

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Алгоритмы и алгоритмические языки

Кафедра дискретной математики и информатики
факультета математики и компьютерных наук

Образовательная программа бакалавриата

01.03.02 Прикладная математика и информатика

Направленность (профиль) программы
Математическое моделирование и вычислительная математика

Форма обучения

очная

Статус дисциплины: входит в часть, формируемую
участниками образовательных отношений

Махачкала, 2022

Рабочая программа дисциплины «Алгоритмы и алгоритмические языки» составлена в 2022 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика от «10» января 2018 г. № 9.


Разработчик(и): кафедра дискретной математики и информатики, Раджабова Наима Шамильевна, к.ф.-м.н., доцент.

Рабочая программа дисциплины одобрена:
на заседании кафедры дискретной математики и информатики от «28» февраля 2022 г.,
протокол № 6;

зав. кафедрой  Магомедов А. М.
(подпись)

и

на заседании Методической комиссии факультета математики и компьютерных наук от
«24» марта 2022 г., протокол № 4;

председатель  Ризаев М. К.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением
«31» марта 2022 г.

Начальник УМУ  Гасангаджиева А. Г.
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина “Алгоритмы и алгоритмические языки” входит в часть ОПОП, формируемую участниками образовательных отношений, образовательной программы бакалавриата по направлению 01.03.02 Прикладная математика и информатика.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с современными знаниями в области языков и методов программирования: методы программирования на базе языка С# (и среды MS Visual Studio), типы данных и операции, управляющие структуры и визуальные компоненты, создание консольных и графических (на основе форм) приложений, объектно-ориентированное и событийно управляемое программирование.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных – ПК-2, ПК-5.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические и лабораторные занятия.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: текущего контроля в форме 3 контрольных работ и промежуточного – в форме зачета.

Объем дисциплины составляет 3 зачетные единицы (108 часов), в том числе в академических часах по видам учебных занятий:

Объем дисциплины в очной форме

Семестр	Учебные занятия						Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)	
	в том числе							
	Контактная работа обучающихся с преподавателем							СРС, в том числе экзамен
	Всего	из них						
Лекции		Лабораторные занятия	Практические занятия	КСР	Консультации			
2	36		14	14			8	зачет
3	72	16	16				40	зачет
Итого:	108	16	30	14			48	

1. Цели освоения дисциплины

Целью освоения дисциплины (модуля) «Алгоритмы и алгоритмические языки» являются:

- получение фундаментальных знаний по основам современных информационных технологий;
- формирование представлений об основах программирования: знакомство со структурами данных, составление алгоритма решения задачи, реализация алгоритма на языке программирования, отладка и тестирование программы.

Задачей изучения дисциплины является создание и развитие практических навыков по использованию фундаментальных теоретических знаний в области реализации современных и перспективных парадигм программирования

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Алгоритмы и алгоритмические языки» относится к части, формируемой участниками образовательных отношений, образовательной программы бакалавриата по направлению 01.03.02 Прикладная математика и информатика.

Дисциплина «Алгоритмы и алгоритмические языки» призвана содействовать знакомству студентов с языками и методами программирования и является курсом, для освоения которого необходимы теоретические знания и практические навыки, полученные по дисциплинам «Основы информатики», «Языки и методы программирования», «Дискретная математика».

Результаты освоения данной дисциплины будут востребованы на занятиях по «Объектно-ориентированным языкам и системам программирования» и при подготовке выпускной квалификационной работы.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код компетенции из ФГОС ВО	Наименование компетенции из ФГОС ВО	Планируемые результаты обучения	Процедура освоения
ПК-2. Способен понимать, совершенствовать и применять современный математический аппарат, фундаментальные	ПК-2.1. Знает принципы построения совершенствования и применения современного математического аппарата.	Знает современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области	Конспектирование лекций, выполнение лабораторных работ, подготовка и представление реферата

<p>концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий</p>		<p>информационных технологий. Умеет собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. Владеет методами разработки алгоритмических и программных решений в области прикладного программирования и имитационных моделей в пакетах прикладных программ.</p>	
	<p>ПК-2.2. Умеет решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой.</p>	<p>Знает современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий. Умеет собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. Владеет методами разработки алгоритмических и программных решений в области прикладного</p>	

		программирования и имитационных моделей в пакетах прикладных программ.	
	ПК-2.3. Имеет практический опыт использования математического аппарата, международных и профессиональные стандарты в области информационных технологий	<p>Знает современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий.</p> <p>Умеет собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям.</p> <p>Владеет методами разработки алгоритмических и программных решений в области прикладного программирования и имитационных моделей в пакетах прикладных программ.</p>	
ПК-5. Способен анализу требований программному обеспечению	ПК-5.1. Знает методы анализа возможностей реализации требований к программному обеспечению.	<p>Знает структуру программы на языке высокого уровня С#, основные управляющие конструкции, значимые (целые, вещественные, логический, символьный, перечислительный, диапазонный) и ссылочные типы.</p> <p>Умеет разрабатывать ясные и надежные алгоритмы для</p>	Конспектирование лекций, выполнение лабораторных работ, подготовка и представление реферата

	<p>несложных задач, избегая рекурсии, вложенных вызовов, непредсказуемых ситуаций, и составлять адекватные программы с обработкой исключительных ситуаций, с отладкой и достаточно полным тестированием.</p> <p>Владеет начальными навыками разработки алгоритмов и программ, приемами выбора простых и структурированных типов для представления несложных информационных объектов, принятых считать «стандартными», навыками составления программ (20-50 строк) с использованием (объявление и вызов) методов.</p>
--	---

<p>ПК-5.2. Умеет использовать возможности существующей программно-технической архитектуры, методологию разработки программного обеспечения и технологии программирования.</p>	<p>Знает структуру программы на языке высокого уровня С#, основные управляющие конструкции, значимые (целые, вещественные, логический, символьный, перечислительный, диапазонный) и ссылочные типы.</p> <p>Умеет разрабатывать ясные и надежные алгоритмы для несложных задач, избегая рекурсии, вложенных вызовов, непредсказуемых ситуаций, и составлять адекватные программы с обработкой исключительных ситуаций, с отладкой и достаточно полным тестированием.</p> <p>Владеет начальными навыками разработки алгоритмов и программ, приемами выбора простых и структурированных типов для представления несложных информационных объектов, принятых считать «стандартными», навыками составления программ (20-50 строк) с использованием (объявление и вызов) методов.</p>
---	--

	<p>ПК-5.3. Имеет практический опыт проведения оценки и обоснование рекомендуемых решений</p>	<p>Знает структуру программы на языке высокого уровня С#, основные управляющие конструкции, значимые (целые, вещественные, логический, символьный, перечислительный, диапазонный) и ссылочные типы.</p> <p>Умеет разрабатывать ясные и надежные алгоритмы для несложных задач, избегая рекурсии, вложенных вызовов, непредсказуемых ситуаций, и составлять адекватные программы с обработкой исключительных ситуаций, с отладкой и достаточно полным тестированием.</p> <p>Владеет начальными навыками разработки алгоритмов и программ, приемами выбора простых и структурированных типов для представления несложных информационных объектов, принятых считать «стандартными», навыками составления программ (20-50 строк) с использованием (объявление и вызов) методов.</p>	
--	--	--	--

4. Объем, структура и содержание дисциплины.

4.1 Объем дисциплины составляет 3 зачетные единицы, 108 академических часов.

4.2 Структура дисциплины

Структура дисциплины в очной форме

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) и Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	
Модуль 1. Базовые конструкции языка C#								
1	Введение в платформу .Net	2	1-2		2	2	2	Прием лабораторных работ, реферат
2	Системные типы данных	2	3-4		4	4	2	Прием лабораторных работ, реферат
3	Управляющие конструкции языка	2	5-7		4	6	2	Прием лабораторных работ
4	Работа со строками	2	8		2	2	2	Прием лабораторных работ.
Итого по модулю 1:			36		14	14	8	Контрольная работа
Итого за семестр			36		14	14	8	Зачет
Модуль 2. Построение методов								
1	Понятие перегрузки методов	3	1-5	6		6	10	Прием лабораторных работ
3	Массивы	3	6-9	4		4	6	Прием лабораторных работ
Итого по модулю 2:			36	10		10	16	Контрольная работа
Модуль 3. Событийно-ориентированное программирование на C#								
1	Основные принципы ООП	3	10-11	2		2	8	Прием лабораторных работ, реферат
2	Наследование и полиморфизм	3	12-13	2		2	6	Прием лабораторных работ.

3	Событийно-управляемое программирование		14-172		2	10	Прием лабораторных работ.
	Итого по модулю 3		36	6		6	24
	Итого за семестр		72	16		16	40
	ИТОГО:		108	16	14	30	48

4.3 Содержание дисциплины, структурированное по разделам и темам

4.3.1 Содержание лекционных занятий по дисциплине

Семестр 3

Модуль 2. Построение методов

Тема 1. Понятие перегрузки методов

Методы и модификаторы параметров. Перегрузка методов. Перегрузка операций.

Тема 2. Массивы

Класс Array. Ступенчатые массивы

Модуль 3. Событийно-ориентированное программирование на C#

Тема 1. Основные принципы ООП

Тип класса C#. Понятия инкапсуляции, наследования и полиморфизма

Тема 2. Наследование и полиморфизм

Родительский класс System.Object. Типы наследования. Переопределение методов: virtual и override. Интерфейсы.

Тема 3. Событийно-ориентированное программирование

Создание меню. Создание основного меню. Контекстное меню. Пример создания обработчика события. Визуальные компоненты. Кнопка (Button). Радиокнопка (RadioButton). Окно флажка (CheckBox). Метка (Label). Гиперссылочная метка (LinkLabel). Текстовое окно (TextBox). Расширенное текстовое окно (RichTextBox). Список с помечаемыми окнами (CheckBoxList). Выпадающий список (ComboBox). Список позиций (ListBox). Перемещение по числовому диапазону (NumericUpDown). Текст с предопределённым форматом (MaskedTextBox). Всплывающее окно (ToolTip). Индикатор процесса (ProgressBar). Измерение времени (DateTimePicker, MonthCalendar, компонент Timer).

4.3.2 Содержание практических занятий по дисциплине

2 Семестр

Модуль 1. Базовые конструкции языка C#

Практическая работа 1. Введение в платформу .Net

Основные принципы построения платформы .Net. Понятия CLR, CTS, CLS.

Практическая работа 2. Системные типы данных

Лабораторная работа 1. Иерархия для типов данных. Внутренние типы. Операции. Преобразования типов.

Практическая работа 3. Управляющие конструкции языка

Лабораторная работа 3. Операции языка. Условный оператор. Оператор выбора. Операторы цикла.

Практическая работа 4. Работа со строками

Лабораторная работа 4. Классы String, StringBuilder. Сравнение строк.

Семестр 3

Модуль 2. Построение методов

Лабораторная работа 1. Перегрузка методов и операций

Лабораторная работа 2. Класс Array. Ступенчатые массивы

Модуль 3. Объектно-ориентированное программирование на C#

Тема 1. Основные принципы ООП

Лабораторная работа 3. Конструирование класса C#.

Тема 2. Наследование и полиморфизм

Лабораторная работа 4. Иерархия типов C#. Переопределение методов: virtual и override. Интерфейсы.

Тема 3. Событийно-управляемое программирование

Лабораторная работа 5. Создание приложений Калькулятор, Блокнот.

Лабораторная работа 6. Создание приложения Анализатор текста.

Лабораторная работа 7. Обработка списков.

4.3.3. Содержание лабораторно занятий по дисциплине

2 семестр

Модуль 1. Базовые конструкции языка C#

Лабораторная работа 1. Сборки, пространства имен и типы. Структура проекта и приложения в среде VisualStudio.

Лабораторная работа 2. Иерархия для типов данных. Внутренние типы. Операции. Преобразования типов.

Лабораторная работа 3. Операции языка. Условный оператор. Оператор выбора. Операторы цикла.

Лабораторная работа 4. Классы String, StringBuilder. Сравнение строк.

Семестр 3

Модуль 2. Построение методов

Лабораторная работа 1. Перегрузка методов и операций

Лабораторная работа 2. Класс Array. Ступенчатые массивы

Модуль 3. Объектно-ориентированное программирование на C#

Лабораторная работа 3. Конструирование класса C#.

Тема 2. Наследование и полиморфизм

Лабораторная работа 4. Иерархия типов C#. Переопределение методов: virtual и override. Интерфейсы.

Тема 3. Событийно-управляемое программирование

Лабораторная работа 5. Создание приложений Калькулятор, Блокнот.

Лабораторная работа 6. Создание приложения Анализатор текста.

Лабораторная работа 7. Обработка списков.

5. Образовательные технологии

В процессе преподавания дисциплины «Алгоритмы и алгоритмические языки» применяются следующие образовательные технологии: лекционно-зачетная система обучения, технология развития критического мышления, изложение теоретических положений в ходе лекционных занятий с применением современного презентационного оборудования, проведение групповых дискуссий. Основная литература, презентации и конспекты лекций предоставляются студентам в электронном виде.

6. Учебно-методическое обеспечение самостоятельной работы студентов

6.1 Виды самостоятельной работы и её контроля

- проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;

- подготовки к отчетам по лабораторным работам;
- подготовки к сдаче промежуточных форм контроля.

Пакет заданий для самостоятельной работы выдается по истечению месяца с начала семестра, определяются предельные сроки их выполнения и сдачи.

Контрольная работа – средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу.

Перед контрольной работой по каждому модулю студент должен самостоятельно повторить и освоить соответствующий теоретический материал по данному модулю, систематизировать необходимые алгоритмы, детально анализировать ранее решенные на практических занятиях задачи и упражнения. Задания по контрольной работе составлены для проверки освоения необходимых умений и навыков решения задач по тематике данного модуля.

Критерии оценки по контрольной работе

Если студент владеет по данному модулю навыками решения типичных задач, то по этому модулю ему выставляются:

- 1) 50 баллов;
- 2) 40 баллов в случае наличия неточностей;
- 3) 20 баллов в случае наличия некоторых допустимых ошибок.

Эти баллы учитываются при выводе общего результата как интегральной оценки, складывающейся из текущего контроля и промежуточного контроля.

Примерное распределение времени самостоятельной работы студентов:

Вид самостоятельной работы	Примерная трудоёмкость, а.ч.		
	Очная	Очно-заочная	заочная
Текущая СРС			
работа с лекционным материалом, с учебной литературой	6		
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	4		
самостоятельное изучение разделов дисциплины			
выполнение домашних заданий, домашних контрольных работ	5		
подготовка к лабораторным работам и практическим занятиям	5		
подготовка к контрольным работам	8		
подготовка к зачету	8		
Творческая проблемно-ориентированная СРС			
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	4		
исследовательская работа, участие в конференциях, семинарах, олимпиадах	4		
Итого СРС:	44		

Учебно-методическое обеспечение СРС

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Проработка лекционного материала	Контрольный фронтальный опрос	См. разделы 7.3, 8, 9 данного документа
2.	Изучение рекомендованной литературы материалов соответствующих форумов интернет	Контрольный фронтальный опрос, прием и представление рефератов.	См. разделы 7.3, 8, 9 данного документа
3.	Подготовка к отчетам по лабораторным работам.	Проверка выполнения работ по опрос по теме работы.	См. разделы 7.3, 8, 9 данного документа
4.	Подготовка к сдаче промежуточных форм контроля	Контрольные работы по каждому модулю.	См. разделы 7.3, 8, 9 данного документа

Типовые задания для самостоятельной работы

1. Нажатием на кнопку организовать диалог по выбору файла любых форматов, поддерживаемых WindowsMediaPlayer, и проиграть его содержимое в окне.

Указания к решению. Поместим на форму кнопку и невидимый объект openFileDialog1, затем для размещения элемента управления WindowsMediaPlayer на форму внесем предварительно его в раздел General панели инструментов: вызовем контекстное меню раздела General панели инструментов, выберем пункт "Choose Items", в открывшемся окне для выбора компонент на вкладке COM Components поставим галочку рядом с Windows Media Player и нажмем ОК. Появившийся в разделе General появится элемент Windows Media Player перетащим на форму (или дважды щелкнем по этому элементу) и растянем его по ширине формы. Во вкладке Properties изменим имя полученного объекта на «WMP1».

В обработчике щелчка по кнопке (подпишемся на щелчок по кнопке) занесем имя выбранного файла в свойство WMP1.URL.

2. XOR – логическая операция, которая принимает значение «истина» только если всего один из аргументов имеет значение «истина».

Значение выражения $a \text{ xor } 0$ равно

- a.
- b.

- c. 0
 - d. 1
 - e. $a*a$
3. XOR – логическая операция, которая принимает значение «истина» только если всего один из аргументов имеет значение «истина». Свойство $a \text{ xor } b \text{ xor } b = a$ можно использовать для:
- a. Обмена значениями двух целых переменных
 - b. Шифрования с ключом b
 - c. Шифрования документов MsOffice
 - d. Установки старшего бита числа a в 0
4. XOR – логическая операция, которая принимает значение «истина» только если всего один из аргументов имеет значение «истина». Чему равно значение выражения $a \text{ xor } a$?
5. Оператор присваивания (=) вычисляет выражение, заданное в его правой части, и присваивает результат переменной, идентификатор которой расположен в левой части. Например, $z := (x + y)/2$. Во избежание ошибок присваивания необходимо следить, чтобы тип выражения был совместим с типом переменной. Под совместимостью типов данных понимается возможность автоматического преобразования значений одного типа данных в значения другого типа данных. Допустимым оператором присваивания для переменной вещественного типа является:
- a. $y := 2.0 * 1000 * 2000 * 2000$
 - b. $y := 1000 * 2000 * 2000$
 - c. $y := 10 * 1000 * 2000 * 200$
 - d. $y := 4 * 1000 * 1000 * 1000$
6. Оператор присваивания (=) вычисляет выражение, заданное в его правой части, и присваивает результат переменной, идентификатор которой расположен в левой части. Например, $z := (x + y)/2$. Во избежание ошибок присваивания необходимо следить, чтобы тип выражения был совместим с типом переменной. Под совместимостью типов данных понимается возможность автоматического преобразования значений одного типа данных в значения другого типа данных. В каких случаях произойдет автоматическое преобразование в операторе присваивания?
- a. Переменной с типом данных Integer присвоить выражение с типом данных Byte
 - b. Выражение с типом данных Integer присвоить переменной с типом данных Real
 - c. Переменной с типом данных Integer присвоить выражение с типом данных Single
 - d. Переменной с типом данных Byte присвоить выражение с типом данных Word
7. Оператор присваивания (:=) вычисляет выражение, заданное в его правой части, и присваивает результат переменной, идентификатор которой расположен в левой части. Например, $z := (x + y)/2$. Во избежание ошибок присваивания необходимо следить, чтобы тип выражения был совместим с типом переменной. Под совместимостью типов данных понимается возможность автоматического преобразования значений одного типа данных в значения другого типа данных. Чему будет равно значение переменной y, равное 3, после выполнения оператора присваивания $y := y \text{ xor } 0$?

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1 Типовые контрольные задания

7.1.1 Темы рефератов для практических занятий

1. Консольные приложения Delphi и C#. Сравнительная характеристика.
2. Компьютерный вывод рекуррентных формул (на примере задачи разбиения).
3. Длинная целочисленная арифметика.
4. Числа Фибоначчи. Вычислительные аспекты.
5. Настольные графические приложения.
6. Проекты с несколькими формами.
7. Компьютерная графика в C#-проектах.
8. OpenGL в проектах C#.
9. Проекты с применением веб-камер. Распознавание цифр.
10. Визуальные компоненты Visual C#.
11. Класс BigInteger в задачах криптографии

7.1.2 Примерные упражнения и задания к лабораторным занятиям и для самопроверки

Задание 1. Найти время вычисления операций +, -, *, /.

Указания к решению. Во-первых, необходим цикл. Во-вторых, пустой цикл или цикл с присвоением, чтобы время этого цикла (время на организацию цикла) отнять. Самое главное – нужно учитывать мультизадачность, что совсем трудно; то, что при разных запусках наша программа запускается в одном потоке – утешение малое, т.к. этому потоку будет отводиться малое время. Целесообразно программу запускать 10 раз (программным путем).

Задание 2. Составить программу с двумя кнопками «Действия с блокнотом» и «Запуск калькулятора» и текстовым контейнером.

При щелчке на первую кнопку запустить программу Блокнот, ввести в первую строку несколько нулей, во вторую - несколько единиц, затем сохранить текст в текущем каталоге в файле 1234.txt (в предположении отсутствия одноименного файла), после чего программа закроет Блокнот.

При щелчке по второй кнопке проверить, запущено ли стандартное приложение Калькулятор, в случае, если не запущено, программа должна запустить его, затем передачей в него кодов нажатых клавиш выполнить эмуляцию видимых действий по суммированию $0+1+2+ \dots + 100$. При щелчке по контейнеру текста выполнить бесконечный вывод в него буквы «А».

7.1.3 Примерный перечень заданий для текущего контроля

Упр. 1. Введите с клавиатуры n , объявите одномерный массив a из n целых элементов и выведите построчно на экран: а) с использованием индексов, б) с помощью оператора `foreach`.

Упражнение 2. Заполнить прямоугольный массив $n*m$ ($n=m=4$) случайными целыми числами $< k=20$ и просуммировать все четные элементы, сумму вывести в формате с 5 знаками. Упражнение 3. Измените решение первого упражнения на вывод отсортированного массива, добавить вывод индекса значения 0 в отсортированном массиве.

Упражнение 4 (см. в тексте лекции). Обратите внимание на аргумент функции `Main`. Аргумент `string [] args` является массивом строк. Это параметры, принимаемые из командной строки (задаются после имени программы с пробельным разделителем). Перейдем в Windows в режим командной строки, используя главное меню. Затем, используя команды `cd`, перейдем в папку, которая содержит исполняемый файл следующей программы. Пусть это `A.exe`. Наберите командной строке `A 12 13 14` и нажмите `Enter` для выполнения.

Упражнение 5. Программа должна удалить из папки `C:\SomeDir` своего компьютера все файлы с расширением `bmp`.

Упражнение 6. Введите предложение и подсчитайте количество слов в нем, учитывая всевозможные разделители.

Упражнение 7. Заданную с клавиатуры строку выведите в верхнем и нижнем регистре.

Упражнение 8. Создание коллекции - строки статуса с панелями; в первую панель выводятся текущие координаты курсора мыши, во вторую – текущее время. Для обеих панелей предусмотреть всплывающие подсказки («координаты мыши» и текущую дату соответственно).

Упражнение 9. При запуске программы на поверхности формы нарисовать $m=10$ рядов квадратных кнопок по $m=10$ кнопок в ряду, размеры кнопки – $30*30$ Затем выполнить дрожание кнопок в 10 итерациях.

Упражнение 10. Создать проект из двух форм. При нажатии на кнопку первой формы предьявляется вторая форма, а первая начинает «прогулку» по невидимой окружности с постепенным усилением прозрачности.

7.1.4 Вопросы к зачету

Семестр 2

1. Основные принципы построения платформы .Net. Понятия CLR, CTS, CLS.
2. Сборки, пространства имен и типы. Структура проекта и приложения в среде VisualStudio.
3. Иерархия для типов данных.

4. Внутренние типы.
5. Преобразования типов.
6. Операции языка.
7. Условный оператор.
8. Оператор выбора.
9. Операторы цикла.
10. Классы String, StringBuilder. Сравнение строк.

Семестр 3

1. Методы и модификаторы параметров.
2. Перегрузка методов.
3. Перегрузка операций.
4. Класс Array.
5. Ступенчатые массивы
6. Тип класса C#.
7. Понятия инкапсуляции, наследования и полиморфизма
8. Родительский класс System.Object.
9. Типы наследования.
10. Переопределение методов: virtual и override.
11. Интерфейсы.
12. Создание основного меню. Контекстное меню.
13. Пример создания обработчика события.
14. Визуальные компоненты. Кнопка (Button). Радиокнопка (RadioButton).
15. Окно флажка (CheckBox).
16. Метка (Label). Гиперссылочная метка (LinkLabel).
17. Текстовое окно (TextBox). Расширенное текстовое окно (RichTextBox).
18. Список с помечаемыми окнами (CheckListBox).
19. Выпадающий список (ComboBox).
20. Список позиций (ListBox).
21. Перемещение по числовому диапазону (NumericUpDown).
22. Текст с предопределённым форматом (MaskedTextBox).
23. Всплывающее окно (ToolTip).
24. Индикатор процесса (ProgressBar).
25. Измерение времени (DateTimePicker, MonthCalendar, компонент Timer).

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- реферат- 10 баллов,
- выполнение лабораторных или практических заданий – 40 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает: письменная контрольная работа – 100 баллов.

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

а) адрес сайта курса

<http://cathedra.dgu.ru/EducationalProcess.aspx?Value=18&id=6>

б) основная литература:

1. Гаврилов, Г.П. Задачи и упражнения по дискретной математике : учебное пособие [Электронный ресурс] / Г. П. Гаврилов ; Гаврилов Г. П. - М. : Физматлит, 2009. - 416 с. - ISBN 978-5-9221-0477-7. Российская государственная библиотека (РГБ) URL: http://нэб.рф/catalog/000199_000009_004393971/ — Режим доступа: http://нэб.рф/catalog/000199_000009_004393971/ (дата обращения: 12.04.2018)
2. Павловская Т.А. Программирование на языке высокого уровня С# [Электронный ресурс] / Т.А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73713.html> (дата обращения: 12.04.2018)
3. Токманцев Т.Б. Алгоритмические языки и программирование: учебное пособие / Токманцев Т.Б.— Е.: Уральский федеральный университет, ЭБС АСВ, 2013. 104—с.
4. **Андреева, Татьяна Анатольевна.** Программирование на языке Pascal : учеб. пособие / Андреева, Татьяна Анатольевна. - М. : Изд-во Интернет-Ун-та Информ. Технологий: БИНОМ. Лаб. знаний, 2006. - 234,[5] с. - (Основы информационных технологий). - ISBN 5-9556-0025-6 (ИНТУИТ.РУ) : 230-00.

в) дополнительная:

5. Иванов И.П. Сборник задач по курсу «Алгоритмы и структуры данных» [Электронный ресурс] : методические указания / И.П. Иванов, А.Ю. Голубков, С.Ю. Скоробогатов. — Электрон. текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2013. — 36 с. —

978-5-7038-3681-1. — Режим доступа: <http://www.iprbookshop.ru/31548.html> (дата обращения: 12.04.2018)

6. Мейер Б. Инструменты, алгоритмы и структуры данных [Электронный ресурс] / Б. Мейер. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 542 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73680.html> (дата обращения: 12.04.2018)

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1) eLIBRARY.RU [Электронный ресурс]: электронная библиотека / Науч. электрон. б-ка. — Москва, 1999 – . Режим доступа: <http://elibrary.ru/defaultx.asp> (дата обращения: 01.04.2019). – Яз. рус., англ.

2) Moodle [Электронный ресурс]: система виртуального обучением: [база данных] / Даг. гос. ун-т. – Махачкала, г. – Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. – URL: <http://moodle.dgu.ru/> (дата обращения: 22.05.2019).

3) Электронный каталог НБ ДГУ [Электронный ресурс]: база данных содержит сведения о всех видах лит, поступающих в фонд НБ ДГУ/Дагестанский гос. ун-т. – Махачкала, 2010 – Режим доступа: <http://elib.dgu.ru>, свободный (дата обращения: 21.05.2019).

Видеокурсы лекций:

- 1) <https://www.coursera.org/>
- 2) <https://www.udacity.com/>
- 3) <https://www.intuit.ru/>

Форумы по компьютерным наукам и программированию:

- 1) www.stackoverflow.com
- 2) <http://www.cyberforum.ru/>
- 3) <http://citforum.ru/http://www.intuit.ru/>

10. Методические указания для обучающихся по освоению дисциплины. При решении лабораторных заданий программистский подход непременно должен присутствовать (без него решение не будет полноценным), однако, он не должен заслонять сугубо математические (доказательство и др.) и алгоритмические (построение, оптимизация, верификация и др.) аспекты.

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

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: Microsoft Visual Studio Express, Microsoft Windows, Ubuntu Linux, Skype.

Также студентам предоставляется доступ к российским и международным электронным библиотекам через компьютеры университета.

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

Имеется необходимая литература в библиотеке, медиапроектор и компьютер для проведения лекций-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением. Вся основная литература предоставляется студенту в электронном формате.