

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной  
математики Кафедра вычислительных технологий

УТВЕРЖДАЮ

Проректор по учебной работе,  
качеству образования – первый  
проректор

Хагуров Т.А.

подпись

«26» мая 2023 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ Б1.О.16 «Конструирование алгоритмов и структур данных»

Направление

подготовки/специальность 02.03.02 **Фундаментальная информатика и  
информационные технологии**

(код и наименование направления подготовки/специальности)

Направленность (профиль) /специализация

Математическое и программное обеспечение компьютерных технологий

Программа подготовки академический бакалавриат

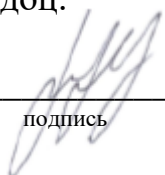
Форма обучения очная

Квалификация выпускника бакалавр

Краснодар 2023

Рабочая программа дисциплины «Конструирование алгоритмов и структур данных» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Программу составил: А.Н. Полетайкин, доц. каф. ИТ, к.т.н., доц.



---

подпись

Рабочая программа дисциплины «Конструирование алгоритмов и структур данных» обсуждена на заседании кафедры вычислительных технологий протокол №8 от «03» мая 2023 г.

Заведующий кафедрой (выпускающей)  
Ю. М. Вишняков



---

подпись

Рабочая программа обсуждена на заседании кафедры вычислительных технологий протокол №8 от «03» мая 2023 г.

Заведующий кафедрой (выпускающей)  
Ю. М. Вишняков



---

подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №5 от «19» мая 2023 г.

Председатель УМК факультета  
А. В. Коваленко



---

подпись

Рецензенты:

Гаркуша О.В., доцент кафедры информационных технологий  
ФБГОУ ВО «Кубанский государственный университет»,  
кандидат физико-математических наук.

Схаляхо Ч.А., доцент КВВУ им.С.М.Штеменко, к.ф.-м.н., доцент

# 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

## 1.1 Цель и задачи освоения дисциплины

Дисциплина «Конструирование алгоритмов и структур данных» рассматривает методы, способы, алгоритмические и программные средства структурного и объектно-ориентированного программирования для решения вычислительных задач, преимущественно обработки структур данных. Изучаются основные понятия, методы и модели программирования, способы алгоритмизации задач программирования статических и динамических структур данных, организации баз данных, модульного тестирования программного обеспечения, средства работы с двумерной графикой. Используются современные инструментальные средства разработки программного обеспечения Microsoft Visual Studio Community.

**Целью** изучения дисциплины «Конструирование алгоритмов и структур данных» является формирование у студентов знаний, умений и практических навыков в области алгоритмизации и программирования структур данных, организации баз данных, проверки качества разработанных программ. Предметом учебной дисциплины являются методы, подходы и инструментальные средства алгоритмизации и программирования.

## 1.2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Конструирование алгоритмов и структур данных» относится к базовой части блока Б1 Дисциплины (модули).

Для изучения дисциплины студент должен владеть знаниями, умениями и навыками по дисциплинам: Основы программирования, Дискретная математика и Алгоритмы вычислительной математики, с которыми дисциплина связана логически и содержательно-методически.

Дисциплина предшествует изучению следующих дисциплин бакалавриата: Теория параллельных алгоритмов, Основы компьютерной графики, Программирование в компьютерных сетях, Информационная безопасность, Криптографические протоколы, Оценка сложности алгоритмов, Распределенные задачи и алгоритмы, Технологии Grid вычислений, Паттерны программирования, Программирование для игровых платформ.

Особенности реализации дисциплины: дисциплина реализуется в смешанной форме на русском языке.

### 1.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующими **профессиональными компетенциями и соотнесенные с ними индикаторы достижения компетенций:**

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине ( <i>знает, умеет, владеет (навыки и/или опыт деятельности)</i> )
<b>ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности</b>	
ОПК-2.1. Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ	Знает: принципы и технологии объектно-ориентированного программирования, основные подходы к разработке программных продуктов, методы проектирования баз данных (БД); Умеет: разрабатывать прикладное программное обеспечение (ПО); проектировать БД в заданной предметной области. Владеет: понятийным аппаратом в области разработки ПО
ОПК-2.2. Знает особенности языков программирования, теорию алгоритмов, умеет составлять программы	Знает: синтаксис и семантику и семиотику языка С#; основные виды структур данных; основные модели данных. Умеет: выделять и классифицировать информационные потоки; программировать структуры данных разных типов; применять методы реляционной алгебры для манипулирования данными. Владеет: навыками разработки ПО; навыками программирования структур данных разных типов; методами и технологиями построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя.
ОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций	Знает: принципы объектно-ориентированного анализа, язык UML, принципы интеграции компонентов ПО Умеет: анализировать предметную область, строить канонические диаграммы UML; программировать графические объекты и графический пользовательский интерфейс. Владеет: принципами объектно-ориентированного проектирования на языке UML, технологиями тестирования и интеграции ПО.
<b>ОПК-6. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</b>	
ОПК-6.1. Знает основные технологии создания и внедрения информационных технологий (ИТ), стандарты управления жизненным циклом ПО	Знает понятие жизненного цикла ИТ и ПО. Умеет выбирать и применять основные технологии создания и внедрения ИТ и ПО, стандарты управления жизненным циклом ИТ и ПО
ОПК-6.2. Способен осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла ПО	Знает состав, структуру организационного обеспечения ИТ и способы организации работ по созданию ПО. Умеет осуществлять организационное обеспечение выполнения работ на всех стадиях жизненного цикла ПО.
ОПК-6.3. Готов к составлению плановой и отчетной документации по управлению проектами создания ПО	Умеет работать с документацией к ПО, создавать документацию к ПО на всех стадиях жизненного цикла ПО Владеет навыками составления плановой и отчетной документации к ПО на стадиях жизненного цикла

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине (знает, умеет, владеет (навыки и/или опыт деятельности))
<b>ПК-1. Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии</b>	
ПК-1.1. Знает основы научно-исследовательской деятельности в области информационных технологий, имеет научные знания в теории информационных систем	Знает: принципы построения структур данных при помощи классов, составление и миграцией методов между классами Умеет: решать проблемы верификации и валидации ПО; тестировать ПО, выполнять рефакторинг Владеет: навыками переработки кода
ПК-1.2. Умеет применять полученные знания в области фундаментальных научных основ теории информации и решать стандартные задачи в собственной научно-исследовательской деятельности	Знает: основы исследования-разработки применительно к конструированию алгоритмов и структур данных; Умеет: разрабатывать прикладное ПО; Владеет: навыками микропроектирования, макропроектирования, предметно-ориентированного проектирования ПО
ПК-1.3. Имеет практический опыт научно-исследовательской деятельности в области информационных технологий	Знает: основы исследования-разработки применительно к конструированию алгоритмов и структур данных; Умеет: проводить исследование ПО и его последовательное развитие Владеет: навыками исследовательской деятельности

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

## 2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 2.1 Распределение трудоемкости дисциплины по видам работ

Общая трудоемкость дисциплины составляет 11 зач.ед. (396 часов), их распределение по видам работ представлено в таблице (для студентов ОФО)

Вид учебной работы	Всего часов	Семестры (часы)			
		III	IV		
<b>Контактная работа, в том числе:</b>	<b>227</b>	<b>120,5</b>	<b>106,5</b>		
<b>Аудиторные занятия (всего):</b>	<b>220</b>	<b>118</b>	<b>102</b>		
Занятия лекционного типа	84	50	34		
Лабораторные занятия	136	68	68		
Занятия семинарского типа (семинары, практические занятия)	–	–	–		
<b>Иная контактная работа:</b>	<b>7</b>	<b>2,5</b>	<b>4,5</b>		
Контроль самостоятельной работы (КСР)	6	2	4		
Промежуточная аттестация (ИКР)	1	0,5	0,5		
<b>Самостоятельная работа, в том числе:</b>	<b>88,6</b>	<b>59,8</b>	<b>28,8</b>		
<i>Курсовая работа</i>	–	–	–		
<i>Проработка учебного (теоретического) материала</i>	60	44	16		
<i>Выполнение индивидуальных заданий (подготовка сообщений, презентаций)</i>	24,6	16,2	8,2		
<i>Реферат</i>	–	–	–		
Подготовка к текущему контролю	4	2	2		

<b>Контроль:</b>		<b>80,4</b>	<b>35,7</b>	<b>44,7</b>		
Подготовка к экзамену		<b>80,4</b>	35,7	44,7		
<b>Общая трудоемкость</b>	<b>час.</b>	<b>396</b>	<b>216</b>	<b>180</b>		
	<b>в том числе контактная работа</b>	<b>227</b>	<b>120,5</b>	<b>106,5</b>		
	<b>зач. ед</b>	<b>11</b>	<b>6</b>	<b>5</b>		

## 2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.  
Разделы дисциплины, изучаемые в III и IV семестрах (очная форма обучения)

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	КСР	ЛР	
<b>III семестр. Объектно-ориентированное программирование структур данных на языке C#</b>						
1.	Алгоритмизация и программирование структур данных	10	2	0,5	4	4
2.	Основные принципы и технологии программирования	10	2		4	4
3.	Объектно-ориентированное программирование (ООП)	10	2		4	4
4.	Язык программирования C# и технология .NET Framework	10	2		4	4
5.	Язык программирования C#	12	4	0,5	4	4
6.	Обработка исключений в C#.	12	4		4	4
7.	Классы и объекты в C#	10	2		4	4
8.	Механизмы наследования	14	4		6	4
9.	Интерфейсы	14	4	0,5	6	4
10.	Классы-коллекции.	14	4		6	4
11.	Перегрузка операций	12	4		4	4
12.	Делегаты и события	12	4	0,5	4	4
13.	Windows Forms	14	4		6	4
14.	Обеспечение качества ПО	14	4		6	4
15.	Унифицированный язык моделирования UML.	9,8	4	–	2	3,8
<b>ИТОГО по разделам дисциплины в III семестре</b>		<b>179,8</b>	<b>50</b>	<b>2</b>	<b>68</b>	<b>59,8</b>
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,5				
Подготовка к текущему контролю		35,7				
<b>Общая трудоемкость по дисциплине в III семестре</b>		<b>216</b>				
<b>IV семестр. Конструирование баз данных</b>						
16.	Введение в базы данных (БД).	8,4	2	0,4	4	2
17.	Моделирование данных	10,3	2	0,3	6	2
18.	Теория нормальных форм	12,3	4	0,3	6	2
19.	Операции над данными в реляционной модели	12,3	4	0,3	6	2
20.	Введение в MS SQL	10,3	2	0,3	6	2
21.	Подмножество языка SQL-DML. Команды модификации данных.	10,3	2	0,3	6	2
22.	Представления и курсоры	8,3	2	0,3	4	2
23.	Хранимые процедуры (Stored Procedure)	10,3	2	0,3	6	2

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	КСР	ЛР	
24.	Триггеры	12,3	4	0,3	6	2
25.	Подмножества языка SQL-TCL. И SQL-DCL	8,3	2	0,3	4	2
26.	Программирование обработки данных на языке С#	12,3	4	0,3	6	2
27.	Трехуровневая архитектура организации БД	9,3	2	0,3	4	3
28.	Постреляционные СУБД	10,1	2	0,3	4	3,8
<b>ИТОГО по разделам дисциплины в IV семестре</b>		<b>134,8</b>	<b>34</b>	<b>4</b>	<b>68</b>	<b>28,8</b>
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0,5				
Подготовка к текущему контролю		44,7				
<b>Общая трудоемкость по дисциплине в IV семестре</b>		<b>180</b>				

## 2.3 Содержание разделов дисциплины:

### 2.3.1 Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представителей работодателей
<b>III семестр. Объектно-ориентированное программирование структур данных на языке С#</b>				
1.	Алгоритмизация и программирование структур данных	Основные определения и понятия. Средства и способы формализации алгоритмов. Базовые канонические структуры алгоритмов. Средства программирования	Тесты	Нет
2.	Основные принципы и технологии программирования	Понятие жизненного цикла программного продукта. Структурное программирование. Модульное программирование. Объектно-ориентированное программирование.	Тесты	Нет
3.	Объектно-ориентированное программирование (ООП)	Основные понятия ООП. Этапы объектно-ориентированного программирования. Достоинства и недостатки ООП. Архитектура классов и объектов.	Тесты	Нет
4.	Язык программирования С# и технология .NET Framework	Платформа .NET. Технология .NET Framework. Компиляция и язык MSIL. Поддержка ЖЦ ПО в рамках подхода .NET	Тесты	Нет
5.	Язык программирования С#	История. Поддерживаемые механизмы. Особенности применения. Основные операторы языка. Универсальная система типизации (UTS). Встроенные типы данных языка С#. Неявные преобразования. Явные преобразования	Тесты	Нет
6.	Обработка исключений в С#.	Конструкция try.catch.finally. Принудительная генерация исключений. Класс Exception и его основные наследники. Свойства класса Exception.	Тесты	Нет
7.	Классы и объекты в С#	Определение классов. Модификаторы прав доступа. Копирующие конструкторы. Уничтожение объектов. Деструктор. Инкапсуляция с использованием свойств типа. Операции класса. Деструкторы. Вложенные типы. Методы. Свойства	Тесты	Нет
8.	Механизмы	Описание класса-потомка. Конструкторы при наследовании. Виртуальные методы. Абстрактные	Тесты	Нет

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представителей работодателей
	наследования	классы. Бесплодные классы. Виды взаимоотношений между классами. Класс object		
9.	Интерфейсы	Правила описания интерфейсов. Интерфейсы. Перечисления. Базовый тип перечисления. Базовый класс System.Enum. Интерфейс IEnumerable.	Тесты	Нет
10.	Классы-коллекции.	Свойства. Функции доступа. Свойства и наследование. Пример объявления свойств в интерфейсах. Статические свойства. Индексаторы. Интерфейсы и индексаторы. Отличия между свойствами и индексаторами	Тесты	Нет
11.	Перегрузка операций	Ключевое слово operator. Унарные операторы. Бинарные операторы. Пример класса с перегруженными методами и операциями	Тесты	Нет
12.	Делегаты и события	Правила определения и использования. Передача делегата через список параметров. Лямбда-выражения. События. Определение события. Механизм событий.	Тесты	Нет
13.	Windows Forms	Пространство имен System.Windows.Forms. Элементы управления. Кнопки и надписи. Практический пример. Отображение координат мыши при ее перемещении.	Тесты	Нет
14.	Обеспечение качества ПО	Виды тестирования. Необходимость. Рефакторинг. Работа с ошибками. Шаблоны тестовых проектов MS Visual Studio. Пять принципов чистых тестов (F.I.R.S.T. Principles). Добавление и выполнение методов модульного теста. Использование классов Assert для модульного тестирования. Типы классов Assert	Тесты	Нет
15.	Унифицированный язык моделирования UML.	Канонический набор диаграмм языка UML. Особенности изображения диаграмм языка UML. Правила графического изображения диаграмм языка UML. Диаграммы вариантов использования UML Особенности построения диаграммы, отношения между актерами, отношения между вариантами использования. Диаграммы классов UML. Особенности изображения классов, виды классов, отношения между классами. Диаграммы поведения. Диаграммы физической архитектуры программ.	Тесты	Нет
<b>IV семестр. Конструирование баз данных</b>				
16.	Введение в базы данных (БД).	Требования к БД. Свойства БД. Классификация БД. Жизненный цикл БД. Системы управления базами данных (СУБД). Компоненты СУБД. Задачи, решаемые СУБД. Классификация СУБД. Примеры СУБД. Модели данных. Классификация моделей данных. Критерии оценки модели данных. Инфологическое проектирование. Модель "сущность-связь" (ER-модель). Составляющие ER-модели. Выделение типов сущностей. Определение связей и атрибутов.	Тесты	Нет
17.	Моделирование данных	Классификация моделей данных. Жизненный цикл БД. Системный анализ предметной области. Представление данных с помощью модели "сущность-связь". Элементы модели. Выделим интересующие нас сущности и связи. Иерархическая модель данных. Сетевая модель	Тесты	Нет



№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представителей работодателей
		данных. Реляционная модель данных		
18.	Теория нормальных форм	1NF - первая нормальная форма. 2NF - вторая нормальная форма. 3NF - третья нормальная форма. BCNF - нормальная форма Бойса-Кодда. Многочисленные зависимости и четвертая нормальная форма (4NF). Зависимости по соединению и пятая нормальная форма (5NF). Ограничения целостности. Двенадцать правил Кодда, которым должна соответствовать реляционная СУБД	Тесты	Нет
19.	Операции над данными в реляционной модели	Реляционная алгебра. Операции обработки кортежей. Операции обработки отношений. Теоретико-множественные операторы обработки отношений. Специальные реляционные операторы обработки отношений. Реляционное исчисление. Запрос на языке реляционного исчисления. Исчисление кортежей. Запрос на языке реляционного исчисления. Исчисление доменов.	Тесты	Нет
20.	Введение в MS SQL	Основные объекты MS SQL. Создание и удаление БД. Использование SQL Server Enterprise Manager. Использование SQL оператора CREATE DATABASE. Типы данных SQL. Создание и удаление таблиц. Создание таблиц с помощью CREATE TABLE. Создание таблиц с помощью SQL Server Enterprise Manager. Создание ограничений для таблиц. Целостность данных. Правила удаления и обновления. Индексы. Создание индексов. Управление индексами. Некоторые сведения об индексах. Правила. Связывание правила со столбцом. Ограничения для правил. Значения по умолчанию. Связывание значения по умолчанию со столбцом. Ограничения для значений по умолчанию	Тесты	Нет
21.	Подмножество языка SQL-DML. Команды модификации данных.	Выборка данных. Оператор выбора SELECT: синтаксис, предложения, примеры. Выборка из нескольких таблиц. Вычисления внутри SELECT. Группировка данных. Сортировка данных. Операция объединения	Тесты	Нет
22.	Представления и курсоры	Использование представлений/ Создание и уничтожение представлений. Операции выборки из представлений. Обновление представлений. Назначение представления. Курсоры. Объявления курсоров. Отличие курсоров от представлений	Тесты	Нет
23.	Хранимые процедуры (Stored Procedure)	Создание хранимых процедур. Удаление хранимых процедур. Модификация хранимых процедур. Правила и значения по умолчанию	Тесты	Нет
24.	Триггеры	Использование триггеров. Срабатывание триггеров. Создание триггеров. Удаление триггеров. Модификация триггеров. DML-триггеры: назначение, форма создания, параметры триггера, правила работы. Использование таблиц deleted и inserted., DML-триггеры и среда CLR. Реализация триггера на языке C#. DDL-триггеры: форма создания, виды, типовые задачи, примеры применения. Logon-trigger: форма создания, виды, типовые задачи, примеры применения. Управление триггерами: активация/деактивация,	Тесты	Нет

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представителей работодателей
		изменение и удаление триггера		
25.	Подмножества языка SQL-TCL. И SQL-DCL	Природа транзакций. Команды управления транзакциями. Последовательность выполнения транзакций. Ограничения транзакций. Журнал транзакций. Последовательность событий при регистрации транзакций. Блокировка в транзакциях. Переопределение блокировок на уровне запросов. Диагностика и решение проблем блокировки. Взаимные блокировки. Объектные и системные привилегии. Операторы GRANT, REVOKE.	Тесты	Нет
26.	Программирование обработки данных на языке C#	Программирование подключения к базе данных Microsoft SQL Server Параметры строки подключения. Свойства подключения. Выполнение команд SQL на языке C# (объект SqlCommand). Типизация результатов выполнения команд SQL на языке C# (объект SqlDataReader). Параметризация запросов SQL на языке C# (объект SqlParameter). Типы параметров. Работа с хранимыми процедурами на языке C#. Программирование операций сохранения в БД и извлечения из БД файлов. Программирование операций с наборами данных. Объекты DataSet, DataTable, DataRow, DataColumn, SqlDataAdapter. Программирование операций с базой данных посредством визуального приложения.	Тесты	Нет
27.	Трехуровневая архитектура организации БД	Основные понятия. Модели взаимодействия клиент-сервер. Мониторы транзакций. Обработка распределенных данных. Структура сервера базы данных. Ограничения реляционных баз данных.	Тесты	Нет
28.	Постреляционные СУБД.	Объектно-ориентированные СУБД. Объектно-ориентированная парадигма. Объектно-ориентированные СУБД. Объектные расширения реляционных СУБД. Язык SQL-3. Объектно-реляционные СУБД.	Тесты	Нет

### 2.3.2. Занятия семинарского типа

Занятия семинарского типа – не предусмотрены.

### 2.3.3. Лабораторные занятия

№ работы	№ раздела	Наименование лабораторных работ	Форма текущего контроля
<b>III семестр. Объектно-ориентированное программирование структур данных на языке C#</b>			
1		Представление данных в ЭВМ и операции над ними в языке C#: среда .NET RunTime и библиотека Framework, процесс компиляции C#-программы, анатомия программы, создание простого консольного приложения на языке C# в среде Microsoft Visual Studio, основные типы данных, преобразования типов	Отчет о выполнении лабораторной работы
2		Программирование алгоритмов с ветвлением: Основные операторы языка C#, встроенные математические функции, вычисление арифметических и логических выражений, консольный ввод-вывод, операторы ветвления, обработка	-//-

		исключений	
3		<u>Программирование алгоритмов циклической структуры:</u> форматирование вывода, операторы циклов C#, итеративная обработка данных	-//-
4		<u>Алгоритмизация обработки одномерных массивов:</u> объявление заданного массива фиксированной длины, инициализацию элементов массива случайными числами, обработка массива циклами for и foreach	-//-
5		<u>Алгоритмизация обработки строк.</u> Пользовательские функции: класс System.String, управляющие последовательности символов, регулярные выражения, программирование пользовательских функций	-//-
6		<u>Работа с файлами:</u> классы FileStream, MemoryStream и NetworkStream, создание потоков ввода-вывода, файловый ввод-вывод строковых данных	-//-
7		<u>Статические структуры данных.</u> Модульные тесты: описание и использование статических структур, ограничения при использовании статических структур, модульное тестирование пользовательских функций с логированием результатов работы каждой функции	-//-
8		<u>Объектно-ориентированное программирование.</u> Использование классов: понятия класса, конструктора, деструктора и метода, уровни доступа к данным, особенности организации программы с использованием классов	-//-
9		<u>Объектная модель обработки матриц:</u> инкапсуляция с использованием традиционных методов доступа и изменения, базовый механизм наследования языка C#, спецификация родительского класса, перегрузка методов и операторов, внутренняя сортировка	-//-
10		<u>Динамические структуры данных.</u> Объектно-ориентированное программирование односвязного списка: Инкапсуляция с использованием свойств типа, виртуальные методы, объектная модель односвязного списка	-//-
11		<u>Принудительное наследование.</u> Абстрактные классы в C#: наследование с применением абстрактных классов, реализация базовых классов при использовании принудительного наследования	-//-
12		<u>Наследование типа «включение-делегирование».</u> Классы-коллекции: полиморфизм посредством абстрактных классов, построенных по принципу «включение-делегирование»	-//-
13		<u>Использование интерфейсов и делегатов:</u> составление и тестирование алгоритмов и объектно-ориентированных программ, использующих стандартные интерфейсы и делегаты	-//-
14		<u>Двумерная графика в C#.</u> Бинарные деревья: графический интерфейс в C#, Пространство имен System.Drawing, классы Graphics, Pen и Brush, бинарные деревья, программирование обходов дерева, копирование и удаление деревьев, прорисовка дерева на форме	-//-
15		<u>Нелинейные структуры данных.</u> Графы: алгоритмы на графах, поиск в глубину и в ширину, кратчайшие пути, связность и достижимость вершин, двойная буферизация графики	-//-
16		<u>Построение графиков функций:</u> библиотека ZedGraph,	-//-

		графическое представление математических функций, масштабирование графиков	
<b>IV семестр. Конструирование баз данных</b>			
1		Анализ предметной области: изучение и описание заданной предметной области на уровне бизнес-процессов и инфопотоков, инфологическое моделирование базы данных (БД), построение моделей сущность-связь, вариантов использования, классов в нотации UML	-//-
2		Даталогическое моделирование реляционной базы данных: разработка даталогической ER-модели реляционной БД, нормализация отношений, документирование БД	-//-
3		Знакомство с СУБД Microsoft SQL Server: установка MS SQL Server и освоение основных инструментов для разработки БД	-//-
4		MS SQL Server: создание базы данных: разработка БД с помощью Enterprise Manager и Query Analyzer; типы данных MS SQL Server; первичные и внешние ключи, индексирование БД; условия на значение; свойства таблиц и полей.	-//-
5		MS SQL Server: формирование области запросов к БД: язык структурированных запросов SQL; разработка представлений и курсоров при помощи Query Analyzer; вывод результатов запросов на экран, в таблицу и файл; разработка сложных объектов SQL Server: хранимые процедуры, правила, триггеры, транзакции.	-//-
6		Разработка БД для заданной предметной области: разработка модификация БД проектируемой информационной системы (ИС) в СУБД MS SQL Server	-//-
7		Разработка области запросов к БД: разработка SQL запросов для выборки данных разной сложности, разработка представлений для формирования выходных данных ИС	-//-
8		Работа с базой данных в автономном режиме: разработка, тестирование, построение и развертывание тестовой базы данных приложения в автономном режиме	-//-
9		Формирование клиент-серверной ИС в MS Visual Studio: установка связи БД SQL Server с приложением, построение главной формы, формирование и настройка пользовательского меню и обеспечение доступа в объектам БД на уровне приложения	-//-
10		Программирование пользовательского интерфейса: разработка форм для ввода и редактирования оперативных данных расширенными средствами MS Visual Studio	-//-
11		Программирование вывода данных из БД: организация вывода данных в форматах XLSX и HTML, разработка отчетов для выходных документов с помощью расширения Crystal Report, экспорт отчетов в файлы doc/docx и pdf	-//-
12		Разработка триггеров: реализация специфических ограничений определения структур данных и манипулирования данными при помощи триггеров	-//-

### 2.3.3 Примерная тематика курсовых работ (проектов)

Учебным планом не предусмотрены.

### 2.3.4 Расчетно-графические задания

Учебным планом не предусмотрены.

## 2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №2 от 25.11.2021
2	Решение задач по объектно-ориентированному программированию структур данных	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №2 от 25.11.2021
3	Моделирование задач обработки данных на языке UML	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №2 от 25.11.2021

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

## 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) – совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология – индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций – анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой

конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
III	Л	Компьютерные презентации и обсуждение	10
	ЛР	Разбор конкретных ситуаций (задач), тренинги по решению задач, компьютерные симуляции (программирование алгоритмов)	10
IV	Л	Компьютерные презентации и обсуждение	10
	ЛР	Разбор конкретных ситуаций (задач), тренинги по решению задач, компьютерные симуляции (программирование алгоритмов)	10
Итого:			40

## 4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### 4.1 Фонд оценочных средств для проведения текущего контроля

#### Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Код и наименование индикатора	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	ОПК-2.1. Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ	Знает: принципы и технологии объектно-ориентированного программирования, основные подходы к разработке программных продуктов, методы проектирования баз данных (БД); Умеет: разрабатывать прикладное программное обеспечение (ПО); проектировать БД в заданной предметной области. Владеет: понятийным аппаратом в области разработки ПО	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.1–1.11, 2.1–2.10
2	ОПК-2.2. Знает особенности языков программирования, теорию алгоритмов, умеет составлять программы	Знает: синтаксис и семантику и семиотику языка C#; основные виды структур данных; основные модели данных. Умеет: выделять и классифицировать информационные потоки; программировать структуры данных разных типов; применять методы реляционной алгебры для манипулирования данными. Владеет: навыками разработки ПО; навыками программирования структур данных разных типов; методами и технологиями построения, сопровождения и модификации БД в соответствии с нуждами конечного пользователя.	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.12–1.21, 2.11–2.32
3	ОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций	Знает: принципы объектно-ориентированного анализа, язык UML, принципы интеграции компонентов ПО Умеет: анализировать предметную область, строить канонические диаграммы UML; программировать графические объекты и графический пользовательский интерфейс. Владеет: принципами объектно-ориентированного проектирования на языке UML, технологиями тестирования и интеграции ПО.	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.51–1.60, 2.46–2.57
4	ОПК-6.1. Знает основные технологии создания и	Знает понятие жизненного цикла ИТ и ПО.	Защита отчетов о	Вопросы 1.1–1.11,



	внедрения информационных технологий (ИТ), стандарты управления жизненным циклом ПО	Умеет выбирать и применять основные технологии создания и внедрения ИТ и ПО, стандарты управления жизненным циклом ИТ и ПО	выполнении лабораторных работ  Тесты	2.1–2.10
5	ОПК-6.2. Способен осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла ПО	Знает состав, структуру организационного обеспечения ИТ и способы организации работ по созданию ПО. Умеет осуществлять организационное обеспечение выполнения работ на всех стадиях жизненного цикла ПО.	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.12–1.21, 2.11–2.32
6	ОПК-6.3. Готов к составлению плановой и отчетной документации по управлению проектами создания ПО	Умеет работать с документацией к ПО, создавать документацию к ПО на всех стадиях жизненного цикла ПО Владеет навыками составления плановой и отчетной документации к ПО на стадиях жизненного цикла	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.51–1.60, 2.46–2.57
7	ПК-1.1. Знает основы научно-исследовательской деятельности в области информационных технологий, имеет научные знания в теории информационных систем	Знает: принципы построения структур данных при помощи классов, составление и миграцией методов между классами Умеет: решать проблемы верификации и валидации ПО; тестировать ПО, выполнять рефакторинг Владеет: навыками переработки кода	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.40–1.50
8	ПК-1.2. Умеет применять полученные знания в области фундаментальных научных основ теории информации и решать стандартные задачи в собственной научно-исследовательской деятельности	Знает: основы исследования-разработки применительно к конструированию алгоритмов и структур данных; Умеет: разрабатывать прикладное ПО; Владеет: навыками микропроектирования, макропроектирования, предметно-ориентированного проектирования ПО	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.22–1.34, 2.33–2.45
9	ПК-1.3. Имеет практический опыт научно-исследовательской деятельности в области информационных технологий	Знает: основы исследования-разработки применительно к конструированию алгоритмов и структур данных; Умеет: проводить исследование ПО и его последовательное развитие Владеет: навыками исследовательской деятельности	Защита отчетов о выполнении лабораторных работ  Тесты	Вопросы 1.35–1.50

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения заданий, лабораторных работ, средств итоговой аттестации (зачет и экзамен в III и IV семестрах).

Оценка успеваемости осуществляется по результатам:

- выполнения лабораторных работ;
- выполнения коллоквиумных работ;
- ответов на теоретические вопросы при защите отчетов о выполнении лабораторных работ;

- выполнения тестовых заданий;
- ответа на зачете (для выявления знания и понимания практического материала дисциплины).
- ответа на экзамене (для выявления знания и понимания теоретического материала дисциплины).

## **Перечень вопросов к зачету и экзамену**

### **1. Объектно-ориентированное программирование структур данных на языке C#**

1. Понятие алгоритма и алгоритмизации. Свойства алгоритма. Базовые канонические структуры алгоритмов. Средства изображения алгоритмов. Примеры.
2. Понятие программы, программирование и языки программирования. Понятие жизненного цикла программного продукта.
3. Цели и основные принципы структурного программирования. Достоинства и недостатки.
4. Цели и основные принципы модульного программирования. Достоинства и недостатки.
5. Основные понятия объектно-ориентированного программирования. Архитектура класса и объекта. Этапы ООП. Достоинства и недостатки ООП. Технология разработки ОО программ.
6. Платформа .NET. Технология .NET Framework. Архитектурная схема .NET Framework и Visual Studio.NET. Библиотека FCL
7. Компиляция в .NET Framework и язык MSIL. Общезыковая спецификация (CLS). Технология выполнения C#-программы. Схема выполнения .NET-приложения в среде CLR.
8. Поддержка жизненного цикла ПО в рамках подхода .NET. Схема выполнения CLR.
9. Язык программирования C#. История. Поддерживаемые механизмы. Особенности применения.
10. Универсальная система типизации (UTS). Встроенные типы данных языка C#.
11. Обработка исключений в C#. Конструкция try.catch.finally. Принудительная генерация исключений.
12. Обработка исключений в C#. Класс Exception и его основные наследники. Свойства класса Exception.
13. Определение классов в языке C#. Модификаторы прав доступа. Создание объектов. Конструкторы. Инициализаторы. Пример.
14. Присваивание и сравнение объектов. Поверхностное и глубокое копирование объектов. Пример. Уничтожение объектов. Деструкторы.
15. Данные классов. Синтаксис описания констант и полей. Спецификаторы полей класса. Статические поля.
16. Методы классов. Синтаксис метода. Параметры методов. Передача параметров. Ключевое слово this. Статические методы.
17. Операции классов. Перегрузка унарных операторов, синтаксис, особенности, пример.
18. Операции классов. Перегрузка бинарных операторов, синтаксис, особенности, пример.
19. Преобразования типов. Неявные преобразования типов, пример.
20. Преобразования типов. Явные преобразования типов, пример
21. Структура цикла foreach, его назначение и отличия от цикла for. Отличительные особенности обработки массивов в циклах for и foreach.
22. Службы инкапсуляции. Концепция инкапсуляции. Инкапсуляция с использованием традиционных методов доступа и изменения.
23. Инкапсуляция с использованием свойств типа. Виды свойств, синтаксис, пример.
24. Инкапсуляция с использованием индексов. Виды индексов, синтаксис, пример. Сравнительный анализ свойств и индексов.

25. Наследование. Спецификатор доступа `protected`. Конструкторы при наследовании. Запрет наследования. Ссылки на объекты базового и дочернего классов.
26. Виртуальные методы. Переопределение виртуальных методов. Пример.
27. Абстрактные классы. Наследование абстрактных классов. Пример.
28. Наследование классов. Виды взаимоотношений между классами.
29. Наследование классов. Класс `object`: методы и особенности их использования.
30. Интерфейсы. Правила описания и использования интерфейсов. Отличия интерфейсов от абстрактных классов. Примеры.
31. Интерфейсы и наследование. Операции `is` и `as`. Интерфейсы и индексы. Примеры.
32. Перечисления. Базовый тип перечисления. Базовый класс `System.Enum`
33. Стандартные интерфейсы. Методы. Свойства. Примеры.
34. Классы-коллекции. Наследование типа «включение-делегирование». Пример.
35. Свойства. Функции доступа. Свойства и наследование. Пример объявления свойств в интерфейсах. Статические свойства
36. Делегаты. Правила определения и использования. Передача делегата через список параметров. Лямбда-выражения. Примеры.
37. События. Определение события. Механизм событий. Примеры.
38. Пространство имен `System.Windows.Forms`
39. Элементы управления `Windows.Forms`: методы и свойства
40. Понятие тестирования. Виды тестов. Виды тестирования программ.
41. Пять принципов чистых тестов (F.I.R.S.T. Principles). Требования к тестам
42. Модульное тестирование: понятие, целесообразность, эффект.
43. Чек-лист тестирования метода. Преимущества Unit-тестирования
44. Признаки некачественного дизайна кода. Рефакторинг
45. Добавление и выполнение методов модульного теста. Использование классов `Assert` для модульного тестирования
46. Добавление и выполнение методов модульного теста. Атрибуты `[TestClass]` и `[TestMethod]`. Классы `Assert`, `CollectionAssert`, `StringAssert`.
47. Добавление и выполнение методов модульного теста. Исключения при тестировании. Атрибут `[ExpectedException]`
48. Добавление и выполнение методов модульного теста. Инициализация и зачистка. Параметр `TestContext`. Результаты тестирования
49. Область применения модульных тестов
50. Шаблоны тестовых проектов MS Visual Studio. Инструменты и библиотеки для тестирования программ.
51. Унифицированный язык моделирования UML. Канонический набор диаграмм языка UML.
52. Особенности изображения диаграмм языка UML. Правила графического изображения диаграмм языка UML.
53. Диаграммы вариантов использования UML. Особенности построения диаграммы, отношения между актерами, отношения между вариантами использования.
54. Диаграммы классов UML. Особенности изображения классов, виды классов, отношения между классами.
55. Унифицированный язык моделирования UML. Диаграммы поведения.
56. Унифицированный язык моделирования UML. Диаграммы физической архитектуры программ.
57. Понятие базы данных (БД). Требования к БД. Свойства БД. Классификация БД. Жизненный цикл БД.
58. Системы управления базами данных (СУБД). Компоненты СУБД. Задачи, решаемые СУБД. Классификация СУБД. Примеры СУБД.
59. Модели данных. Классификация моделей данных. Критерии оценки модели данных
60. Инфологическое проектирование. Модель "сущность-связь" (ER-модель). Составляющие ER-модели. Выделение типов сущностей. Определение связей и атрибутов.

## 2. Конструирование баз данных

1. Введение в базы данных. Жизненный цикл БД
2. Системы управления базами данных. Классификация СУБД. Компоненты СУБД
3. Трехуровневая архитектура баз данных. Модели данных. Классификация моделей данных.
4. Классификация моделей данных. Представление данных с помощью модели "сущность-связь". Степень и кратность связи в ER-модели.
5. Понятие модели данных как прообраза БД. Требования к БД. Свойства БД. Классификация БД.
6. Дореляционные модели данных. Иерархическая и сетевая модель данных. Операции. Примеры.
7. Реляционная модель данных. Понятие реляционного отношения. Основные компоненты реляционного отношения. Свойства отношений.
8. Реляционная модель данных. Ключи отношений. Организация связей между таблицами. Классификация связей. Связи внутри таблицы.
9. Реляционная модель данных. Структура таблицы в реляционной базе данных. Свойства реляционных отношений.
10. Реляционная модель данных. Операции над данными в реляционной модели данных. Достоинства и недостатки реляционной модели данных.
11. Нормализация отношений. Аномалии модификации данных. Примеры. Решение проблем аномалий.
12. Теория нормальных форм. Понятие нормализации и функциональной зависимости.
13. Теория нормальных форм. 1НФ. Основные определения. Пример.
14. Теория нормальных форм. 2НФ. Основные определения. Пример.
15. Теория нормальных форм. 3НФ. Основные определения. Пример.
16. Теория нормальных форм. БКНФ. Основные определения. Пример.
17. Теория нормальных форм. 4НФ. Основные определения. Пример.
18. Теория нормальных форм. 5НФ. Основные определения. Пример.
19. Реляционная модель данных. Ограничения целостности сущностей и связей.
20. Двенадцать правил Кодда, которым должна соответствовать реляционная СУБД.
21. Реляционная алгебра. Операции обработки кортежей. Операции обработки отношений. Примеры.
22. Реляционная алгебра. Теоретико-множественные операторы обработки отношений. Примеры.
23. Реляционная алгебра. Специальные реляционные операторы обработки отношений. Примеры.
24. Реляционное исчисление. Запрос на языке реляционного исчисления. Исчисление кортежей.
25. Реляционное исчисление. Запрос на языке реляционного исчисления. Исчисление доменов.
26. Язык SQL, его структура, стандарты, история развития.
27. Подмножество языка SQL-DML. Операторы INSERT, UPDATE, DELETE: синтаксис, особенности, примеры.
28. Оператор выбора SELECT: синтаксис, предложения, примеры.
29. Подмножество языка SQL-DDL. Операторы CREATE, ALTER, DROP: синтаксис, особенности, примеры.
30. Поддержка ссылочной целостности данных.
31. Представления, их виды и способы построения. Примеры представлений.
32. Подмножество языка SQL-DCL. Объектные и системные привилегии. Операторы GRANT, REVOKE.
33. Триггеры в MS SQL Server: понятие, назначение, виды и фазы триггеров.

34. Триггеры в MS SQL Server. DML-триггеры: назначение, форма создания, параметры триггера, привила работы.
35. DML-триггеры: Обработка исключений и функции об ошибках. Примеры
36. DML-триггеры: использование таблиц deleted и inserted, примеры.
37. DML-триггеры типа FOR: виды, типовые задачи, примеры применения.
38. DML-триггеры типа AFTER: виды, типовые задачи, примеры применения.
39. DML-триггеры типа INSTEAD OF: виды, типовые задачи, примеры применения.
40. DML-триггеры и среда CLR. Реализация триггера на языке C#
41. DDL-триггеры: форма создания, виды, типовые задачи, примеры применения.
42. Logon-trigger: форма создания, виды, типовые задачи, примеры применения.
43. Управление триггерами: активация/деактивация, изменение и удаление триггера
44. СУБД PostgreSQL: история возникновения и развития, функциональность и особенности установки, настройки и применения
45. Объекты БД в СУБД PostgreSQL. Способы и особенности создания объектов БД в PostgreSQL.
46. Типы данных СУБД PostgreSQL: назначение, сравнительный анализ, особенности применения.
47. СУБД Microsoft SQL Server: история возникновения и развития.
48. Технология разработки базы данных посредством Microsoft SQL Server.
49. Программирование подключения к базе данных Microsoft SQL Server на языке C#. Параметры строки подключения. Свойства подключения.
50. Выполнение команд SQL на языке C# (объект SqlCommand).
51. Типизация результатов выполнения команд SQL на языке C# (объект SqlDataReader).
52. Параметризация запросов SQL на языке C# (объект SqlParameter). Типы параметров.
53. Работа с хранимыми процедурами на языке C#.
54. Программирование операций сохранения в БД и извлечения из БД файлов.
55. Программирование операций с наборами данных. Объекты DataSet, DataTable, DataRow, DataColumn, SqlDataAdapter.
56. Программирование операций с базой данных посредством визуального приложения.
57. Организация форм Windows. Программирование взаимодействия между формами.

## Перечень практических заданий к зачету и экзамену

### 1. Объектно-ориентированное программирование структур данных на языке C#

#### Задача 1:

1. Реализовать классом динамическую структуру для хранения и обработки целых чисел.
2. Наполнить структуру данными. Данные вводить с клавиатуры.
3. Проанализировать хранимые данные без разрушения структуры.
4. Вывести результат анализа на консоль и в файл. Имя файла вводить с клавиатуры.
5. Обеспечить вывод данных с разрушением структуры.
6. Обеспечить обработку исключений на всех этапах вычислений.
7. Разработать класс Test для модульного тестирования реализаций всех методов класса.
8. Представить листинг программы и результатное содержание консоли.

#### Задача 2:

1. Создать абстрактный класс *Array*, содержащий такие члены:
  - *N* – число элементов в массиве;

- *Arr* – указатель на массив;
  - *Init()* – метод инициализации элементов массива случайными числами;
  - *Calc()* – абстрактный метод подсчета числового показателя на основе данных массива;
  - *PrintArray()* – абстрактный метод вывода массива на экран;
  - *Processing()* – абстрактный метод, выполняющий обработку массива.
2. Создать производный класс *Vector*, реализующий заданную обработку одномерного массива.
  3. Создать производный класс *Matrix*, реализующий заданную обработку двумерного массива.
  4. Обеспечить обработку исключений на всех этапах вычислений.
  5. Разработать класс *Test* для модульного тестирования реализаций методов *Calc*.
  6. Представить листинг программы и результатное содержание консоли.

### Задача 3:

1. Составить программу для обработки структурированных данных с помощью классов-коллекций. В программе:
  - определить базовый класс как абстрактный;
  - ввести в абстрактный класс дополнительные абстрактные методы для изменения значений полей класса.
  - в производных классах переопределить наследуемый метод для специализированной операции изменения полей класса.
2. Создать класс по принципу «включение-делегирование», содержащий коллекцию производных от базового абстрактного класса объектов. Описать следующие методы:
  - добавления в коллекцию новых объектов;
  - изменения свойств и полей имеющихся объектов по номеру объекта в коллекции;
  - удаление из коллекции объекта по его номеру;
  - вывода информации об имеющихся в коллекции объектах.
3. Обеспечить обработку исключений на всех этапах вычислений.
4. Разработать модульный тест для проверки работоспособности метода обработки данных.
5. Представить листинг программы и результатное содержание консоли.

## 2. Конструирование баз данных

1. **[10 баллов]** Для заданной предметной области разработать инфологическую модель данных (ER-diagram).
2. **[10 баллов]** Для заданной предметной области разработать физическую модель реляционной БД. Провести ее доказательную нормализацию не менее чем в 3НФ.
3. **[10 баллов]** На языке SQL написать команды для построения структуры БД
4. **[10 баллов]** На языке SQL написать команды для управления данными БД каждой таблицы:
  - добавление данных (по 2–3 записи);
  - изменение всех полей данных записи, идентифицированной значением первичного ключа;
  - удаление записи, идентифицированной значением первичного ключа.
5. **[10 баллов]** На языке SQL написать команды для создания 1-2 триггеров для обеспечения целостности данных. Разработать триггер для архивирования манипуляций с оперативными данными.

6. **[10 баллов]** На языке SQL написать сложный запрос, реализующий реляционные операции соединения, выборки, проекции и переименования, а также одну или несколько групповых операций.
7. **[10 баллов]** На языке C# написать код для выполнения разработанных команд SQL и вывода на консоль результатов их выполнения.

Возможные варианты практического задания:

- |               |               |
|---------------|---------------|
| 1, 2, 3, 5, 7 | 2, 3, 4, 5, 7 |
| 1, 2, 4, 5, 7 | 2, 3, 5, 6, 7 |
| 1, 2, 5, 6, 7 | 2, 4, 5, 6, 7 |

Примерные варианты предметной области:

1. Обменный пункт: сотрудники пункта, виды валют, курсы валют, операции обмена.
2. Ювелирный магазин: названия изделий, комитенты (кто сдал изделия на комиссию), журнал сдачи изделий на продажу, журнал покупки изделий.
3. Поликлиника: врачи, пациенты, виды болезней, журнал приёма пациентов.

### Примеры типовых тестовых заданий

#### III семестр. Объектно-ориентированное программирование структур данных на языке C#

Разрешено попыток: 2

Ограничение по времени: 1 ч.

Метод оценивания: Средняя оценка

№	Задания:	Варианты ответы:	Ответы:
	Укажите корректную реализацию класса:	<p>a.     using System;  namespace Lab_1  {  class Program  {  static void Main(string args)  {  // Вывод предложения на экран  System.Console.WriteLine("Привет,  C#!\n");  Console.ReadKey();  }  }  }</p> <p>b.     sing System;  namespace Lab_34  {  class Progr  {  static void Main(string[] args)  {  // Вывод предложения на экран  System.Console.WriteLine("Привет,  C#!\n");</p>	a

	<pre>         Console.ReadKey();     } } }  c.    using System32; namespace Lab_21{     class Proga      {         static void Main(string[] args)         {             // Вывод предложения на экран             System.WriteLine("Привет, С#!\n");             Console.Key();         }     } } </pre>	
Укажите тип системного типа System.UInt64:	<ul style="list-style-type: none"> <li>a. Float</li> <li>b. UInt</li> <li>c. Ulong</li> <li>d. Ushort</li> <li>e. Double</li> </ul>	c
Укажите корректное неявное преобразование без потери данных:	<ul style="list-style-type: none"> <li>a. short Code1 = 7; byte Code2; Code2 = Code1;</li> <li>b. short Code1 = 7; long Code2; Code2 = Code1;</li> <li>c. int Code1 = 7; uint Code2; Code2 = Code1;</li> </ul>	b
Укажите верную идентификацию типа bool:	<ul style="list-style-type: none"> <li>a. boll a = 0;</li> <li>b. bool a = false;</li> <li>c. bool a = not;</li> <li>d. bool a = y;</li> </ul>	b
Что выведет данная программа? int a = 35; a/=5+7; a+=3; if(a<5)Console.WriteLine("YES"); else Console.WriteLine("NO");	<ul style="list-style-type: none"> <li>a. YES</li> <li>b. NO</li> <li>c. Нет верного ответа</li> </ul>	b
Виды тестов:	<ul style="list-style-type: none"> <li>a. Ручной</li> <li>b. Самостоятельный</li> <li>c. Автоматический</li> <li>d. Регулируемый</li> </ul>	a,c
Виды тестирования:	<ul style="list-style-type: none"> <li>a. Модульное</li> <li>b. Постепенного спуска</li> </ul>	a,c,e,h



	<ul style="list-style-type: none"> <li>c. Стрессовое</li> <li>d. Блочное</li> <li>e. Регрессивное</li> <li>f. Резонирующее</li> <li>g. Объектно-ориентированное</li> <li>h. Системное</li> </ul>	
Цена ошибки в ходе разработки и поддержании ПО:	<ul style="list-style-type: none"> <li>a. Не изменяется при несвоевременном их обнаружении.</li> <li>b. Экспоненциально уменьшается при несвоевременном их обнаружении.</li> <li>c. Экспоненциально возрастает при несвоевременном их обнаружении.</li> </ul>	c
Некачественный дизайн кода определяется по ряду признаков. Каких?	<ul style="list-style-type: none"> <li>a. Ненужная сложность</li> <li>b. Прозрачность</li> <li>c. Жесткость</li> <li>d. Читательность</li> <li>e. Чрезмерная оригинальность</li> </ul>	a,c
Методы утверждения в модульных тестах:	<ul style="list-style-type: none"> <li>a. Assert.AreSame</li> <li>b. Assert.IsBetter</li> <li>c. Assert.IsNull</li> <li>d. Assert.IsFalse</li> </ul>	a, d
В случае неудачного завершения теста возникает исключение:	<ul style="list-style-type: none"> <li>a. AssertInconclusiveException</li> <li>b. AssertFailedException</li> <li>c. Assert.ThrowsException</li> <li>d. Assert.Inconclusive</li> </ul>	b
Для улучшения качества кода программных приложений применяют:	<ul style="list-style-type: none"> <li>a. Реконструкцию</li> <li>b. Ребилдинг</li> <li>c. Рефакторинг +</li> <li>d. Реформацию</li> <li>e. Реабилитацию</li> </ul>	c

Полная версия теста расположена в ЭИОС по ссылке: <https://openedu.kubsu.ru/mod/quiz/view.php?id=24648>

#### IV семестр. Конструирование баз данных

Разрешено попыток: 2

Ограничение по времени: 1 ч.

Метод оценивания: Средняя оценка

№	Задания:	Варианты ответы:	Ответы:
	Укажите уровни системы управления базами данных. Выберите один или несколько ответов	<ul style="list-style-type: none"> <li>1. Уровень абстракции</li> <li>2. Физический уровень</li> <li>3. Уровень внешних моделей</li> <li>4. Программный уровень</li> <li>5. Концептуальный уровень</li> <li>6. Уровень представления</li> <li>7. Системный уровень</li> </ul>	3, 4, 5
	Внутренний уровень архитектуры СУБД...	<ul style="list-style-type: none"> <li>1. Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации</li> <li>2. Наиболее близок к пользователю, описывает способ размещения данных на устройствах хранения информации</li> <li>3. Наиболее близок к пользователю, описывает обобщенное представление данных</li> </ul>	1

	4. Наиболее близок к физическому, описывает способ размещения данных в логической структуре базы данных 5. Нет правильного ответа	
В каком порядке должны следовать уровни проектирования БД?	1. Концептуальный – физический - логический 2. Физический – логический - концептуальный 3. Концептуальный – логический - физический 4. Внешний – физический- концептуальный	3
Операция формирования нового отношения К степени $k_1+k_2$ , содержащего все возможные сочетания кортежей отношений $K_1$ степени $k_1$ и $K_2$ степени $k_2$ , называется	1. Объединением 2. Вычитанием 3. Произведением 4. Пересечением 5. Соединением	3
Если существует функциональная зависимость неключевого атрибута от составного ключа, то говорят, что существует	1. Многозначная зависимость 2. Полная функциональная зависимость 3. Частичная функциональная зависимость 4. Транзитивная зависимость 5. Функциональная взаимозависимость	2
Операторы IN, BETWEEN, LIKE относятся к	1. Реляционным операторам 2. Логическим операторам 3. Специальным операторам 4. Агрегатным функциям 5. Нет правильного варианта	3
Что понимается под некоторой абстракцией, которая, позволяет пользователям и разработчикам трактовать данные, как сведения, содержащие не только данные, но и взаимосвязь между ними?	Открытый вопрос	Модель данных
Укажите число известных нормальных форм:	Открытый вопрос	6

Полная версия теста расположена в ЭИОС по ссылке: <https://openedu.kubsu.ru/mod/quiz/view.php?id=27048>

## Критерии оценивания на зачете

Оценка “зачтено” - Практические задания выполнены в срок в объеме не менее 80%. Студент демонстрирует правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при аргументации ответов на вопросы при защите лабораторных.

Оценка «не зачтено» - Практические задания не выполнены либо предоставлены не в срок в объеме менее 60%, Студент демонстрирует наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и

инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа. Для лиц с нарушениями слуха:
- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

## Критерии оценивания на экзамене

### 1. Объектно-ориентированное программирование структур данных на языке C#

Итоговая оценка за экзамен [0, 100]:

$$E = \frac{T + P + K}{2}$$

**T** – оценка за теорию [0, 50];

- 0-10 – материал отсутствует, либо изложен скудно, во многом ошибочен, в основном **не соответствует сути заданного вопроса**;
- 10-20 – материал изложен фрагментарно, целостность теории прослеживается с трудом, фиксируется множество ошибочных ответов, либо **не соответствующих сути заданного вопроса**;
- 20-30 – ответы не полные, местами ошибочные, в целом прослеживается целостность теории;
- 30-40 – материал изложен в полной мере на уровне курса лекций (конспект, презентации);
- 40-50 – даны развернутые ответы, демонстрирующие глубокие знания вопроса.

**P** – оценка за задачу [0, 50];

0-10 баллов за каждое из 5 заданий.

За каждое задание:

- 0 – отсутствует;
- 1-2 – содержит грубые логические и **синтаксические** ошибки;
- 3-5 – содержит мелкие логические и **синтаксические** ошибки;
- 6-8 – содержит мелкие логические ошибки;
- 9-10 – работает корректно.

**K** – оценка за коллоквиум [0, 100];

100-балльная оценка E	4-хбалльная оценка за экзамен
-----------------------	-------------------------------

90–100	"Отлично"
75–89	"Хорошо"
60–74	"Удовлетворительно"
0–59	"Неудовлетворительно"

## 2. Конструирование баз данных

Итоговая оценка за экзамен [0, 100]:

$$E = \frac{T + P + K}{2}$$

**T** – оценка за теорию [0, 50];

- 0-10 – материал отсутствует, либо изложен скудно, во многом ошибочен, в основном не соответствует сути заданного вопроса;
- 10-20 – материал изложен фрагментарно, целостность теории прослеживается с трудом, фиксируется множество ошибочных ответов, либо не соответствующих сути заданного вопроса;
- 20-30 – ответы не полные, местами ошибочные, в целом прослеживается целостность теории;
- 30-40 – материал изложен в полной мере на уровне курса лекций (конспект, презентации);
- 40-50 – даны развернутые ответы, демонстрирующие глубокие знания вопроса.

**P** – оценка за задачу [0, 50];

Задания 1, 3, 5	Задания 2, 6	Критерий
0	0	выполнение задания отсутствует
1–2	1–3	содержит грубые логические и синтаксические ошибки
3–5	4–7	содержит мелкие логические и синтаксические ошибки
6–8	8–12	содержит мелкие логические ошибки
9–10	13–15	задание выполнено корректно

**K** – оценка за коллоквиум [0, 100];

100-балльная оценка E	4-хбалльная оценка за экзамен
90–100	"Отлично"
75–89	"Хорошо"
60–74	"Удовлетворительно"
0–59	"Неудовлетворительно"

## 5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

### 5.1 Основная литература:

1. Полное руководство по языку программирования C# 8.0 и платформе .NET Core 3. URL: <https://metanit.com/sharp/tutorial/>.
2. Мацкевич А.Г. Лекции по курсу: Информационные технологии с изложением основ программирования на языке C#. Ч. 1 : учеб. пособие. – М.: МТУСИ, 2016.

3. Самохвалов Э.Н. Введение в проектирование и разработку приложений на языке программирования С# : учебное пособие / Э.Н. Самохвалов, Г.И. Ревунков, Ю.Е. Гапанюк. – М.: МГТУ им. Н.Э. Баумана, 2018.
4. Павловская Т.А. Программирование на языке высокого уровня С# : учебное пособие. 3-е изд. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021.
5. Горелов С.В. Современные технологии программирования: разработка Windows-приложений на языке С#. В 2 томах. Т.1 : учебник ; под ред. П. Б. Лукьянова. – М.: Прометей, 2019.
6. Горелов С.В. Современные технологии программирования: разработка Windows-приложений на языке С#. В 2 томах. Т.2 : учебник ; под ред. П. Б. Лукьянова. – М.: Прометей, 2019.

## 5.2 Дополнительная литература:

7. Мурадханов С.Э., Широков А.И. Информатика и программирование: объектно-ориентированное программирование (на основе языка С#) : учебник. – М.: Издательский Дом МИСиС, 2015.
8. Синюк В.Г., Рязанов Ю.Д. Алгоритмы и структуры данных : лабораторный практикум. Учеб. пособие. – Белгород : Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2013.
9. Назаренко, П.А. Алгоритмы и структуры данных : Учеб. пособие. – Самара : Поволжский государственный университет телекоммуникаций и информатики, 2015.
10. Батищев Р.В. Структуры и алгоритмы обработки данных. Часть 1 : Учеб. Пособие. – Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2014.
11. Бишоп Дж., Хорспул Н. С# в кратком изложении; Пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2005.
12. Рихтер Дж. CLR via С#. Программирование на платформе Microsoft .NET, 2008.
13. Framework 2.0 на языке С#. Мастер-класс. / Пер. с англ. – М.: Издательство «Русская Редакция» : СПб.: Питер , 2007.
14. Шилдг Герберт. Полный справочник по С#. : Пер. с англ. – М. : Издательский дом "Вильямс", 2004.
15. Джесс Либерти. Программирование на С# : Создание .NET приложений. 2-е изд. / Пер. с англ. – М.: Издательство «Символ-Плюс» : СПб.: Питер , 2012.
16. ГОСТ 34. «Информационная технология. Комплекс стандартов».
17. ГОСТ 19. «Единая система программной документации».
18. Дейтел Х. С#: Пер. с англ. / Дейтел Х., Дейтел П., Листфилд Дж., Нието Т., Мегер Ш., Златкина М., СПб.: БХВ-Петербург, 2006.
19. Троелсен Э. Язык программирования С# 2005 и платформа .NET 2.0, 3-е издание. : Пер. с англ. – М.: ООО "И.Д. Вильямс", 2007.

## 5.3. Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

## 5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

*Электронно-библиотечные системы (ЭБС):*

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>

4. ЭБС «ZNANIUM.COM» [www.znanium.com](http://www.znanium.com)

5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

#### *Профессиональные базы данных*

1. Scopus <http://www.scopus.com/>

2. ScienceDirect <https://www.sciencedirect.com/>

3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>

4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>

5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>

6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>

7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>

8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>

9. Springer Journals: <https://link.springer.com/>

10. Springer Journals Archive: <https://link.springer.com/>

11. Nature Journals: <https://www.nature.com/>

12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>

13. Springer Materials: <http://materials.springer.com/>

14. Nano Database: <https://nano.nature.com/>

15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>

16. "Лекториум ТВ" <http://www.lektorium.tv/>

17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

#### *Информационные справочные системы*

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

#### *Ресурсы свободного доступа*

1. КиберЛенинка <http://cyberleninka.ru/>;

2. Американская патентная база данных <http://www.uspto.gov/patft/>

3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;

4. Федеральный портал "Российское образование" <http://www.edu.ru/>;

5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;

6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .

7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;

8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;

9. Служба тематических толковых словарей <http://www.glossary.ru/>;

10. Словари и энциклопедии <http://dic.academic.ru/>;

11. Образовательный портал "Учеба" <http://www.ucheba.com/>;

12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы [http://xn--273--84d1f.xn--p1ai/voprosy\\_i\\_otvety](http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety)

#### *Собственные электронные образовательные и информационные ресурсы КубГУ*

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>

2. Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>

3. Среда модульного динамического обучения <http://moodle.kubsu.ru>

4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

## **6. Методические указания для обучающихся по освоению дисциплины**

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, лабораторных работ, контрольной работы, зачета и экзамена.

Важнейшим этапом курса является самостоятельная работа по дисциплине с использованием указанных литературных источников и методических указаний автора курса.

Виды и формы СР, сроки выполнения, формы контроля приведены выше в данном документе.

Для лучшего освоения дисциплины при защите ЛР студент должен ответить на несколько вопросов из лекционной части курса.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

## **7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине**

### **7.1 Перечень информационных технологий**

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекций и практических занятий.

### **7.2 Перечень необходимого программного обеспечения**

1. OS Windows, MS Office
2. Microsoft Visual Studio Community.
3. СУБД PostgreSQL
4. СУБД MS SQL Server Developer Edition.
5. Lucidchart – онлайн-платформа.
6. Star UML.
7. Антивирус.

## **8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа (ауд. 129, 131, А305.)	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер	PowerPoint.
Учебные аудитории для проведения	Мебель: учебная мебель	Аудитория, (кабинет) –

занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации (ауд. 147,148)	Технические средства обучения: экран, проектор, компьютер	компьютерный класс
Учебные аудитории для проведения лабораторных работ. Лаборатория 102,105,106	Мебель: учебная мебель Технические средства обучения: компьютер	Лаборатория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	OS Windows, MS Office Microsoft Visual Studio Community. СУБД MS SQL Server Developer Edition. СУБД PostgreSQL Lucidchart – онлайн-платформа. Star UML. Антивирус.
Помещение для самостоятельной работы обучающихся (ауд. 105, 148,150)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	OS Windows, MS Office Microsoft Visual Studio Community. СУБД PostgreSQL СУБД MS SQL Server Developer Edition. Lucidchart – онлайн-платформа. Star UML. Антивирус.