

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Институт вычислительной математики и информационных технологий



Программа дисциплины

Программирование на С# Б1.В.ДВ.03.01

Направление подготовки: 01.04.02 - Прикладная математика и информатика

Профиль подготовки: Математическое моделирование

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2019

Автор(ы): Тумаков Д.Н.

Рецензент(ы): Бахтиева Л.У.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Плещинский Н. Б.

Протокол заседания кафедры № ____ от "____" 20 ____ г.

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:
Протокол заседания УМК № ____ от "____" 20 ____ г.

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
 - 6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения
 - 6.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания
 - 6.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы
 - 6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)
 - 7.1. Основная литература
 - 7.2. Дополнительная литература
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
2.	Тема 2. Методы и параметры. Массивы. Использование ссылочных типов данных.	1	0	0	6	14
3.	Тема 3. Основы объектно-ориентированного программирования. Создание и удаление объектов. Наследование в C#. Операции, делегаты, события.	1	0	0	4	14
4.	Тема 4. Свойства и индексаторы. Атрибуты. Агрегации, пространства имен, сборки и модули.	1	0	0	4	14
Итого			0	0	18	54

4.2 Содержание дисциплины

Тема 1. Обзор платформы MS.NET. Обзор языка C#. Использование структурных переменных. Операторы и исключения.

Общеязыковая среда выполнения упрощает разработку приложений, обеспечивает правильное и безопасное выполнение, поддерживает несколько языков программирования и следит за ресурсами. Эта среда также называется управляемой средой, она управляет памятью, потоками, организует удаленное взаимодействие. В среде CLR автоматически работает ряд сервисов.

Пространство имен System содержит фундаментальные классы и определяет наиболее часто используемые типы данных, события, интерфейсы, атрибуты и исключения. Также, классы преобразования данных, манипуляций с параметрами .

System.Collections содержит списки, хэш-таблицы и другие виды группировки данных.

ADO.NET - следующее поколение технологии ADO, обеспечивает расширенную поддержку отсекленной программной модели БД, поддерживает работу с XML. Классы для работы с ADO.NET находятся в System.Data, с XML - в пространстве имен System.XML.

ASP.NET - программная структура, основанная на среде выполнения, работающей на сервере, которая позволяет создавать мощные Web приложения. Классы расположены в System.Web. Обеспечивается поддержка XML Web сервисов, необходимых для распределенной разработки.

Все приложения работают с теми или иными данными. Разработчику C# необходимо понимать как хранятся и обрабатываются данные в приложении. Обычно данные хранятся в переменных, которые необходимо объявить до их использования. При объявлении переменной под неё резервируется некоторое количество памяти в зависимости от типа переменной и объявляется имя. После объявления переменной можно присваивать ей значения.

Тема 2. Методы и параметры. Массивы. Использование ссылочных типов данных.

При разработке большинства приложений их разделяют на функциональные модули, так как маленькие разделы кода удобнее для понимания, разработки и отладки. Кроме того, это позволяет несколько раз использовать одни и те же участки кода для других приложений.

В C# приложение состоит из классов, которые содержат именованные блоки кода, называемые методами. Метод - это член класса, который может осуществлять действия или вычислять значения.

Параметры позволяют передавать информацию из одного метода в другой. При объявлении метода можно задать список его параметров, если список пустой, то это означает, что метод не имеет параметров.

В C# программа состоит из классов, содержащих методы. Метод может выполнять действие или вычислять значение.

Метод - это набор операторов, которые выполняются вместе. В C# все методы принадлежат какому-либо классу. Для создания метода необходимо задать его имя, определить список параметров и тело метода.

Массивы хороший инструмент группировки данных. Однако, массивы хранят фиксированное количество объектов, а иногда заранее не известно, сколько потребуется объектов. И в этом случае намного удобнее применять коллекции. Еще один плюс коллекций состоит в том, что некоторые из них реализует стандартные структуры данных, например, стек, очередь, словарь, которые могут пригодиться для решения различных специальных задач.

Большая часть классов коллекций содержится в пространствах имен `System.Collections` (простые необобщенные классы коллекций), `System.Collections.Generic` (обобщенные или типизированные классы коллекций) и `System.Collections.Specialized` (специальные классы коллекций). Также для обеспечения параллельного выполнения задач и многопоточного доступа применяются классы коллекций из пространства имен `System.Collections.Concurrent`.

Класс `List<T>` является самым простым из классов коллекций. Его можно использовать практически так же, как массив, ссылаясь на существующий в коллекции `List<T>` элемент с использованием обычной для массивов системы записи с квадратными скобками и индексом элемента. Можно добавить элемент к концу коллекции `List<T>`, воспользовавшись имеющимся в ее классе методом `Add`, которому предоставляется добавляемый элемент. Размер коллекции увеличивается `List<T>` автоматически.

Класс коллекций `LinkedList<T>` реализует двусвязный список. В каждом элементе списка содержится значение элемента со ссылкой на следующий элемент списка (свойство `Next`) и его предыдущий элемент (свойство `Previous`).

В классе `LinkedList<T>` записи, присущие массивам, не поддерживаются. Вставка элементов осуществляется отличным от `List<T>` способом. Можно воспользоваться методом `AddFirst` для вставки элемента в начало списка с перемещением предыдущего первого элемента дальше по списку и установки в качестве значения его свойства `Previous` ссылки на новый элемент. Аналогично этому для вставки элемента в конец списка можно воспользоваться методом `AddLast`. Для вставки элемента перед указанным элементом списка или после него можно воспользоваться методами `AddBefore` и `AddAfter`.

Тема 3. Основы объектно-ориентированного программирования. Создание и удаление объектов. Наследование в C#. Операции, делегаты, события.

Ключевым словом в ООП является класс. Все языки программирования могут обращаться с общими данными и методами. Эта возможность помогает избежать дублирования. Главная концепция программирования - не писать один и тот же код дважды. Программы без дублирования лучше и понятнее, так как содержат меньше кода. ООП переводит эту концепцию на новый уровень, он позволяет описывать классы (множества объектов), которые делают общими и структуру, и поведение. Классы не ограничиваются описанием конкретных объектов, они также могут описывать и абстрактные вещи.

Объект - это конкретный представитель класса. Его определяет три характеристики: уникальность, поведение и состояние. Уникальность - это характеристика, определяющая отличие одного объекта от другого. Поведение определяет то, чем объект может быть полезен, что он может делать. Поведение объекта классифицирует его. Объекты разных классов отличаются своим поведением. Состояние описывает внутреннюю работу объекта то, что обеспечивает его поведение. Хорошо спроектированный объект оставляет свою состояния недоступным. Нас не интересует, как он это делает, нам важно то, что он умеет это делать.

Процесс получения информации о типе во время выполнения называется рефлексией. В пространстве имен `System.Reflection` содержатся классы и интерфейсы, которые позволяют получить информацию о типах, методах и полях. Класс `System.Type` содержит методы для получения информации об объявлении типа: о конструкторах, полях, методах, событиях и свойствах класса.

Конструкторы - это специальные методы, которые используются для инициализации объектов при создании. Если конструктор не описан в классе, то запускается конструктор по умолчанию.

Наследование - это свойство объектно-ориентированной системы наследовать данные и функциональность базового класса. Можно в класс-потомок к методам и полям родительского класса добавить необходимые поля и методы. Класс-потомок может замещать методы родительского класса. Надо помнить, при изменении родительского класса, класс-потомок может оказаться не рабочим.

Тема 4. Свойства и индексаторы. Атрибуты. Агрегации, пространства имен, сборки и модули.

Агрегации используются для группировки объектов вместе в иерархию объектов, которую можно неоднократно использовать. Агрегации определяют связь целое/часть между объектами, не классами. В одном случае в этой связи время жизни "целого" и "части" могут быть не связаны. В этом случае агрегация называется агрегацией по ссылке. В случае, когда время жизни "целого" и "части" связаны друг с другом, агрегация называется агрегацией по значению. В агрегации "целое" - это всего лишь класс, который используется для группировки "частей"-классов в единое целое, т.е. "целый" класс реально не существует.

Модификаторы доступа определяют возможность доступа к элементам класса, таким как методы и свойства. При разработке класса необходимо явно указывать модификатор доступа у каждого члена класса.

Свойства - это наборы функций, которые могут быть доступны клиенту таким же способом, как открытые поля класса. Идея свойства заключается в методе или паре методов, которые ведут себя с точки зрения клиентского кода как поле.

Индексация позволяет индексировать объекты таким же способом, как массив или коллекцию, позволяет обращаться к членам класса как к массиву.

Атрибуты - инструмент для добавления метаданных в классы, для изменения поведения объекта во время выполнения, для получения информации о транзакции объекта, также используется для передачи дополнительной информации, носящей описательный характер, разработчику. Атрибут - тег для предоставления информации во время выполнения о поведении программных элементов, таких как классы, перечисления, сборки.

Информация об атрибутах хранится в метаданных, связанных с элементом. Атрибуты можно применять к разным программным элементам: сборкам, модулям, классам, структурам, перечислениям, конструкторам, методам, свойствам, полям, событиям, интерфейсам, параметрам, возвращаемым значениям и делегатам.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301).

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений".

Положение от 29 декабря 2018 г. № 0.1.1.67-08/328 "О порядке проведения текущего контроля успеваемости и промежуточной аттестации обучающихся федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет".

Положение № 0.1.1.67-06/241/15 от 14 декабря 2015 г. "О формировании фонда оценочных средств для проведения текущей, промежуточной и итоговой аттестации обучающихся федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет"".

Положение № 0.1.1.56-06/54/11 от 26 октября 2011 г. "Об электронных образовательных ресурсах федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/66/16 от 30 марта 2016 г. "Разработки, регистрации, подготовки к использованию в учебном процессе и удаления электронных образовательных ресурсов в системе электронного обучения федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/11/16 от 25 января 2016 г. "О балльно-рейтинговой системе оценки знаний обучающихся в федеральном государственном автономном образовательном учреждении высшего образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/91/13 от 21 июня 2013 г. "О порядке разработки и выпуска учебных изданий в федеральном государственном автономном образовательном учреждении высшего профессионального образования "Казанский (Приволжский) федеральный университет"".

6. Фонд оценочных средств по дисциплине (модулю)

6.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения

Этап	Форма контроля	Оцениваемые компетенции	Темы (разделы) дисциплины
Семестр 1			
	Текущий контроль		
1	Письменное домашнее задание	ПК-3	1. Обзор платформы MS.NET. Обзор языка С#. Использование структурных переменных. Операторы и исключения.
2	Письменное домашнее задание	ПК-3	2. Методы и параметры. Массивы. Использование ссылочных типов данных. 3. Основы объектно-ориентированного программирования. Создание и удаление объектов. Наследование в С#. Операции, делегаты, события.
3	Письменное домашнее задание	ПК-4	4. Свойства и индексаторы. Атрибуты. Агрегации, пространства имен, сборки и модули.
	Зачет	ПК-3, ПК-4	

6.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Форма контроля	Критерии оценивания				Этап	
	Отлично	Хорошо	Удовл.	Неуд.		
Семестр 1						
Текущий контроль						
Письменное домашнее задание	Правильно выполнены все задания. Продемонстрирован высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.	Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий.	Задания выполнены более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.	Задания выполнены менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.	1 2 3	
	Зачтено	Не зачтено				
Зачет	Обучающийся обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справился с выполнением заданий, предусмотренных программой дисциплины.	Обучающийся обнаружил значительные пробелы в знаниях основного учебно-программного материала, допустил принципиальные ошибки в выполнении предусмотренных программой заданий и не способен продолжить обучение или приступить по окончании университета к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине.				

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Семестр 1

Текущий контроль

1. Письменное домашнее задание

Тема 1

1. Что такое платформа MS.NET? Каковы её преимущества?
2. Перечислите основные понятия платформы .NET.
3. Охарактеризуйте компоненты MS.NET Framework.
4. Что такое общеязыковая среда выполнения? Какие ее основные сервисы?
5. Перечислите основные классы библиотеки .NET Framework.
6. Откуда начинается выполнение приложения C#?
7. Когда приложение заканчивает работу?
8. Сколько классов может содержать приложение C#?
9. Сколько методов Main может содержать приложение?
10. Как прочитать данные, введенные пользователем с клавиатуры?

2. Письменное домашнее задание

Темы 2, 3

1. Объясните что такое методы и почему они важны?
2. Опишите три возможных пути передачи параметров и соответствующие ключевые слова C#.
3. Когда создаются и уничтожаются локальные переменные?
4. Что входит в сигнатуру метода?
5. В чем отличия коллекций от массивов?
6. Перечислите основные свойства и методы класса System.Array.
7. Приведите примеры описания массивов и коллекций.
8. Как передавать и возвращать массивы и коллекции из методов.
9. Объясните принцип работы цикла foreach.
10. Скажите о плюсах и минусах использования двусвязных списков.

11. Приведите практические примеры эффективного использования рассмотренных коллекций.

3. Письменное домашнее задание

Тема 4

1. Может ли арифметическое присваивание ($+=$, $-=$, $*=$, $/=$ и $\%=$) быть перегружено?

2. В каком случае операция преобразования типа должна быть явной?

3. Что такое делегат?

4. Как можно подписаться на событие?

5. Каким образом можно вызвать метод, подписавшийся на событие?

6. Объясните концепцию абстракции, и почему она важна для программной инженерии?

7. Назовите два принципа инкапсуляции.

8. Опишите наследование в контексте ООП.

9. Что такое полиморфизм? Как он связан с ранним и поздним связыванием?

10. Опишите разницу между интерфейсами, абстрактными классами и конкретными классами.

Зачет

Вопросы к зачету:

1. Что такое платформа MS.NET? Каковы её преимущества?

2. Перечислите основные понятия платформы .NET.

3. Охарактеризуйте компоненты MS.NET Framework.

4. Что такое общеязыковая среда выполнения? Какие ее основные сервисы?

5. Перечислите основные классы библиотеки .NET Framework.

6. Откуда начинается выполнение приложения C#?

7. Когда приложение заканчивает работу?

8. Сколько классов может содержать приложение C#?

9. Сколько методов Main может содержать приложение?

10. Как прочитать данные, введенные пользователем с клавиатуры?

11. В каком пространстве имен находится класс Console?

12. Что произойдёт при необработанном в приложении исключении?

13. Что такое общеязыковая система типов?

14. Может ли структурная переменная иметь значение null?

15. Можно ли не инициализировать переменные в C#? Почему?

16. Можно ли потерять данные в результате неявного преобразования?

17. Перечислите основные группы операторов, выделив их характерные особенности.

18. Что такое исключение? Какие классы для обработки исключительных ситуаций вы знаете?

19. Как организовать обработку исключительной ситуации?

20. Объясните что такое методы и почему они важны?

21. Опишите три возможных пути передачи параметров и соответствующие ключевые слова C#.

22. Когда создаются и уничтожаются локальные переменные?

23. Что входит в сигнатуру метода?

24. В чем отличия коллекций от массивов?

25. Перечислите основные свойства и методы класса System.Array.

26. Приведите примеры описания массивов и коллекций.

27. Как передавать и возвращать массивы и коллекции из методов.

28. Объясните принцип работы цикла foreach.

29. Скажите о плюсах и минусах использования двусвязных списков.

30. Приведите практические примеры эффективного использования рассмотренных коллекций.

31. Объясните концепцию абстракции, и почему она важна для программной инженерии?

32. Назовите два принципа инкапсуляции.

33. Опишите наследование в контексте ООП.

34. Что такое полиморфизм? Как он связан с ранним и поздним связыванием?

35. Опишите разницу между интерфейсами, абстрактными классами и конкретными классами.

6.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

В КФУ действует балльно-рейтинговая система оценки знаний обучающихся. Суммарно по дисциплине (модулю) можно получить максимум 100 баллов за семестр, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов.

Для зачёта:

56 баллов и более - "зачтено".

55 баллов и менее - "не зачтено".

Для экзамена:

86 баллов и более - "отлично".

71-85 баллов - "хорошо".

56-70 баллов - "удовлетворительно".

55 баллов и менее - "неудовлетворительно".

Форма контроля	Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций	Этап	Количество баллов
Семестр 1			
Текущий контроль			
Письменное домашнее задание	Обучающиеся получают задание по освещению определённых теоретических вопросов или решению задач. Работа выполняется письменно дома и сдаётся преподавателю. Оцениваются владение материалом по теме работы, аналитические способности, владение методами, умения и навыки, необходимые для выполнения заданий.	1 2 3	15 20 15
Зачет	Зачёт нацелен на комплексную проверку освоения дисциплины. Обучающийся получает вопрос (вопросы) либо задание (задания) и время на подготовку. Зачёт проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.		50

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

7.1 Основная литература:

1. Александрова И.Л., Тумаков Д.Н. Программирование на языке С#: учебное пособие. 2-ое издание. - Казань, Изд-во Казанского университета, 2017. - 112 с. (ISBN 978-5-00019-944-5) Режим доступа: https://shelly.kpfu.ru/e-ksu/docs/F_2033489788/Tumakov_Programmирование_na_yazyke_C_.pdf
2. Гуриков, С.Р. Введение в программирование на языке Visual C#: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2013. - 448 с.: 70x100 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-91134-738-3, 500 Режим доступа: <http://www.znaniy.com/bookread.php?book=404441>
3. Кузин, А.В. Программирование на языке Си / А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с.: 70x100 1/16. - (Высшее образование) (Обложка) ISBN 978-5-00091-066-5 - Режим доступа: <http://znaniy.com/catalog/product/505194>

7.2. Дополнительная литература:

1. Залогова, Л.А. Основы объектно-ориентированного программирования на базе языка С# [Электронный ресурс] : учебное пособие / Л.А. Залогова. ? Электрон. дан. ? Санкт-Петербург : Лань, 2018. ? 192 с. ? Режим доступа: <https://e.lanbook.com/book/106731>
2. Солдатенко, И.С. Практическое введение в язык программирования Си [Электронный ресурс] : учебное пособие / И.С. Солдатенко, И.В. Попов. ? Электрон. дан. ? Санкт-Петербург : Лань, 2018. ? 132 с. ? Режим доступа: <https://e.lanbook.com/book/109619>

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

- Microsoft Virtual Academy - <http://www.microsoftvirtualacademy.com/training-courses/exciting-programming-c-sharp-rus>
Информатика: Курс лекций. - <http://znaniy.com/catalog.php?bookinfo=204273>
Методы программирования. Компьютерные вычисления - <http://znaniy.com/bookread.php?book=350418>
Практикум по курсу "Алгоритмизация и программирование" - http://libweb.ksu.ru/ebooks/09_63.pdf
Руководство по программированию на С# - <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>

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

Вид работ	Методические рекомендации
лабораторные работы	Содержание лабораторных работ посвящено изучению основ алгоритмизации вычислительных процессов, изучению синтаксических и семантических конструкций языка программирования высокого уровня C#. Рассматриваются принципы программирования разветвляющихся и циклических вычислительных процессов, обработки массивов и файлов, использования структурированных типов данных, подпрограмм с особенностями передачи в них параметров. Приводятся различные приемы программирования, в том числе использование указателей и ссылок, динамически распределяемой памяти. Все работы ориентированы на применения структурного подхода и модульности, представлены принципы работы с библиотечными ресурсами. В каждой лабораторной работе приводится пример выполнения типового задания с учетом предъявляемых требований. Необходимо изучить теорию по заданной теме, только после этого приступать к лабораторным заданиям.
самостоятельная работа	Содержание самостоятельных работ посвящено изучению основ алгоритмизации вычислительных процессов, изучению синтаксических и семантических конструкций языка программирования высокого уровня C#. Рассматриваются принципы программирования разветвляющихся и циклических вычислительных процессов, обработки массивов и файлов, использования структурированных типов данных, подпрограмм с особенностями передачи в них параметров. Приводятся различные приемы программирования, в том числе использование указателей и ссылок, динамически распределяемой памяти. Все работы ориентированы на применения структурного подхода и модульности, представлены принципы работы с библиотечными ресурсами.
письменное домашнее задание	Содержание домашних работ также посвящено изучению основ алгоритмизации вычислительных процессов, изучению синтаксических и семантических конструкций языка программирования высокого уровня C#. Рассматриваются принципы программирования разветвляющихся и циклических вычислительных процессов, обработки массивов и файлов, использования структурированных типов данных, подпрограмм с особенностями передачи в них параметров. Домашние задания выполняются после того, как выполнены все лабораторные по заданной теме.
зачет	Проверка теоретических знаний производится в виде опроса по любым вопросам, входящим в программу. Вопрос, как правило, формулируется в виде просьбы написать на листе бумаги короткий (10-15 строк) код, иллюстрирующий то или иное понятие языка программирования. Обычно задается от двух до пяти вопросов. Студенты, не сдавшие теоретическую часть, автоматически не допускаются к выполнению практической части зачета.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Освоение дисциплины "Программирование на С#" предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows Professional 7 Russian

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Освоение дисциплины "Программирование на С#" предполагает использование следующего материально-технического обеспечения:

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи;
- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 01.04.02 "Прикладная математика и информатика" и магистерской программе "Математическое моделирование".