
- b) параллелизм;
- c) последовательность;
- d) иерархия;

4. 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;

5. Что не включает в себя качество ПО:

- a) эффективность;
- b) мобильность;
- c) практичность;

d) управление доступом;

6. Массив-это:

- a) упорядоченная последовательность переменных разных типов;
- b) упорядоченная последовательность констант одного типа;
- c) упорядоченная последовательность переменных одного типа;
- d) нет верного ответа;

7. К структурированным типам относятся

- 1) вещественный, 2) файл, 3) множество,
- 4) символьный, 5) диапазон, 6) целый

8. Какой из представленных не является этапом разработки программы:

- a) Разработка или выбор алгоритма;
- b) Постановка задачи;
- c) Сертификация;
- d) Тестирование и отладка программы;

9. Почему приведение типов данных может быть не безопасно.

- a) Вы можете навсегда изменить значение переменной.
- b) Некоторые преобразования не определены компилятором, такие как - преобразование символа в целое.
- c) Вы можете временно потерять часть данных - таких, как отсечение десятичной части чисел с плавающей точкой.
- d) нет никаких опасностей.

10. Укажите неправильно записанную операцию отношения

- a) >=
- b) <=
- c) !=
- d) все операторы записаны правильно

11. Что из нижеперечисленного не является прототипом функции?

- a) void funct();
- b) double funct(char x)
- c) int funct(char x, char y);
- d) char x();

12. Какие из следующих функций являются встроенными?

- a) нет правильного ответа
- b) inline void foo() { }
- c) inline: void foo() { }
- d) void foo() inline { }

13. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI

- 1. int main(void)
- 2. int main()
- 3. void main()
- 4. void main(void)

14. Какие преобразования типов данных не возможны без потери данных?

- 1. char to float
- 2. все перечисленные преобразования не возможны

3. float to int
 4. int to float
15. Оператор if else позволяет определить действие ...
 1. только для ложного условия
 2. для истинного и ложного условий
 3. только для истинного условия
 16. Если условие оператора выбора ложное, то:
 1. выполняется следующий оператор, сразу после оператора if
 2. программа завершает работу
 3. выполняется тело оператора выбора
 17. В каком случае лучше всего использовать приведение типов данных?
 1. при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой
 2. во всех выше указанных случаях
 3. чтобы изменить тип возвращаемого значения функции
 4. чтобы разрешить программе использовать только целые числа
 18. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?
 1. 10
 2. 11
 3. 13
 4. 12
 19. Какой порядковый номер последнего элемента массива, размер массива 19?
 1. порядковый номер определяется программистом
 2. 18
 3. 19

Методические материалы, характеризующих процедуры оценивания

Тесты, индивидуальные задания.

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 работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Высокий уровень
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки	Продвинутый уровень

		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	
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	<i>Пороговый уровень</i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<i>Компетенции не сформированы</i>

3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. Что не является обязательным элементом объектной модели:

- а) абстрагирование;
- б) инкапсуляция;
- с) устойчивость;
- д) модульность;

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=3715>

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