МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Донской казачий государственный институт пищевых технологий и бизнеса (филиал) федерального государственного бюджетного образовательного учреждения «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)»

УТВЕРЖДАЮ
И.о. директора ДКГИПТиБ (филиала)
Г.В. Павлятенко
23 апреля 2024 г.

Рабочая программа дисциплины (модуля)

Б1.О.03.03 МОДУЛЬ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ
Программирование

Кафедра:	Физика, математика и информационные технологии
Направление подготовки:	09.03.03 Прикладная информатика
Направленность (профиль):	Управление проектами в области информационных технологий
Квалификация выпускника:	Бакалавр
Форма обучения:	очно-заочная
Общая трудоемкость:	144 часов/4 з.е.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 00AA7853CEFA35F0837BA866654F3503A6
Владелец: Павлятенко Геннадий Владимирович
Действителен: с 08.12.2025 до 03.03.2027

Ростов-на-Дону, 2024 г.

Программу составил(и):
ст. преподаватель Ковалев Д.В.

Рабочая программа дисциплины (модуля)
"Программирование"
разработана составлена на основании учебного плана, утвержденного ученым советом 28 марта 2024 г. протокол № 9 в соответствии
с ФГОС ВО Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Руководитель ОПОП
доц., канд. экон. наук, доц. Мурат Е.П._____ _____

Рабочая программа обсуждена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от 23 апреля 2024 г. № 8
И.о. зав. кафедрой Мурат Елена Петровна _____

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Цели:

Цели освоения дисциплины "Программирование":
• обеспечить студентов базовыми знаниями в области разработки программных продуктов;
• заложить основы для последующих курсов, посвященных созданию современных информационных систем;
• познакомить студентов с прогрессивными парадигмами программирования и механизмами их реализации в программных продуктах;
• обучить студентов применению современных интегрированных инструментальных сред, предназначенных для разработки программ в интерактивном режиме;
• привить студентам навыки исследовательской работы, предполагающей самостоятельное изучение специфических инструментов и средств, необходимых для решения именно той конкретной проблемы, которая в качестве задачи поставлена перед ними.

1.2. Задачи:

-развитие и систематизация знаний, относящихся к математическим основам информатики, к принципам организации и функционирования программных и аппаратных средств вычислительной техники;
- расширение опыта создания, редактирования, хранения, передачи информационных объектов различного типа с помощью современных программных и коммуникационных средств с соблюдением соответствующих правовых и этических норм и требований информационной безопасности.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

Цикл (раздел) ОП: Б1.О

Дисциплина относится к обязательной части ОПОП и обязательна для освоения.

Связь с предшествующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Основы алгоритмизации и программирования	1	ОПК-7.1, ОПК-7.2, ОПК-7.3
2	Разработка программных приложений	2	ОПК-7.1, ОПК-7.2, ОПК-7.3
3	Основы информационных технологий	1	ОПК-2.1, ОПК-2.2, ОПК-2.3, УК-1.1, УК-1.2, УК-1.3
4	Пакеты прикладных программ для профессиональной деятельности	2	ОПК-2.1, ОПК-2.2, ОПК-2.3

Связь с последующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Операционные системы	5	ОПК-2.1, ОПК-2.2, ОПК-2.3
2	Ознакомительная практика	4	УК-7.1, УК-7.2, УК-7.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, УК-4.1, УК-4.2, УК-4.3
3	Интеллектуальные информационные системы	6	ОПК-2.1, ОПК-2.2, ОПК-2.3
4	Технологическая (проектно-технологическая) практика	6	УК-1.1, УК-1.2, УК-1.3, УК-2.1, УК-2.2, УК-2.3, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-3.5, УК-3.6, УК-5.1, УК-5.2, УК-5.3, УК-8.1, УК-8.2, УК-8.3, УК-8.4, УК-8.5, УК-8.6, УК-9.1, УК-9.2, УК-9.3, УК-10.1, УК-10.2, УК-10.3, ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-3.1, ОПК-3.2, ОПК-3.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК-5.3, ОПК-6.1, ОПК-6.2, ОПК-6.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-8.1, ОПК-8.2, ОПК-8.3, ОПК-9.1, ОПК-9.2, ОПК-9.3, УК-4.4, УК-4.5, УК-4.6, УК-6.1, УК-6.2, УК-6.3

Распределение часов дисциплины

Семестр (<Курс>.<Семестр на курсе>)	**3 (2.1)**		Итого	
Недель	16 5/6			
Вид занятий	УП	РП	УП	РП

Лекции	4	4	4	4
Лабораторные	20	20	20	20
В том числе электрон.	60	60	60	60
Итого ауд.	24	24	24	24
Контактная работа	24	24	24	24
Сам. работа	84	84	84	84
Часы на контроль	36	36	36	36
Итого	144	144	144	144

Вид промежуточной аттестации:

Экзамен 3 семестр

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций и индикаторов их

ОПК-2:Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

ОПК-2.1: Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы

ОПК-2.2: Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии

ОПК-2.3: Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-7.1: Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий

ОПК-7.2: Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

ОПК-7.3: Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименования разделов, тем, их краткое содержание и результаты освоения /вид занятия/	Семестр	Часов	Интеракт.	Прак. подг.	Индикаторы достижения компетенции	Оценочные средства
	Раздел 1.Программирование						
1.1	**Тема 1. Основные понятия объектно-ориентированного программирования Краткое содержание Основные понятия объектно-ориентированного программирования. Основные принципы ООП. Классы, поля, методы. Статусы доступа. Объекты. Конструкторы, деструкторы. Перегрузка**	3	2	0	0	ОПК-2.1,ОПК-7.1	Устный опрос

	стандартных операций в С++. Статические члены класса. Дружественные функции. Предполагаемые результаты обучения: Знать: основные понятия и принципы объектно-ориентированного программирования /Лек/						
1.2	**Тема 1. Основные понятия объектно-ориентированного программирования Краткое содержание Основные понятия объектно-ориентированного программирования. Основные принципы ООП. Классы, поля, методы. Статусы доступа. Объекты. Конструкторы, деструкторы. Перегрузка стандартных операций в С++. Статические члены класса. Дружественные функции. Предполагаемые результаты обучения: Уметь: использовать средства и принципы объектно-ориентированного программирования для решения задач профессиональной деятельности Владеть: навыками использования средств и принципов объектно-ориентированного программирования для решения задач профессиональной деятельности /Лаб/**	3	6	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.3	**Тема 1. Основные понятия объектно-ориентированного программирования Краткое содержание Основные понятия объектно-ориентированного программирования. Основные принципы ООП. Классы, поля, методы. Статусы доступа. Объекты. Конструкторы, деструкторы. Перегрузка стандартных операций в С++. Статические члены класса. Дружественные функции. Предполагаемые результаты обучения: Знать: основные понятия и принципы объектно-ориентированного программирования Уметь: использовать средства и принципы объектно-ориентированного программирования для решения задач профессиональной деятельности Владеть: навыками использования средств и принципов объектно-ориентированного программирования для решения**	3	15	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	задач профессиональной деятельности /Ср/						
1.4	**Тема 2. Наследование и полиморфизм** **Краткое содержание:** **Наследование классов. Статусы доступа при наследовании. Виртуальные функции. Абстрактные классы. Полиморфизм** **Предполагаемые результаты обучения:** **Знать: принципы наследования и полиморфизма** **/Лек/**	3	2	0	0	ОПК-2.1,ОПК-7.1	Устный опрос
1.5	**Тема 2. Наследование и полиморфизм** **Краткое содержание:** **Наследование классов. Статусы доступа при наследовании. Виртуальные функции. Абстрактные классы. Полиморфизм** **Предполагаемые результаты обучения:** **Уметь: использовать принципы наследования и полиморфизма при составлении программ для решения задач профессиональной деятельности** **Владеть: навыками использования принципов наследования и полиморфизма при составлении программ для решения задач профессиональной деятельности /Лаб/**	3	10	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.6	**Тема 2. Наследование и полиморфизм** **Краткое содержание:** **Наследование классов. Статусы доступа при наследовании. Виртуальные функции. Абстрактные классы. Полиморфизм** **Предполагаемые результаты обучения:** **Знать: принципы наследования и полиморфизма** **Уметь: использовать принципы наследования и полиморфизма при составлении программ для решения задач профессиональной деятельности** **Владеть: навыками использования принципов наследования и полиморфизма при составлении программ для решения задач профессиональной деятельности /Ср/**	3	18	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки
1.7	**Тема 3. Обработка исключений** **Краткое содержание:** **Обработка ошибок. Группировка и перехват исключений. Управление ресурсами. Исключения и эффективность. Альтернативные методы обработки ошибок. Стандартные исключения.** **Предполагаемые результаты**	3	18	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	обучения: **Знать: понятия исключения и способов его обработки** **Уметь: использовать обработку исключений при составлении программ для решений задач профессиональной деятельности** **Владеть: навыками использования обработки исключений при составлении программ для решений задач профессиональной деятельности /Ср/**						
1.8	**Тема 4. Шаблоны** **Краткое содержание:** **Шаблоны классов. Определение шаблона. Инстанцирование. Параметры шаблонов и проверка типов. Шаблоны функций. Специализация. Наследование и шаблоны.** **Предполагаемые результаты обучения:** **Знать: правила организации шаблонов классов и функций** **Уметь: использовать шаблоны при написании программ для решения задач профессиональной деятельности** **Владеть: навыками использования шаблонов при написании программ для решения задач профессиональной деятельности /Ср/**	3	18	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки
1.9	**Тема 5. Библиотека стандартных шаблонов STL** **Краткое содержание:** **Библиотека стандартных шаблонов STL. Общие сведения (понятия контейнеров, итераторов и объектов -функций). Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры). Итераторы (итератор как обобщение указателя, классы итераторов). Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)** **Предполагаемые результаты обучения:** **Уметь: использовать возможности библиотеки стандартных шаблонов STL при написании программ для решения задач профессиональной деятельности** **Владеть: навыками использования возможностей библиотеки стандартных шаблонов STL при написании программ для решения задач профессиональной деятельности /Лаб/**	3	4	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.10	**Тема 5. Библиотека стандартных шаблонов STL**	3	15	0	0	ОПК-2.1,ОПК-	Вопросы для самоподготовки

	Краткое содержание: **Библиотека стандартных шаблонов STL. Общие сведения (понятия контейнеров, итераторов и объектов -функций). Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры). Итераторы (итератор как обобщение указателя, классы итераторов). Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)** **Предполагаемые результаты обучения:** **Знать: объекты и средства библиотеки стандартных шаблонов STL** **Уметь: использовать возможности библиотеки стандартных шаблонов STL при написании программ для решения задач профессиональной деятельности** **Владеть: навыками использования возможностей библиотеки стандартных шаблонов STL при написании программ для решения задач профессиональной деятельности /Ср/**					2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	
1.11	**Экзамен** **Результаты освоения:** **ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;** **ОПК-2.1: Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы** **ОПК-2.2: Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для**	3	36	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы к экзамену. Итоговое тестирование

	решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии **ОПК-2.3: Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности** **ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения;** **ОПК-7.1: Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий** **ОПК-7.2: Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки**						

	программирования и языки работы с базами данных, среды, разработки информационных систем и технологий ОПК-7.3: Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач /Экзамен/						

Перечень применяемых активных и интерактивных образовательных технологий:
Информационные технологии

Личностно ориентированная технология, способ организации самостоятельной деятельности учащихся, направленный на решение задачи учебного проекта

Компьютерная технология обучения

Основана на использовании информационных технологий в учебном процессе. Реализация данной технологии осуществляется посредством компьютера и иных мультимедийных средств. Использование компьютерных технологий делает учебный процесс не только современным и познавательным, но интересным для обучающихся

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная (внеаудиторная) работа обучающегося (далее - СРС) – это планируемая учебная, практическая, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во
внеаудиторное время (свободное от аудиторных учебных занятий) по заданию и при методическом руководстве педагогического работника, отнесенного к профессорско-преподавательскому составу (далее – преподаватель) и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях, но без их непосредственного участия.
СРС по заданию и при методическом руководстве преподавателя и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях, реализуется во время групповых консультаций и (или) индивидуальной работы обучающихся с преподавателями Университета и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях (в том числе индивидуальных консультаций), а также во время текущего контроля выполнения заданий, отнесенных к самостоятельной работе обучающихся.
Целью СРС является овладение формирование компетенций через овладение знаниями, умениями и навыками профессиональной деятельности по направлению подготовки (специальности). Самостоятельная работа является обязательной для каждого обучающегося.
Формы самостоятельной работы обучающихся определяются преподавателями кафедр Университета при разработке рабочих программ дисциплин (модулей), рабочих программ практик, НИР, программ государственной итоговой (итоговой) аттестации, методических указаний по выполнению практических, лабораторных работ, написанию курсовых работ/проектов и ВКР в соответствии с их содержанием.
В университете оборудованы специальные помещения для самостоятельной работы обучающихся. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Перечень компетенций с указанием этапов формирования индикаторов их достижения в процессе освоения ОПОП

ОПК-2:Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

Недостаточный уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, представления информации, используемые для решения задач профессиональной деятельности, и принципы их работы

Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, для решения задач профессиональной деятельности;

Владеет навыками работы с данными с помощью информационных технологий, для решения задач профессиональной деятельности

Пороговый уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации используемые для решения задач профессиональной деятельности, и принципы их работы

Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, для решения задач профессиональной деятельности;

Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, для решения задач профессиональной деятельности

Продвинутый уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии), используемые для решения задач профессиональной деятельности, и принципы их работы

Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности;

Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, для решения задач профессиональной деятельности

Высокий уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы

Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии

Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Недостаточный уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии);

Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения;

Фрагментарно владеет навыками разработки оригинальных алгоритмов и компьютерных программ

Пороговый уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ;

Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения;

Владеет навыками разработки оригинальных алгоритмов и компьютерных программ

Продвинутый уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования, среды разработки информационных систем и технологий

Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения;

Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения;

Высокий уровень:

Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий

Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

6.2. Шкала оценивания в зависимости от уровня сформированности компетенций

Уровень сформированности компетенций

Характеристики индикаторов достижения компетенций	**1. Недостаточный: компетенции не сформированы.**	**2. Пороговый: компетенции сформированы.**	**3. Продвинутый: компетенции сформированы.**	**4. Высокий: компетенции сформированы.**
Знания:	Знания отсутствуют.	Сформированы базовые структуры знаний.	Знания обширные, системные.	Знания твердые, аргументированные, всесторонние.
Умения:	Умения не сформированы.	Умения фрагментарны и носят репродуктивный характер.	Умения носят репродуктивный характер применяются к решению типовых заданий.	Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий.
Навыки:	Навыки не сформированы.	Демонстрируется низкий уровень самостоятельности практического навыка.	Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка.

Описание критериев оценивания

Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие умения выполнять практические задания, предусмотренные программой дисциплины; - отсутствие готовности (способности) к дискуссии и низкая степень контактности.	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной программой дисциплины; - умение без грубых ошибок решать практические задания, которые следует выполнить.	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала; -способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; - правильные и конкретные, без грубых ошибок ответы на поставленные вопросы; - умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины; - наличие собственной обоснованной позиции по обсуждаемым вопросам. Возможны незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и теории; - логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.

		в ответах на дополнительные вопросы.	
Оценка «незачет», «неудовлетворительно»	**Оценка «зачтено/удовлетворительно», «удовлетворительно»**	**Оценка «зачтено/хорошо», «хорошо»**	**Оценка «зачтено/отлично», «отлично»**

Оценочные средства, обеспечивающие диагностику сформированности компетенций, заявленных в рабочей программе по дисциплине (молулю) для проведения промежуточной аттестации

ОЦЕНИВАНИЕ УРОВНЯ ЗНАНИЙ: Теоретический блок вопросов. Уровень освоения программного материала, логика и грамотность изложения, умение самостоятельно обобщать и излагать материал.
1. Недостаточный уровень
Знает процессы, методы поиска, сбора, хранения, обработки, представления информации, используемые для решения задач профессиональной деятельности, и принципы их работы
Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения;
Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии);
Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, для решения задач профессиональной деятельности;
Владеет навыками работы с данными с помощью информационных технологий, для решения задач профессиональной деятельности
Фрагментарно владеет навыками разработки оригинальных алгоритмов и компьютерных программ
2. Пороговый уровень
Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ;
Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения;
Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, для решения задач профессиональной деятельности
Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, для решения задач профессиональной деятельности;
Владеет навыками разработки оригинальных алгоритмов и компьютерных программ
Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации используемые для решения задач профессиональной деятельности, и принципы их работы
3. Продвинутый уровень
Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, для решения задач профессиональной деятельности
Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения;
Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии), используемые для решения задач профессиональной деятельности, и принципы их работы
Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования, среды разработки информационных систем и технологий
Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности;
Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения;

4. Высокий уровень
Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий
Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности
Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии
Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы
Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий
Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

6.3. Оценочные средства текущего контроля (примерные темы докладов, рефератов, эссе)

Вопросы для устного опроса

Тема 1

1. Какая основная идея объектно-ориентированного программирования?
2. В чем преимущества объектно-ориентированного программирования перед процедурным стилем?
3. В чем состоит принцип абстракции в ООП?
4. Что такое инкапсуляция?
5. Что такое класс в ООП?
6. Какие существуют статусы доступа к членам класса?
7. Как объявляется конструктор класса?
8. Какие виды конструкторов класса бывают?
9. Как объявляется деструктор класса?
10. Как реализуется принцип инкапсуляции в классе?
11. Что такое перегрузка функций?
12. Что такое перегрузка операторов?
13. Что такое дружественная функция?
14. Что такое объект класса?

Тема 2

1. Что такое наследование классов?
2. Какие компоненты базового класса присутствуют в производном классе?
3. Какие методы базового класса не наследуются?
4. Каким образом можно использовать конструктор базового класса?
5. Что происходит при наследовании, если в базовом классе есть метод такой же, как и в производном?
6. От чего зависит статус доступа к членам производного класса?
7. Что такое абстрактные классы?
8. Для чего нужны абстрактные классы в ООП?
9. Что такое интерфейсы?
10. Чем интерфейс отличается от абстрактного класса?
11. Что такое чисто виртуальная функция?
12. Что такое полиморфизм?

Вопросы для самоподготовки

Тема 1

1. В чем преимущества объектно-ориентированного программирования перед процедурным стилем?
2. Какие средства языка C++ поддерживают принцип абстракции?
3. Что относится к членам класса?

4. Что определяют поля класса?
5. Что определяют методы класса?
6. Когда вызываются конструкторы класса?
7. Когда вызываются деструкторы класса?
8. Каково назначение дружественной функции класса?
9. Что такое статические поля класса?
10. Что такое статические методы класса?
11. Как создать объект класса?

Тема 2
1. Для чего нужны чисто виртуальные функции?
2. Как объявляется чисто виртуальная функция?
3. Что происходит при вызове виртуальной функции через полиморфный объект?
4. В каких случаях деструктор следует объявлять виртуальным?
5. Что такое таблица виртуальных методов?
6. Для чего нужны интерфейсы в ООП?
7. Как в C++ используется множественное наследование?
8. В чем отличия множественного наследования от единичного наследования?
9. Что такое алмаз смерти в множественном наследовании?
10. Какие существуют проблемы множественного наследования в крупных проектах?

Тема 3
1. Для чего используется блок try при обработке исключений?
2. Для чего используется операция throw при обработке исключений?
3. Для чего используется блок catch при обработке исключений?
4. После какого блока должен следовать перехватчик?
5. Сколько может быть обработчиков для одного исключения?
6. В каком порядке просматриваются обработчики при выполнении программы?
7. Допускается ли вложенность блоков try при обработке исключительных ситуаций?
8. В чем заключается существенное отличие вызова конкретного обработчика от вызова обычной функции?
9. Когда возникает исключение ArgumentOutOf RangeException?
10. Когда возникает исключение OverflowException?

Тема 4
1. Сколько раз определяется шаблон функции?
2. Для каких встроенных типов данных могут использоваться шаблоны функций?
3. Если вы собираетесь использовать некоторый шаблон класса, как узнать какие операции требуются?
4. Что такое меташаблоны?
5. Как создать объект шаблонного класса?
6. Как описываются шаблоны классов с параметрами по умолчанию?
7. Как можно инстанцировать параметры по умолчанию?
8. Поддерживают ли механизм наследования шаблоны классов?
9. В чем достоинства использования шаблонов?
10. В чем недостатки использования шаблонов?

Тема 5
1. На основе какого механизма строится STL?
2. Почему алгоритмы не включены непосредственно в контейнеры?
3. Какие существуют три типа итераторов?
4. С помощью каких итераторов осуществляется доступ к отдельным элементам?
5. Как осуществляется итерация по контейнеру вперед?
6. Как осуществляется итерация по контейнеру назад?
7. Как осуществляется работа с random access iterator итератором?
8. На какие три основные группы можно поделить STL-алгоритмы?
9. Как используется контейнер vector?
10. Как используется контейнер list?
11. Как используется контейнер map?

ОС лабораторных работ
Темы лабораторных работ:
1. Классы, методы, конструкторы, деструкторы, объекты.
2. Наследование классов.
3. Работа с библиотекой STL: vector

6.4. Оценочные средства промежуточной аттестации.
Оценочные средства промежуточной аттестации
Вопросы к экзамену

ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;
Итоговое тестирование:
1. Класс - это ...
а) любой тип данных, определяемый пользователем;
б) структура, для которой в программе имеются функции работы с нею;
в) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки.

2. Членами класса могут быть ...
) только переменные и функции, объявленные как private;
б только переменные, объявленные как private;
в) только функции, объявленные как private;
г) как переменные, так и функции, могут быть объявлены и как private и как public;
д) только переменные и функции, объявленные как public.

3.Что называется конструктором?
а) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса;
б) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса);
в) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса;
г) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса.

4. Объект - это ...
а) переменная, содержащая указатель на класс;
б) класс, который содержит в себе данные и методы их обработки;
в) экземпляр класса.

5. Что называется деструктором?
а) метод, который уничтожает объект;
б) метод, который удаляет объект;
в) метод, который освобождает память, занимаемую объектом;
г) системная функция, которая освобождает память, занимаемую объектом.

6. Что называется наследованием?
а) Это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить;
б) Это механизм переопределения методов базового класса.
в) Это механизм, посредством которого производный класс получает все поля базового класса.
г) Это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы.

7. Какое ключевое слово должно быть использовано, чтобы указать, что объект не является модифицируемым и любая попытка изменения этого типа является ошибкой:
а) readonly
б) const
в) static
г) undeclare

8. Для приведенного ниже объявления класса и фрагмента кода перечислите все идентификаторы, являющиеся именами объектов класса:

Объявление класса	Фрагмент кода
class ThisObj	ThisObj obj1;
{	ThisObj obj2;
public:	obj1.F1(2);
void F1(int n);	int t;
void F2();	double y;
private:	
int x;	
}	

а) x, F1, F2
б) obj1, obj2
в) ThisObj
г) t, y
9. Выберите правильные утверждения:
а) конструктор должен явно вызываться всегда перед объявлением объекта
б) конструктор наследуется, но должен быть перегружен
в) у конструктора могут быть параметры
г) конструктор вызывается автоматически при объявлении объекта
д) объявление каждого класса должно содержать свой конструктор

в) деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
г) деструктор наследуется, но должен быть перегружен;
д) деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом.

11. Выберите правильное объявление производного класса:
а) class MoreDetails: public Details;
б) class MoreDetails: public class Details;
в) class MoreDetails:: Details;
г) class MoreDetails: class(Details);

12. Вызовет данный код ошибку компиляции?
class Rectangle {public: int a, b; int sum (); int square (); ~Rect (); };
а) ошибки нет, все записано верно.
б) ошибка являются: имя деструктора не может начинаться с заглавной буквы.
в) ошибка являются: имя деструктора должно совпадать с именем класса.
г) ошибка являются: никакой идентификатор в С ++ не может начинаться со знака «~».

13. Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void:
а) virtual void SomeFunction (int *x);
б) void SomeFunction (int x) virtual;
в) virtual SomeFunction (int x);
г) virtual void SomeFunction (int x);

14. Укажите правильное использование оператора friend:
а) class A {int_friend CountPass (); private: short i;};
б) class A {public: friend int H :: CountPass (); private: short i;};
в) class A {public: int A1 :: CountPass (); friend: short i;};
г) class A {public: friend int H :: q; short i;};

15. В программе описан абстрактный класс А и производный от этого класса класс А1. Какая из записей заведомо неверна?
а) A *a = new A;
б) A1 a1;
в) A1 a1; A &a = a1;
г) A1 a1; A1 a2;

Знать:
1. Перечислите основные принципы объектно-ориентированного программирования
2. Дайте определение принципа абстракции в ООП
3. Дайте определение класса в ООП
4. Дайте определение конструктора класса
5. Перечислите виды конструкторов класса
6. Дайте определение деструктора класса
7. Дайте определение наследования классов
8. Дайте определение абстрактного класса
9. Дайте определение понятия «полиморфизм»
10. Дайте определение число виртуальной функции
11. Дайте определение абстрактного класса
12. Дайте определение исключения
13. Дайте определение шаблона класса
14. Дайте определение STL
15. Перечислите элементы, образующие ядро библиотеки STL

Уметь:
1. Сформулируйте основную идею объектно-ориентированного программирования.
2. Опишите, как реализуется принцип инкапсуляции в классе
3. Дайте характеристику членам класса.
4. Дайте характеристику перегрузки операторов.
5. Дайте характеристику статических полей класса.
6. Дайте характеристику статических методов класса.
7. Сформулируйте концепцию программирования с защитой от ошибок
8. Охарактеризуйте главную идею обработки исключительных ситуаций
9. Охарактеризуйте формы записи обработчиков исключений
10. Охарактеризуйте права доступа при наследовании класса: private, protected, public
11. Сформулируйте назначение дружественных функций класса.
12. Охарактеризуйте назначение шаблонов класса
13. Охарактеризуйте назначение контейнеров в STL
14. Охарактеризуйте назначение итераторов в STL
15. Охарактеризуйте назначение алгоритмов в STL

Владеть:

1. Разработать класс описания книги с закрытыми полями: название, автор(ы), год издания, издательство, количество страниц

2. В приведенном классе определите поля класса:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```

3. В приведенном классе определите статус полей класса:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```

4. В приведенном классе определите конструктор без параметров в классе:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```

5. В приведенном классе определите конструктор копирования в классе:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```

6. В приведенном классе определите конструктор с параметрами:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
```

Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
7. В приведенном классе определите деструктор:
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
8. В приведенном классе определите сеттер:
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
9. В приведенном классе определите геттер:
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
10. В приведенном классе определите статический метод:
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
11. В приведенном классе определите конструктор с параметрами:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```
12. В приведенном классе определите тип возвращаемого значения метода doSomething():
```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```
13. Для приведенного класса создайте объект(экземпляр) этого класса с именем detail1 со значением параметра 12:
```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```
14. Для приведенного класса для объекта(экземпляра) этого класса с именем detail3 запишите вызов метода doSomething:
```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```
15. Для приведенного класса для объекта с именем выведите в консоль его вес (weight):
```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
protected:
Detail();
Detail(const Detail&);
public: Detail (int);
```

```
static void performAction();
int getSerial() const;
double getWeight() const;
void doSomething();
~Detail(); };
```

ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения
Итоговое тестирование:
1. Выберите правильное утверждение:
а) если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям;
б) если элементы класса объявлены как private, то они недоступны ни наследникам класса, ни внешним функциям;
в) если элементы объявлены как public, то они доступны наследникам класса, но не внешним функциям.

2. Возможность и способ обращения производного класса к элементам базового определяется ...
а) ключами доступа: private, public, protected в заголовке объявления производного класса
б) только ключом доступа protected в заголовке объявления производного класса
в) ключами доступа: private, public, protected в теле производного класса
г) ключами доступа: private, public, protected в теле базового класса

3. Выберите правильное соответствие между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового:
а) ключ доступа - public; в базовом классе: private;
б) ключ доступа - protected или public ; в базовом классе: protected;
в)ключ доступа - private; в базовом классе: public;

1) права доступа в производном классе - protected
2) права доступа в производном классе - private
3)права доступа в производном классе - public

4. Выберите правильное утверждение:
а) одна функция может быть дружественной нескольким классам
б) дружественная функция не может быть обычной функцией, а только методом другого класса
в) дружественная функция не может быть методом другого класса

5. Шаблон функции - это...
а) определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров
б) прототип функции, в котором вместо имен параметров указан условный тип
в) определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
г) определение функции, в которой типу обрабатываемых данных присвоено условное обозначение

6. Выберите правильное утверждение:
а) по умолчанию члены класса имеют атрибут public
б) по умолчанию члены класса имеют атрибут private
в) члены класса имеют доступ только к элементам public;

7. Переопределение операций имеет вид:
а) имя_класса, ключевое слово operation, символ операции
б) имя_класса, ключевое слово operator, список аргументов
в) имя_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы
г) имя_класса, два двоеточия, ключевое слово operator, символ операции

8. Для доступа к элементам объекта используются:
а) при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»
б) при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
в) при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
г) при обращении через имя объекта – точка, при обращении через указатель – операция «->»

9. Полиморфизм – это ...
а) средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
б) средство, позволяющее в одном классе использовать методы с одинаковыми именами
в) средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
г) средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов

10. Полиморфизм в объектно-ориентированном программировании реализуется:
а) через виртуальные функции и шаблоны
б) через механизмы перегрузки (функций и операций) и шаблоны
в) через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны
г) через механизмы перегрузки (функций и операций) и виртуальные функции

11. Виртуальными называются функции:
а) функции производного класса, переопределенные относительно базового класса
б) функции базового класса, которые не используются в производном классе
в) функции базового класса, которые не могут быть переопределены в базовом классе
г) функции базового класса, которые могут быть переопределены в производном классе

12. В программе описан класс и объект:
class A {public: int a, b, c; };
A * obj;
Как обратиться к атрибуту c?
а) obj.c
б) obj A -> -> c
в) obj-> c
г) obj-> A.c

13. Какая из перечисленных функций не может быть конструктором?
а) String (const int a)
б) String ();
в) String (String & s)
г) void String ()

14. Отметьте правильное утверждение для абстрактного класса для языка C ++.
а) Класс, у которого все методы чисто виртуальные, называется абстрактным.
б) Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
в) Невозможно создать объект абстрактного класса.
г) В абстрактном классе не описываются методы вообще.

15. В приведенном фрагменте программы, возможно, содержится ошибка. Выбрать среди приведенных ответов правильный:
```
class A
{ int x;
public: void SetX (int _x) {x=_x;}
}
A *a;
SetX(10);
```
а) для доступа следует написать a.SetX(10);
б) для доступа следует написать A.SetX(10);
в) для доступа следует написать a->SetX(10);
г) для доступа следует написать A->SetX(10);
д) все правильно.

16. Дружественная функция - это ...
а) функция другого класса, среди аргументов которой есть элементы данного класса
б) функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса;
в) функция, являющаяся членом класса и объявленная с атрибутом friend;
г) функция, которая в другом классе объявлена как дружественная данному

17. Укажите правильный вариант доступа к членам объектов, описанных следующим образом: class my {char s;public: double Z; int f (int c, int d) {return c + d;}; } T1, T2;
а) T1.s = ‘#’;
б) T2-> f (2,1);
в) T1.Z = 23.1;
г) my.T2-> s = ‘L‘.

Знать:
1. Дайте определение инкапсуляции в ООП
2. Дайте определение объекта класса в ООП
3. Перечислите классы доступа к членам класса
4. Дайте определение перегрузки функции
5. Дайте определение дружественной функции
6. Дайте определение интерфейса
7. Назовите главную идею обработки исключительных ситуаций.
8. Перечислите блоки, используемые при обработке исключений
9. Дайте определение шаблона функции
10. Перечислите встроенные типы данных, которые могут использоваться в шаблонах функций
11. Дайте определение меташаблона
12. Перечислите последовательные контейнеры в библиотеке STL
13. Перечислите ассоциативные контейнеры в библиотеке STL
14. Перечислите псевдо-контейнеры в библиотеке STL
15. Перечислите основные алгоритмы в библиотеке STL

Уметь:
1. Охарактеризуйте преимущества объектно-ориентированного программирования перед процедурным стилем
2. Дайте характеристику статусам доступа к членам класса.
3. Охарактеризуйте назначение классов.
4. Охарактеризуйте назначение конструкторов в классе
5. Охарактеризуйте назначение деструкторов в классе
6. Проанализируйте непредвиденные ситуации, которые могут возникать в процессе работы программы.
7. Дайте характеристику исключения IndexOutOfRangeException
8. Проанализируйте преимущества использования шаблонов
9. Проанализируйте недостатки использования шаблонов
10. Проанализируйте достоинства STL подхода
11. Проанализируйте недостатки STL подхода
12. Дайте характеристику типам итераторов в STL
13. Проанализируйте, на основе какого механизма строится STL
14. Охарактеризуйте основные группы STL-алгоритмов
15. Проанализируйте механизм наследования шаблонов классов

Владеть:
1. В приведенном примере класса MobilePhone, определите, какие методы являются конструкторами:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
2. В приведенном примере класса MobilePhone, определите, какие методы являются деструкторами:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
3. В приведенном примере класса MobilePhone, определите, какие методы реализуют перегрузку операции:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
4. В приведенном примере класса MobilePhone, определите, какие методы являются дружественными функциями:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
```

```
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
5. В приведенном примере класса MobilePhone, определите, значение какого типа является результатом геттера getOwnNumber:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
6. В приведенном примере класса MobilePhone, определите, значение какого типа является результатом метода getAllKnownNetworks:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
7. Для приведенного примера класса MobilePhone, запишите, как будет выглядеть объявление метода перегрузки операции сложения (+):
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
8. Для приведенного примера класса MobilePhone, запишите, как будет выглядеть вызов перегруженной операции:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```

9. Для приведенного примера класса MobilePhone, запишите, как будет выглядеть вызов дружественной функции:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
11. Для приведенного примера класса MobilePhone, определите, какое действие будет выполнять дружественная функция:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
12. Для приведенного примера класса MobilePhone напишите объявление конструктора без параметров:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
13. Для приведенного примера класса MobilePhone для объекта класса с именем phone1 напишите, как будет производиться вызов метода placeCall:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
14. Для приведенного примера класса MobilePhone для объекта класса с именем phone1 напишите, как будет записано создание объекта phone2 как копии phone1:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
```

```
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```
15. Для приведенного примера класса MobilePhone объекта класса с именем phone1 напишите, как будет производиться вызов метода getAllKnownNetworks:
```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
MobilePhone (const MobilePhone &);
const MobilePhone & operator=(const MobilePhone &);
int getOwnNumber() const;
int popLastCall();
void redial();
void placeCall(int number);
friend ostream& operator<<(ostream&, const MobilePhone&);
~MobilePhone ();
static Network* getAllKnownNetworks();
};
```

6.5. Примерная тематика курсовых работ (проектов)

Учебным планом не предусмотрено

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

Методические рекомендации по работе с конспектом лекций

Просмотрите конспект сразу после занятий. Пометьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.
Работа с рекомендованной литературой:
При работе с основной и дополнительной литературой целесообразно придерживаться такой последовательности. Сначала прочитать весь заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом. Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать. План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала. Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов: - план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения, - текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника, - свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом, - тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу. В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации по подготовке к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение практического занятия предполагает, например: ¯ индивидуальные выступления студентов с сообщениями по какому-либо вопросу изучаемой темы; ¯ фронтальное обсуждение рассматриваемой проблемы, обобщения и выводы; ¯ решение задач и упражнений по образцу; ¯ решение вариантных задач и упражнений; решение ситуационных производственных (профессиональных) задач; ¯ проектирование и моделирование разных видов и компонентов профессиональной деятельности. выполнение контрольных работ; работу с тестами. При подготовке к практическим занятиям студентам рекомендуется: внимательно ознакомиться с тематикой практического занятия; прочесть конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; проверить свои знания, отвечая на вопросы для самопроверки; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в тетради. Все письменные задания выполнять в рабочей тетради. Практические занятия развивают у студентов навыки

самостоятельной работы по решению конкретных задач.

Методические рекомендации по подготовке к лабораторным работам

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в изучаемой дисциплине. Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение лабораторной работы предполагает: – изучение теоретического материала по теме лабораторной работы (по вопросам изучаемой темы); – выполнение необходимых расчетов и экспериментов; – оформление отчета с заполнением необходимых таблиц, построением графиков, подготовкой выводов по проделанным экспериментам и теоретическим расчетам; – по каждой лабораторной работе проводится контроль: проверяется содержание отчета, проверяется усвоение теоретического материала. Контроль усвоения теоретического материала является индивидуальным.

Методические указания по выполнению отчёта к лабораторным работам

Основным требованием по выполнению лабораторных и практических работ является полное исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения и профессиональной подготовки студентов.
Методические указания обеспечивают комплексный подход в учебной работе студентов, единство и преемственность требований к оформлению результатов работы на разных этапах обучения. С единых позиций приведены основные требования по структуре, оформлению и содержанию отчета по лабораторным и практическим работам.
Структура отчёта:
- цель работы;
- краткие теоретические сведения;
- ход выполнения работы;
- выводы.
Дополнительными элементы:
- приложения;
- библиографический список.
Требования к содержанию отчёта:
1. Титульный лист
В верхнем поле листа указывают полное наименование учебного заведения.
В среднем поле указывается вид работы, в данном случае лабораторная или практическая работа с указанием курса, по которому она выполнена, и ниже ее название. Название работы приводится без слова тема и в кавычки не заключается.
Далее ближе к правому краю титульного листа указывают фамилию, инициалы и группу учащегося, выполнившего работу, а также фамилию, инициалы преподавателя, принявшего работу.
В нижнем поле листа указывается место выполнения работы и год ее написания (без слова год).
2. Цель работы должна отражать тему работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до 0,5 страницы.
3. Краткие теоретические сведения. В этом разделе излагается краткое теоретическое описание изучаемой в работе темы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для дальнейшей обработки полученных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.
4. Ход выполнения работы. В данном разделе подробно излагается методика выполнения работы, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.
5. Выводы по работе - кратко излагаются результаты работы, полученные в результате выполнения работы, а также краткий анализ полученных результатов.
Отчет по лабораторной работе оформляется на листе формата А4. Допускается оформление отчета по лабораторной работе в электронном виде средствами Microsoft Office. Текст работы должен быть напечатан через полтора интервала шрифтом Times New Roman, кегль – 12. Поля должны оставаться по всем четырем сторонам печатного листа: левое – не менее 30 мм, правое – не менее 10, нижнее – не менее 20 и верхнее – не 15 мм.
Для защиты лабораторной работы студент должен подготовить отчет, провести самостоятельную работу, иметь отметку о проверенном отчете.
Результаты определяются по пятибалльной системе оценок.

Методические рекомендации по выполнению реферата

Реферат – письменная работа объемом 8–10 страниц. Это краткое и точное изложение сущности какого-либо вопроса, темы. Тему реферата студент выбирает из предложенных преподавателем или может предложить свой вариант. В реферате нужны развернутые аргументы, рассуждения, сравнения. Содержание темы излагается объективно от имени автора. Функции реферата. Информативная, поисковая, справочная, сигнальная, коммуникативная. Степень выполнения этих функций зависит от содержательных и формальных качеств реферата и для каких целей их использует. Требования к языку реферата. Должен отличаться точностью, краткостью, ясностью и простотой.
Структура реферата:

1. Титульный лист
2. Оглавление (на отдельной странице). Указываются названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. Введение. Аргументируется актуальность исследования, т.е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками, перечисляются положения, которые должны быть обоснованы. Обязательно формулируются цель и задачи реферата.
4. Основная часть. Подчиняется собственному плану, что отражается в разделении текста на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала. В случае если используется чья-либо неординарная мысль, идея, то обязательно нужно сделать ссылку на того автора, у кого взят данный материал.
5. Заключение. Последняя часть научного текста. В краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования.
6. Приложение. Может включать графики, таблицы, расчеты.
7. Библиография (список литературы). Указывается реально использованная для написания реферата литература. Названия книг располагаются по алфавиту с указанием их выходных данных. Общие требования к построению, содержанию и оформлению».
При проверке реферата оцениваются:
 знание фактического материала, усвоение общих представлений, понятий, идей;
 характеристика реализации цели и задач исследования;
 степень обоснованности аргументов и обобщений;
 качество и ценность полученных результатов;
 использование литературных источников;
 культура письменного изложения материала;
 культура оформления материалов работы.
Правила написания научных текстов (реферат, дипломная работа):
Здесь приводятся рекомендации по консультированию студентов относительно данного вида самостоятельной работы. Во время консультаций руководителю следует предложить к обсуждению следующие вопросы.
• Какова истинная цель Вашего научного текста – это поможет Вам разумно распределить свои силы и время.
• Важно разобраться, кто будет «читателем» Вашей работы.
• Начинать писать серьезную работу следует не раньше, чем возникнет ощущение, что по работе с источниками появились идеи, которыми можно поделиться.
• Должна быть идея, а для этого нужно научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного).
• Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно, а также стремясь структурировать свой текст.
• Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Методические рекомендации по выполнению контрольных работ

Контрольная работа выполняется по вариантам. На бланке указывается факультет, курс, группа, ФИО студента. Вопросы строятся на основе тестовых и ситуативных заданий. В тестовых заданиях, выбирается правильный(ые) ответ(ы). При решении ситуативных заданий выбирается правильная последовательность действий в рассматриваемой ситуации. Проверка контрольной работы позволяет выявить и исправить допущенные студентами ошибки, указать, какие вопросы дисциплины ими недостаточно усвоены и требуют доработки. Студент должен внимательно ознакомиться с письменными замечаниями преподавателя и приступить к их исправлению, для чего еще раз повторить соответствующий материал.

Методические рекомендации по подготовке к коллоквиуму

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:
- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.
Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы. Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на

самостоятельную подготовку к коллоквиуму студенту отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.
Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

Методические рекомендации по устному опросу/самоподготовке

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных сигналов, воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств. В случае необходимости следует рекомендовать еще раз внимательно разобраться в материале. Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение решать задачи или пройти тестирование по пройденному материалу. Однако преподавателю следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Методические рекомендации по подготовке к семинарским занятиям

Одним из видов внеаудиторной самостоятельной работы является подготовка к семинарским занятиям. Семинар – форма учебно-практических занятий, при которой студенты обсуждают сообщения, доклады и рефераты, выполненные ими по результатам учебных или научных исследований под руководством преподавателя. Преподаватель в этом случае является координатором обсуждений темы семинара, подготовка к которому является обязательной. Поэтому тема семинара и основные источники обсуждения предъявляются до обсуждения для детального ознакомления, изучения. Цели обсуждений направлены на формирование навыков профессиональной полемики и закрепление обсуждаемого материала. Семинар – это такая форма организации обучения, при которой на этапе подготовки доминирует самостоятельная работа учащихся с учебной литературой и другими дидактическими средствами над серией вопросов, проблем и задач, а в процессе семинара идут активное обсуждение, дискуссии и выступления учащихся, где они под руководством преподавателя делают обобщающие выводы и заключения. Семинар предназначен для углубленного изучения дисциплины, овладения методологией научного познания, то главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли.

Методические рекомендации по подготовке к эссе

Одним из видов самостоятельной работы студентов является написание творческой работы по заданной либо согласованной с преподавателем теме. Творческая работа (эссе) представляет собой оригинальное произведение объемом 500-700 слов, посвященное какой-либо значимой классической либо современной проблеме в определенной теоретической и практической области. Творческая работа не является рефератом и не должна носить описательный характер, большое место в ней должно быть уделено аргументированному представлению своей точки зрения студентами, критической оценке рассматриваемого материала и проблематики, что должно способствовать раскрытию творческих и аналитических способностей. Цели написания эссе – научиться логически верно и аргументировано строить устную и письменную речь; работать над углублением и систематизацией своих философских знаний; овладеть способностью использовать основы знаний для формирования мировоззренческой позиции. Приступая к написанию эссе, изложите в одном предложении, что именно вы будете утверждать и доказывать (свой тезис).

Методические рекомендации по подготовке к докладу

Для подготовки доклада необходимо выбрать актуальную тему. Желательно, чтобы тема была интересна докладчику и вызывала желание качественно подготовить материалы. Подготовка доклада предполагает: определение цели доклада; подбор необходимого материала, определяющего содержание доклада; составление плана доклада, распределение собранного материала в необходимой логической последовательности.
Композиция доклада имеет вступление, основную часть и заключение.
Вступление должно содержать: название доклада; сообщение основной идеи; современную оценку предмета изложения; краткое перечисление рассматриваемых вопросов; интересную для слушателей форму изложения. Основная часть, в которой необходимо раскрыть суть темы, обычно строится по принципу отчёта. Задача основной части: представить достаточно данных для того, чтобы слушатели заинтересовались темой.
Заключение – чёткое обобщение и краткие выводы по излагаемой теме.

Методические рекомендации по подготовке к собеседованию

Собеседование – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.
Цель собеседования: проверка усвоения знаний; умений применять знания; сформированности профессионально значимых личностных качеств.
Подготовка к собеседованию предполагает повторение пройденного материала и приобретение навыка свободного владения терминологией и фактическими данными по определенному разделу дисциплины.

Методические рекомендации по подготовке к тестированию/ итоговому тестированию

Тестирование/итоговое тестирование – это система стандартизированных заданий, определенного содержания и упорядоченных в рамках определенной стратегии предъявления, позволяющая качественно оценить структуру и эффективно измерить уровень знаний, умений и навыков по учебной дисциплине (разделу). Тестирование позволяет оценить у студентов уровень знаний фактического материала, умения логически мыслить, способности к рефлексии и творческому подходу к решению поставленной задачи на каждом этапе их обучения.
При самостоятельной подготовке к тестированию студенту необходимо:
- четко выяснить все условия тестирования заранее (сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов);
- проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- проработать информационный материал по разделу дисциплине;
- конспективно выписать из соответствующих разделов учебной дисциплины основные понятия, ключевые идеи, теоретические концепции и подходы;
- повторить теоретический материал лекций, самостоятельно изученной учебной и научной литературы, семинарских занятий.
При выполнении тестовых заданий рекомендуется:
- внимательно полностью прочитать вопрос и предлагаемые варианты ответов, выбрать правильные из них (их может быть несколько);
- при затруднении при ответе на сложный для студента вопрос целесообразно переходить к другим вопросам теста. По завершению ответов на все последующие вопросы теста необходимо вернуться к трудному вопросу и дать на него продуманный ответ;
- по окончании тестирования следует проверить правильность данных ответов на вопросы теста, чтобы избежать механических ошибок.

Методические рекомендации по подготовке к экзамену

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине. Экзаменационная сессия – это серия экзаменов, установленных учебным планом. Между экзаменами интервал 2-4 дня, в течение студент систематизирует уже имеющиеся знания. На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки вопросы. Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

Методические рекомендации по подготовке к зачету/ зачету с оценкой

В ходе подготовки к зачету/зачету с оценкой студент, в первую очередь, должен систематизировать знания, полученные в ходе изучения дисциплины. К зачету/зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине. В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:
- программой дисциплины;
- перечнем знаний, умений и навыков, которыми студент должен владеть;
- тематическими планами лекций, семинарских занятий;
- учебниками, учебными пособиями по дисциплине, а также электронными ресурсами;
- перечнем вопросов к зачету/зачету с оценкой.
После этого у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине. Систематическое выполнение учебной работы на лекциях и лабораторных занятиях позволит успешно освоить дисциплину и создать хорошую базу для сдачи зачета/зачета с оценкой.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература

7.1.1. Основная литература	
Л.1.1	Барков И. А. Объектно-ориентированное программирование [Электронный ресурс]:учебник для вузов. - Санкт-Петербург: Лань, 2023. - 700 с. – Режим доступа: https://e.lanbook.com/book/329549
Л.1.2	Степанов П. П., Кабанов А. А., Никонов В. А., Павлюченко Т. С., Обухова К. В. Объектно-ориентированное программирование [Электронный ресурс]:учебное пособие. - Омск: Омский государственный технический университет (ОмГТУ), 2021. - 112 с. – Режим доступа: https://biblioclub.ru/index.php?page=book&id=700657
Л.1.3	Скворцова Л. А., Бирюкова А. А., Гусев К. В. Объектно-ориентированное программирование на языке С++: Практикум [Электронный ресурс]:. - Москва: РТУ МИРЭА, 2021. - 146 с. – Режим доступа: https://e.lanbook.com/book/176540
7.1.2. Дополнительная литература	
Л.2.1	Быкадорова Е. А. Программирование. Практикум [Электронный ресурс]:. - Санкт-Петербург: Лань, 2020. - 60 с. – Режим доступа: https://e.lanbook.com/book/148278
7.2. Лицензионное и свободно распространяемое программное обеспечение в том числе отечественного производства	
7.2.1	Microsoft Windows 10
7.2.2	Microsoft Office 2013 Standard
7.2.3	Kaspersky Endpoint Security
7.2.4	Microsoft Visual Studio Professional 2017
7.3. Перечень профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет	
7.3.1	Электронно-библиотечная система "Лань". Режим доступа: https://e.lanbook.com/
7.3.2	Электронно-библиотечная система "Университетская библиотека онлайн". Режим доступа: https://biblioclub.ru/
7.3.3	Электронно-библиотечная система "BOOK.ru". Режим доступа: https://book.ru/
7.3.4	Единое окно доступа к образовательным ресурсам. Режим доступа: http://window.edu.ru/
7.3.5	ПЛАТФОРМА ОНЛАЙН-ОБРАЗОВАНИЯ «РАЗУМ». Режим доступа: https://razoom.mgutm.ru/
7.3.6	Электронно-библиотечная система "Юрайт". Режим доступа: https://biblio-online.ru/
7.3.7	"Электронная библиотека учебников" . Режим доступа: http://studentam.net/

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-28 - Компьютерный класс : 15 персональных компьютеров, объединённых в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийная установка «Toshiba» с экраном 3,5 м. Лицензионные программы:. ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01,. WinDjView, КОМПАС-3D LT, 1С Предприятие 8.1, Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7 ─ Эксперт Системс» (сетевая версия). CorelDRAW Graphics Suite X7. GRAPHISOFT ArchiCAD 15, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, QBacic, Pascal, Delphi, Kaspersky Endpoint Security. IQlib. Все компьютеры подключены к сети Internet

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Организация образовательного процесса для лиц с ограниченными возможностями осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн. 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. Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей. Предусмотрена возможность обучения по индивидуальному графику, при составлении которого возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доц. Мурат Е.П._____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2025 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич_____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2025 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доц. Мурат Е.П._____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич_____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доц. Мурат Е.П._____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2027 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич_____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2027 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доц. Мурат Е.П._____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2028 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич_____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2028 г. № __
Зав. кафедрой Нечушкин Александр Юрьевич _____