

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Арзамасский филиал ННГУ - Факультет естественных и математических наук

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 15 от 24.12.2025 г.

Рабочая программа дисциплины
Объектно-ориентированное программирование

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
09.03.03 - Прикладная информатика

Направленность образовательной программы
Прикладная информатика в экономике

Форма обучения
очно-заочная

г. Арзамас

2026 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.05 Объектно-ориентированное программирование относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-2: Способен осуществлять проектирование программного обеспечения ИС и разрабатывать техническую документацию на его компоненты	<p>ПК-2.1: Демонстрирует знание современных языков и систем программирования, технологий проектирования программного обеспечения</p> <p>ПК-2.2: Демонстрирует умение сформулировать требования к разрабатываемому программному обеспечению, выполнить его реализацию и оформить техническую документацию на его компоненты</p> <p>ПК-2.3: Имеет практический опыт проектирования программного обеспечения конкретной ИС и разработки технической документации на ее компоненты</p>	<p>ПК-2.1: Знать основные принципы и технологии разработки программного обеспечения, методы и средства сборки модулей и компонентов программного обеспечения; разработки процедур для развертывания программного обеспечения, методы и средства миграции и преобразования данных, методы создания пользовательских интерфейсов</p> <p>ПК-2.2: Уметь разрабатывать программный код на языках программирования высокого уровня, осуществлять отладку программ, оформлять техническую документацию; использовать выбранную среду программирования для разработки процедур интеграции программных модулей, проводить оценку работоспособности программного обеспечения</p> <p>ПК-2.3: Иметь практический опыт</p>	<p>Практическое задание</p> <p>Собеседование</p> <p>Тест</p>	<p>Зачёт: Контрольные вопросы</p> <p>Экзамен: Контрольные вопросы</p>

		<p>разработки исходного кода, тестирования программного обеспечения, сборки модулей и компонентов программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов</p>		
<p>ПК-8: Способен разрабатывать лингвистическое, информационное и программное обеспечение ИС (ИИС) и сопровождающую его документацию</p>	<p>ПК-8.1: Демонстрирует знание современных языков и систем программирования, формализмов описания знаний на концептуальном и инфологическом уровнях, требований к технической документации на все виды обеспечения ИС (ИИС)</p> <p>ПК-8.2: Применяет современные языки и системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях при разработке лингвистического, информационного и программного обеспечения ИИС и сопровождающей ее документации</p> <p>ПК-8.3: Имеет практический опыт разработки лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей ее документации</p>	<p>ПК-8.1: Знать современное состояние и принципиальные возможности языков программирования и использующих его систем программирования; Уметь устанавливать, тестировать, испытывать и использовать программные средства, ставить задачи и разрабатывать алгоритм их решения, используя программное обеспечение</p> <p>Владеть навыками разработки и отладки программ, основными шаблонами проектирования программных систем с использованием технологии программирования.</p> <p>ПК-8.2: Знать возможности программного обеспечения для проведения анализа социально-экономических задач и процессов с применением методов системного анализа и математического моделирования. Уметь разрабатывать основные программные документы; работать с современными системами программирования, включая объектно-ориентированные. Владеть приемами разработки прикладных программ на различных языках.</p>	<p>Собеседование Практическое задание Тест</p>	<p>Зачёт: Контрольные вопросы</p> <p>Экзамен: Контрольные вопросы</p>

		<p>ПК-8.3: Знать особенности осуществления разработки лингвистического, информационного и программного обеспечения конкретной ИС Уметь разрабатывать программное обеспечение ИС и сопровождающую его документацию Владеть способностью осуществлять разработку лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей его документацию.</p>		
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очно-заочная
Общая трудоемкость, з.е.	8
Часов по учебному плану	288
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	34
- занятия семинарского типа (практические занятия / лабораторные работы)	34
- КСР	3
самостоятельная работа	181
Промежуточная аттестация	36 Экзамен, Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабора	Всего	

	о з ф о	о з ф о	торные работы), часы		о з ф о
			о з ф о	о з ф о	
Тема 1. Объектно-ориентированный подход в программировании	13	2	2	4	9
Тема 2. Классы и объекты. Отношения.	13	2	2	4	9
Тема 3. Языки ООП и их классификация	13	2	2	4	9
Тема 4. Наследование и полиморфизм	13	2	2	4	9
Тема 5. Основы программирования на языке ООП	13	2	2	4	9
Тема 6. Методы и классы .	12	2	2	4	8
Тема 7. Операции и операторы	13	2	2	4	9
Тема 8. Массивы.	12	2	1	3	9
Тема 9. Коллекции и словари	10	1	1	2	8
Тема 10. Язык запросов LINQ	12	1	2	3	9
Тема 11. Классы и их основные элементы	12	2	1	3	9
Тема 12. События и данные в классе	12	1	2	3	9
Тема 13. Отношения между классами	12	2	1	3	9
Тема 14. Структуры и перечисления	12	1	2	3	9
Тема 15. Разработка консольных приложений	12	2	1	3	9
Тема 16. Формы и работа с ними	11	1	2	3	8
Тема 17. Управление формами	10	1	1	2	8
Тема 18. Диалоговые окна	11	1	2	3	8
Тема 19. Формы и графика	11	2	1	3	8
Тема 20. Файлы. Классы для работы с файлами	10	1	1	2	8
Тема 21. Классы и базы данных	12	2	2	4	8
Аттестация	36				
КСР	3			3	
Итого	288	34	34	71	181

Содержание разделов и тем дисциплины

Тема 1. Объектно-ориентированный подход в программировании

Введение в объектно-ориентированное программирование. Понятие объекта и класса в ООП.

Инкапсуляция, наследование и полиморфизм в ООП.

Тема 2. Классы и объекты. Отношения.

Определение классов и объектов в ООП: синтаксис, структура и особенности. Инкапсуляция данных: понятие видимости и модификаторов доступа (private, protected, public). Полиморфизм: виды (параметрический, ad-hoc, конструкционный), реализация и использование в программах.

Наследование и его виды: одиночное, множественное, виртуальное и абстрактное. Интерфейсы и их реализация в программировании. Использование конструкторов и деструкторов в классах. Статические члены класса и их особенности. Перегрузка операторов и её особенности. Определение отношений в ООП, их виды и характеристики. Реализация отношений наследования, агрегации и композиции в классах.

Особенности виртуальных и абстрактных отношений в наследовании. Применение отношений для построения иерархий классов и принципов композиции.

Тема 3. Языки ООП и их классификация

Классификация языков программирования по степени поддержки ООП: полностью объектно-

ориентированные, частично объектно-ориентированные и не объектно-ориентированные. Примеры языков программирования каждого из классов: Java, C#, Python, PHP, JavaScript, Swift, Objective-C, Delphi и другие.

Основные особенности и преимущества каждого из классов языков программирования. Рекомендации по выбору языка программирования в зависимости от задачи и предпочтений разработчика.

Тема 4. Наследование и полиморфизм

Наследование в ООП: определение, основные понятия и принципы работы. Виды наследования: простое, множественное и гибридное. Полиморфизм в ООП: определение и основные концепции.

Реализация полиморфизма в разных языках программирования.

Абстрактные классы и интерфейсы: понятие, использование и взаимодействие. Проблемы наследования и способы их решения. Ковариантность и контрвариантность в наследовании: понятия и применение.

Тема 5. Основы программирования на языке ООП

Основные принципы объектно-ориентированного программирования, включая такие понятия, как переменные, типы данных, ветвление, циклы и массивы.

Классы и объекты, инкапсуляция, наследование и полиморфизм. Алгоритмы и структуры данных.

Паттерны проектирования.

Тема 6. Методы и классы.

Понятия класса, объекта, метода и свойства в ООП. Инкапсуляция: сокрытие данных и обеспечение доступа к ним через методы. Полиморфизм: использование одного интерфейса для выполнения

различных задач. Наследование: создание новых классов на основе существующих. Абстрактные классы и методы: предоставление общей структуры для подклассов без определения конкретных

деталей. Виртуальные методы: обеспечение динамического связывания во время выполнения

программы. Интерфейсы: определение общих наборов методов, которые должны быть реализованы

классами. Перегрузка методов: использование нескольких методов с одним именем и разными

параметрами. Конструкторы и деструкторы: специальные методы, которые выполняются при создании и уничтожении объектов. Статические методы и поля: доступ к которым можно получить без создания

объекта класса.

Тема 7. Операции и операторы

Оператор создания экземпляра класса (new). Используется для создания новых объектов в программе.

Операторы доступа к свойствам и методам класса (точка и стрелки). Позволяют получить доступ к свойствам и вызвать методы класса для конкретного объекта. Оператор присваивания (знак равно).

Используется для присвоения значений переменным и объектам. Бинарные операторы (сложение, умножение, сравнение и т.д.). Выполняют различные математические и логические операции над объектами.

Тернарный оператор (? :). Применяется для условного выполнения операторов в зависимости от значения некоторого условия. Унарные операторы (-, !, ~). Меняют знак числа или логического значения, инвертируют биты числа или логического значения.

Составные операторы ((), {}). Используются для группировки выражений и указания порядка выполнения операций. Приоритеты операторов. Определяют порядок выполнения операций в выражении. Операция sizeof. Возвращает размер указанного типа или переменной в байтах.

Тема 8. Массивы.

Введение в массивы в C#

Объявление и инициализация массивов в C#

Работа с многомерными массивами в C#

Использование циклов для обработки массивов в C#

Применение LINQ для работы с массивами в C#

Передача массивов в качестве аргументов функций в C#

Возврат массивов из функций в C#

Сортировка массивов в C# с помощью стандартных алгоритмов

Создание и использование многопоточных приложений для работы с массивами в C#

Безопасный доступ к элементам массивов с использованием индексов в C#
Обработка исключений при работе с массивами в C#
Сериализация и десериализация массивов в .NET Framework в C#
Организация динамического выделения памяти для массивов в C#
Вложенные и анонимные массивы в C#
Рекурсивные алгоритмы работы с массивами на C#
Тема 9. Коллекции и словари
Введение в коллекции и словари в C#
Объявление и инициализация коллекций и словарей в C#
Работа с многопоточными коллекциями и словарями в C#
Использование циклов и LINQ для обработки коллекций и словарей в C#
Передача коллекций и словарей в качестве аргументов функций в C#
Возврат коллекций и словарей из функций в C#
Сортировка коллекций и словарей с помощью стандартных алгоритмов в C#
Безопасный доступ к элементам коллекций и словарей с использованием индексов и итераторов в C#
Обработка исключений при работе с коллекциями и словарями в C#
Сериализация и десериализация коллекций и словарей .NET Framework в C#
Организация динамического выделения памяти для коллекций и словарей в C#
Вложенные и анонимные коллекции и словари в C#
Рекурсивные алгоритмы работы с коллекциями и словарями на C#
Тема 10. Язык запросов LINQ
Введение в язык запросов LINQ.
Основные принципы работы языка LINQ.
Работа с данными в LINQ: методы расширения и стандартные операции.
Работа с коллекциями в LINQ: преобразование, фильтрация и сортировка.
Работа с наборами данных в LINQ: объединение, пересечение и разность.
Применение LINQ to Objects для работы с объектами в памяти.
Применение LINQ to SQL для работы с базами данных.
Использование LINQ to XML для работы с XML-документами.
Применение LINQ to Entity Framework для работы с объектно-ориентированными моделями данных.
Использование асинхронного LINQ (Async/Await) для асинхронной обработки данных.
Обработка ошибок и исключений в LINQ.
LINQ и обобщенное программирование (Generics).
LINQ как средство для создания эффективных и гибких приложений.
Тема 11. Классы и их основные элементы
Понятие класса в C#.
Создание класса.
Члены класса.
Статические члены класса.
Свойства класса.
Инкапсуляция в классах.
Конструкторы класса.
Деструкторы класса.
Методы класса.
Наследование классов.
Виртуальные и абстрактные классы.
Полиморфизм в C#.
Индексаторы и делегаты в классах.
Перегрузка операторов в классах.
Тема 12. События и данные в классе

Общие сведения о событиях в C#.

Обработка событий в классах.

Применение событий для коммуникации между объектами.

Классы данных в C#.

Структура классов данных.

Реализация интерфейсов в классах данных.

Использование модификаторов доступа для защиты данных классов.

Применение конструкторов и деструкторов для работы с данными.

Перегрузка операций в классах данных.

Приведение типов данных в классах.

Тема 13. Отношения между классами

Основы отношений между классами в C#.

Понятие ассоциации и агрегации в объектно-ориентированном проектировании.

Отношение композиции в C#: особенности и применение.

Отношение наследования в C#: базовые и производные классы.

Отношение агрегации в C#: составные и агрегированные объекты.

Отношение зависимости между классами: причины и следствия.

Управление зависимостями между классами с помощью внедрения зависимостей.

Тема 14. Структуры и перечисления

Обзор структур в C#.

Использование структур для представления данных.

Структуры как альтернатива классам.

Особенности работы со структурами.

Перечисления в C#: определение и использование.

Базовые типы перечислений.

Модификаторы доступа для перечислений.

Тема 15. Разработка консольных приложений

Консольные приложения в C#: основы создания и работы.

Структура консольного приложения.

Ввод и вывод данных в консольных приложениях.

Обработка исключений в консольных приложениях.

Использование операторов ветвления и циклов в консольных приложениях.

Работа с файлами в консольных приложениях: чтение, запись и открытие файлов.

Многопоточность в консольных приложениях C#.

Установка пользовательских атрибутов и работа с ними в консольных приложениях.

Тема 16. Формы и работа с ними

Создание и работа с формами в C#.

Элементы управления формой и их свойства.

События элементов управления формой.

Обработка событий формы.

Диалоговые окна в C# формах.

Валидация данных в формах.

Работа с графикой в формах C#.

Многопоточное программирование в формах C#.

Тема 17. Управление формами

Основные элементы управления формой в C#.

Работа со свойствами элементов управления формой.

Обработка событий элементов управления формой.

Создание диалоговых окон в формах C#.

Валидация данных на форме.

Многопоточная работа с формами C#.

Графические элементы управления в формах C#.

Тема 18. Диалоговые окна

Что такое диалоговые окна и зачем они нужны.

Какие бывают виды диалоговых окон.

Как создавать диалоговые окна в различных программах.

Как управлять элементами диалогового окна.

Как обрабатывать события, связанные с диалоговыми окнами.

Как использовать диалоги для ввода и вывода данных.

Как работать с графическими элементами в диалоговых окнах.

Как создавать многооконные приложения с использованием диалогов.

Как тестировать и отлаживать диалоговые окна.

Как оптимизировать работу с диалоговыми окнами для повышения производительности приложений.

Тема 19. Формы и графика

Введение в формы и графику в C#.

Основы создания форм в C#.

Элементы управления формами и их свойства.

События элементов управления формами.

Обработка событий форм.

Графические элементы управления и работа с ними.

Обработка графики на формах.

Многопоточное программирование с формами.

Валидация данных на формах.

Создание диалоговых окон на формах.

Тема 20. Файлы. Классы для работы с файлами

Класс File - основные методы для работы с файлами.

Класс DirectoryInfo - работа с каталогами файловой системы.

Класс Path - работа с путями к файлам и каталогам.

Класс StreamReader - чтение текстовых файлов.

Класс StreamWriter - запись в текстовые файлы.

Класс BinaryWriter - запись бинарных данных в файлы.

Класс BinaryReader - чтение бинарных данных из файлов.

Классы FileStream, MemoryStream - работа с потоками данных.

Класс FileInfo - информация о файлах.

Работа с ошибками при чтении и записи файлов.

Тема 21. Классы и базы данных

Подключение к базе данных с использованием классов.

Управление подключениями с использованием классов.

Создание объектов классов для работы с базой данных.

Использование методов и свойств классов для выполнения запросов к базе данных.

Обработка ошибок, возникающих при работе с классами и базой данных.

Создание и использование методов внутри классов для работы с базой данных.

Реализация паттерна проектирования Repository с использованием классов и базы данных.

Использование наследования и полиморфизма в классах для работы с различными базами данных.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:
Электронные курсы, созданные в системе электронного обучения ННГУ:

Объектно-ориентированное программирование, <https://e-learning.unn.ru/course/view.php?id=7975>.

Иные учебно-методические материалы:

Учебно-методические документы, регламентирующие самостоятельную работу, адреса доступа к документам:

<https://arz.unn.ru/sveden/document/>

https://arz.unn.ru/pdf/Metod_all_all.pdf

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-2:

1. Разработка консольного приложения.
2. Работа с переменными и константами простых типов.
3. Инициализация и реализация одномерных массивов.
4. Вычисление агрегированных показателей на одномерных массивах.
5. Управление потоком выполнения в программе с использованием условного оператора, оператора выбора, цикла for, цикла while и цикла do...while.
6. Создание иерархии классов для работы с каталогами файловой системы и файлами.
7. Перегрузка методов класса.
8. Наследование реализации и создание иерархии классов.
9. Обработка данных в файлах с использованием потоков ввода-вывода.
10. Разработка классов для работы с каталогами файловой системы и файлами.

5.1.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-8:

Задание 1.

Составить на любом языке программирования консольное приложение, которое содержит описание класса **Time** (время), который должен содержать:

Класс должен включать:

1. Закрытые свойства для хранения часов и минут
2. Методы доступа к закрытым свойствам
3. Конструктор или несколько конструкторов, для создания экземпляров класса
4. Метод отображения на экране времени в формате (чч:мм)

Программа должна делать следующее:

1. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
2. Задать им следующие значения (2ч 30м, 5ч 15м, 3ч 45м)

3. Вывести на экран время, хранящееся во всех объектах.
4. Рассчитать разницу в днях между 1 и 2 объектами и вывести ее на экран.

Задание 2.

Составить на любом языке программирования консольную программу, которая содержит описание класса

Date - дата (год, месяц, день)

Класс должен включать:

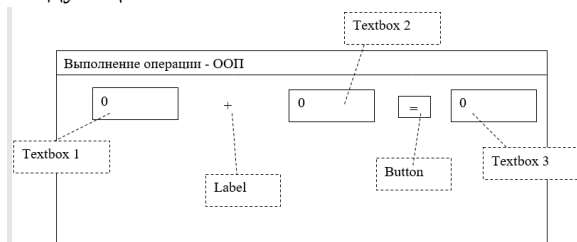
1. Закрытые свойства для хранения год, месяц, день.
2. Методы доступа к закрытым свойствам.
3. Конструктор или несколько конструкторов, для создания объектов класса.
4. Метод - показать на экране время в формате (дд/мм/гг)
5. Метод - рассчитать количество дней с начала года до даты

```
public int Days( )
```

1. Программа должна делать следующее:
2. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
3. Задать им следующие значения (1.5.2001 5.2.2002 13.7.2001)
4. Вывести на экран даты, хранящиеся во всех объектах.

1. Рассчитать разницу в днях между 1 и 3 объектами и вывести ее на экран

1. Написать программу, которая, используя объект класса производного от класса Form, выводит следующее окно:



0

При нажатии кнопки Button, текстовое поле Textbox3 должно показать результат выбранной в ListBox операции над числами, введенными в Textbox1 и Textbox2

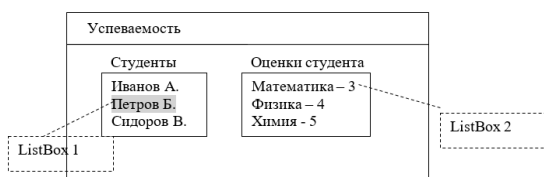
При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

2. Написать программу, которая, используя объект класса производного от класса Form, позволяет просматривать информацию об успеваемости студентов по годам:

ФИО студента	Математика	Физика	Химия
Иванов А.	4	5	4

Петров Б.	3	4	5
Сидоров В.	5	5	5

Форма окна программы:



При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
отлично	выставляется студенту, если представленная работа выполнена полностью без ошибок и недочетов;
хорошо	выставляется студенту, если представленная работа выполнена полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов;
удовлетворительно	выставляется студенту, если представленная им работа выполнена правильно не менее чем на 2/3 всей работы или в работе допущены не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и трех недочетов, при наличии четырех-пяти недочетов;
неудовлетворительно	выставляется студенту, если число ошибок и недочетов в работе превысило норму для оценки 3 или правильно выполнено менее 2/3 всей работы.

5.1.3 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-2:

1. Объясните, в чём заключаются основные концепции ООП.
2. Каковы преимущества ООП?
3. Что такое полиморфизм и как он поддерживается в выбранном вами языке кодирования?
4. Объясните различия между классами и объектами.
5. Опишите два различия между процедурным программированием и ООП.
6. Какие ограничения имеет ООП?

5.1.4 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ПК-8:

1. Понятие типа. Константы и переменные. Операции и формы их записи.
2. Сортировка одномерного массива методом обмена.
3. Логический тип данных. Диапазон значений логического типа данных. Логические операции.
4. Сортировка одномерного массива методом вставки.
5. Целочисленные типы данных. Прямой, обратный и дополнительный код целочисленных типов данных. Диапазоны значений целочисленных типов данных.
6. Сортировка одномерного массива методом выбора.
7. Целочисленные типы данных. Операции над целочисленными типами данных.
8. Задача о минимаксе.
9. Вещественные типы данных. Операции над вещественными типами данных.
10. Открытые массивы в языке .
11. Вещественные типы данных. Представление вещественных типов данных на примере модельного типа.
12. Многомерные статические массивы в языке .
13. Вещественные типы данных. Представление вещественных типов данных в памяти компьютера. Диапазоны значений вещественных типов данных.
14. Примеры задач на обработку массивов.
15. Символьный тип данных. Литералы символьного типа.

Критерии оценивания (оценочное средство - Собеседование)

Оценка	Критерии оценивания
отлично	Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности, грамотный научный язык; ответ самостоятельный.
хорошо	Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.
удовлетворительно	Ответ полный, но при этом допущена существенная ошибка или неполный, несвязный ответ.
неудовлетворительно	Ответ обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

5.1.5 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-2:

1. Что такое объектно-ориентированное программирование (ООП)?
 - а) Основано на реальных объектах, инкапсулирует данные и

- обеспечивает повторное использование кода.
- б) Основано на математических формулах и алгоритмах.
 - в) Основано на использовании функций и структур данных.
2. Что такое инкапсуляция?
- а) Объединение данных с методами, работающими с этими данными.
 - б) Создание абстрактных классов для группировки похожих методов.
 - в) Разделение кода на независимые блоки.
3. Что такое наследование?
- а) Создание нового класса на основе существующего.
 - б) Добавление новых методов в существующий класс.
 - в) Изменение существующих методов класса.
4. Что такое полиморфизм?
- а) Возможность иметь несколько форм одной сущности.
 - б) Использование разных типов данных в одном коде.
 - в) Создание нескольких экземпляров одного класса.
5. Что такое класс в C#?
- а) Набор инструкций для выполнения определённой задачи.
 - б) Абстрактный шаблон для создания объектов.
 - в) Структура данных для хранения информации.
6. Что такое метод в C#?
- а) Функция для выполнения определённой задачи.
 - б) Набор инструкций для изменения состояния объекта.
 - в) Способ доступа к данным объекта.
7. Что такое конструктор в C#?
- а) Функция для инициализации объекта.
 - б) Метод для добавления данных в объект.
 - в) Способ доступа к свойствам объекта.
8. Что такое деструктор в C#?
- а) Функция для уничтожения объекта.
 - б) Метод для удаления данных из объекта.
 - в) Способ доступа к методам объекта.
9. Что такое интерфейс в C#?
- а) Абстрактный класс без реализации.
 - б) Набор методов, доступных для всех классов.
 - в) Способ доступа к свойствам объекта.
10. Что такое абстрактный класс в C#?
- а) Класс, содержащий абстрактные методы.
 - б) Класс, который нельзя создать.
 - в) Класс, который можно использовать для создания объектов.

5.1.6 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-8:

1. Какую роль выполняет наследование?

- а) расширять поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе;
- б) расширять поведение производного класса (класса-потомка), наследуя основную функциональность в исходных файлах;
- в) обеспечивать взаимодействие класса-родителя и класса-потомка.

2. Объектно-ориентированное программирование - это:

- а) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.
- б) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является отдельно существующей структурой (функцией, переменной, процедурой).
- в) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых может быть приведен к общему виду (структуре) программы.

3. Классы — это:

- а) аналоги переменных и структур в процедурном программировании.
- б) это абстракция описывающая методы, свойства, ещё не существующих объектов.
- в) это абстракция описывающая методы, свойства существующих объектов.

4. Три основных принципа построения классов – это:

- а) инкапсуляция, полиморфизм, наследование.
- б) наследование, распределенность, структурируемость.
- в) полиморфизм, инкапсуляция, стандартизация.

5. Инкапсуляция – это:

- а) механизм объектно-ориентированного подхода, позволяющий расширить и/или изменить структуру уже существующего (родительского) класса, путём написания нового класса (потомка), который полностью наследует все свойства и методы и добавляет свои.
- б) это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.
- в) это свойство объектов скрывать некоторые свои данные и способы их обработки (методы) от окружающей его цифровой среды.

6. Три основных уровня инкапсуляции:

- а) open, closed и average.
- б) public, protected и private.
- в) public, not public и average.

7. Конструктор предназначен для ...

- а) инициализации данных класса при создании нового экземпляра.
- б) удаления объекта заданного класса.
- в) определения структуры класса.

8. Какое из описаний конструктора будет верным?

- а) ~Student (char *);
- б) Student::~Student ()
- в) Student (char *);

9. Какое из описаний деструктора будет верным?

- а) ~Student (char *);
- б) Student::~~Student ()
- в) Student (char *);

10. В заданной строке определяется имя класса и имя метода (функции) класса void Student::show (void), а именно:

- а) имя класса – Student, метода - show.
- б) имя класса – show, метода - void.
- в) имя класса – show, метода - Student.

11. В заданном классе определите методы и объекты класса

```
class Students {  
public:  
    void set_name(std::string);  
    std::string get_name();  
    void set_last_name(std::string);  
    std::string get_last_name();  
private:  
    std::string name;  
    std::string last_name;
```

```
};
```

- а) методы класса – name, last_name; объекты класса – set_name, get_name, set_last_name, get_last_name.
- б) методы класса – set_name, get_name, set_last_name, get_last_name; объекты класса – name, last_name.
- в) методы класса – void, string; объекты класса – private, public.

12. Можно ли в производном классе объявлять новые поля и методы?

- а) Нет;
- б) Да.

13. Обязательно ли делать поля класса приватными?

- а) Нет;
- б) Да.

14. Что такое виртуальная функция?

- а) пользовательская функция;
- б) функция, которая вызывает сама себя;
- в) функция базового класса, переопределенная в одном или в нескольких производных классах;
- г) функция дочернего класса, переопределенная в одном или в нескольких производных классах;

15. Полиморфизм – это...

- способность функции обрабатывать данные разных типов;
- способность функции вызывать методы раздела private любой другой функции программы;
- свойство базового класса;
- способ описания методов класса.

16. Метод это:

- Функция класса, получающая в качестве обязательного параметра указатель на объект
- Структура, хранящая указатели this, parent, supper
- Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
- Определенный программистом абстрактный тип данных

17. Объект это:

- Функция класса, получающая в качестве обязательного параметра указатель на объект
- Структура, хранящая указатели this, parent, supper
- Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
- Определенный программистом абстрактный тип данных

18. Для создания контекстного меню используется компонент класса:

1. TButton
2. TEdit
3. TCheckBox
4. TPopupMenu
5. TBitBtn

19. Для создания многострочного редактируемого текстового поля используется компонент класса:

- TLabel
- TMainMenu
- TMemo
- TEdit
- TForm

20. Для создания главного меню используется компонент класса:

1. TLabel
2. TMenu
3. TEdit
4. TMainMenu
5. TPopupMenu

21. Для создания однострочного редактируемого текстового поля используется компонент класса:

1. TEdit
2. TMemo
3. TLabel
4. TPopupMenu
5. TPanel

22. Свойство, отвечающее за высоту формы:

1. Top
2. Size
3. Style
4. Width
5. Height

23. Типы данных бывают:

1. Integer, real, char, for
2. Float, integer, if, string
3. Integer, float, char, string, boolean
4. Integer, real, printf, string
5. For, scanf, integer, float

24. Для создания строки состояния используется компонент класса:
1. TProgressBar
 2. TCoolBar
 3. TToolBar
 4. TStatusBar
 5. TTrackBar
25. Какое свойство компонента класса TEdit отвечает за текст находящийся в нём:
1. String
 2. Text
 3. Lines
 4. Caption
 5. Edit
26. Вкладка Events в Object Inspector предназначена для...
1. Изменения свойств компонентов
 2. Редактирования кода программы
 3. Поиска компонентов
 4. Просмотра структуры проекта
 5. Определения событий
27. Для выравнивания текста по правому или левому краю у компонента класса TEdit имеется свойство...
1. Align
 2. AutoSize
 3. Font
 4. Alignment
 5. Left
28. Для того чтобы сделать любой компонент невидимым предназначено свойство...
1. Visible
 2. Enable
 3. Style
 4. Transparent
 5. Glyph
29. Ограничение на количество вводимых символов для компонента Edit задаётся в свойстве...
1. Text
 2. Left
 3. Font
 4. AutoSize
 5. MaxLength
30. Когда нажата кнопка мыши, генерируется событие:
1. OnMouseMove
 2. OnMouseUp
 3. OnMouseDown
 4. OnMouseWheel
 5. OnClick
31. Событие OnCreate генерируется, когда...
1. Окно закрывается
 2. Окно создается
 3. Окно изменяет размеры
 4. Пользователь щелкнул по форме
 5. Окно меняет цвет
32. Вкладка Properties в Object Inspector предназначена для...
1. Редактирования кода программы

2. Поиска компонентов
3. Просмотра структуры проекта
4. Определения событий
5. Изменения свойств компонентов

33. Функция, выполняющая преобразование числа в строку имеет вид:

1. StrToInt(s)
2. FloatToStr(n)
3. Chr(n)
4. IntToStr(a,s)
5. StrToFloat(s)

34. Свойство формы, отвечающее за заголовок окна.

1. Caption
2. Text
3. Name
4. Font
5. String

35. Что будет выведено на экран при выполнении фрагмента кода...

```
int c;
```

```
int a=5, b=2;
```

```
c = a/b;
```

```
cin >> a << " / " << b << " = " << c << endl;
```

- 5 / 2 = 2
- 5 / 2 = 2,5
- 5 / 2 = 3
- 5, 2, 5 / 2
- 5, 2, 2,5

36. Что будет выведено на экран при выполнении фрагмента кода...

```
double c;
```

```
double a=5, b=2;
```

```
c = a/b;
```

```
cin >> a << " / " << b << " = " << c << endl;
```

- 5 / 2 = 2
- 5 / 2 = 2,5
- 5 / 2 = 3
- 5, 2, 5 / 2
- 5, 2, 2,5

37. Какое имя получит обработчик события соответствующий щелчку мыши на кнопке TButton?

- void TForm1::Button1Enter(TObject *Sender)
- void TForm1::Button1MouseDown(TObject *Sender, TMouseButton Button, TShiftState Shift, int X, int Y)
- void TForm1::Button1KeyPress(TObject *Sender, wchar_t &Key)
- void TForm1::Button1Click(TObject *Sender)
- void TForm1::Button1MouseEnter(TObject *Sender)

38. Если имеется код `int x; cin >> x;` и вводится 1.2, то что будет в переменной x?

- 1,2
- 2
- 2,0
- 1,2
- 1

39. В программе на языке C++ обязательно имеется функция...

- head
- start
- prime
- main
- finish

40. Какое свойство компонента TCheckBox определяет состояние флажка?

- Enabled
- Visible
- Checked
- Hint
- State

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
отлично	85-100% правильных ответов;
хорошо	66-84 % правильных ответов;
удовлетворительно	50-65 % правильных ответов;
неудовлетворительно	меньше 50 %.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	неудовлетворительно	удовлетворительно	хорошо	отлично
	не зачтено	зачтено		
<u>Знания</u>	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок
<u>Умения</u>	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме
<u>Навыки</u>	При решении стандартных задач не	Имеется минимальный набор	Продemonстрированы базовые навыки при	Продemonстрированы навыки при решении

	продемонстрированы базовые навыки. Имели место грубые ошибки	навыков для решения стандартных задач с некоторыми недочетами	решении стандартных задач с некоторыми недочетами	нестандартных задач без ошибок и недочетов
--	--------------------------------------------------------------	---------------------------------------------------------------	---------------------------------------------------	--------------------------------------------

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-2

1. Что такое объектно-ориентированное программирование?
2. Определение класса в C# и понятие объектов.
3. Функциональные компоненты класса и их типы.
4. Конструкторы и их роль в создании объектов.
5. Деструкторы и их назначение.
6. Перегрузка функциональных элементов и операций.
7. Константные объекты и функции, ключевое слово this.
8. Наследование классов и его особенности.
9. Виртуальные функции и полиморфизм.
10. Неоднозначность при множественном наследовании и виртуальный базовый класс.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-8

1. Статические одномерные массивы.
2. Основные методы строкового типа данных.
3. Алгоритм бинарного поиска.
4. Строковый тип данных, сравнение строк
5. Алгоритм линейного поиска
6. Понятие типа. Константы и переменные. Операции и формы их записи
7. Сортировка одномерного массива методом обмена

8. Логический тип данных. Диапазон значений логического типа данных. Логические операции
9. Сортировка одномерного массива методом вставки.
10. Целочисленные типы данных. Прямой, обратный и дополнительный код целочисленных типов данных. Диапазоны значений целочисленных типов данных
11. Сортировка одномерного массива методом выбора
12. Целочисленные типы данных. Операции над целочисленными типами данных
13. Задача о минимаксе
14. Вещественные типы данных. Операции над вещественными типами данных
15. Открытые массивы
16. Вещественные типы данных. Представление вещественных типов данных на примере модельного типа
17. Многомерные статические массивы в языке Pascal
18. Вещественные типы данных. Представление вещественных типов данных в памяти компьютера. Диапазоны значений вещественных типов данных
19. Примеры задач на обработку массивов
20. Символьный тип данных. Литералы символьного типа
21. Динамические одномерные массивы
22. Процедуры. Процедуры-функции
23. Метод пошаговой детализации
24. Потоки ввода/вывода
25. Пример синтаксического анализа
26. Классификация программного обеспечения
27. Задача о расстановке скобок
28. Связный список и его реализация
29. Трансляция. Виды трансляций
30. Стек и его реализация в языке

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Ответ полный и правильный на основании изученной теории; теоретический материал и решение поставленных задач изложены в необходимой логической последовательности, грамотный научный язык; ответ самостоятельный. Могут быть допущены две-три несущественные ошибки, исправленные по требованию преподавателя.
не зачтено	Ответ обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

5.3.3 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-2

1. Основные парадигмы ООП: наследование, полиморфизм, инкапсуляция.
2. Понятие класса и объекта (экземпляра класса).
3. Содержимое класса.
4. Особенности языка программирования C#.
5. Типы данных.
6. Статическая типизация и преобразования типов.
7. Циклы.
8. Конструкторы и инициализация объектов.
9. Определение нескольких конструкторов.
10. Параметры по умолчанию.
11. Деструктор.
12. Наследование классов.
13. Множественное наследование.
14. Инкапсуляция.
15. Спецификаторы доступа.
16. Геттеры и сеттеры.
17. Файл заголовков и исходных кодов класса.
18. Ключевое слово `this`.
19. Статические функции.
20. Виртуальные функции.
21. Указатели.
22. Получение значения по адресу.
23. Арифметические операции с указателями.
24. Указатели на массивы.
25. Указатели на функции.
26. Динамические объекты.
27. Динамические массивы.
28. Многомерные динамические массивы.
29. Абстрактные классы.
30. Перегрузка операторов.
31. Исключения.
32. Вложенные исключения.
33. Классы исключений.

- 34.Раздельная обработка исключений.
- 35.Шаблоны функций.
- 36.Шаблоны классов.
- 37.Строки.
- 38.Преобразование в строку.
- 39.Преобразование из строки.
- 40.Сравнение строк.
- 41.Поиск подстроки в строке.
- 42.Контейнер deque.
- 43.Контейнер list.
- 44.Файловые потоки.

5.3.4 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-8

1. Архитектура компьютера
2. Понятие предметной области. Объекты и классы
3. Основные идеи и компоненты .Net технологии
4. Последовательность создания и выполнения программ на платформе .Net.
5. Сборка (assembly) и ее структура в .Net технологии. Основные части и их назначение
6. Структура программы на языке ООП.
7. Описания классов на языке ООП.
8. Встроенные типы языка ООП. Неявное и явное приведение (преобразование) встроенных типов
9. Описание и использование методов класса. Перегрузка методов класса
10. Базовый класс System.Object. Назначение, методы
11. Операции языка ООП. Перегрузка операций в создаваемых классах
12. Программирование явного и неявного преобразования пользовательских типов (собственных классов).
13. Динамические одномерные массивы.
14. Процедуры. Процедуры-функции.
15. Метод пошаговой детализации.
16. Потоки ввода/вывода.
17. Пример синтаксического анализа.

18. Классификация программного обеспечения.
19. Задача о расстановке скобок.
20. Связный список и его реализация.
21. Трансляция. Виды трансляций.
22. Стек и его реализация в языке.
23. Архитектура компьютера.
24. Понятие предметной области. Объекты и классы.
25. Основные идеи и компоненты .Net технологии.
26. Последовательность создания и выполнения программ на платформе .Net.
27. Сборка (assembly) и ее структура в .Net технологии. Основные части и их назначение.
28. Структура программы на языке ООП.
29. Описания классов на языке ООП.
30. Встроенные типы языка ООП. Неявное и явное приведение (преобразование) встроенных типов.
31. Описание и использование методов класса. Перегрузка методов класса.
32. Базовый класс System.Object. Назначение, методы.
33. Операции языка ООП. Перегрузка операций в создаваемых классах.
34. Программирование явного и неявного преобразования пользовательских типов (собственных классов).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
отлично	Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности, грамотный научный язык; ответ самостоятельный.
хорошо	Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

Оценка	Критерии оценивания
удовлетворительно	Ответ полный, но при этом допущена существенная ошибка или неполный, несвязный ответ.
неудовлетворительно	Ответ обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Зыков С. В. Программирование. Объектно-ориентированный подход : учебник и практикум / С. В. Зыков. - Москва : Юрайт, 2023. - 155 с. - (Высшее образование). - ISBN 978-5-534-00850-0. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=845198&idb=0>.
2. Казанский Александр Анатольевич. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. - 2-е изд. - Москва : Юрайт, 2024. - 192 с. - (Высшее образование). - URL: <https://urait.ru/bcode/537364> (дата обращения: 15.08.2024). - ISBN 978-5-534-12338-8 : 909.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=910934&idb=0>.
3. Тузовский Анатолий Федорович. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. - Москва : Юрайт, 2024. - 213 с. - (Высшее образование). - URL: <https://urait.ru/bcode/537332> (дата обращения: 15.08.2024). - ISBN 978-5-534-16316-2 : 989.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=903277&idb=0>.

Дополнительная литература:

1. Лозовский В. В. Объектно-ориентированное программирование / Лозовский В. В., Морозов В. А., Гололобов А. А. - Москва : РТУ МИРЭА, 2024. - 107 с. - Книга из коллекции РТУ МИРЭА - Информатика. - ISBN 978-5-7339-2120-4., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=896946&idb=0>.
2. Унгер А. Ю. Объектно-ориентированное программирование : учебное пособие / Унгер А. Ю. - Москва : РТУ МИРЭА, 2022. - 84 с. - Книга из коллекции РТУ МИРЭА - Информатика. - ISBN 978-5-7339-1628-6., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=828336&idb=0>.
3. Якимов С. П. Объектно-ориентированное программирование: лаб. практикум / Якимов С. П., Резова Н. Л., Алехина А. Е. - Омск : СибАДИ, 2023. - 94 с. - Книга из коллекции СибАДИ - Информатика., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=973797&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

Лицензионное программное обеспечение: Операционная система Windows.

Лицензионное программное обеспечение: Microsoft Office.

Профессиональные базы данных и информационные справочные системы
Российский индекс научного цитирования (РИНЦ), платформа Elibrary: национальная
информационно-аналитическая система. Адрес доступа: http://elibrary.ru/project_risc.asp

ГАРАНТ. Информационно-правовой портал [Электронный ресурс].– Адрес доступа:
<http://www.garant.ru>

MathSciNet: информационно-библиографическая и реферативная база данных по математике, в
т.ч. прикладной математике и статистике. Электронная версия Mathematical Reviews. Адрес
доступа: <http://www.ams.org/mathscinet>

Math-Net.Ru: Общероссийский математический портал. Адрес доступа: <http://www.mathnet.ru/>

Свободно распространяемое программное обеспечение:

программное обеспечение LibreOffice;

программное обеспечение Yandex Browser;

программное обеспечение Paint.NET;

программное обеспечение 1С:

* "Бухгалтерия предприятия", редакция 3.0, см. <http://v8.1c.ru/buhv8/> ,

* "Управление торговлей", редакция 11.1, см. <http://v8.1c.ru/trade/> ,

* "Зарплата и управление персоналом", редакция 3.0, см. <http://v8.1c.ru/hrm/> ,

* "Управление небольшой фирмой", редакция 1.5, см. <http://v8.1c.ru/small.biz/> ,

* "ERP Управление предприятием 2.0", см. <http://v8.1c.ru/erp/> .

* "Бухгалтерия государственного учреждения", редакция 1.0, см. <http://v8.1c.ru/stateacc/> ,

* "Зарплата и кадры государственного учреждения", редакция 1.0, <http://v8.1c.ru/statehrm/> .

программное обеспечение PascalABC.NET

Электронные библиотечные системы и библиотеки:

Электронная библиотечная система "Лань" <https://e.lanbook.com/>

Электронная библиотечная система "Консультант студента" <http://www.studentlibrary.ru/>

Электронная библиотечная система "Юрайт" <http://www.urait.ru/ebs>

Электронная библиотечная система "Znanium" <http://znanium.com/>

Электронно-библиотечная система Университетская библиотека ONLINE <http://biblioclub.ru/>

Фундаментальная библиотека ННГУ www.lib.unn.ru/

Сайт библиотеки Арзамасского филиала ННГУ. – Адрес доступа: lib.arz.unn.ru

Ресурс «Массовые открытые онлайн-курсы Нижегородского университета им. Н.И. Лобачевского»
<https://mooc.unn.ru/>

Портал «Современная цифровая образовательная среда Российской Федерации»
<https://online.edu.ru/public/promo>

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 09.03.03 - Прикладная информатика.

Автор(ы): Сазанов Александр Анатольевич.

Рецензент(ы): Статуев Алексей Анатольевич, кандидат педагогических наук.

Заведующий кафедрой: Нестерова Лариса Юрьевна, кандидат педагогических наук.

Программа одобрена на заседании методической комиссии от 10.12.2025, протокол № 10.