

Рабочая программа дисциплины

Объектно-ориентированное программирование

<i>Направление подготовки</i>	Информационные системы и технологии
<i>Код</i>	09.03.02
<i>Направленность (профиль)</i>	Информационные технологии и большие данные
<i>Квалификация (степень) выпускника</i>	бакалавр

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-4
Профессиональные	-	ПК-6

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-4	ПК-4. Способен применять методы и средства проектирования программного обеспечения и баз данных	ПК-4.1. Адаптация бизнес-процессов к возможностям типовой ИС. Разработка модели бизнес-процессов. Проектирование и дизайн типовой ИС. Моделирование бизнес-процессов ПК-4.2. Сбор данных о запросах и потребностях заказчика применительно к типовой ИС. Документирование собранных данных в соответствии с регламентами организации. ПК-4.3. Согласование и утверждение требований к типовой ИС. ПК-4.4. Разработка прототипов ИС на базе типовой ИС в соответствии с требованиями. ПК-4.5. Интеграция ИС с существующими ИС заказчика
ПК-6	ПК-6. Способен осуществлять администрирование процесса управления безопасностью сетевых устройств и программного обеспечения.	ПК-6.1. Определяет параметры безопасности и защиты программного обеспечения сетевых устройств ПК-6.2. Понимает принципы обеспечения безопасности и защиты программного обеспечения сетевых устройств. ПК-6.3. Выполняет установку и настройку специализированных программных средств обеспечения безопасности, настройку параметров безопасности операционных систем сетевых устройств. ПК-6.4. Понимает принципы обеспечения безопасности и защиты программного обеспечения сетевых устройств.

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	ПК-4		
	- методы оценки объемов и сроков выполнения работ, технологии выполнения работ в организации; - архитектуру, устройство и функционирование вычислительных систем, коммуникационное оборудование, сетевые протоколы; - основы современных операционных систем, основы современных систем управления базами данных, устройство и функционирование современных ИС; - теорию баз данных, системы хранения и анализа баз данных; - основы программирования, современные объектно-ориентированные языки программирования, современные структурные языки программирования, языки современных бизнес-приложений; - современные методики тестирования разрабатываемых ИС, современные стандарты информационного взаимодействия систем; - программные	- работать с современными системами программирования, конструировать программное обеспечение и базы данных, разрабатывать основные программные документы; - оценивать объемы и сроки выполнения работ; - разрабатывать руководство программиста к модифицированным элементам типовой ИС; - разрабатывать руководство администратора к модифицированным элементам типовой ИС; - разрабатывать руководство пользователя к модифицированным элементам типовой ИС; - разрабатывать ТЗ.	- навыками конструирования программного обеспечения и баз данных; - навыками разработки интерфейсов обмена данными, форматов обмена данными, технологий обмена данными между ИС и существующими системами в соответствии с техническим заданием.

	средства и платформы инфраструктуры информационных технологий организаций; - современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP, ITIL, ITSM); - основы теории систем и системного анализа; - методики описания и моделирования бизнес-процессов, средства моделирования бизнес-процессов; - системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; - отраслевую нормативную техническую документацию.		
Код компетенции	ПК-6		
	Основные понятия и принципы функционирования информационных систем, используемых для автоматизации бизнес-процессов и организационного управления. Методы и технологии проектирования, разработки и сопровождения информационных систем. Стандарты и методологии управления проектами	Анализировать потребности организации и разрабатывать технические задания для создания или модификации информационных систем. Проектировать архитектуру информационных систем с учетом специфики бизнес-процессов организации. Использовать современные инструменты и технологии для разработки и внедрения информационных систем. Осуществлять управление	Навыками работы с инструментами разработки и сопровождения информационных систем, включая языки программирования и платформы. Умением применять методологии управления проектами, такие как Agile, Scrum или Waterfall, в контексте разработки ИС.

	в области информационных технологий. Основные бизнес-процессы и задачи, подлежащие автоматизации в организации. Правила и подходы к тестированию, внедрению и сопровождению информационных систем.	проектами по созданию и сопровождению информационных систем, включая планирование, контроль и оценку результатов. Проводить обучение пользователей и обеспечивать техническую поддержку информационных систем.	Компетенцией в области анализа и оптимизации бизнес-процессов с использованием информационных технологий. Способностью разрабатывать и внедрять решения для автоматизации задач организационного управления. Умением работать в команде, взаимодействовать с различными заинтересованными сторонами и управлять проектами на всех этапах их жизненного цикла.
--	---	---	--

4. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» к части, формируемой участниками образовательных отношений учебного плана ОПОП.

Данная дисциплина взаимосвязана с другими дисциплинами, такими как: «Алгоритмизация и методы программирования», «Операционные системы и среды».

В рамках освоения программы бакалавриата выпускники готовятся к решению задач профессиональной деятельности следующих типов:

Профиль (направленность) программы установлена путем ее ориентации на сферу профессиональной деятельности выпускников: Информационные технологии и большие данные

5. Объем дисциплины

<i>Виды учебной работы</i>	<i>Формы обучения</i>
	<i>Очно-заочное</i>
Общая трудоемкость: зачетные единицы/часы	8/288
Контактная работа:	
Занятия лекционного типа	24
Занятия семинарского типа	32
Промежуточная аттестация: экзамен, зачет	18,1
Самостоятельная работа (СРС)	213,9

6. Содержание дисциплины (модуля), структурированное по темам / разделам с указанием отведенного на них количества академических часов и видов учебных занятий

6.1. Распределение часов по разделам/темам и видам работы

6.1.1. Очно-заочная форма обучения

№ п/п	Раздел/тема	Виды учебной работы (в часах)						
		Контактная работа						Самосто ятельная работа
		Занятия лекционного типа		Занятия семинарского типа				
		Лекции	Иные учебные занятия	Практи ческие занятия	Семи нары	Лабора торные работы	Иные	
1.	Введение в C++ и основы синтаксиса. Управляющие конструкции	2			2			18
2.	Функции в C++	2			2			18
3.	Массивы и строки	2			2			18
4.	Указатели и ссылки	2			2			18
5.	Основы ООП: классы и объекты	2			3			18
6.	Инкапсуляция и абстракция данных	2			3			18
7.	Наследование и композиция	2			3			18
8.	Полиморфизм и виртуальные функции	2			3			18
9.	Шаблоны (Templates)	2			3			18
10.	Исключения и обработка ошибок	2			3			18
11.	Управление памятью и умные указатели	2			3			18
12.	Файловый ввод-вывод и работа с потоками	2			3			15,9
	Промежуточная аттестация	18,1						
	Итого	24			32			213,9

6.2 Программа дисциплины, структурированная по темам / разделам

6.2.1 Содержание лекционного курса

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
----------	---------------------------------	-------------------------------

1.	Введение в C++ и основы синтаксиса. Управляющие конструкции	История и особенности языка C++. Структура программы: <code>main()</code> , директивы препроцессора (<code>#include</code>). Компиляция и выполнение программы. Базовые типы данных (<code>int</code> , <code>float</code> , <code>double</code> , <code>char</code> , <code>bool</code> , <code>void</code>). Переменные и константы. Операторы и выражения. Ввод/вывод (<code>cin</code> , <code>cout</code>). Условные операторы (<code>if</code> , <code>else if</code> , <code>else</code> , <code>switch</code>). Циклы (<code>for</code> , <code>while</code> , <code>do-while</code>). Операторы управления циклом (<code>break</code> , <code>continue</code>). Примеры решения типовых задач (поиск максимума, суммы, факториала).
2.	Функции в C++	Объявление и определение функций. Параметры (по значению, по ссылке, по указателю). Возвращаемое значение. Перегрузка функций. Рекурсия. Область видимости переменных. Статические переменные.
3.	Массивы и строки	Одномерные и многомерные массивы. Инициализация массивов. Строки в стиле C (<code>char[]</code>) и класс <code>std::string</code> . Операции над строками. Поиск и сортировка в массивах (линейный поиск, пузырьковая сортировка).
4.	Указатели и ссылки	Понятие указателя. Операторы <code>*</code> и <code>&</code> . Арифметика указателей. Указатели и массивы. Ссылки. Передача параметров по ссылке и по значению. Динамическое выделение памяти (<code>new/delete</code>). Проблемы утечек памяти.
5.	Основы ООП: классы и объекты	Принципы ООП (инкапсуляция, наследование, полиморфизм). Классы и объекты. Конструкторы (включая конструктор по умолчанию, копирующий) и деструкторы. Модификаторы доступа (<code>public</code> , <code>private</code> , <code>protected</code>). Статические члены класса.
6.	Инкапсуляция и абстракция данных	Инкапсуляция данных. Геттеры и сеттеры. Константные методы. Дружественные функции и классы (<code>friend</code>). Константные объекты. Принципы проектирования интерфейсов классов.
7.	Наследование и композиция	Типы наследования (<code>public</code> , <code>protected</code> , <code>private</code>). Конструкторы и деструкторы при наследовании. Множественное наследование. Виртуальное наследование. Композиция и агрегация.
8.	Полиморфизм и виртуальные функции	Перегрузка операторов (унарных и бинарных). Виртуальные функции. Абстрактные классы и чистые виртуальные функции (<code>= 0</code>). Раннее и позднее связывание. Виртуальные деструкторы.
9.	Шаблоны (Templates)	Шаблоны функций. Шаблоны классов. Специализация шаблонов. Стандартная библиотека шаблонов (STL): контейнеры (<code>vector</code> , <code>list</code> , <code>map</code> , <code>set</code>), итераторы, алгоритмы (<code>sort</code> , <code>find</code> , <code>copy</code>).
10.	Исключения и обработка ошибок	Механизм исключений (<code>try</code> , <code>catch</code> , <code>throw</code>). Стандартные исключения (<code>std::exception</code> и его

		наследники). Создание собственных исключений. Гарантия безопасности исключений.
11.	Управление памятью и умные указатели	Проблемы ручного управления памятью. Умные указатели (std::unique_ptr, std::shared_ptr, std::weak_ptr). Принцип RAII (Resource Acquisition Is Initialization). Управление ресурсами.
12.	Файловый ввод-вывод и работа с потоками	Содержание: работа с файлами (ifstream, ofstream, fstream). Режимы открытия файлов. Чтение и запись данных. Бинарные и текстовые файлы. Обработка ошибок ввода-вывода.

6.2.2 Содержание практических занятий

№ п/п	Наименование раздела дисциплины	Содержание практических занятий
1.	Введение в C++ и основы синтаксиса. Управляющие конструкции	История и особенности языка C++. Структура программы: main(), директивы препроцессора (#include). Компиляция и выполнение программы. Базовые типы данных (int, float, double, char, bool, void). Переменные и константы. Операторы и выражения. Ввод/вывод (cin, cout). Условные операторы (if, else if, else, switch). Циклы (for, while, do-while). Операторы управления циклом (break, continue). Примеры решения типовых задач (поиск максимума, суммы, факториала).
2.	Функции в C++	Объявление и определение функций. Параметры (по значению, по ссылке, по указателю). Возвращаемое значение. Перегрузка функций. Рекурсия. Область видимости переменных. Статические переменные.
3.	Массивы и строки	Одномерные и многомерные массивы. Инициализация массивов. Строки в стиле C (char[]) и класс std::string. Операции над строками. Поиск и сортировка в массивах (линейный поиск, пузырьковая сортировка).
4.	Указатели и ссылки	Понятие указателя. Операторы * и &. Арифметика указателей. Указатели и массивы. Ссылки. Передача параметров по ссылке и по значению. Динамическое выделение памяти (new/delete). Проблемы утечек памяти.
5.	Основы ООП: классы и объекты	Принципы ООП (инкапсуляция, наследование, полиморфизм). Классы и объекты. Конструкторы (включая конструктор по умолчанию, копирующий) и деструкторы. Модификаторы доступа (public, private, protected). Статические члены класса.
6.	Инкапсуляция и абстракция данных	Инкапсуляция данных. Геттеры и сеттеры. Константные методы. Дружественные функции и классы (friend). Константные объекты. Принципы проектирования интерфейсов классов.

7.	Наследование и композиция	Типы наследования (public, protected, private). Конструкторы и деструкторы при наследовании. Множественное наследование. Виртуальное наследование. Композиция и агрегация.
8.	Полиморфизм и виртуальные функции	Перегрузка операторов (унарных и бинарных). Виртуальные функции. Абстрактные классы и чистые виртуальные функции (= 0). Раннее и позднее связывание. Виртуальные деструкторы.
9.	Шаблоны (Templates)	Шаблоны функций. Шаблоны классов. Специализация шаблонов. Стандартная библиотека шаблонов (STL): контейнеры (vector, list, map, set), итераторы, алгоритмы (sort, find, copy).
10.	Исключения и обработка ошибок	Механизм исключений (try, catch, throw). Стандартные исключения (std::exception и его наследники). Создание собственных исключений. Гарантия безопасности исключений.
11.	Управление памятью и умные указатели	Проблемы ручного управления памятью. Умные указатели (std::unique_ptr, std::shared_ptr, std::weak_ptr). Принцип RAII (Resource Acquisition Is Initialization). Управление ресурсами.
12.	Файловый ввод-вывод и работа с потоками	Содержание: работа с файлами (ifstream, ofstream, fstream). Режимы открытия файлов. Чтение и запись данных. Бинарные и текстовые файлы. Обработка ошибок ввода-вывода.

6.2.3 Содержание самостоятельной работы

№ п/п	Наименование раздела дисциплины	Содержание самостоятельной работы
1.	Введение в C++ и основы синтаксиса. Управляющие конструкции	История и особенности языка C++. Структура программы: main(), директивы препроцессора (#include). Компиляция и выполнение программы. Базовые типы данных (int, float, double, char, bool, void). Переменные и константы. Операторы и выражения. Ввод/вывод (cin, cout). Условные операторы (if, else if, else, switch). Циклы (for, while, do-while). Операторы управления циклом (break, continue). Примеры решения типовых задач (поиск максимума, суммы, факториала).
2.	Функции в C++	Объявление и определение функций. Параметры (по значению, по ссылке, по указателю). Возвращаемое значение. Перегрузка функций. Рекурсия. Область видимости переменных. Статические переменные.
3.	Массивы и строки	Одномерные и многомерные массивы. Инициализация массивов. Строки в стиле C (char[]) и класс std::string. Операции над строками. Поиск и сортировка в массивах (линейный поиск, пузырьковая сортировка).

4.	Указатели и ссылки	Понятие указателя. Операторы * и &. Арифметика указателей. Указатели и массивы. Ссылки. Передача параметров по ссылке и по значению. Динамическое выделение памяти (new/delete). Проблемы утечек памяти.
5.	Основы ООП: классы и объекты	Принципы ООП (инкапсуляция, наследование, полиморфизм). Классы и объекты. Конструкторы (включая конструктор по умолчанию, копирующий) и деструкторы. Модификаторы доступа (public, private, protected). Статические члены класса.
6.	Инкапсуляция и абстракция данных	Инкапсуляция данных. Геттеры и сеттеры. Константные методы. Дружественные функции и классы (friend). Константные объекты. Принципы проектирования интерфейсов классов.
7.	Наследование и композиция	Типы наследования (public, protected, private). Конструкторы и деструкторы при наследовании. Множественное наследование. Виртуальное наследование. Композиция и агрегация.
8.	Полиморфизм и виртуальные функции	Перегрузка операторов (унарных и бинарных). Виртуальные функции. Абстрактные классы и чистые виртуальные функции (= 0). Раннее и позднее связывание. Виртуальные деструкторы.
9.	Шаблоны (Templates)	Шаблоны функций. Шаблоны классов. Специализация шаблонов. Стандартная библиотека шаблонов (STL): контейнеры (vector, list, map, set), итераторы, алгоритмы (sort, find, copy).
10.	Исключения и обработка ошибок	Механизм исключений (try, catch, throw). Стандартные исключения (std::exception и его наследники). Создание собственных исключений. Гарантия безопасности исключений.
11.	Управление памятью и умные указатели	Проблемы ручного управления памятью. Умные указатели (std::unique_ptr, std::shared_ptr, std::weak_ptr). Принцип RAII (Resource Acquisition Is Initialization). Управление ресурсами.
12.	Файловый ввод-вывод и работа с потоками	Содержание: работа с файлами (ifstream, ofstream, fstream). Режимы открытия файлов. Чтение и запись данных. Бинарные и текстовые файлы. Обработка ошибок ввода-вывода.

7. Текущий контроль по дисциплине (модулю) в рамках учебных занятий

В рамках текущего контроля преподаватель самостоятельно может проводить следующие мероприятия:

№ п/п	Контролируемые разделы (темы)	Наименование оценочного средства
1.	Введение в C++ и основы синтаксиса. Управляющие конструкции	Опрос, информационный проект.
2.	Функции в C++	Опрос, информационный проект,

		тестирование.
3.	Массивы и строки	Опрос, информационный проект.
4.	Указатели и ссылки	Опрос, информационный проект.
5.	Основы ООП: классы и объекты	Опрос, информационный проект, тестирование.
6.	Инкапсуляция и абстракция данных	Опрос, информационный проект.
7.	Наследование и композиция	Опрос, информационный проект.
8.	Полиморфизм и виртуальные функции	Опрос, тестирование.
9.	Шаблоны (Templates)	Опрос, тестирование.
10.	Исключения и обработка ошибок	Опрос, информационный проект.
11.	Управление памятью и умные указатели	Опрос, информационный проект.
12.	Файловый ввод-вывод и работа с потоками	Опрос, тестирование.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

8.1. Основная литература:

1. Демаков, А. В. Объектно-ориентированное программирование : учебное пособие / А. В. Демаков, А. А. Квасников, С. П. Куксенко. — Томск : Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2022. — 190 с. — ISBN 978-5-4332-0302-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/152823.html>
2. Ситняковская, Е. И. Основные понятия объектно-ориентированного программирования: справочник с примерами на C++ : учебно-методическое пособие / Е. И. Ситняковская, Е. В. Янченко. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2024. — 27 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/149528.html>
3. Объектно-ориентированное программирование. В 3-х частях. Ч.1 : учебное пособие / П. П. Степанов, А. А. Кабанов, В. А. Никонов, Т. С. Павлюченко. — Омск : Омский государственный технический университет, 2021. — 112 с. — ISBN 978-5-8149-3301-0 (ч.1), 978-5-8149-3300-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124850.html>
4. Достовалов, Д. Н. Объектно-ориентированный анализ и проектирование. Задачи и примеры на C++ : учебное пособие / Д. Н. Достовалов, О. В. Лауферман. — Новосибирск : Новосибирский государственный технический университет, 2022. — 74 с. — ISBN 978-5-7782-4708-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL:

<https://www.iprbookshop.ru/126577.html>

8.2. Дополнительная литература:

1. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход : учебное пособие / С. В. Зыков. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 187 с. — ISBN 978-5-4497-0926-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146342.html>
2. Толмачев, Д. Е. Методы объектно-ориентированного анализа и программирования в управлении экономическими системами : учебное пособие / Д. Е. Толмачев, А. А. Тарасьев, В. Ф. Турыгина. — Екатеринбург : Издательство Уральского университета, 2023. — 130 с. — ISBN 978-5-7996-3683-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/157207.html>
3. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : Сибирский федеральный университет, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/100067.html>

8.3. Периодические издания

1. Вестник Московского государственного технического университета имени Н.Э. Баумана. Серия Естественные науки. ISSN 1812-3368. <https://www.iprbookshop.ru/23124.html>
2. Открытые Системы. СУБД. ISSN 1028-7493. <https://www.iprbookshop.ru/76383.html>
3. Информационные технологии моделирования и управления. ISSN 1813-9744. <https://www.iprbookshop.ru/43350.html>

9. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины (модуля)

1. Федеральный портал «Российское образование» <http://www.edu.ru>
2. Электронно-библиотечная система «Научная электронная библиотека eLIBRARY.RU» <https://www.elibrary.ru>
3. Электронно-библиотечная система IPR BOOKS <https://www.iprbookshop.ru>
4. Электронная библиотечная система <https://biblioclub.ru>

10. Методические указания для обучающихся по освоению дисциплины (модуля)

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности – лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

1. Работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;

2. Внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
3. Выполнение самостоятельных практических работ;
4. Подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.
2. Интенсивная подготовка должна начаться не позднее, чем за месяц до экзамена.
3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. Microsoft Windows Server;
 2. Семейство ОС Microsoft Windows;
 3. Libre Office свободно распространяемый офисный пакет с открытым исходным кодом;
 4. Информационно-справочная система: Система КонсультантПлюс (КонсультантПлюс);
 5. Информационно-правовое обеспечение Гарант: Электронный периодический справочник «Система ГАРАНТ» (Система ГАРАНТ);
- Перечень используемого программного обеспечения указан в п.12 данной рабочей программы дисциплины.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

12.1. Учебная аудитория для проведения учебных занятий, предусмотренных образовательной программой, оснащенная оборудованием и техническими средствами обучения.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя, проектор, экран, колонки

Перечень лицензионного программного обеспечения, в том числе отечественного производства:

Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Яндекс Браузер, LibreOffice, МТС Линк.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду ММУ.

12.2. Помещение для самостоятельной работы обучающихся.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя; компьютеры в сборе для обучающихся; колонки; проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства:

Windows Server 2016, Windows 10, Microsoft Office, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Яндекс Браузер, LibreOffice, МТС Линк, Gimp, Paint.net, AnyLogic, Inkscape.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ММУ.

13. Образовательные технологии, используемые при освоении дисциплины

Для освоения дисциплины используются как традиционные формы занятий – лекции (типы лекций – установочная, вводная, текущая, заключительная, обзорная; виды лекций – проблемная, визуальная, лекция конференция, лекция консультация); и семинарские (практические) занятия, так и активные и интерактивные формы занятий - деловые и ролевые игры, решение ситуационных задач и разбор конкретных ситуаций.

На учебных занятиях используются технические средства обучения мультимедийной аудитории: компьютер, монитор, колонки, настенный экран, проектор, микрофон, пакет программ Microsoft Office для демонстрации презентаций и медиафайлов, видеопроектор для демонстрации слайдов, видеосюжетов и др. Тестирование обучаемых может осуществляться с использованием компьютерного оборудования университета.

13.1. В освоении учебной дисциплины используются следующие традиционные образовательные технологии:

- чтение проблемно-информационных лекций с использованием доски и видеоматериалов;
- семинарские занятия для обсуждения, дискуссий и обмена мнениями;
- контрольные опросы;

- консультации;
- самостоятельная работа студентов с учебной литературой и первоисточниками;
- подготовка и обсуждение рефератов (проектов), презентаций (научно-исследовательская работа);
- тестирование по основным темам дисциплины.

13.2. Активные и интерактивные методы и формы обучения

Из перечня видов: («мозговой штурм», анализ НПА, анализ проблемных ситуаций, анализ конкретных ситуаций, инциденты, имитация коллективной профессиональной деятельности, разыгрывание ролей, творческая работа, связанная с освоением дисциплины, ролевая игра, круглый стол, диспут, беседа, дискуссия, мини-конференция и др.) используются следующие:

- диспут;
- анализ проблемных, творческих заданий, ситуационных задач;
- ролевая игра;
- круглый стол;
- мини-конференция;
- дискуссия;
- беседа.

13.3. Особенности обучения инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ)

При организации обучения по дисциплине учитываются особенности организации взаимодействия с инвалидами и лицами с ограниченными возможностями здоровья (далее – инвалиды и лица с ОВЗ) с целью обеспечения их прав. При обучении учитываются особенности их психофизического развития, индивидуальные возможности и при необходимости обеспечивается коррекция нарушений развития и социальная адаптация указанных лиц.

Выбор методов обучения определяется содержанием обучения, уровнем методического и материально-технического обеспечения, особенностями восприятия учебной информации студентами с инвалидностью и студентами с ограниченными возможностями здоровья и т.д. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение и дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены печатными и электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Автономная некоммерческая организация высшего образования
«МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ»

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ**

Объектно-ориентированное программирование

<i>Направление подготовки</i>	Информационные системы и технологии
<i>Код</i>	09.03.02
<i>Направленность (профиль)</i>	Информационные технологии и большие данные
<i>Квалификация (степень) выпускника</i>	бакалавр

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-4
Профессиональные	-	ПК-6

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-4	ПК-4. Способен применять методы и средства проектирования программного обеспечения и баз данных	ПК-4.1. Адаптация бизнес-процессов к возможностям типовой ИС. Разработка модели бизнес-процессов. Проектирование и дизайн типовой ИС. Моделирование бизнес-процессов ПК-4.2. Сбор данных о запросах и потребностях заказчика применительно к типовой ИС. Документирование собранных данных в соответствии с регламентами организации. ПК-4.3. Согласование и утверждение требований к типовой ИС. ПК-4.4. Разработка прототипов ИС на базе типовой ИС в соответствии с требованиями. ПК-4.5. Интеграция ИС с существующими ИС заказчика
ПК-6	ПК-6. Способен осуществлять администрирование процесса управления безопасностью сетевых устройств и программного обеспечения.	ПК-6.1. Определяет параметры безопасности и защиты программного обеспечения сетевых устройств ПК-6.2. Понимает принципы обеспечения безопасности и защиты программного обеспечения сетевых устройств. ПК-6.3. Выполняет установку и настройку специализированных программных средств обеспечения безопасности, настройку параметров безопасности операционных систем сетевых устройств. ПК-6.4. Понимает принципы обеспечения безопасности и защиты программного обеспечения сетевых устройств.

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	ПК-4		
	- методы оценки объемов и сроков выполнения работ, технологии выполнения работ в организации; - архитектуру, устройство и функционирование вычислительных систем, коммуникационное оборудование, сетевые протоколы; - основы современных операционных систем, основы современных систем управления базами данных, устройство и функционирование современных ИС; - теорию баз данных, системы хранения и анализа баз данных; - основы программирования, современные объектно-ориентированные языки программирования, современные структурные языки программирования, языки современных бизнес-приложений; - современные методики тестирования разрабатываемых ИС, современные стандарты информационного взаимодействия систем; - программные средства и платформы инфраструктуры информационных	- работать с современными системами программирования, конструировать программное обеспечение и базы данных, разрабатывать основные программные документы; - оценивать объемы и сроки выполнения работ; - разрабатывать руководство программиста к модифицированным элементам типовой ИС; - разрабатывать руководство администратора к модифицированным элементам типовой ИС; - разрабатывать руководство пользователя к модифицированным элементам типовой ИС; - разрабатывать ТЗ.	- навыками конструирования программного обеспечения и баз данных; - навыками разработки интерфейсов обмена данными, форматов обмена данными, технологий обмена данными между ИС и существующими системами в соответствии с техническим заданием.

	технологий организаций; - современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP, ITIL, ITSM); - основы теории систем и системного анализа; - методики описания и моделирования бизнес-процессов, средства моделирования бизнес-процессов; - системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; - отраслевую нормативную техническую документацию.		
Код компетенции	ПК-6		
	Основные понятия и принципы функционирования информационных систем, используемых для автоматизации бизнес-процессов и организационного управления. Методы и технологии проектирования, разработки и сопровождения информационных систем. Стандарты и методологии управления проектами в области информационных технологий.	Анализировать потребности организации и разрабатывать технические задания для создания или модификации информационных систем. Проектировать архитектуру информационных систем с учетом специфики бизнес-процессов организации. Использовать современные инструменты и технологии для разработки и внедрения информационных систем. Осуществлять управление проектами по созданию и сопровождению информационных систем, включая планирование,	Навыками работы с инструментами разработки и сопровождения информационных систем, включая языки программирования и платформы. Умением применять методологии управления проектами, такие как Agile, Scrum или Waterfall, в контексте разработки ИС. Компетенцией в области анализа и оптимизации

	Основные бизнес-процессы и задачи, подлежащие автоматизации в организации. Правила и подходы к тестированию, внедрению и сопровождению информационных систем.	контроль и оценку результатов. Проводить обучение пользователей и обеспечивать техническую поддержку информационных систем.	бизнес-процессов с использованием информационных технологий. Способностью разрабатывать и внедрять решения для автоматизации задач организационного управления. Умением работать в команде, взаимодействовать с различными заинтересованными сторонами и управлять проектами на всех этапах их жизненного цикла.
--	--	--	---

3.2. Критерии оценки знаний студентов

Шкала оценивания	Индикаторы достижения	Показатели оценивания результатов обучения
ОТЛИЧНО/Зачтено	Знает:	- студент глубоко и всесторонне усвоил материал, уверенно, логично, последовательно и грамотно его излагает, опираясь на знания основной и дополнительной литературы, - на основе системных научных знаний делает квалифицированные выводы и обобщения, свободно оперирует категориями и понятиями.
	Умеет:	- студент умеет самостоятельно и правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, используя научные понятия, ссылаясь на нормативную базу.
	Владеет:	- студент владеет рациональными методами (с использованием рациональных методик) решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении продемонстрировал навыки - выделения главного, - связкой теоретических положений с требованиями руководящих документов, - изложения мыслей в логической последовательности, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.

ХОРОШО/Зачтено	Знает:	- студент твердо усвоил материал, достаточно грамотно его излагает, опираясь на знания основной и дополнительной литературы, - затрудняется в формулировании квалифицированных выводов и обобщений, оперирует категориями и понятиями, но не всегда правильно их верифицирует.
	Умеет:	- студент умеет самостоятельно и в основном правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, не в полной мере используя научные понятия и ссылки на нормативную базу.
	Владеет:	- студент в целом владеет рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; - При решении смог продемонстрировать достаточность, но не глубинность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
УДОВЛЕТВОРИТЕЛЬНО/Зачтено	Знает:	- студент ориентируется в материале, однако затрудняется в его изложении; - показывает недостаточность знаний основной и дополнительной литературы; - слабо аргументирует научные положения; - практически не способен сформулировать выводы и обобщения; - частично владеет системой понятий.
	Умеет:	- студент в основном умеет решить учебно-профессиональную задачу или задание, но допускает ошибки, слабо аргументирует свое решение, недостаточно использует научные понятия и руководящие документы.
	Владеет:	- студент владеет некоторыми рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; - При решении продемонстрировал недостаточность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
НЕУДОВЛЕТВОРИТЕЛЬНО/Не	Знает:	- студент не усвоил значительной части материала; - не может аргументировать научные положения; - не формулирует квалифицированных выводов и обобщений; - не владеет системой понятий.
	Умеет:	студент не показал умение решать учебно-профессиональную задачу или задание.

	Владеет:	не выполнены требования, предъявляемые к навыкам, оцениваемым “удовлетворительно”.
--	----------	--

При ответе на вопросы в рамках прохождения промежуточной аттестации (зачет/зачет с оценкой/экзамен) допускается вольная формулировка ответа, по смыслу раскрывающая содержание ответа, указанного в фонде оценочных средств, в качестве верного ответа.

4. Типовые контрольные задания (закрытого, открытого и иного типа) для проведения промежуточной аттестации, необходимые для оценки достижения компетенции, соотнесенной с результатами обучения по дисциплине

ТЕСТИРОВАНИЕ

3 СЕМЕСТР ПК-4

1. Какой оператор используется для вывода данных в C++?
 а) **cout**
 б) print
 в) printf
 г) write
2. Какой тип данных используется для хранения целых чисел?
 а) **int**
 б) float
 в) double
 г) char
3. Какой оператор используется для ввода данных?
 а) **cin**
 б) input
 в) read
 г) scanf
4. Какой цикл выполняется хотя бы один раз?
 а) for
 б) while
 в) **do-while**
 г) foreach
5. Как объявить функцию без возвращаемого значения?
 а) **void function()**
 б) null function()

- в) empty function()
 - г) function()
6. Что такое перегрузка функций?
- а) Несколько функций с одним именем, но разными параметрами**
 - б) Повторное объявление одной функции
 - в) Вызов функции внутри себя
 - г) Создание функции в функции
7. Как объявить массив из 10 целых чисел?
- а) int array[10];**
 - б) array int[10];
 - в) int[10] array;
 - г) int array = new int[10];
8. Что возвращает оператор sizeof?
- а) Размер в байтах**
 - б) Количество элементов
 - в) Адрес памяти
 - г) Тип данных
9. Что такое указатель?
- а) Переменная, хранящая адрес памяти**
 - б) Ссылка на функцию
 - в) Специальный тип данных
 - г) Константа
10. Как выделить память динамически?
- а) malloc
 - б) new**
 - в) alloc
 - г) create
11. Что такое класс в C++?
- а) Шаблон для создания объектов**
 - б) Функция
 - в) Структура данных
 - г) Оператор
12. Какой модификатор доступа делает члены класса доступными только внутри класса?
- а) public
 - б) private**
 - в) protected
 - г) internal
13. Что такое конструктор?
- а) Метод для инициализации объекта**
 - б) Метод для уничтожения объекта
 - в) Специальный тип данных
 - г) Оператор
14. Что такое деструктор?
- а) Метод, вызываемый при уничтожении объекта**
 - б) Метод для копирования объекта
 - в) Функция для создания объекта
 - г) Специальный оператор
15. Что такое инкапсуляция?
- а) Скрытие данных и методов внутри класса**
 - б) Наследование классов

- в) Переопределение методов
- г) Создание объектов

3 СЕМЕСТР ПК-6

1. Как объявить константную функцию?
 - а) function const()
 - б) function() const**
 - в) const function()
 - г) constant function()
2. Что такое дружественная функция?
 - а) Функция, имеющая доступ к приватным членам класса**
 - б) Статическая функция
 - в) Виртуальная функция
 - г) Конструктор
3. Какой оператор используется для доступа к членам объекта через указатель?
 - а) .
 - б) ->**
 - в) ::
 - г) #
4. Что такое статический член класса?
 - а) Член, общий для всех объектов класса**
 - б) Член, доступный только в конструкторе
 - в) Член, созданный при вызове метода
 - г) Член, удаляемый при уничтожении объекта
5. Как передать параметр по ссылке?
 - а) int x
 - б) int& x**
 - в) int* x
 - г) ref int x
6. Какой тип данных используется для логических значений?
 - а) bool**
 - б) logic
 - в) boolean
 - г) flag
7. Как завершить выполнение функции и вернуть значение?
 - а) exit
 - б) return**
 - в) stop
 - г) end
8. Какой оператор используется для проверки условий?
 - а) if**
 - б) while
 - в) for
 - г) switch
9. Что делает оператор break в цикле?
 - а) Прерывает выполнение цикла**
 - б) Продолжает выполнение следующей итерации

- в) Завершает программу
 - г) Вызывает исключение
10. Как объявить строку в стиле C?
- а) `char str[100];`**
 - б) `string str;`
 - в) `char* str;`
 - г) `text str;`
11. Что такое рекурсия?
- а) Вызов функции самой себя**
 - б) Повторение цикла
 - в) Создание объекта внутри функции
 - г) Передача функции в качестве параметра
12. Как освободить динамически выделенную память?
- а) `free`
 - б) `delete`**
 - в) `release`
 - г) `clear`
13. Какой оператор используется для доступа к членам класса через объект?
- а) `.`**
 - б) `->`
 - в) `::`
 - г) `#`
14. Что произойдёт, если не освободить динамически выделенную память?
- а) Утечка памяти**
 - б) Ошибка компиляции
 - в) Автоматическое освобождение
 - г) Ничего не произойдёт
15. Какой заголовочный файл нужно подключить для использования `cout` и `cin`?
- а) `<iostream>`**
 - б) `<stdio.h>`
 - в) `<conio.h>`
 - г) `<string>`

4 СЕМЕСТР ПК-4

1. Какой тип наследования делает публичные члены базового класса защищёнными в производном классе?
- а) `public`
 - б) `protected`**
 - в) `private`
 - г) `virtual`
2. Что такое виртуальный деструктор?
- а) Деструктор, объявленный с ключевым словом `virtual`**
 - б) Деструктор без тела
 - в) Деструктор в абстрактном классе
 - г) Деструктор с параметрами
3. Как объявить чистую виртуальную функцию?
- а) `virtual void func();`
 - б) `virtual void func() = 0;`**

- в) `pure virtual void func();`
- г) `abstract void func();`
- 4. Что такое шаблон функции?
 - а) **Функция, которая может работать с разными типами данных**
 - б) Функция с фиксированными параметрами
 - в) Виртуальная функция
 - г) Функция-конструктор
- 5. Как объявить шаблон класса?
 - а) `class<T> MyClass;`
 - б) **`template<typename T> class MyClass;`**
 - в) `generic class MyClass;`
 - г) `template class MyClass;`
- 6. Какой контейнер STL хранит элементы в виде связанного списка?
 - а) `vector`
 - б) **`list`**
 - в) `map`
 - г) `set`
- 7. Что делает алгоритм `sort` из STL?
 - а) **Сортирует элементы контейнера**
 - б) Ищет элемент в контейнере
 - в) Удаляет дубликаты
 - г) Обращает порядок элементов
- 8. Как обработать исключение в C++?
 - а) `try { ... } except { ... }`
 - б) **`try { ... } catch { ... }`**
 - в) `if { ... } else { ... }`
 - г) `handle { ... } error { ... }`
- 9. Какой умный указатель гарантирует уникальность владения объектом?
 - а) **`std::unique_ptr`**
 - б) `std::shared_ptr`
 - в) `std::weak_ptr`
 - г) Обычный указатель
- 10. Что такое RAII?
 - а) **Принцип управления ресурсами через время жизни объектов**
 - б) Тип данных
 - в) Алгоритм сортировки
 - г) Механизм наследования
- 11. Какое ключевое слово используется для автовывода типа переменной?
 - а) `var`
 - б) **`auto`**
 - в) `dynamic`
 - г) `infer`
- 12. Как записать цикл `for` на основе диапазона?
 - а) `for (int i = 0; i < n; i++)`
 - б) **`for (auto x : container)`**
 - в) `foreach (x in container)`
 - г) `loop (x in container)`
- 13. Что такое лямбда-выражение?
 - а) **Анонимная функция**
 - б) Виртуальный метод
 - в) Шаблон функции
 - г) Тип данных

14. Что означает move-семантика?
- а) **Перенос владения ресурсами вместо копирования**
 - б) Копирование объекта
 - в) Удаление объекта
 - г) Создание ссылки
15. Какой оператор создаёт rvalue-ссылку?
- а) &
 - б) **&&**
 - в) *
 - г) #
16. Как закрыть файловый поток?
- а) close(file);
 - б) **file.close();**
 - в) end(file);
 - г) stop(file);
17. Что такое nullptr в C++?
- а) **Нулевой указатель, безопасный аналог NULL**
 - б) Ключевое слово для объявления константы
 - в) Функция для освобождения памяти
 - г) Тип данных
18. Что означает ключевое слово constexpr?
- а) **Выражение может быть вычислено на этапе компиляции**
 - б) Константная функция
 - в) Переменная только для чтения
 - г) Виртуальная функция
19. Как передать объект в функцию без копирования (используя move-семантику)?
- а) По значению
 - б) По константной ссылке
 - в) **По rvalue-ссылке (&&)**
 - г) Через указатель
20. Что такое лямбда-захват по значению?
- а) **Копирование переменных в лямбда-выражение**
 - б) Передача переменных по ссылке
 - в) Создание указателя на переменную
 - г) Игнорирование переменных

4 СЕМЕСТР ПК-6

1. Что такое абстрактный класс?
- а) **Класс, содержащий хотя бы одну чистую виртуальную функцию**
 - б) Класс без методов
 - в) Класс с приватными членами
 - г) Шаблон класса
2. Какой тип наследования используется по умолчанию для классов?
- а) public
 - б) **private**
 - в) protected
 - г) Нет значения по умолчанию

3. Что такое полиморфизм в ООП?
 - а) **Способность объекта обрабатывать данные разных типов**
 - б) Создание новых классов
 - в) Инкапсуляция данных
 - г) Наследование методов
4. Как вызывать метод базового класса из производного?
 - а) `base::method()`
 - б) **`BaseClass::method()`**
 - в) `super::method()`
 - г) `parent::method()`
5. Что такое перегрузка операторов?
 - а) **Определение поведения операторов для пользовательских типов**
 - б) Изменение приоритета операторов
 - в) Создание новых операторов
 - г) Удаление операторов
6. Какой контейнер STL реализует ассоциативный массив?
 - а) `vector`
 - б) `list`
 - в) **`map`**
 - г) `queue`
7. Что возвращает функция `size()` у контейнера STL?
 - а) **Количество элементов в контейнере**
 - б) Размер в байтах
 - в) Максимальный размер
 - г) Индекс последнего элемента
8. Как проверить, пуст ли контейнер STL?
 - а) `if (container == NULL)`
 - б) **`if (container.empty())`**
 - в) `if (size == 0)`
 - г) `if (!container)`
9. Что такое итератор в STL?
 - а) **Объект, позволяющий перемещаться по элементам контейнера**
 - б) Функция сортировки
 - в) Тип данных
 - г) Контейнер
10. Какой умный указатель поддерживает совместное владение объектом?
 - а) `std::unique_ptr`
 - б) **`std::shared_ptr`**
 - в) `std::weak_ptr`
 - г) Обычный указатель
11. Что делает `std::weak_ptr`?
 - а) **Слабая ссылка на объект, управляемый `shared_ptr`**
 - б) Уникальный указатель
 - в) Обычный указатель
 - г) Итератор
12. Какой заголовочный файл нужен для использования умных указателей?
 - а) **`<memory>`**
 - б) `<pointer>`
 - в) `<smart>`
 - г) `<util>`
13. Что такое специализация шаблона?
 - а) **Определение отдельной реализации шаблона для конкретного типа**

- б) Наследование шаблона
 - в) Перегрузка шаблона
 - г) Создание экземпляра шаблона
14. Как выбросить исключение в C++?
- а) raise exception;
 - б) throw exception;**
 - в) emit exception;
 - г) send exception;
15. Что такое множественное наследование?
- а) Наследование от нескольких базовых классов**
 - б) Наследование с модификатором virtual
 - в) Наследование шаблонов
 - г) Последовательное наследование
16. Какой заголовочный файл нужен для работы с файловым вводом-выводом?
- а) <fstream>**
 - б) <file>
 - в) <iofile>
 - г) <stream>
17. Как открыть файл для чтения?
- а) file.open("file.txt", ios::write);
 - б) file.open("file.txt", ios::in);**
 - в) file.read("file.txt");
 - г) open("file.txt");
18. Что делает метод eof() у файлового потока?
- а) Проверяет, достигнут ли конец файла**
 - б) Закрывает файл
 - в) Читает следующий символ
 - г) Перемещает указатель в начало файла
19. Какой режим используется для добавления данных в конец файла?
- а) ios::in
 - б) ios::out
 - в) ios::app**
 - г) ios::trunc
20. Что произойдёт, если попытаться прочитать из несуществующего файла?
- а) Программа аварийно завершится
 - б) Будет установлен флаг ошибки потока**
 - в) Файл автоматически создастся
 - г) Ничего не произойдёт