

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шавалович

Должность: Ректор

Дата подписания: 16.10.2023 14:59:02

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f9fa4304cc

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**имени академика М.Д. Миллионщикова**

**«УТВЕРЖДАЮ»**

**Первый проректор**

**И.Г. Гаирабеков**



**РАБОЧАЯ ПРОГРАММА**

**дисциплины**

**«Программирование»**

**Направление подготовки**

**09.03.03 Прикладная информатика**

**Направленность (профиль)**

**«Прикладная информатика в экономике»**

**Квалификация**

**бакалавр**

**Год начала подготовки – 2021**

**Грозный – 2021**

## 1. Цели и задачи дисциплины

Цель преподавания дисциплины «Программирование» состоит в освоении студентами, получающими квалификацию бакалавра, подходов к составлению алгоритмов решения задач различной степени сложности, а также овладении навыками создания программ на примере среды разработки приложений Microsoft Visual Studio.

Задачами дисциплины являются: изучение основополагающих понятий и правил программирования; разработка алгоритмов обработки данных различной структуры, освоение правил создания и организации пользовательского интерфейса.

## 2. Место дисциплины в структуре образовательной программы

Учебная дисциплина «Программирование» относится к обязательной части ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика (квалификация «бакалавр»).

Для освоения дисциплины «Программирование» студент должен обладать знаниями и умениями, приобретенными в результате освоения предшествующих дисциплин:

- информатика;
- теория информации.

Дисциплина «Программирование» является предшествующей и необходимой для изучения следующих дисциплин:

- разработка и стандартизация программного обеспечения;
- программирование в экономических системах;
- технологии разработки программных приложений.

## 3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУВ)
<b>Общепрофессиональные</b>		
<b>ОПК-1.</b> Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	<b>ОПК-1.1.</b> Применяет основы математики, вычислительной техники и программирования в профессиональной деятельности. <b>ОПК-1.2.</b> Решает стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования. <b>ОПК-1.3.</b> Использует методы теоретического и	<b>Знать:</b> методы и приемы решения практических задач в профессиональной деятельности с помощью информационных систем. <b>Уметь:</b> решать базовые задачи обработки данных в профессиональной деятельности. <b>Владеть:</b> общей подготовкой для решения практических задач в области информационных технологий.

	экспериментального исследования объектов 19 профессиональной деятельности	
<b>ОПК-2.</b> Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	<b>ОПК-2.1.</b> Выбирает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. <b>ОПК-2.2.</b> Применяет современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	<b>Знать:</b> методы автоматизации операций обработки информации с помощью информационных технологий. <b>Уметь:</b> осуществлять постановку задач и использовать различные алгоритмы обработки информации. <b>Владеть:</b> навыками работы с программными средствами, осуществляющими обработку информации.
<b>ОПК-7.</b> Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<b>ОПК-7.1.</b> Применяет языки программирования и языки работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. <b>ОПК-7.2.</b> Программирует, выполняет отладку и тестирование прототипов программно-технических комплексов задач	<b>Знать:</b> принципы, базовые концепции технологий программирования; основные этапы и принципы создания программного продукта. <b>Уметь:</b> составлять алгоритмы решения задач различной структуры и оформлять их в соответствии с синтаксическими правилами языка программирования; разрабатывать и тестировать программные компоненты информационных систем. <b>Владеть:</b> приемами разработки пользовательского интерфейса приложения, обеспечивающего оптимальное функционирование программы и удобство пользователя; приемами отладки приложений, поиска ошибок и обработки исключений.

#### 4. Объем дисциплины и виды учебной работы

Таблица 2

Вид учебной работы	Всего часов/ зач. ед.		Семестры						
	ОФО	ЗФО	3	4	5	3	4	5	
			ОФО			ЗФО			
<b>Контактная работа (всего)</b>	<b>200/5,8</b>	<b>56/1,6</b>	<b>68/2</b>	<b>64/1,8</b>	<b>68/2</b>	<b>20/0,56</b>	<b>16/0,48</b>	<b>20/0,56</b>	
В том числе:									
Лекции	100/2,9	28/0,8	34/1	32/0,9	34/1	10/0,28	8/0,24	10/0,28	
Практические занятия	-	-	-	-	-	-	-	-	
Практическая подготовка	-	-	-	-	-	-	-	-	
Лабораторные работы	100/2,9	28/0,8	34/1	32/0,9	34/1	10/0,28	8/0,24	10/0,28	
<b>Самостоятельная работа (всего)</b>	<b>160/4,2</b>	<b>304/8,4</b>	<b>51/1,5</b>	<b>48/1,2</b>	<b>61/1,5</b>	<b>92/2,5</b>	<b>108/3</b>	<b>104/2,9</b>	
В том числе:									
Курсовая работа (проект)	-	-	-	-	-	-	-	-	
Расчетно-графические работы	-	-	-	-	-	-	-	-	
ИТР	-	-	-	-	-	-	-	-	
Рефераты	-	-	-	-	-	-	-	-	
Доклады с презентациями	50/1,4	90/2,5	17/0,5	16/0,4	17/0,5	36/1	36/1	18/0,5	
<i>И (или) другие виды самостоятельной работы:</i>									
Подготовка к лабораторным работам	60/1,4	106/2,9	17/0,5	16/0,4	27/0,5	20/0,5	36/1	50/1,4	
Подготовка к практическим занятиям	-	-	-	-	-	-	-	-	
Подготовка к зачету	33/0,9	72/2	17/0,5	16/0,4	-	36/1	36/1	-	
Подготовка к экзамену	17/0,5	36/1	-	-	17/0,5	-	-	36/1	
<b>Вид отчетности</b>			<b>зач.</b>	<b>зач.</b>	<b>экз.</b>	<b>зач.</b>	<b>зач.</b>	<b>экз.</b>	
<b>Общая трудоемкость дисциплины</b>	<b>ВСЕГО в часах</b>	<b>360</b>	<b>360</b>	<b>360</b>			<b>360</b>		
	<b>ВСЕГО в зач. единицах</b>	<b>10</b>	<b>10</b>	<b>10</b>			<b>10</b>		

#### 5. Содержание дисциплины

##### 5.1. Разделы дисциплины и виды занятий

Таблица 3

№ п/п	Наименование раздела дисциплины по семестрам	Часы лекционных занятий		Часы лабораторных занятий		Часы практических (семинарских) занятий		Всего часов	
		ОФО	ЗФО	ОФО	ЗФО	ОФО	ЗФО	ОФО	ЗФО
<b>3 семестр</b>									
1.	Основы алгоритмизации и программирования	20	4	10	4	-	-	30	8
2.	Основы синтаксиса языка программирования C#	14	6	24	6	-	-	38	12
<b>4 семестр</b>									

3.	Обработка данных в C#	8	4	26	6	-	-	34	10
4.	Программные компоненты: разработка, отладка и тестирование	24	4	6	2	-	-	30	6
<b>5 семестр</b>									
5.	Программирование приложений на языке C#	16	4	16	4	-	-	32	8
6.	Работа с источниками данных	18	6	18	6			36	12

## 5.2. Лекционные занятия

Таблица 4

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Основы алгоритмизации и программирования	<p>Понятие алгоритма. Формы представления алгоритмов. Блок-схемы, используемые в графическом представлении. Свойства алгоритмов.</p> <p>Основные этапы программирования решения задачи.</p> <p>Языки программирования низкого и высокого уровня. Трансляторы.</p> <p>Основные понятия ООП.</p> <p>Модели предметной области в процедурном и объектно-ориентированном программировании.</p> <p>Жизненный цикл программы.</p> <p>Понятие системы и среды программирования.</p> <p>Особенности современных сред разработки приложений (IDE).</p>
2.	Основы синтаксиса языка программирования C#	<p>Типы данных в Visual C#. Объявление переменных. Инкремент и декремент. Операторы сравнения и логические операторы.</p> <p>Условные ветвления в C# (операторы if и switch).</p> <p>Реализация классических и итерационных циклов в C# (операторы for и while).</p> <p>Модальные и немодальные диалоговые окна. Окна вывода сообщений MessageBox.</p>
3.	Обработка данных в C#	<p>Работа с массивами в C#: объявление массивов, основные операции с массивами. Одномерные и двумерные массивы.</p> <p>Обработка строк текста в C#. Основные функции обработки строк.</p> <p>Основные функции обработки даты и времени в C#.</p>

4.	Программные компоненты: разработка, отладка и тестирование	<p>Понятие и назначение отладки программы. Виды ошибок в программе. Понятие трассировки. Точки останова. Ошибки времени выполнения программы. Обработка исключений. Инструкция try...catch.</p> <p>Понятие, назначение и виды библиотек в программировании.</p> <p>Понятие и механизм работы API-функций.</p> <p>Типизация данных в программировании.</p> <p>Предмет и цель программной инженерии как области знаний. Основные принципы разработки программных продуктов.</p> <p>Уровни требований к программным продуктам.</p> <p>Понятие и основные принципы конфигурационного управления.</p> <p>Виды тестирования программного обеспечения. Верификация и валидация.</p> <p>Разработка приложений для различных областей применения. Стил и методы программирования.</p>
5.	Программирование приложений на языке C#	<p>Модульное программирование: пользовательские процедуры и функции.</p> <p>Графика в программировании. Карандаш и кисть.</p> <p>Методы рисования графических примитивов: прямоугольников, эллипсов, многоугольников.</p> <p>Понятие и состав интегрированной среды разработки приложений (IDE). Примеры и краткая характеристика популярных сред разработки.</p>
6.	Работа с источниками данных	<p>Работа с файлами и каталогами в C#: основные методы.</p> <p>Работа с базами данных в C#. Технология ADO.NET: основные используемые объекты.</p> <p>Взаимодействие приложения на языке C# с MS Excel.</p> <p>Делегаты в C#.</p>

### 5.3. Лабораторные занятия

Таблица 5

№ п/п	Наименование раздела дисциплины	Наименование лабораторных работ
1.	Основы алгоритмизации и программирования	<p>Графическая форма представления алгоритмов. Блок-схемы.</p> <p>Составление алгоритмов решения линейных задач.</p> <p>Алгоритмы разветвляющейся структуры (условия).</p> <p>Решение задач с условиями.</p> <p>Алгоритмы циклической структуры. Решение задач с циклами.</p>

2.	Основы синтаксиса языка программирования C#	<p>Знакомство со средой разработки приложений Microsoft Visual Studio. Приемы создания интерфейса приложения. Элементы управления, размещаемые на форме.</p> <p>Переменные и типы данных в Visual C#. Основы синтаксиса языка.</p> <p>Разработка линейных программ на языке C#.</p> <p>Алгоритмы с условными ветвлениями. Конструкция if.</p> <p>Алгоритмы с условными ветвлениями. Конструкция switch.</p> <p>Программирование циклов в языке C#: алгоритмы с повторяющимися действиями.</p> <p>Реализация классических и итерационных циклов в C#.</p> <p>Использование окон вывода сообщений MessageBox.</p>
3.	Обработка данных в C#	<p>Массивы. Типичные операции при работе с массивами.</p> <p>Использование массивов для хранения входных данных и результатов вычислений.</p> <p>Обработка текстовой информации в C#. Функции обработки строк.</p> <p>Автоматизация операций по обработке текста.</p> <p>Работа с датами и временем.</p>
4.	Программные компоненты: разработка, отладка и тестирование	<p>Методы отладки приложения. Трассировка, точки останова.</p> <p>Обработка исключений, инструкция try...catch.</p>
5.	Программирование приложений на языке C#	<p>Модульное программирование. Разработка пользовательских функций и процедур.</p> <p>Графика: рисование в программировании. Создание графических примитивов.</p> <p>Разработка приложений для различных областей применения.</p>
6.	Работа с источниками данных	<p>Работа с файлами. Использование файлов для обмена данными с приложением. Работа с каталогами как объектами файловой системы.</p> <p>Связь приложений на C# с базами данных.</p> <p>Взаимодействие приложений на C# с базами данных с помощью запросов на языке SQL.</p> <p>Взаимодействие приложения на языке C# с MS Excel.</p>

#### 5.4. Практические (семинарские) занятия: нет

Таблица 6

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	-	-

#### 6. Самостоятельная работа студентов по дисциплине

Способ организации самостоятельной работы: подготовка презентации на 12-15 слайдов с устным докладом по заданной тематике; разработка приложений по заданию преподавателя.

3 семестр

Описание одного языка программирования (на выбор студента): его особенности, достоинства и недостатки, сферы применения, история развития.

Pascal	Delphi	Python
Algol	Java	Swift
Fortran	Objective-C	Visual Basic
COBOL	Perl	LISP
C++	PHP	C#

#### **4 семестр**

Тематика докладов с презентациями:

1. Сферы применения ЯП: программирования для локальных ПК, для мобильных устройств, для сети Интернет.
2. Современные среды программирования: сравнительный анализ.
3. Язык программирования C++: история возникновения, особенности, наследие.
4. Язык программирования C#: история возникновения, особенности, перспективы.
5. Язык программирования Java: история возникновения, характерные особенности, сферы применения.
6. Логическая парадигма программирования: основные понятия, характерные особенности, области применения, достоинства и недостатки.
7. Функциональная парадигма программирования: основные понятия, характерные особенности, области применения, достоинства и недостатки.
8. Сравнительный анализ функционального, логического и процедурного подхода в программировании.
9. Понятие и назначение библиотек в программировании.
10. Парадигма объектно-ориентированного программирования: основные понятия, характерные особенности, области применения, достоинства и недостатки.
11. Инкапсуляция, наследование и полиморфизм в ООП.
12. Среда программирования Visual Studio.
13. Основные принципы технологии Microsoft .NET.
14. Методы тестирования программного обеспечения. Классификация ошибок в программном обеспечении
15. Правила и подходы к построению интерфейса пользователя.

#### **5 семестр**

Тематика докладов с презентациями:

1. Создание веб-страниц и программирование для Internet.
2. Язык разметки веб-страниц HTML.
3. Язык PHP: особенности, характеристики и сферы применения.
4. Язык XML и обширные возможности его применения. Стандарты XML.
5. ОС Android: история возникновения, особенности развития. Разработка приложений для Android.
6. Язык запросов SQL: стандарты, функции, особенности. Связь и взаимодействие приложений с базами данных.
7. Приемы отладки программ и обработки исключений. Оптимизация приложений.
8. Групповая разработка проекта.
9. Создание инсталляционного пакета приложения.



10. Управление данными. Технология ADO.NET.
11. Создание интернет-приложений. Технология ASP.NET.
12. Классы в ООП: поля, методы и свойства.
13. Создание и удаление классов и экземпляров классов.
14. Понятие и состав интегрированной среды разработки приложений (IDE).

Примеры популярных сред разработки.

15. Краткая характеристика основных современных сред разработки приложений.

### **Примеры заданий**

Программа-калькулятор. Разработать программу для выполнения простейших арифметических действий с числами – сложения, вычитания, умножения, деления. Интерфейс программы продумать максимально приближенным к классическому калькулятору.

Программа-лотерея. Разработать игру, в которой компьютер генерирует случайное число в диапазоне от 0 до 100. Четыре игрока делают ставки, пытаясь угадать число. Выигрывает тот, чья ставка оказалась ближе всего к «загаданному» компьютером числу.

Программа «Тренажер памяти». Компьютер генерирует случайную последовательность из 7 цифр. Игроку дается 4 секунды на запоминание, затем компьютер скрывает последовательность. Задача игрока – воспроизвести комбинацию цифр. Игра включает 10 заданий, возможно усложнение условий.

### **Учебно-методическое обеспечение для самостоятельной работы студентов:**

1. Котов, О.М. Язык С#. Краткое описание и введение в технологии программирования [Электронный ресурс]: учебное пособие / О.М. Котов. — Электрон. текстовые данные. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014. – 208 с. – Режим доступа: <http://www.iprbookshop.ru/68524.html> (ЭБС «IPRbooks»).

2. Биллиг, В.А. Основы программирования на С# [Электронный ресурс] / В. А. Биллиг. – 2-е изд. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 574 с. – Режим доступа: <http://www.iprbookshop.ru/73695.html> (ЭБС «IPRbooks»).

## **7. Оценочные средства**

### **7.1. Вопросы к рубежным аттестациям**

#### **3 семестр**

*К 1-ой рубежной аттестации:*

1. Понятие алгоритма. Формы представления алгоритмов. Блок-схемы, используемые в графическом представлении.
2. Свойства алгоритмов.
3. Основные этапы программирования решения задачи.
4. Типы данных в Visual C#. Объявление переменных. Инкремент и декремент.

Операторы сравнения и логические операторы.

5. Условные ветвления в С# (операторы if и switch).
6. Рассказать о некоторых основных компонентах, применяемых для построения интерфейса в среде Visual Studio (button, label, textBox, comboBox, radioButton, checkBox).

*К 2-ой рубежной аттестации:*

1. Реализация классических и итерационных циклов в C# (операторы for и while).
2. Модальные и немодальные диалоговые окна. Окна вывода сообщений MessageBox.
3. Языки программирования низкого и высокого уровня. Трансляторы.
4. Основные понятия ООП: объект, класс, инкапсуляция, наследование, полиморфизм.
5. Основные понятия ООП: свойство, метод, событие, оператор, переменная.
6. Модели предметной области и парадигмы программирования.
7. Процедурно-ориентированное программирование.
8. Диаграммы UML в объектно-ориентированном программировании.

#### **4 семестр**

*К 1-ой рубежной аттестации:*

1. Работа с массивами в C#: объявление массивов, основные операции с массивами. Одномерные и двумерные массивы.
2. Обработка строк текста в C#. Основные функции обработки строк: Length, IndexOf, Trim, Remove, ToUpper.
3. Обработка строк текста в C#. Основные функции обработки строк: LastIndexOf, Substring, Insert, Replace, ToLower.
4. Основные функции обработки даты и времени в C#.
5. Понятие и назначение отладки программы. Виды ошибок в программе. Понятие трассировки. Точки останова.
6. Ошибки времени выполнения программы. Обработка исключений. Инструкция try...catch.
7. Понятие, назначение и виды библиотек в программировании.
8. Понятие и механизм работы API-функций.

*Ко 2-ой рубежной аттестации:*

1. Типизация данных в программировании.
2. Предмет и цель программной инженерии как области знаний. Основные принципы разработки программных продуктов.
3. Уровни требований к программным продуктам.
4. Понятие и основные принципы конфигурационного управления.
5. Виды тестирования программного обеспечения. Верификация и валидация.
6. Разработка приложений для различных областей применения. Стиль и методы программирования.

#### **5 семестр**

*К 1-ой рубежной аттестации:*

1. Модульное программирование: пользовательские процедуры и функции в C#.
2. Графика в программировании. Карандаш и кисть.
3. Методы рисования графических примитивов: прямоугольников, эллипсов, многоугольников.
4. Основные методы работы с файлами. Использование файлов для обмена данными с приложением.
5. Работа с каталогами как объектами файловой системы: основные методы.

*Ко 2-ой рубежной аттестации:*

1. Работа с базами данных в C#.
2. Технология ADO.NET: основные используемые объекты.

3. Взаимодействие приложения на языке C# с MS Excel.
4. Понятие и состав интегрированной среды разработки приложений (IDE).

Примеры популярных сред разработки.

5. Краткая характеристика основных современных сред разработки приложений.
6. Делегаты в C#.

Помимо проверки знания теоретического материала, на аттестации / экзамене студентам предлагаются практические задания по разделам дисциплины.

Образец билетов рубежной аттестации:

<b>Грозненский Государственный Нефтяной Технический Университет</b> им. акад. М.Д. Миллионщикова <b>Кафедра «Информационные технологии»</b> Дисциплина «Программирование» 1-я рубежная аттестация	
<b>Группа:</b>	<b>Семестр: 3</b>
<b>Билет №</b>	
1. Основные этапы программирования решения задачи.	
2. Вводятся три числа. Программа должна указать, является ли их сумма двузначным числом.	
<b>Преподаватель</b> _____	

<b>Грозненский Государственный Нефтяной Технический Университет</b> им. акад. М.Д. Миллионщикова <b>Кафедра «Информационные технологии»</b> Дисциплина «Программирование» 2-я рубежная аттестация	
<b>Группа:</b>	<b>Семестр: 3</b>
<b>Билет №</b>	
1. Реализация классических и итерационных циклов в C# (операторы for и while).	
2. Пользователь вводит число N. Найти сумму и произведение чисел в диапазоне от N до 2N.	
<b>Преподаватель</b> _____	

<b>Грозненский Государственный Нефтяной Технический Университет</b> им. акад. М.Д. Миллионщикова <b>Кафедра «Информационные технологии»</b> Дисциплина «Программирование» 1-я рубежная аттестация	
<b>Группа:</b>	<b>Семестр: 4</b>
<b>Билет №</b>	
1. Работа с массивами в C#: объявление массивов, основные операции с массивами. Одномерные и двумерные массивы.	
2. Вводится массив из 6 целых чисел. Требуется вычислить произведение элементов, являющихся нечетными числами.	
<b>Преподаватель</b> _____	

**Грозненский Государственный Нефтяной Технический Университет  
им. акад. М.Д. Миллионщикова  
Кафедра «Информационные технологии»  
Дисциплина «Программирование»  
2-я рубежная аттестация**

**Группа:**

**Семестр: 4**

**Билет №**

1. Виды тестирования программного обеспечения. Верификация и валидация.
2. Вводится слово. Определить его длину, вторую и предпоследнюю букву.

**Преподаватель** \_\_\_\_\_

**Грозненский Государственный Нефтяной Технический Университет  
им. акад. М.Д. Миллионщикова  
Кафедра «Информационные технологии»  
Дисциплина «Программирование»  
1-я рубежная аттестация**

**Группа:**

**Семестр: 5**

**Билет №**

1. Работа с каталогами как объектами файловой системы: основные методы.
2. Вводятся три числа. Программа должна указать, является ли их сумма двузначным числом.

**Преподаватель** \_\_\_\_\_

**Грозненский Государственный Нефтяной Технический Университет  
им. акад. М.Д. Миллионщикова  
Кафедра «Информационные технологии»  
Дисциплина «Программирование»  
2-я рубежная аттестация**

**Группа:**

**Семестр: 5**

**Билет №**

1. Работа с базами данных в C#.
2. Пользователь вводит числа А и В. Найти сумму и произведение чисел в диапазоне от А до В.

**Преподаватель** \_\_\_\_\_

## **7.2. Вопросы к зачету / экзамену**

### **3 семестр**

#### **Вопросы к зачету**

1. Понятие алгоритма. Формы представления алгоритмов. Блок-схемы, используемые в графическом представлении.
2. Свойства алгоритмов.
3. Основные этапы программирования решения задачи.
4. Типы данных в Visual C#. Объявление переменных. Инкремент и декремент. Операторы сравнения и логические операторы.
5. Условные ветвления в C# (операторы if и switch).
6. Реализация классических и итерационных циклов в C# (операторы for и while).
7. Модальные и немодальные диалоговые окна. Окна вывода сообщений MessageBox.
8. Языки программирования низкого и высокого уровня. Трансляторы.

9. Основные понятия ООП: объект, класс, инкапсуляция, наследование, полиморфизм.

10. Основные понятия ООП: свойство, метод, событие, оператор, переменная.

11. Модели предметной области и парадигмы программирования.

12. Процедурно-ориентированное программирование.

13. Диаграммы UML в объектно-ориентированном программировании.

#### **4 семестр**

##### **Вопросы к зачету**

1. Работа с массивами в C#: объявление массивов, основные операции с массивами. Одномерные и двумерные массивы.

2. Обработка строк текста в C#. Основные функции обработки строк: Length, IndexOf, Trim, Remove, ToUpper.

3. Обработка строк текста в C#. Основные функции обработки строк: LastIndexOf, Substring, Insert, Replace, ToLower.

4. Основные функции обработки даты и времени в C#.

5. Понятие и назначение отладки программы. Виды ошибок в программе. Понятие трассировки. Точки останова.

6. Ошибки времени выполнения программы. Обработка исключений. Инструкция try...catch.

7. Понятие, назначение и виды библиотек в программировании.

8. Понятие и механизм работы API-функций.

9. Типизация данных в программировании.

10. Предмет и цель программной инженерии как области знаний. Основные принципы разработки программных продуктов.

11. Уровни требований к программным продуктам.

12. Понятие и основные принципы конфигурационного управления.

13. Виды тестирования программного обеспечения. Верификация и валидация.

14. Разработка приложений для различных областей применения. Стиль и методы программирования.

#### **5 семестр**

##### **Вопросы к экзамену**

1. Модульное программирование: пользовательские процедуры и функции в C#.

2. Графика в программировании. Карандаш и кисть.

3. Методы рисования графических примитивов: прямоугольников, эллипсов, многоугольников.

4. Основные методы работы с файлами. Использование файлов для обмена данными с приложением.

5. Работа с каталогами как объектами файловой системы: основные методы.

6. Работа с базами данных в C#.

7. Технология ADO.NET: основные используемые объекты.

8. Взаимодействие приложения на языке C# с MS Excel.

9. Понятие и состав интегрированной среды разработки приложений (IDE).  
Примеры популярных сред разработки.

10. Краткая характеристика основных современных сред разработки приложений.

11. Делегаты в C#.

Образец билета к зачету:

<b>Грозненский Государственный Нефтяной Технический Университет</b> им. акад. М.Д. Миллионщикова <b>Кафедра «Информационные технологии»</b> <b>Дисциплина «Программирование»</b>	
<b>Группа:</b>	<b>Семестр: 3</b>
<b>Билет №</b>	
1. Ошибки времени выполнения программы. Обработка исключений. Инструкция try...catch.	
2. Вводится массив из 6 целых чисел. Требуется вычислить произведение элементов, являющихся нечетными числами.	
Подпись преподавателя _____	Подпись заведующего кафедрой _____

Образец билета к экзамену:

<b>Грозненский Государственный Нефтяной Технический Университет</b> им. акад. М.Д. Миллионщикова <b>Кафедра «Информационные технологии»</b> <b>Дисциплина «Программирование»</b>	
<b>Группа:</b>	<b>Семестр: 4</b>
<b>Билет №</b>	
1. Модульное программирование: пользовательские процедуры и функции в C#.	
2. Пользователь вводит два положительных числа: M и N. Найти произведение чисел в диапазоне от M до N, исключив сами значения M и N.	
Подпись преподавателя _____	Подпись заведующего кафедрой _____

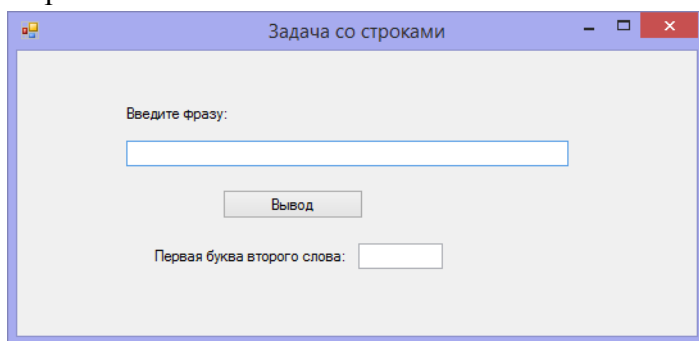
### 7.3. Текущий контроль

Образец типового задания для лабораторных занятий

Лабораторная работа на тему «Обработка текстовой информации в C#.

#### Функции обработки строк»

Пользователь вводит любую фразу. Программа должна выделить и вывести первую букву второго слова.



#### Программный код

```
private void button1_Click(object sender, EventArgs e)
{
```

str – введенная фраза, s – искомая буква; n – позиция пробела в строке как разделителя слов.

```
    string str, s;
    int n;
    str = textBox1.Text;
```

Значение n находится с помощью метода IndexOf, определяющего индекс с отсчетом от нуля первого вхождения указанного символа в данной строке.

```
    n = str.IndexOf(" ");
```

С помощью метода Substring выделяется 1 символ с позиции, следующей сразу за пробелом (n+1).

```
s = str.Substring(n + 1, 1);  
textBox2.Text = s;  
}
```

**Для самостоятельного выполнения**

Выделить из введенной пользователем фразы последнее слово в отдельное текстовое поле.

**7.4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания**

**Таблица 7**

Планируемые результаты освоения компетенции	Критерии оценивания результатов обучения				Наименование оценочного средства
	менее 41 баллов (неудовлетворительно)	41-60 баллов (удовлетворительно)	61-80 баллов (хорошо)	81-100 баллов (отлично)	
<b>ОПК-1:</b> Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности					
<b>Знать:</b> методы и приемы решения практических задач в профессиональной деятельности с помощью информационных систем.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	Комплект заданий для выполнения лабораторных работ, темы докладов с презентациями, вопросы по темам / разделам дисциплины
<b>Уметь:</b> решать базовые задачи обработки данных в профессиональной деятельности.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
<b>Владеть:</b> общей подготовкой для решения практических задач в области информационных технологий.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	
<b>ОПК-2:</b> Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности					
<b>Знать:</b> методы автоматизации операций обработки информации с помощью информационных технологий.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	Комплект заданий для выполнения лабораторных работ, темы докладов с презентациями, вопросы по темам /
<b>Уметь:</b> осуществлять постановку задач и использовать различные алгоритмы обработки информации.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	



<b>Владеть:</b> навыками работы с программными средствами, осуществляющими обработку информации.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	разделам дисциплины
<b>ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения</b>					
<b>Знать:</b> принципы, базовые концепции технологий программирования; основные этапы и принципы создания программного продукта.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	Комплект заданий для выполнения лабораторных работ, темы докладов с презентациями, вопросы по темам / разделам дисциплины
<b>Уметь:</b> составлять алгоритмы решения задач различной структуры и оформлять их в соответствии с синтаксическими правилами языка программирования; разрабатывать и тестировать программные компоненты информационных систем.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
<b>Владеть:</b> приемами разработки пользовательского интерфейса приложения, обеспечивающего оптимальное функционирование программы и удобство пользователя; приемами отладки приложений, поиска ошибок и обработки исключений.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

## **8. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом. На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту-инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или зачете). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебные пособия для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1) для инвалидов и лиц с ограниченными возможностями здоровья **по зрению:**

- **для слепых:** задания для выполнения на семинарах и практических занятиях оформляются рельефно-точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту; обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

- **для слабовидящих:** обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

2) для инвалидов и лиц с ограниченными возможностями здоровья **по слуху:**

- **для глухих и слабослышащих:** обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;

- **для слепоглухих** допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

3) для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

4) для инвалидов и лиц с ограниченными возможностями здоровья, **имеющих**

### **нарушения опорно-двигательного аппарата:**

- для лиц с нарушениями опорно-двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимся.

## **9. Учебно-методическое и информационное обеспечение дисциплины**

### **9.1. Литература**

1. Агапов, В.П. Основы программирования на языке С# [Электронный ресурс]: учебное пособие / В.П. Агапов. – Электрон. текстовые данные. – М.: Московский государственный строительный университет, ЭБС АСВ, 2012. – 128 с. – Режим доступа: <http://www.iprbookshop.ru/16366.html> (ЭБС «IPRbooks»).

2. Поляков, А.Ю. Программирование [Электронный ресурс]: практикум / А.Ю. Поляков, А.Ю. Полякова, Е.Н. Перышкова. – Электрон. текстовые данные. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2015. – 55 с. – Режим доступа: <http://www.iprbookshop.ru/55494.html> (ЭБС «IPRbooks»).

3. Котов, О.М. Язык С#. Краткое описание и введение в технологии программирования [Электронный ресурс]: учебное пособие / О.М. Котов. — Электрон. текстовые данные. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014. – 208 с. – Режим доступа: <http://www.iprbookshop.ru/68524.html> (ЭБС «IPRbooks»).

4. Биллиг, В.А. Основы программирования на С# [Электронный ресурс] / В. А. Биллиг. – 2-е изд. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 574 с. – Режим доступа: <http://www.iprbookshop.ru/73695.html> (ЭБС «IPRbooks»).

### **9.2. Методические указания для обучающихся по освоению дисциплины (приложение)**

## **10. Материально-техническое обеспечение дисциплины**

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

Перечень материально-технических средств учебной аудитории для проведения занятий по дисциплине:

- учебная аудитория, доска;
- стационарные компьютеры;
- мультимедийный проектор;
- настенный экран.

### **10.2. Помещения для самостоятельной работы**

Учебная аудитория для самостоятельной работы – 4-01.

**Методические указания по освоению дисциплины  
«Программирование»**

**1. Методические указания для обучающихся по планированию и организации времени, необходимого для освоения дисциплины**

Изучение рекомендуется начать с ознакомления с рабочей программой дисциплины, ее структурой и содержанием разделов (модулей), фондом оценочных средств, ознакомиться с учебно-методическим и информационным обеспечением дисциплины.

Дисциплина «Программирование» состоит из шести связанных между собой разделов, обеспечивающих последовательное изучение материала.

Обучение по дисциплине «Программирование» осуществляется в следующих формах:

1. Аудиторные занятия (лекции, лабораторные занятия).
2. Самостоятельная работа студента (подготовка к лекциям, лабораторным занятиям, доклады с презентациями, индивидуальная консультация с преподавателем).

Учебный материал структурирован и изучение дисциплины производится в тематической последовательности. Каждому лабораторному занятию и самостоятельному изучению материала предшествует лекция по данной теме. Обучающиеся самостоятельно проводят предварительную подготовку к занятию, принимают активное и творческое участие в обсуждении теоретических вопросов, разборе проблемных ситуаций и поисков путей их решения.

Описание последовательности действий обучающегося:

При изучении курса следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

1. После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10- 15 минут).
2. При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10-15 минут).
3. В течение недели выбрать время для работы с литературой в электронной библиотечной системе (по 1 часу).
4. При подготовке к лабораторному занятию повторить основные понятия по теме, изучить примеры. Решая конкретную ситуацию, – предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить 1-2 задачи.

**2. Методические указания по работе обучающихся во время проведения лекций**

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах. Лекции обычно излагаются в традиционном или в проблемном стиле. Для студентов в большинстве случаев в проблемном стиле. Проблемный стиль позволяет стимулировать активную познавательную деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию.

Во время лекционных занятий рекомендуется вести конспектирование учебного

материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления, выводы и практические рекомендации.

Работая над конспектом лекций, необходимо использовать также литературу, которую дополнительно рекомендовал преподаватель. Тематика лекций дается в рабочей программе дисциплины.

### **3. Методические указания обучающимся по подготовке к лабораторным занятиям**

На лабораторных занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения задач, уметь находить полезный дополнительный материал по тематике занятий.

Студенту рекомендуется следующая схема подготовки к лабораторному занятию:

1. Ознакомиться с планом занятия, который отражает содержание предложенной темы.

2. Проработать конспект лекций.

3. Прочитать основную и дополнительную литературу.

В процессе подготовки к лабораторным занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов отношение к конкретной проблеме.

4. Выполнить домашнее задание.

5. При затруднениях сформулировать вопросы к преподавателю.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы, выступать и участвовать в коллективном обсуждении вопросов изучаемой темы, правильно выполнять практические задания, которые даются в фонде оценочных средств дисциплины.

### **4. Методические указания обучающимся по организации самостоятельной работы**

Цель организации самостоятельной работы по дисциплине «Программирование» – это углубление и расширение знаний в области программирования; формирование навыка и интереса к самостоятельной познавательной деятельности.

Самостоятельная работа обучающихся является важнейшим видом освоения содержания дисциплины, подготовки к практическим занятиям и к рубежной аттестации. Самостоятельная работа носит исследовательский характер, что послужит в будущем основанием для написания выпускной квалификационной работы, практического применения полученных знаний.

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению, с учетом потребностей и возможностей личности.

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет студентам развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивать высокий уровень успеваемости в период обучения, получить навыки повышения профессионального уровня.

Подготовка к лабораторному занятию включает, кроме проработки конспекта и презентации лекции, поиск литературы (по рекомендованным спискам и самостоятельно).

При подготовке к контрольной работе (рубежной аттестации) обучающийся должен повторять пройденный материал в строгом соответствии с учебной программой, используя конспект лекций и литературу, рекомендованную преподавателем. При

необходимости можно обратиться за консультацией и методической помощью к преподавателю.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий – на лекциях, лабораторных занятиях;
- в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Виды СРС и критерии оценок

(по балльно-рейтинговой системе ГГНТУ, СРС оценивается в 15 баллов)

1. Доклад с презентацией
2. Подготовка к лабораторным занятиям

Темы для самостоятельной работы прописаны в рабочей программе дисциплины. Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

**Составитель:**

Доцент кафедры  
«Информационные технологии»

/ Мачуева Д.А. /

**СОГЛАСОВАНО:**

Зав. кафедрой  
«Информационные технологии»

/ Моисеенко Н.А. /

Зав. выпускающей кафедрой «ИСЭ»

/ Магомаева Л.Р. /

Директор ДУМР

/ Магомаева М.А. /